

Ομάδα: ΓΕΝΙΚΑ - GENERAL

- * ΟΛΑ [ALL]
- * ΑΡΙΘΜΗΤΙΚΑ [ARITHMETIC FUNCTIONS]
- * ΕΝΤΟΛΕΣ ΕΙΚΟΝΩΝ [BITMAP COMMANDS]
- * ΕΝΤΟΛΕΣ ΙΣΤΟΥ [BROWSER COMMANDS]
- * ΚΟΙΝΕΣ ΦΟΡΜΕΣ [COMMON DIALOGUES]
- * ΕΝΤΟΛΕΣ ΚΟΝΣΟΛΑΣ [CONSOLE COMMANDS]
- * ΣΤΑΘΕΡΕΣ [CONSTANTS]
- * ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ [DATABASES]
- * ΟΡΙΣΜΟΙ [DEFINITIONS]
- * ΕΓΓΡΑΦΑ [DOCUMENTS]
- * ΓΡΑΦΙΚΑ 2Δ [DRAWING 2D]
- * ΧΕΙΡΙΣΜΟΣ ΑΡΧΕΙΩΝ [FILE OPERATIONS]
- * ΡΟΗ ΠΡΟΓΡΑΜΜΑΤΟΣ [FLOW CONTROL]
- * ΔΙΕΡΜΗΝΕΥΤΗΣ [INTERPRETER]
- * ΧΕΙΡΙΣΜΟΣ ΤΜΗΜΑΤΩΝ [MODULE COMMANDS]
- * ΕΝΤΟΛΕΣ ΔΕΙΚΤΗ [MOUSE COMMANDS]
- * ΧΕΙΡΙΣΤΕΣ ΤΗΣ ΤΥΠΩΣΕ [OPERATORS IN PRINT]
- * ΕΚΤΥΠΩΣΕΙΣ [PRINTINGS]
- * ΟΘΟΝΗ ΚΑΙ ΑΡΧΕΙΑ [SCREEN AND FILES]
- * ΗΧΟΙ ΚΑΙ ΤΑΙΝΙΕΣ [SOUNDS AND MOVIES]
- * ΕΝΤΟΛΕΣ ΣΩΡΟΥ [STACK COMMANDS]
- * ΑΛΦΑΡΙΘΜΗΤΙΚΑ [STRING FUNCTIONS]
- * ΣΤΟΧΟΙ ΚΑΙ ΕΠΙΛΟΓΗ [TARGET AND MENU]
- * ΜΕΤΑΒΛΗΤΕΣ ΣΥΣΤΗΜΑΤΟΣ [VARS READ ONLY]

Ομάδα: ΔΙΕΡΜΗΝΕΥΤΗΣ - INTERPRETER

Αναγνωριστικό: ΑΞΙΩΣΗ [ASSERT]

ΑΞΙΩΣΗ X=10, Z>40

ΥΠΟΛΟΓΙΖΕΙ ΟΛΕΣ ΤΙΣ ΣΥΝΘΗΚΕΣ ΚΑΙ ΓΙΑ ΟΣΕΣ ΒΡΕΙ ΝΑ ΜΗΝ ΙΚΑΝΟΠΟΙΟΥΝΤΑΙ ΔΙΝΕΙ ΛΑΘΟΣ ΚΑΙ ΔΕΙΧΝΕΙ ΠΟΙΕΣ ΣΥΝΘΗΚΕΣ ΔΕΝ ΙΚΑΝΟΠΟΙΗΘΗΚΑΝ.

ΔΙΑΦΥΓΗ ΟΧΙ - ΚΑΝΕΙ ΑΝΕΝΕΡΓΕΣ ΤΙΣ ΑΞΙΩΣΕΙΣ

Αναγνωριστικό: ΑΡΓΑ [SLOW]

ΑΡΓΑ

Μέσα σε πρόγραμμα πρέπει να δώσουμε: ΘΕΣΕ ΑΡΓΑ

Δίνει χρόνο στο σύστημα, οπότε έχει μια μικρή καθυστέρηση.

Αναγνωριστικό: APXH [START]

1) Επανεκκίνηση Ψυχρή (COLD RESTART)

APXH

2) Επανεκκίνηση (WARM RESTART)

APXH "", ""

3) Όπως η (2) αλλά με αλλαγή βασικής γραμματοσειράς

APXH "COURIER"

4) Όπως η (2) αλλά με εκτέλεση εντολών

APXH "", "ΜΠΙΠ:ΜΠΙΠ"

Παρατήρηση

1) Αν δώσουμε όνομα γραμματοσειράς, τέτοιο που να μην υπάρχει στην λίστα γραμματοσειρών του υπολογιστή που τρέχουμε την γλώσσα, τότε επανέρχεται η εξ ορισμού γραμματοσειρά.

2) η εντολή αυτή σε καμία περίπτωση δεν αλλάζει το μέγεθος του περιθωρίου καθώς και των γραμμάτων. Γι' αυτές τις περιπτώσεις δες ΤΥΠΟΣ & ΠΑΡΑΘΥΡΟ

3) Σε καμία περίπτωση από της παραπάνω δεν σβήνει ο σωρός τιμών, δες την ΑΔΕΙΑΣΕ

Αναγνωριστικό: ΒΟΗΘΕΙΑ [HELP]

1) Εμφάνιση βοήθειας

CTRL+F1

ΒΟΗΘΕΙΑ

2)

ΒΟΗΘΕΙΑ φραση\$

πχ "ΕΝΤΟΛΕΣ" ή "ΣΥΝΑΡΤΗΣΕΙΣ"

3)

ΒΟΗΘΕΙΑ εντολή

4) Ανάγνωση από το αρχείο βοήθειας mdb (αν το έχουμε, αλλιώς δείτε το 5)

\\ Όλα τα αναγνωριστικά σε αλφαβητική σειρά

Φόρμα 80,50

Άδειασε

Έγγραφο Όλα\$

ΝέαΓραμμή\$={

}

Κατάλογος Εφαρμογή.Κατ\$ \\ AppDir\$ δίνει το φάκελο που βρίσκεται η εφαρμογή του διερμηνευτή

\\ ετοιμάζουμε το όνομα της βάσης για το κλείσιμο στο τέλος

\\ είναι προαιρετικό, δηλαδή μπορούμε να μη το βάλουμε, και να μην φτιάξουμε την Βάση\$
Βάση\$=Κατ\$+Πεζ\$("HELP2000")+".mdb"

Ανάκτηση "HELP2000", "SELECT * FROM COMMANDS ORDER BY DESCRIPTION"

Διάβασε N

Για i=1 έως N {

Ανάκτηση "HELP2000", "SELECT * FROM COMMANDS ORDER BY DESCRIPTION", i

Πέτα \\ διώχνει ένα στοιχείο από το σωρό (από τη κορυφή)

Διάβασε Ελλ\$, Κείμενο\$, Αγγ\$, νούμερο_ομάδας

\\ το 22 είναι η ομάδα Γενικά - που περιλαμβάνει τις εντολές ανά ομάδα

Αν νούμερο_ομάδας<>22 Τότε {

Ανάκτηση "HELP2000", "GROUP", 1, "GROUPNUM ", νούμερο_ομάδας

Πέτα 2 : Διάβασε τίτλος_ελλ_αγγ\$

Ελλ\$=Αλλαγή\$(" ", " ", Ελλ\$)

Αν Δεξί\$(Ελλ\$,1)="(" Τότε Ελλ\$=Ελλ\$+"")

\\ ο χαρ\$(160) είναι το διάστημα που δεν "σπάει", λογαριάζετα σαν γράμμα.

Αναφορά Ελλ\$ + " (" +ΑριστερόΜέρος\$(τίτλος_ελλ_αγγ\$, " "+Χαρ\$(160))+")"

Όλα\$=Ελλ\$ + " (" +ΑριστερόΜέρος\$(τίτλος_ελλ_αγγ\$,

" "+Χαρ\$(160))+")"+ΝέαΓραμμή\$

}

}

Τύπωσε

Σημ 1 : Πρόχειρο Όλα\$ ' στο Πρόχειρο

Κατάλογος Χρήστη \\ επιστροφή στο κατάλογο χρήστη

Σώσε.Έγγραφο Όλα\$, "Greek.txt"

Σύστημα "WordPad", Παράθεση\$(Κατ\$+"Greek.txt")

Κλείσε Βάση Βάση\$ ' Θα κλείσει όπως και να έχει στο τερματισμό, αλλά εδώ γίνεται όταν το θέλουμε εμείς!

5) Ανάγνωση από το αρχείο dat/

Εγγραφο ολα\$, κειμ\$

Φόρτωσε.Εγγραφο ολα\$, Εφαρμογή.Κατ\$+"help2000utf8.dat"

α=Τιμή(Παράγραφος\$(ολα\$, 1))

β=Τιμή(Παράγραφος\$(ολα\$, 2+α))

Για i=1 Έως β

μ=Τιμή(ΔεξίΜέρος\$(Παράγραφος\$(ολα\$, 2+α+i+2*β), "!"))+1

β\$=μεσ\$(ΑριστερόΜέρος\$(Παράγραφος\$(ολα\$, 2+α+i+2*β), "!"), 2)

Αν Δεξί\$(β\$, 1)="(" Τότε β\$+="")

γραμμή\$=β\$+" (" + ΑριστερόΜέρος\$(Παράγραφος\$(ολα\$, μ), " "+χαρ\$(160))+")"

Τύπωσε #-2, γραμμή\$

κειμ\$=γραμμή\$+{

}

Επόμενο i

Σώσε.Έγγραφο κειμ\$,"Greek.txt"
Σύστημα "WordPad", Παράθεση\$(Κατ\$+"Greek.txt")

Αναγνωριστικό: ΓΡΗΓΟΡΑ [FAST]
ΓΡΗΓΟΡΑ
ΓΡΗΓΟΡΑ !

\\ Παράδειγμα
ΦΟΡΜΑ 80,60
ΑΝΑΛΥΤΗΣ
Κ=1
ΑΑ=10
ΔΙΑΒΑΣΕ ? Κ, ΑΑ
ΑΝ Κ=1 ΤΟΤΕ {
 ΘΕΣΕ ΑΡΓΑ
 ΤΥΠΩΣΕ "ΑΡΓΑ"
} ΑΛΛΙΩΣ.ΑΝ Κ= 2 ΤΟΤΕ {
 ΘΕΣΕ ΓΡΗΓΟΡΑ
 ΤΥΠΩΣΕ "ΓΡΗΓΟΡΑ"
} ΑΛΛΙΩΣ {
 ΘΕΣΕ ΓΡΗΓΟΡΑ !
 ΤΥΠΩΣΕ "ΜΕΓΙΣΤΟ"
}

ΑΝΑΝΕΩΣΗ ΑΑ ' ΟΡΙΖΟΥΜΕ ΧΡΟΝΙΚΟ ΔΙΑΣΤΗΜΑ ΚΑΙ ΓΙΝΕΤΑΙ ΚΑΙ ΣΥΓΧΡΟΝΙΣΜΟΣ
' ΜΙΚΡΟ ΔΙΑΣΤΗΜΑ ΠΙΟ ΓΡΗΓΟΡΗ ΑΝΑΝΕΩΣΗ ΠΙΟ ΑΡΓΟ ΤΟ ΑΠΟΤΕΛΕΣΜΑ.
ΓΙΑ Ι=1 ΕΩΣ 3000 {
 ΤΥΠΩΣΕ "x";
}
ΤΥΠΩΣΕ ΦΟΡΤΟΣ
ΑΝΑΝΕΩΣΗ
ΑΝΑΜΟΝΗ 1000

τρέξτε το με το ΑΡΓΑ και με το ΓΡΗΓΟΡΑ και το ΓΡΗΓΟΡΑ !
Στην γραμμή εντολών δίνονται άμεσα οι εντολές Αργά, Γρήγορα και Γρήγορα !
Δες την Θέσε

Αναγνωριστικό: ΔΙΑΓΡΑΦΗ [REMOVE]

ΔΙΑΓΡΑΦΗ

Διαγράφει το τελευταίο τμήμα που δημιουργήθηκε. Την επόμενη φορά θα διαγράψει από τα εναπομείναντα το τελευταίο που δημιουργήθηκε.

Διαγραφή φορτωμένης βιβλιοθήκης (όλο το όνομα όπως δώσαμε στην Όρισε μέχρι την τελεία, χωρίς αυτήν)

Διαγραφή "user32"

Αναγνωριστικό: ΔΙΑΚΟΠΤΕΣ [SWITCHES]

Στη κονσόλα:

οι αλλαγές γράφονται στη registry των Windows. Με Break, ή εντολή Αρχή ή αν ξεκινήσουμε άλλο περιβάλλον, θα διαβαστούν πάλι, οι τιμές των διακοπτών, με τις αλλαγές που κάναμε.

Διακόπτες "+DEC -DIV -PRI"

Σε τμήμα:

οι αλλαγές συμβαίνουν μόνο για το φορτωμένο περιβάλλον. Με Break, ή εντολή Αρχή ή αν ξεκινήσουμε άλλο περιβάλλον, θα διαβαστούν πάλι οι παλιές σωσμένες τιμές των διακοπτών.

Θέσε Διακόπτες "+DEC -DIV -PRI"

Στην γραμμή εντολών των Windows, όταν καλούμε την M2000 (και σε συντόμευση, στη θέση για παραμέτρους)

οι αλλαγές θα είναι τοπικές.

Διακόπτες:

TXT

με + οι συγκρίσεις αλφαριθμητικών γίνονται βάσει κειμένου (> < >= <= <> =)

με - οι συγκρίσεις αλφαριθμητικών γίνονται δυαδικά **εξ ορισμού είναι αυτό** (> < >= <= <> =)

DEC

με - εμφανίζει την τελεία ως χαρακτήρα υποδιαστολής

με + εμφανίζει το τοπικό χαρακτήρα υποδιαστολής (στα ελληνικά είναι το κόμμα εκτός και αν έχει αλλαχθεί) ***εξ ορισμού είναι αυτό***

στο κώδικα είναι πάντα η τελεία.

Αν δοθεί εντολή Τοπικό 1032 θα αλλάξει προσωρινά η υποδιαστολή σε "," μέχρι την επόμενη Διακόπτες ή επόμενη Τοπικό, ή αν πατήσουμε το πλήκτρο Break, ή με εντολή Αρχή στη κονσόλα, ή Θέσε Αρχή στο κώδικα.

DIV

αλλάζει τα mod και div ή υπόλοιπο, υπόλ και δια

με + δουλεύει όπως στην Python και έχουν το ίδιο επίπεδο με το * και το / (στο δεξιό μέρος) δηλαδή το 10 υπόλοιπο 2.3 *2 θα είναι αυτό: (10 υπόλοιπο 2.3) *2

με - με ακέραιους δουλεύει όπως στην VB6, ενώ με πραγματικούς (δεν υποστηρίζει η Vb6) δίνει ακέραιο στο δια και πραγματικό στο υπόλοιπο. ***εξ ορισμού είναι αυτό***

εδώ δεν έχουμε ίδιο επίπεδο με το * και /, δηλαδή το 10 υπόλοιπο 2.3 *2 θα είναι αυτό: 10

υπόλοιπο (2.3 *2)

PRI

αλλάζει την προτεραιότητα εκτέλεσης των KAI και H

πχ στο Τύπωσε Αληθής Η Ψευδής KAI Ψευδής

με - όλα εκτελούνται με τη σειρά

άρα αληθής ή ψευδής δίνει αληθής και ψευδής δίνει ψευδής ***εξ ορισμού είναι αυτό***

με + το KAI (λογικός πολλαπλασιασμός) εκτελείται πριν το Η (λογική πρόσθεση)

άρα το ψευδής KAI ψευδής θα δώσει Ψευδής και θα εκτελεστεί το Αληθής Η Ψευδής άρα θα δώσει το Αληθής.

REC

Όριο αναδρομής για συναρτήσεις μόνο (δες και το Όριο.Αναδρομής)

με - θέτει το όριο στο 300 (το χρησιμοποιούμε αν θέλουμε να τρέξουμε την M2000 μέσα από άλλη εφαρμογή)

με + θέτει το όριο στο 3260 *** εξ ορισμού είναι αυτό***

η αλλαγή απλά αλλάζει τον κόφτη εσωτερικά, πέρα από τον οποίο δίνει σφάλμα. Η αλλαγή θα φανεί με την Monitor μετά από Break (με Αρχή ή με πάτημα του Break), και θα είναι μόνιμη, μέχρι την επόμενη αλλαγή.

FOR

' δεν αποθηκεύεται μόνιμα. με Break επανέρχεται στο -FOR

με - λειτουργία νορμάλ (όλα τα For (Για) εκτελούνται τουλάχιστον μια φορά)

με + γίνεται έλεγχος ώστε αν το βήμα δεν θα φτάσει ποτέ την τελική τιμή να μη εκτελεστεί το

For (Για)

DIM

' δεν αποθηκεύεται μόνιμα. με Break επανέρχεται στο -DIM

με - λειτουργία νορμάλ, όλοι οι πίνακες δημιουργούνται με τόσα στοιχεία όσα ορίζουμε.

με + και εφόσον η βάση πινάκων είναι 0, κάθε διάσταση του πίνακα που δεν είναι μηδενική παίρνει συν ένα στοιχείο

SEC

' Αποθηκεύεται μόνιμα (secure names), οι κλήσεις τμημάτων γίνεται με εσωτερικά διαφορετικά ονόματα

με - λειτουργία όπως στην έκδοση 8. (για μικρά ονόματα μεταβλητών είναι πιο γρήγορο, αλλά έχει λάθη, αν και όχι σημαντικά)

με + λειτουργία με μέγιστη ασφάλεια (δεν έχει λάθη, ανώτερη λογική) - είναι η κανονική λειτουργία

η διαφορά φαίνεται στο πρόγραμμα στην αγγλική έκδοση εδώ: SWITCHES
βλέπουμε τις ρυθμίσεις με την εντολή MONITOR από την κονσόλα ή Θέσε Monitor από τμήμα.

RDB

' δεν αποθηκεύεται μόνιμα. με BREAK δεν αλλάζει η τυχόν αλλαγή

με - σταματάει την περικοπή στο 13 δεκαδικό στους διπλούς (double) όπως υπήρχε πριν την 9.6 έκδοση

με + ξεκινάει τη περικοπή στο 13 δεκαδικό στους διπλούς (double) όπως υπήρχε πριν την 9.6 έκδοση

Εξ ορισμού είναι στο -RDB

TAB

με +TAB ο διορθωτής χρησιμοποιεί το tab για να βάζει το χαρακτήρα 9

με -TAB ο διορθωτής χρησιμοποιεί το tab για να βάζει διαστήματα

το μήκος το ορίζουμε με την εντολή Σ ! 6 για 6 χαρακτήρες ή 3 "m" πλάτος για τον χαρακτήρα 9

ο χαρακτήρας 9 όταν μπαίνει μέσα στο κείμενο δίνει μεταβλητό μήκος ανάλογα με τη θέση στη γραμμή που ξεκίνησε

SBL

+SBL δίνει Αληθής/Ψευδής ως τιμή στην Τύπωσε για αριθμούς τύπου λογικός

-SBL δίνει -1/0 για τιμή στη για αριθμούς τύπου λογικός λογικός

Οι τιμές στις σταθερές Αληθής/Αληθές/Ψευδής/Ψευδές είναι τύπου διπλός με τιμές -1/-1/0/0

Οι συγκρίσεις δίνουν πάντα τύπο λογικό.

Για να πάρουμε λογικό από μεταβλητή αριθμητική που δεν είναι λογικός θέλουμε μια σύγκριση

A=Αληθής : Τύπωσε A=Αληθής

θα μας δώσει Αληθής αν ο διακόπτης είναι +SBL ή -1 αν είναι -SBL

INP

με +INP η μεταβλητή ΠΕΔΙΟ γυρίζει 13 όταν πατήσουμε το enter

με -INP η μεταβλητή ΠΕΔΙΟ γυρίζει 1 όταν πατήσουμε το enter (εξ ορισμού)

MDB

με +MDB το σύστημα παίρνει τη βοήθεια από το Help2000.mdb

με -MDB το σύστημα παίρνει τη βοήθεια από το help2000utf8.dat

NBS

με -NBS το η εξαγωγή στο πρόχειρο από τον διορθωτή αλλάζει το TAB με NBS (non break space)

εξ ορισμού είναι το +NBS

Ειδικό διακόπτες για την κλήση από τη γραμμή εντολών των Windows

Test

+TEST και -TEST

εφαρμόζει την διαδικασία ελέγχου με οποιοδήποτε πρόγραμμα και αν αυτό ξεκινάει άμεσα με τη φόρτωση.

Norun

-NORUN φορτώνει το πρόγραμμα αλλά δεν το τρέχει (δεν τρέχει εντολές εκτός από αυτές που φορτώνουν τμήματα και συναρτήσεις)

Διακόπτες που εφαρμόζουν τις αλλαγές στην επόμενη εκτέλεση του διεμνηυτή ή αν πατήσουμε το Break, ή αν δώσουμε την εντολή Αρχή

Linespace

+LINESPACE 150 ορίζει διάστιχο 10 εικονοστοιχείων (αν το εικονοστοιχείο είναι 15 twips - στα 96 dpi έχουμε $1440/96=15$ όπου 1440 twips είναι η λογική ίντσα - πραγματική στον εκτυπωτή- και dpi είναι στην οθόνη τα εικονοστοιχεία στην ίντσα). Τιμές από 0 έως 600 (για 15 dpi), ακέραιες τιμές.

-LINESPACE κάνει το διάστιχο 0

Size

+SIZE 12 κάνει τα γράμματα 12άρια, παίρνει τιμές από 8 έως 48 (μόνο ακέραιες τιμές)

-SIZE κάνει τα γράμματα 15άρια από οποιαδήποτε τιμή έχουν στις ρυθμίσεις

Font

-FONT επιλέγει την Monospac821Greek BT γραμματοσειρά (αν υπάρχει)

Bold

+BOLD χρησιμοποιεί από την αρχή τα bold γράμματα της γραμματοσειράς

-BOLD χρησιμοποιεί την κανονική γραμματοσειρά

Pen

+PEN 5 βάζει την τιμή 5 από τα βασικά χρώματα 0 έως 15. Αν το Paper είναι 5 τότε βάζει το 16-5

-PEN κάνει την Πένα 0 και την Οθόνη 7

Paper

+PAPER 0 βάζει την τιμή 0 από τα βασικά χρώματα 0 έως 15. Αν το Pen είναι 0 τότε βάζει το 16-0

-PAPER κάνει την Πένα 0 και την Οθόνη 7

Dark

+DARK κάνει την εξαγωγή σε Html με επιλογή χρωμάτων για σκούρο φόντο

+DARK κάνει την εξαγωγή σε Html με επιλογή χρωμάτων για φωτεινό φόντο

Casesensitive

+CASESENSITIVE όταν αναζητεί αρχεία κοιτάει αν συμπίπτουν τα πεζά/κεφαλαία

-CASESENSITIVE το αντίστροφο.

Αν δοθεί η εντολή αλλάζει άμεσα (όχι στην επόμενη εκκίνηση)

Greek

+Greek κάνει τα μηνύματα στους διαλόγους του περιβάλλοντος να εμφανίζονται στα ελληνικά

-Greek κάνει τα μηνύματα στους διαλόγους του περιβάλλοντος να εμφανίζονται στα Αγγλικά

\\ Παράδειγμα

Φόρμα 80, 50

Πίνακας Βάση 0, β\$(2)

β\$(0)="-DEC", "+DEC"

γενική αα=κάθε(β\$())

Ενώ αα {

Για άλλο=ψευδές έως αληθές {

Τύπωσε "Διακόπτες "+Παράθεση\$(Πίνακας\$(αα))

```

Θέσε Διακόπτες Πίνακας$(αα)
Κάνε μοφ1$()="{0:1:"+γραφή$(-στήλη)+"}{1:2:"+γραφή$(-στήλη)+"}"
Αν άλλο Αλλιώς Τοπικό 1032 : Τύπωσε "Τοπικό ";Τοπικό
Τύπωσε μορφή$(μοφ1$(), -1.66, 2.45)
Τύπωσε -1.66, 2.45
Αν άλλο Αλλιώς Τοπικό 1033 : Τύπωσε "Τοπικό ";Τοπικό
Τύπωσε μορφή$(μοφ1$(), -1.66, 2.45)
Τύπωσε -1.66, 2.45
Τύπωσε $(3), γραφή$(.66), γραφή$(-.45), $(0)," ποτέ δεν αλλάζει" ' χωρίς 0 και πάντα
τελεία.
Τύπωσε $(3), γραφή$(.66,0), γραφή$(-.45, 0), $(0)," ποτέ δεν αλλάζει" ' $(3) σημαίνει
δεξιά στοίχιση
Τύπωσε $(3), γραφή$(.66,1033), γραφή$(-.45, 1033), $(0)," ποτέ δεν αλλάζει" '
χρησιμοποιεί το τοπικό άμεσα
Τύπωσε $(3), γραφή$(.66,"") , γραφή$(-.45,"")
' $(0) είναι η εξ ορισμού στοίχιση, στα αλφαριθμητικά έχουν αριστερή στοίχιση και
μπορούν να επεκταθούν σε άλλες στήλες
' ενώ οι αριθμοί έχουν δεξιά στοίχιση
Τύπωσε $(0),.66, -.45
}
}

```

```

\\ Εδώ ελέγχουμε την εξαγωγή της μεταβλητής ΠΕΔΙΟ
\\ αντέγραψε αυτό σε ένα τμήμα
\\ αν θέλουμε να ξεχωρίζουμε το enter από το κάτω βελάκι πρέπει να δώσουμε αυτό:
ΣΗΜ : ΘΕΣΕ ΔΙΑΚΟΠΤΕΣ "+INP"
ΟΘΟΝΗ
Α$=ΠΕΔΙΟ$("ΑΛΦΑ",10)
ΠΕΔΙΟ 10,10 ΩΣ Α$
ΤΥΠΩΣΕ ΠΕΔΙΟ
ΤΥΠΩΣΕ Α$

```

Αναγνωριστικό: ΔΟΚΙΜΗ [TEST]

1) Η εντολή ΔΟΚΙΜΗ εμφανίζει μια φόρμα όπου βλέπουμε μια προς μια τις εντολές που εκτελούνται.

Μπορούμε να τη δώσουμε και μέσα σε κώδικα

2) Δοκιμή όνομα_τμήματος (τρέχει το τμήμα με φόρμα ελέγχου)

3) Η απενεργοποίηση γίνεται με το κλείσιμο της φόρμας η με την εντολή:

Δοκιμή !

4) Δοκιμή "Όνομα στο τίτλο της φόρμας ελέγχου"

5) Όπως η 4 αλλά μπορούμε να βάλουμε μια σειρά παραμέτρων που θα φαίνονται στην Τύπωσε της φόρμας

\\ Μια πιο νέα χρήση με τη χρήση της συνάρτησης δοκιμή()

\\ απλά πατάμε το Enter και βλέπουμε το κώδικα να τρέχει μέχρι το επόμενο σημείο

\\ όπου πάλι σταματάει η ροή.

Δοκιμή "Αρχή", Δοκιμή("ok"), α

Για α=1 έως 100 {

Αναμονή 1

Τύπωσε α

Αν α=41 Τότε Δοκιμή "ok" : Δοκιμή Δοκιμή("ok2")

Αν α=81 Τότε Δοκιμή "ok2": Δοκιμή

}

Δοκιμή !

Αναγνωριστικό: ΕΚΔΟΣΗ [VERSION]

ΕΚΔΟΣΗ

Τυπώνει τον αριθμό έκδοσης της γλώσσας (υπάρχει και η μεταβλητή ΕΚΔΟΣΗ για έλεγχο μέσα από το πρόγραμμα.

Αναγνωριστικό: ΕΛΕΓΧΟΣ [MONITOR]

ΕΛΕΓΧΟΣ στη γραμμή εντολών

ή ΘΕΣΕ ΕΛΕΓΧΟΣ σε πρόγραμμα

Εμφανίζει πληροφορίες σχετικές με τον τρέχον διερμηνευτή

Αναγνωριστικό: ΚΑΘΑΡΟ [CLEAR]

ΚΑΘΑΡΟ

Καθαρίζει όλες τις μεταβλητές αλλά όχι τα φορτωμένα τμήματα του προγράμματός μας.

Μέσα σε τμήμα καθαρίζει τις στατικές από το τμήμα αυτό και σε κάθε άλλο που καλέστηκε από αυτό.

δες παράδειγμα στα αγγλικά

Παράδειγμα που δείχνει πως εκκινούμε τη Διαγραφή σε ένα αντικείμενο

```

κλάση Αλφα {
    Διαγραφή {
        Τύπωσε "οκ"
    }
}
κ->Αλφα()
\\ μπορούμε να εκκινήσουμε τη διαγραφή
\\ αν δεν υπάρχει άλλος δείκτης στο αντικείμενο
Καθαρό κ
\\ τώρα βάζουμε άλλο αντικείμενο
κ->Αλφα()
\\ Το τμήμα Διαγραφή θα ξεκινήσει αυτόματα
\\ μόλις το κ διαγραφεί στο τέλος του τμήματος
\\ αν δεν υπάρχει άλλος δείκτης να δείχνει στο
\\ ίδιο αντικείμενο

```

Αναγνωριστικό: ΚΛΕΙΔΙ [FKEY]
κλειδιά από F1 =1 έως F12=12 και shift+F1 =13 έως shift+ F12=24

```

1)
ΚΛΕΙΔΙ ΚΑΘΑΡΟ
  Σβήνουμε όλα τα κλειδιά
2)
ΚΛΕΙΔΙ 1, εντολές_σε_αλφαριθμητικο$
ΚΛΕΙΔΙ 1 {εντολές που χρωματίζονται στον διορθωτή }
3)
ΚΛΕΙΔΙ 1
  Μας δίνει την περιγραφή του
4)
ΚΛΕΙΔΙ
  Μας δίνει όλες τις περιγραφές στα κλειδιά που έχουμε ενεργοποιήσει
5) ΚΛΕΙΔΙ 1,""
ΚΛΕΙΔΙ 1 {}
Διαγράφουμε αυτό το κλειδί

```

Τα κλειδιά διαβάζονται με την ΑΝΑΝΕΩΣΗ, σε περίπτωση που είμαστε στον μεταφραστή γραμμής (δεν τρέχουμε κάποιο τμήμα), στην ΣΑΡΩΣΗ όταν υπάρχει ένας τουλάχιστον στόχος. Ο κώδικας του κλειδιού παρεμβάλλεται και εκτελείται

Προσοχή το κλειδί F1 έχει δυο λειτουργίες ανάλογα με το αν έχει πατηθεί το CTRL. Με το CTRL+F1 ανοίγουμε την άμεση βοήθεια.

Αναγνωριστικό: ΚΟΝΣΟΛΑ [DOS]

Όταν έχουμε χρησιμοποιήσει την εντολή ΧΡΗΣΤΗΣ αυτή η εντολή επιστρέφει λάθος

1) ΚΟΝΣΟΛΑ [, χιλιοστά δευτερολέπτου καθυστέρηση]

Άνοιγμα παραθύρου DOS

2) ΚΟΝΣΟΛΑ "ΕΝΤΟΛΗ" [, χιλιοστά δευτερολέπτου καθυστέρηση]

Άνοιγμα παραθύρου DOS, εκτέλεση, κράτημα (περιμένει να γράψουμε exit)

3) ΚΟΝΣΟΛΑ "ΕΝΤΟΛΗ" [, χιλιοστά δευτερολέπτου καθυστέρηση];

Εκτέλεση εντολής χωρίς άνοιγμα παραθύρου.

Αναγνωριστικό: ΛΙΣΤΑ [LIST]

1) ΛΙΣΤΑ ΧΡΗΣΤΩΝ

εμφανίζει λίστα χρηστών (δες ΧΡΗΣΤΗΣ)

2) ΛΙΣΤΑ

Εμφάνιση μεταβλητών και πινάκων. Οι μεταβλητές φαίνονται με τα περιεχόμενά τους ενώ οι πίνακες με τον αριθμό στοιχείων τους. Αν μια αλφαριθμητική μεταβλητή περιέχει εικόνα τότε φαίνεται ως περιεχόμενο το cDIB.

Τα ονόματα των μεταβλητών έχουν πρόθεμα το όνομα του τμήματος όπου δημιουργήθηκαν: A.B=5 σημαίνει ότι η μεταβλητή B έχει τιμή 5 και δημιουργήθηκε στο τμήμα A

3) ΛΙΣΤΑ !

δίνει το ίδιο αλλά με αναλογική γραφή

4)ΛΙΣΤΑ κ

εξάγει στο κ αρχείο

ανοίξε κατ\$+"μεταβλητές.txt" για εξαγωγή ως κ

λίστα κ

κλείσε κ

εγγραφο κκκ\$

φορτωσε.εγγραφο κκκ\$, κατ\$+"μεταβλητές.txt",1

διορθωσε κκκ\$

5) Λιστα com

6) Λιστα com στο ΑλφαΤμήμα

εξάγει από ένα αντικείμενο τους ορισμούς των αντικειμένων

και αν δώσουμε το δεύτερο θα πάρουμε την εξαγωγή σε τμήμα στην αρχή (με Όρισε εντολές)

Υπάρχει η Επίλεξε.αντικείμενο η οποία απλά φέρνει στο μενού επιλογών όλα τα ονόματα.

(πιθανόν να αλλαχθεί αυτή η εντολή)

Αναγνωριστικό: NEO [NEW]

NEO

Διαγράφει όλα τα φορτωμένα τμήματα ώστε να φορτώσουμε ή να συγγράψουμε νέα τμήματα, νέο πρόγραμμα. Επίσης διαγράφει και τα ΚΛΕΙΔΙΑ αν έχουμε ορίσει τέτοια (Δες ΚΛΕΙΔΙ)

Παρατήρηση

Δεν σβήνει τυχόν μεταβλητές που παράμεναν στην μνήμη (χρειάζεται η εντολή ΚΑΘΑΡΟ)

Αναγνωριστικό: ΟΡΙΟ.ΑΝΑΔΡΟΜΗΣ [RECURSION.LIMIT]

ΟΡΙΟ.ΑΝΑΔΡΟΜΗΣ

1000

Χωρίς παράμετρο δίνει το προηγούμενο νούμερο (δεν σημαίνει 1000 κλήσεις)

με παράμετρο 0 δεν υπάρχει πια όριο!

με κάθε άλλο υπάρχει περίπου ο αριθμός κλήσεων που βάζουμε.

Αυτό το όριο είναι για την κλήση ρουτίνας.

εξ ορισμού οι ρουτίνες έχουν όριο 10000 κλήσεις (βάθος κλήσεων), αλλά μπορούμε να το αλλάξουμε πχ 50000 αρκεί να μην έχουμε μπλοκ { } μεταξύ των κλήσεων. Τα μπλοκ κάνουν χρήση του στοίβας του συστήματος.

Οι συναρτήσεις έχουν όριο τις 3375 κλήσεις (βάθος κλήσεων)

π.χ

```
FUNCTION F {  
  READ FF  
  IF FF=0 THEN {  
    =1  
  } ELSE {  
    =F(FF-1)*FF  
  }  
}
```

αν ξεχάσουμε το -1 στο = τότε θα γίνει η μια κλήση μετά την άλλη χωρίς τελειωμό.

Λάθος =F(FF)*FF
Σωστό =F(FF-1)*FF

Τα τμήματα δεν έχουν λειτουργία αναδρομής.

Αναγνωριστικό: ΠΕΡΙ [ABOUT]

1) Παράθυρο βοήθειας

ΠΕΡΙ τίτλος\$, περιγραφη\$

2)

ΠΕΡΙ τίτλος, πλατος_σε_twips, περιγραφη\$

3)

ΠΕΡΙ τίτλος, πλατος_σε_twips, υψος_σε_twips, περιγραφη\$

4)

ΠΕΡΙ

Κλείνει το παράθυρο (το οποίο κλείνει και με δεξί κλικ στο ποντίκι)

Μπορούμε να μετακινήσουμε το παράθυρο με το αριστερό πλήκτρο του ποντικιού πατημένο.

Αυτή η εντολή χρησιμοποιείται για να προκαλέσουμε την προσοχή του χρήστη σε ένα ιδιαίτερο μήνυμα που περιγράφει κάποιον χειρισμό.

Μπορούμε να έχουμε βοήθεια και με το control F1

{ }

5)

ΠΕΡΙ ! τίτλος, πλατος_σε_twips, υψος_σε_twips, περιγραφη\$

Δεν θα ανοίξει αμέσως η φόρμα αλλά όταν πατήσουμε control+F1

6)

ΠΕΡΙ ! ""

Σβήνουμε την βοήθεια

7) ΠΕΡΙ ΔΕΙΞΕ "κάτι" ή Περί Δείξε αν έχει προετοιμαστεί η φόρμα βοήθειας.

Με την συνάρτηση Περί Κάλεσε { } φτιάχνουμε μια Επίλεξε Με Περί\$Τέλος Επιλογής, όπου σε κάθε Με "θέμα" ακολουθεί μια Περί με τίτλο και Περιγραφή, όπου στο θέμα υπάρχουν [] με θέματα με πολλές λέξεις ή μία, και αυτά με κλικ οδηγούνται μέσω πάλι της Περί\$ στην συνάρτησης Περί Κάλεσε { } και έτσι έχουμε μια δομή υπερκειμένου με πολλές σελίδες!

Αν το χρησιμοποιήσουμε στη κονσόλα, με εντολές εκτός στοιχείων ελέγχων πχ στην Εισαγωγή, θα προετοιμάσουμε τη σελίδα που θέλουμε με την Περί !

Αν το χρησιμοποιήσουμε σε φόρμα (και η φόρμα και τα στοιχεία ελέγχου έχουν το γεγονός About), μόλις δώσουμε το Ctrl+F1 θα κληθεί το γεγονός About και αν θέλουμε βάζουμε μια Περί όπως τα 1 έως 4, ή έχουμε δώσει πιο πριν την Περί Κάλεσε { } και απλά επιλέγουμε ένα από τα θέματα με την Περί Δείξε "θέμα". Ο κώδικας της Περί Κάλεσε αποθηκεύεται ξεχωριστά και ισχύει μέχρι την επόμενη αλλαγή, με νέα Περί Κάλεσε, ή με την καθαρίζουμε με την Περί ! "" . Αν

είμαστε στην κονσόλα, δηλαδή δεν τρέχει κάποιο τμήμα, με την εντολή Βοήθεια, βγάζουμε εκτός την συνάρτηση αυτή και βλέπουμε τη βοήθεια της M2000. Το ίδιο συμβαίνει αν ανοίξουμε το διορθωτή για τμήμα, θα γυρίσει η βοήθεια σε αυτήν του διερμηνευτή για τις εντολές του, και γενικότερα τα αναγνωριστικά του.

Αναγνωριστικό: ΠΛΗΚΤΡΟΛΟΓΙΟ [KEYBOARD]

Πληκτρολόγιο Φόρτωσε ' φορτώνει/επιλέγει το ελληνικό πληκτρολόγιο

Πληκτρολόγιο Φόρτωσε "00000408" ' φορτώνει/επιλέγει το 00000408 Πληκτρολόγιο που είναι το ελληνικό

Πληκτρολόγιο Φόρτωσε "00000410" ' Ιταλικό

Πληκτρολόγιο Πέτα "00000410"

Πληκτρολόγιο ! 'ανοίγει ένα πληκτρολόγιο οθόνης

Πληκτρολόγιο "? 1+2", 13

στέλνει τους χαρακτήρες στο Πληκτρολόγιο

Πληκτρολόγιο 65, 66, 67, "φ", 255

μπορεί να στέλνει ASCII κώδικα ή χαρακτήρες

Αυτά που στέλνει τα διαβάζει η ίδια εφαρμογή όταν έρθει η σειρά για ΕΙΣΑΓΩΓΗ ή για ΠΕΔΙΟ

Στο ΠΕΔΙΟ βάζουμε τιμή πριν την καλέσουμε, οπότε η χρησιμότητα εδώ είναι στην εισαγωγή!

Μπορούμε να δούμε τι πληκτρολόγια σχετίζονται με το τοπικό (ο αριθμός που δηλώνει γλώσσα)
? Τοπικό\$(94)

Με την εντολή τοπικό αλλάζουμε τη γλώσσα. Μπορεί να αλλάξει και με τη λέξη Ελληνικά η οποία βάζει και το ελληνικό πληκτρολόγιο

Τοπικό 1032

? Τοπικό

Πληκτρολόγιο "1212"

εισαγωγή "Τιμή X=";X

τυπωσε "X=";X

Αναγνωριστικό: ΠΡΟΧΕΙΡΟ [CLIPBOARD]

ΠΡΟΧΕΙΡΟ "AAAAAAAAA"

ΤΥΠΩΣΕ ΠΡΟΧΕΙΡΟ\$

Η πρώτη εντολή βάζει στο πρόχειρο το αλφαριθμητικό και η δεύτερη το παίρνει. Θα μπορούσε κανείς να βάλει εδώ ένα αλφαριθμητικό με δομή σωρού. Δες Σωρος
Επίσης αν βάλουμε αλφαριθμητικό με εικόνα τότε η εντολή στέλνει την εικόνα στο πρόχειρο

Η προχειρο\$ δεν παίρνει εικόνα δες την προχειρο.εικονα\$

Πρόχειρο A ' στέλνει την τιμή το A ως αλφαριθμητικό στο πρόχειρο. Αν το A έχει εικόνα τότε το στέλνει στο πρόχειρο

Πρόχειρο A ως "bmp" 'στέλνει το A ως εικόνα (αν είναι εικόνα bitmap ή metafile)

Πρόχειρο A ως "emf" 'στέλνει το A ως εικόνα metafile (αν είναι εικόνα metafile)

Διαβάζουμε εικόνες με τα Πρόχειρο.Εικόνα και Πρόχειρο.Σχέδιο (για metafile)

Αναγνωριστικό: ΠΡΩΤΟΤΥΠΟ [PROTOTYPE]

Πρωτότυπο {

Τύπωσε "ok"

} ως Αλφα\$ ' δούλεύει και χωρί; \$

Τύπωσε Μήκος(Αλφα\$)

Ένθεση Αλφα\$

Αναγνωριστικό: ΣΕΝΑΠΙΟ [SCRIPT]

μόνο από τον μεταφραστή γραμμής εντολών

Σενάριο αλφαριθμητική_παράσταση ' τρέχει τις εντολές στη παράσταση στον μεταφραστή γραμμής εντολών

(ο μεταφραστής δεν υποστηρίζει επαναλήψεις)

Σενάριο όνομα_τμήματος ' πρέπει να είναι γενικό το τμήμα

Το τμήμα ότι φτιάχνει παραμένει!

Μέσα από το μεταφραστή προγράμματος μπορούμε να καλέσουμε ένα σενάριο με την Θέσε

Θέσε Σενάριο όνομα_τμήματος

\\ Παράδειγμα

\\ Τοπικά φτιαγμένα σενάρια, ως γενικά!

\\ οι μεταβλητές/συναρτήσεις/τμήματα

\\ παραμένουν μετά τη κλήση (δεν γίνεται αυτό στα κανονικά τμήματα)

Τμήμα Γενικό [001] {

Σταθερές N=5, Z=50

Συνάρτηση Για\$ {

= "Για χαρά"

}

```

}
Τμήμα Γενικό [002] {
    \ \ ότι φτιάχνουμε θα είναι γενικό
    \ \ και προσωρινό για το τμήμα που κάλεσε το σενάριο
    Σταθερές N=55, Z=550
    Συνάρτηση Γεια$ {
        ="Καλημέρα"
    }
}
Θέσε σενάριο [001]
Αρχη()
Καθαρό
Βάλε "[002]"
\ \ διαβάζουμε με την γράμμα$ γιατί στη υπόλοιπη γραμμή της θέσε...
\ \ δεν διαβάζονται οι μεταβλητές του τμήματος, μόνο ο σωρός είναι κοινός.
Θέσε σενάριο "σενάριο "+γράμμα$
Αρχη()

```

```

Ρουτίνα Αρχη()
    Τύπωσε N, Z, Γεια$()
Τέλος Ρουτίνας

```

```

Αναγνωριστικό: ΣΗΜ    [REM]
ΣΗΜ : ΤΥΠΩΣΕ "ΑΥΤΗ Η ΓΡΑΜΜΗ ΔΕΝ ΘΑ ΕΚΕΛΕΣΤΕΙ ΑΛΛΑ ΘΑ ΕΧΕΙ ΧΡΩΜΑ"

ΣΗΜ {
    ΤΥΠΩΣΕ "ΑΥΤΟ ΤΟ ΜΠΛΟΚ ΔΕΝ ΘΑ ΕΚΤΕΛΕΣΤΕΙ, ΚΑΙ ΔΕΝ ΕΧΕΙ ΧΡΩΜΑ"
    \ \ ΕΙΝΑΙ ΜΙΑ ΣΗΜΕΙΩΣΗ ΠΟΛΛΩΝ ΓΡΑΜΜΩΝ
}

```

```

Αναγνωριστικό: ΣΥΓΓΡΑΦΕΑΣ    [WRITER]
ΣΥΓΓΡΑΦΕΑΣ
Τυπώνει το όνομα του συγγραφέα της γλώσσας M2000

```

Αναγνωριστικό: ΣΥΓΓΡΑΦΗ [EDIT]

ΣΥΓΓΡΑΦΗ ή Σ

1) Δημιουργία Τμήματος

Σ αλφα

Αν δεν γράψουμε κάτι στο τμήμα τότε δεν δημιουργείται τμήμα

Σ αλφα, 50

Μας μεταφέρει τον δρομέα κειμένου στην θέση 50

2) Σ "αλφα.txt"

Επεξεργασία αρχείου κειμένου απ' ευθείας από τον δίσκο. Πρέπει να έχει κατάληξη gsb ή txt.

3) Δημιουργία συνάρτησης (καθολικής) Σ α(ή Σ α\$(ή Σ α%(ή όπου Σ βάλε ΣΥΓΓΡΑΦΗ

4) Σ

Ανοίγει την λίστα γραμμών εντολών (από την οποία με τα βελάκια επιλέγουμε το προηγούμενο, όταν είμαστε στην γραμμή εντολών). Δουλεύει μόνο στη γραμμή εντολών (Ούτε με ΘΕΣΕ δεν μπορούμε να την εκτελέσουμε μέσα από τμήμα), γιατί ελέγχει αν είναι σε αυτό το σημείο και μετά ανοίγει την λίστα. Δεν δίνει λάθος αν την έχουμε μέσα σε τμήματα.

5) Σ ! 4

θέτει στους 4 χαρακτήρες την εσοχή, και το πλάτος του TAB χαρακτήρα μόνο για την διόρθωση

Παρατήρηση

Όταν ανοίγουμε τον διορθωτή (Editor) καμία αλλαγή δεν γίνεται στην οθόνη. Το πλαίσιο κειμένου βρίσκεται σε ξεχωριστό επίπεδο. Υπάρχει κάθετη μπάρα ολίσθησης όταν οι γραμμές είναι περισσότερες, αλλά συνήθως είναι κρυμμένη και για να εμφανιστεί πρέπει να πάμε στην άκρη δεξιά το ποντίκι. Μπορούμε όμως να την εμφανίσουμε όταν ολισθαίνουμε τις γραμμές χωρίς αυτήν. Αυτό γίνεται με το κράτημα του αριστερού πλήκτρου του ποντικιού και την μετακίνηση προς τα εκεί που θέλουμε. Αν δεν έχουμε αναδίπλωση λέξεων ενεργή τότε οι χαρακτήρες εκτός οθόνης εμφανίζονται καθώς πάμε τον δρομέα. Με το ποντίκι μπορούμε να μεταφέρουμε το δείκτη και αυτόματα θα έχουμε οριζόντια ολίσθηση. Δεν υπάρχει για το λόγο αυτό οριζόντια μπάρα ολίσθησης.

συντομεύσεις που υπάρχουν και στο αναδυόμενο μενού

ESC - τερματίζουμε τον διορθωτή.

shift ESC - τερματίζουμε χωρίς αλλαγές.

F1 - ενεργοποιεί την αναδίπλωση λέξεων ή την απενεργοποιεί ανάλογα.

F2 - αναζητεί προς τα πάνω το μαρκαρισμένο κείμενο και αν το βρει το μεταβαίνει εκεί

F3 - αναζητεί προς τα κάτω το μαρκαρισμένο κείμενο και αν το βρεί μεταβαίνει εκεί

ctrl + F1 ανοίγει την βοήθεια (επειδή το ίδιο στοιχείο χρησιμοποιείται και για εισαγωγή κειμένου σε μεταβλητή, σε εκείνη την περίπτωση η βοήθεια είναι αυτή που φτιάχτηκε με την ΠΕΡΙ)

(δεν χρειάζεται να είναι στα αγγλικά το πληκτρολόγιο για να κάνουμε τους συνδυασμούς)

ctrl + X αποκοπή
ctrl + C αντιγραφή
ctrl + V επικόλληση

Άλλες συντομεύσεις που δεν φαίνονται στο αναδυόμενο μενού

shift + βελάκι - μαρκάρουμε το κείμενο

ctrl + A

- Επιλογή όλων

shift ctrl + A

- Επιλογή κανένα

ctrl + Z

- Αναίρεση (υπάρχουν απεριόριστα επίπεδα αναίρεσης)

ctrl + Y

- Αναίρεση αναίρεσης (καταγράφονται τα ctrl+z μέχρι να γράψουμε κάτι καινούργιο)

Προσοχή το Break ("Pause/Break") πλήκτρο είναι το "Reset" του μεταφραστή (ευτυχώς βγαίνει μήνυμα)

Περί "μεταφοράς και απόθεσης" ή "σύρε και άφησε". Το να αφήσουμε από άλλο στοιχείο κάποιο κείμενο είναι αποδεκτό πάντα. Το να σύρουμε όμως είναι απενεργοποιημένο και πρέπει να ενεργοποιηθεί από το αναδυόμενο μενού. Η χρήση του είναι πιο απλή εδώ. Στο μαρκαρισμένο κείμενο κάνουμε διπλό κλικ μέχρι που εμφανίζεται το βελάκι με το τετράγωνο. Αν θέλουμε μπορούμε χωρίς να πατάμε κανένα πλήκτρο να μετακινήσουμε τον δρομέα με το ποντίκι ακόμα και στις γραμμές που δεν βλέπουμε (θα τις δούμε όπως κάνουμε την μετακίνηση). και απλά κάνουμε ένα κλικ και γίνεται η μεταφορά. Για να σύρουμε εκτός αυτού του στοιχείου σε άλλο πρόγραμμα πρέπει να κρατάμε κάτω το αριστερό πλήκτρο του ποντικιού. Σε κάθε περίπτωση το ESC αναιρεί τη κατάσταση της μεταφοράς και απόθεσης.

Αν θέλουμε να περάσουμε ένα κείμενο σε μια μεταβλητή τότε δες ΕΙΣΑΓΩΓΗ

Αναγνωριστικό: ΣΥΣΤΗΜΑ [WIN]

ανοίγουμε την αριθμομηχανή των Windows

ΣΥΣΤΗΜΑ "CALC"

ΣΥΣΤΗΜΑ "VIBER:" ' (ανοίγει την εφαρμογή Viber που έχουμε κατεβάσει με το Microsofy Store)

ανοίγουμε τον κατάλογο που ο διερμηνευτής σώνει και φορτώνει τμήματα
ΣΥΣΤΗΜΑ ΚΑΤ\$

ανοίγουμε τον επεξεργαστή κειμένου Wordpad
ΣΥΣΤΗΜΑ "WORDPAD"

ΣΥΣΤΗΜΑ "WORDPAD", "μονοπάτι και όνομα αρχείου χωρίς διαστήματα ή άλλες εντολές"
ΣΥΣΤΗΜΑ "WORDPAD", ΠΑΡΑΘΕΣΗ\$("μονοπάτι και όνομα αρχείου με διαστήματα εδώ")

Με την εντολή ΣΥΣΤΗΜΑ ή ΣΥΣ τρέχουμε μια εξωτερική εφαρμογή.

Αν θέλεις να τρέξεις ένα άλλο πρόγραμμα gsb (πρόγραμμα του διερμηνευτή) παράλληλα με το τρέχον πρόγραμμα δες την ΧΡΗΣΗ όπου μας δίνει τρόπο να επικοινωνήσουμε με αυλό δεδομένων.

Δες ΑΥΛΟΣ

Αναγνωριστικό: ΣΩΣΕ [SAVE]

Από την έκδοση 9 σώνει τα επιπέδου 0 (γενικά) τμήματα και συναρτήσεις ως γενικά. Εκτός και αν χρησιμοποιήσουμε το Σώσε Τοπικά

Από την έκδοση 7 σώνει σε utf-8, ενώ διατηρεί την παλιά κωδικοποίηση στα παλιά αρχεία.

Επίσης μετονομάζει το παλιό αρχείο πριν γράψει το νέο με το ίδιο όνομα, με αλλαγή κατάληξης bck

1) Απλό σώσιμο όλων των τμημάτων στον δίσκο

ΣΩΣΕ ΤΟΠΙΚΑ ονομα

ΣΩΣΕ ονομα

ΣΩΣΕ "ονομα με διαστήματα"

στον τρέχον κατάλογο σώνουμε το ονομα.gsb

2) Όπως η (1) αλλά να τρέχει απ' ευθείας ένα τμήμα αμέσως μετά την φόρτωση

ΣΩΣΕ ονομα, τμημα1

3) Όπως τα (1) και (2) αλλά κωδικοποιημένα με το βασικό κλειδί, έτσι ώστε να μην εμφανίζει τα περιεχόμενα των τμημάτων.

ΣΩΣΕ ονομα @, τμημα1

ΣΩΣΕ ονομα @

4) Όπως το (1) αλλά κωδικοποιημένο με δικό μας κωδικό

ΣΩΣΕ "ονομα" @@ "ΚΩΔΙΚΟΣΜΟΥ"

Ακόμα και να φορτώσουμε ένα τέτοιο τμήμα αυτό δεν αναγνωρίζεται εάν στην εντολή φόρτωσης δεν δώσουμε τον κατάλληλο κωδικό ΚΩΔΙΚΟΣΜΟΥ

Αναγνωριστικό: ΤΑΞΙΝΟΜΗΣΗ [SORT]

Για έγγραφα και καταστάσεις και πίνακες

ΤΑΞΙΝΟΜΗΣΗ [φθίνουσα] έγγραφο\$ [, αρχή, τέλος [, θέση για αρχή σύγκρισης]]

Στα έγγραφα ο εξ ορισμού τρόπος σύγκρισης είναι η σύγκριση ως κείμενο.

Μπορεί να θέλουμε όμως μια ειδική περίπτωση όπου έχει σημασία η γλώσσα στη ταξινόμηση, και ο στανταρ τρόπος να μην είναι αυτό που θέλουμε!

Μπορούμε να επέμβουμε με εντολή Μέθοδο (το έγγραφο είναι αντικείμενο)

1)

\ \ χρησιμοποίησε κάποιο πλήκτρο γιατί η Αναφορά σταματάει το κείμενο για να διαβαστεί!

Έγγραφο άλφα\$={ένα

δύο

τρία

τέσσερα

πέντε

έξι

επτά

οκτώ

εννιά

δέκα

}

Ταξινόμηση άλφα\$, 1, 10

Αναφορά άλφα\$

Τύπωσε "*****"

Ταξινόμηση φθίνουσα άλφα\$, 1, 10

Αναφορά άλφα\$

Τύπωσε "*****"

Ταξινόμηση άλφα\$, 1, 5

Αναφορά άλφα\$

Εδώ αλλάζουμε το τρόπο σύγκρισης. Υπάρχουν τέσσερις επιλογές SetTextCompare (default), SetBinaryCompare, SetDatabaseCompare και η τελευταία που παρουσιάζεται εδώ και είναι η μόνη που δέχεται προεραϊτικά τιμή (αν δεν δώσουμε θα πάρει την διαθέσιμη τιμή τοπικού αναγνωριστικού (πχ το έχουμε δώσει στη φόρτωση από το αρχείο).)

Έγγραφο α\$={péché

sin

peach

pêche

}

Τύπωσε "Με 1036 Γαλλικά"

Μέθοδος α\$, "SetLocaleCompare", 1036

Ταξινόμηση Αύξουσα α\$, 1,4
Αναφορά α\$
Τύπωσε "Με 1033 Αγγλικά"
Μέθοδος α\$, "SetLocaleCompare", 1033
Ταξινόμηση Αύξουσα α\$, 1,4
Αναφορά α\$

2)
Έγγραφο α\$={1234βήτα
4567άλφα
1212ζήτα
9921έψιλον
}

Ταξινόμηση φθίνουσα α\$, 1,4, 5
Αναφορά α\$

3)
Ταξινόμηση Άυξουσα inventory1 ως κείμενο
Ταξινόμηση Φθίνουσα inventory1 ως κείμενο
Ταξινόμηση Άυξουσα inventory1 ως κείμενο, τρόπος
Ταξινόμηση Φθίνουσα inventory1 ως κείμενο, τρόπος
Τρόπος:
0 Σύγκριση Δυαδική
1 Σύγκριση Κειμένου
2 Σύγκριση βάσεων δεδομένων
αριθμός τοπικού (πχ 1032 είναι τα ελληνικά)

Ταξινόμηση Άυξουσα inventory1 ως αριθμός
Ταξινόμηση Φθίνουσα inventory1 ως αριθμός
Ταξινόμηση inventory1 ως κείμενο, 1
Ταξινόμηση inventory1 ως κείμενο, 0
Ταξινόμηση Inventory1 ως αριθμός, 1
Ταξινόμηση inventory1 ως αριθμός, 0

Κατάσταση α="sin","pêche","péché","peach"
Ταξινόμηση α ως κείμενο, 1
Τύπωσε α, "κανονική αύξουσα"
Ταξινόμηση α ως κείμενο, 0
Τύπωσε α, "κανονική φθίνουσα"
Ταξινόμηση αύξουσα α ως κείμενο, 1036
Τύπωσε α, 1036

Ταξινόμηση αύξουσα α ως κείμενο, 1033
Τύπωσε α, 1033

Αναγνωριστικό: ΤΕΛΟΣ [END]

ΤΕΛΟΣ

Τερματίζει την λειτουργία του μεταφραστή.

Παρατήρηση

Η εντολή ΤΕΛΟΣ είναι κατανοητή μόνο από τον μεταφραστή γραμμής και όχι από τον μεταφραστή τμημάτων. Για να εκτελέσουμε όμως την εντολή αυτή μέσα από ένα τμήμα τότε πρέπει έμμεσα να στείλουμε την εντολή αυτή με την εντολή ΘΕΣΕ

ΘΕΣΕ ΤΕΛΟΣ

Αναγνωριστικό: ΦΟΡΤΩΣΕ [LOAD]

Από την έκδοση 9, αποθηκεύονται τα φορτώματα, και όταν ζητάμε ξανά να φορτώσουν διαβάζονται από τα αποθηκευμένα, μπορούμε να αναιρέσουμε το διάβασμα των αποθηκευμένων με την Φόρτωσε Νέο

(και Νέα παίρνει)

Με Φόρτωσε Τμήματα θα φορτώσει χωρίς να τρέξει εντολές για τμήματα (θα τις πετάξει)

Υπάρχει και ο συνδυασμός Φόρτωσε Νέα Τμήματα

Από την έκδοση 7 μπορούμε να ανοίξουμε UTF-8, UTF-16 και ANSI αρχεία (τα gsb αρχεία της M2000 είναι αρχεία κειμένου, εκτός από τα κωδικοποιημένα)

1) Φόρτωση τμημάτων

ΦΟΡΤΩΣΕ "ονομα" [&& "αλλο ονομα"

ΦΟΡΤΩΣΕ ονομα && ονομα3 && "άλλο ονομα"

Αν είχαμε σώσει και άμεσης εκτέλεσης εντολή τότε αυτή θα ενεργοποιηθεί

Αν είχαμε σώσει κωδικοποιημένα τα τμήματα του αρχείου "ονομα", με το βασικό κλειδί τότε δεν θα μπορούμε να βλέπουμε τον κώδικα αυτών

2) Φόρτωση κωδικοποιημένων τμημάτων με δικό μας κωδικό

ΦΟΡΤΩΣΕ "ονομα", "ΚΩΔΙΚΟΣΜΟΥ"

ΦΟΡΤΩΣΕ ονομα, "ΚΩΔΙΚΟΣΜΟΥ"

Παρατήρηση.

Μπορούμε να κάνουμε φόρτωση τμημάτων μέσα από άλλα τμήματα. Όταν το τμήμα που κάλεσε

την ΦΟΡΤΩΣΕ τερματίσει τότε διαγράφονται και τα φορτωμένα τμήματα της φόρτωσης.
Θα μπορούσε κανείς να έχει δυο ή περισσότερες παραλλαγές ενός τμήματος και να επιλέγει ανάλογα να φορτώνει αυτή που χρειάζεται τη δεδομένη στιγμή.
Επίσης κάποια τμήματα θα μπορούσαν να είχαν μόνο εντολές καταχώρισης τιμών στο σωρό και με την φόρτωσή τους να μας παρείχαν αυτές τις τιμές.

Μπορούμε να φορτώσουμε κώδικα στα ελληνικά αν έχει στην αρχή το ΚΛΑΣΗ ή ΤΜΗΜΑ ή ΣΥΝΑΡΤΗΣΗ

Ομάδα: ΧΕΙΡΙΣΜΟΣ ΤΜΗΜΑΤΩΝ - MODULE COMMANDS

Αναγνωριστικό: ΑΛΤ [HALT]

Μπορούμε οπουδήποτε στο πρόγραμμα να δώσουμε το ΑΛΤ

Θα μας βγάλει στη κονσόλα, και μπορούμε να συνεχίσουμε με το Συνέχισε

Στη κονσόλα με το Λίστα βλέπουμε όλες τις μεταβλητές και μπορούμε να τις αλλάξουμε. Τις βλέπουμε με το πρόθεμά τους (αν είναι τοπικές)

Αναγνωριστικό: ΑΥΛΟΣ [PIPE]

ΑΥΛΟΣ ΟΝΟΜΑΑΥΛΟΥ\$, ΤΙΜΗ1, ΤΙΜΗ2\$, ΤΙΜΗ3

PIPE PIPENAME, DATATOSEND1, DATATOSEND2\$, DATATOSENDN....

MODULE B {WINDOW 12, 6000,4000

FORM 32, 25

FORM

SHOW

MOTION.W 500,2000

PRINT "THIS IS THE BASE"

' θα χρησιμοποιήσω το TRG.GSB σαν πελάτη. Του στέλνω και μερικές πληροφορίες (μια που δεν φαίνεται είναι το όνομα του αυλού για να μου στείλει κάτι

USE TRG.GSB 1,"MESSAGEGETXT",3 TO K\$ AS L

THREAD L INTERVAL 200

C=0

PAR\$=""

NM\$=""

GETOUT=FALSE

EVERY 200 {

```

C=C+1
PRINT C
IF INKEY$="" THEN EXIT
IF C=1000 THEN EXIT
' K$ παίρνει τιμή από το νήμα L και περιέχει
IF K$<>"" THEN {
    STACK K$ ' εδώ ρίχνω τον σωρό K$ στον κύριο σωρό τιμών
    STACK ' εδώ μου δείχνει τι τιμές έχει ο σωρός
    READ NM$, PAR$
    FLUSH ' εδώ καθαρίζω τον σωρό
    IF PAR$="N" THEN GETOUT =TRUE
    C=0
}
IF GETOUT THEN EXIT
}
PRINT "FINISH"
}

```

Παρακάτω είναι το πρόγραμμα TRG.GSB ή ο πελάτης.

Απλή περιγραφή:

Τρέχει το C και βγαίνει στην γραμμή εντολών όπου μπορούμε να γράψουμε FEED (να ταΐσουμε...δηλαδή να επιστρέψουμε τιμές στο αρχικό πρόγραμμα (την βάση, ή τον εξυπηρετητή). Δείτε ότι κάνει έλεγχο δοκιμής σφάλματος. Η επικοινωνία είναι ασύγχρονη. Εμείς στέλνουμε αλλά το πρόγραμμα δεν λαμβάνει πάντα...και ορισμένες φορές απορρίπτει τα δεδομένα. Κάποια στιγμή όμως θα λάβει. Όμως μπορούμε να χρησιμοποιήσουμε μια μεταβλητή που θα μετράει τον χρόνο, ώστε σε ένα μέγιστο χρονικό διάστημα να αποφασίσουμε πως το άλλο πρόγραμμα δεν μπορεί να λάβει τα δεδομένα!

Αν γράψουμε TERM τότε στέλνουμε ένα μήνυμα τέλους στον εξυπηρετητή

Αυτό που δεν έχω γράψει εδώ (είναι άσκηση) είναι το πως ο εξυπηρετητής θα ξαναστείλει και άλλα δεδομένα πέρα από τα αρχικά. Η λύση είναι να δημιουργήσουμε στον πελάτη έναν αυλό και να στείλουμε το αναγνωριστικό του στον εξυπηρετητή (αφού αυτό μπορεί να γίνει). Όταν λοιπόν πάρει ο εξυπηρετητής το αναγνωριστικό του αυλού τότε στέλνει και αυτός με σύστημα δοκιμής σφάλματος.

Αν έχουμε να στείλουμε ένα μεγάλο αλφαριθμητικό θα πρέπει να αλλάξουμε στρατηγική. Οι αυλοί δεν μπορούν να δεχτούν μεγάλα αλφαριθμητικά, επειδή το σύστημα δημιουργεί έναν ενδιάμεσο σταθμό καταχώρησης με το μέγιστο που ορίζουμε σε μνήμη και παραμένει διαθέσιμος συνέχεια, που σημαίνει ότι η μνήμη αυτή αφαιρείται από το σύστημα. Έτσι ένας αυλός με μεγάλη καταχώρηση μνήμης θα έφερνε μάλλον δυστυχία στο σύστημα..(θα αργούσε ή θα κόλλαγε).

Αναγνωριστικό: ΔΙΑΚΟΠΗ [STOP]

Ανοίγει ένα παράθυρο διαλόγου και ρωτάει αν θέλουμε προσωρινά να διακόψουμε το

πρόγραμμα (δεν διακόπτει τα νήματα). Αν απαντήσουμε θετικά τότε βγαίνει στην κονσόλα και δείχνει το όνομα του χρήστη και ζητάει εντολές. Οι εντολές είναι σαν να τις δίνουμε στη γραμμή που είναι το πρόγραμμα. Με έξοδο γυρνάμε στην εκτέλεση του προγράμματος. Μπορούμε να δούμε με την Λίστα την λίστα μεταβλητών, και με το Τμήματα ? τα υπάρχοντα τμήματα. Μπορούμε να ανοίξουμε τη βοήθεια ή τον έλεγχο (Δοκιμή). Στον έλεγχο υπάρχει παρόμοιος τρόπος να δώσουμε εντολές, όταν εμφανίζεται το σύμβολο > στην εισαγωγή κειμένου στο κάτω μέρος της φόρμας. Χρησιμοποιούμε το Delete για να αλλάξουμε το σύμβολο εισαγωγής σε > από ?

Απλά γράφουμε τη λέξη Διακοπή όπου θέλουμε να σταματήσει.

ΔΙΑΚΟΠΗ

εμφανίζει

ονομα_χρήστη> Έξοδος (για να επιστρέψουμε στην εκτέλεση)

Η διαφορά με το ALT είναι ότι το τελευταίο βγάζει στον μεταφραστή γραμμής της κονσόλας, οπότε δεν μπορούμε να δώσουμε εντολές εκεί που τρέχει το πρόγραμμα. Μπορούμε όμως να πειράξουμε γενικές μεταβλητές, ή να φορτώσουμε κάποιο πρόγραμμα και να το τρέξουμε

Αναγνωριστικό: ΔΙΑΦΥΓΗ [ESCAPE]

ΔΙΑΦΥΓΗ ΟΧΙ

ΔΙΑΦΥΓΗ ΝΑΙ

Η εντολή αυτή μας επιτρέπει η όχι να μπορούμε να διακόπτουμε την ροή του προγράμματος με το πλήκτρο ESC.

Για καλύτερο έλεγχο της ροής του προγράμματός σας δοκιμάστε την εντολή ΔΟΚΙΜΗ

Όταν γράφουμε κείμενο πολλών γραμμών όπου το Escape χρησιμεύει για να τερματίσουμε τον διορθωτή τότε λειτουργεί παρόλο που δεν θα λειτουργήσει για διακοπή προγράμματος.

Αναγνωριστικό: ΕΝΘΕΣΗ [INLINE]

α\$=Παράθεση\$(1,"αλφαριθμητικό",2,3)

Ένθεση "Σειρά "+α\$ 'βάζει τη γραμμή Σειρά 1, "αλφαριθμητικό", 2, 3

Η ένθεση παρεμβάλλει κώδικα που παίρνει από ένα αλφαριθμητικό και εκτελείται άμεσα. Μπορεί κανείς να βάλει σε έναν πίνακα εναλλακτικές ομάδες εντολών και να εκτελέσει κατά περίπτωση μια. Η θα μπορούσε ένα πρόγραμμα να λάβει από αυλό σειρά εντολών για να εκτελέσει.

```
X=1
Τμήμα Άλφα {
    X++
    Τύπωσε X
}
Ένθεση Κώδικα Άλφα, Άλφα
```

Μπορούμε να ξέρουμε πότε υπάρχει ένα τμήμα ή μια συνάρτηση χωρίς να την καλέσουμε με χρήση της συνάρτησης Τμήμα()

Τύπωσε Τμήμα(Άλφα) \ \ δίνει ένα για τον προηγούμενο κώδικα

Αναγνωριστικό: ΕΝΩΣΕ [\[LINK\]](#)
\ \ ενώνουμε μεταβλητές με άλλες μεταβλητές

```
Ένωσε λίστα_μεταβλητών στο λίστα_μεταβλητών
Ένωσε λίστα_μεταβλητών στη λίστα_μεταβλητών
' νέο από 9.3 αν. 33 για δημιουργία νέας μεταβλητής ακόμα και αν υπάρχει ίδιο όνομα
Ένωσε λίστα_μεταβλητών τοπικά στο λίστα_μεταβλητών
Ένωσε λίστα_μεταβλητών τοπικά στη λίστα_μεταβλητών
```

```
\ \ απλό παράδειγμα
A$="ένα"
Ένωσε A$ στο B$
Τύπωσε B$
```

```
\ \ απλό παράδειγμα χωρίς την ένωση (κάνει εδώ ακριβώς το ίδιο)
A$="ένα"
Βάλε &A$ \ \ βάζει την ισχυρή αναφορά στο σωρό
\ \ δεν μπορούμε έτσι να συνδέσουμε υπάρχουσα B$ θα βγει λάθος
Διάβασε &B$ \ \ διαβάζει την ισχυρή αναφορά και συνδέει με την B$
Τύπωσε B$
```

```
A$="123345"
Ένωσε A$ στο B
Τύπωσε B
B+=200
Τύπωσε A$
```

```
\ \ παράδειγμα με χρήση της ισχυρής αναφοράς και της ένωσης
α$="{=1000*αριθμός+αριθμός}"
```

```

\\ η α$ έχει ισχνή αναφορά συνάρτησης (έχει το μπλοκ εντολών {} )
Ένωσε ισχνη α$ στο αα()
Τύπωσε αα(10,-100)
M=300
β$=ισχνή$(M)
\\ η β$ είναι ισχνή αναφορά στην M
Ένωσε ισχνη β$ στο Δ
Τύπωσε Δ
Δ+=5000
Τύπωσε M, M=Δ, Εκφρ(β$)
\\ μπορούμε με μια τελεία μετά το β$ να την χρησιμοποιούμε άμεσα
β$.+=500
Τύπωσε M, M=Δ, Εκφρ(β$)

```

```

Ομάδα Βήτα {
    μετρητής=0
    Συνάρτηση A1 (χ, ψ) {
        .μετρητής++
        =χ**2+ψ
    }
}
Τμήμα Δες_Αυτό (&φ()) {
    Τύπωσε φ(11,3)
}
Ένωσε Βήτα.A1() στη ζ()
Τύπωσε ζ(10,2), Βήτα.μετρητής
Δες_Αυτό &Βήτα.A1()
Δες_Αυτό Οκν$(Βήτα.A1(αριθμός, 30))
Δες_Αυτό Οκν$(Βήτα.A1(20, 30))
Τύπωσε ζ(12,4), Βήτα.μετρητής

```

Αναγνωριστικό: ΛΑΘΟΣ [ERROR]

1) ΑΜΕΣΗ ΔΙΑΚΟΠΗ ΠΡΟΓΡΑΜΜΑΤΟΣ ΚΑΙ ΜΕΤΑΦΡΑΣΤΗ

ΛΑΘΟΣ 0

2) ΔΙΑΚΟΠΗ ΠΡΟΓΡΑΜΑΤΟΣ ΜΕ ΕΜΦΑΝΙΣΗ ΑΡΙΘΜΟΥ

ΛΑΘΟΣ 100

3) ΔΙΑΚΟΠΗ ΜΕ ΕΜΦΑΝΙΣΗ ΦΡΑΣΗΣ ΚΑΘΩΣ ΚΑΙ ΕΜΦΑΝΙΣΗ ΘΕΣΗΣ ΚΛΗΣΕΩΝ ΤΜΗΜΑΤΩΝ

ΛΑΘΟΣ " πρόβλημα με το"

Μπορούμε να μεταβούμε σε μια θέση με την ΣΥΓΓΡΑΦΗ ή απλά Σ ως εξής

Σ ονομα_τμηματος, θέση

ΑΔΕΙΑΣΕ ΛΑΘΟΣ ' το χρησιμοποιούμε πριν την ΛΑΘΟΣ αν θέλουμε να μην προστεθεί το μήνυμα σε υπάρχον μήνυμα

Αναγνωριστικό: ΝΗΜΑ [THREAD]

Νήμα στη Μ2000 είναι κώδικας σε Τμήμα ή σε Συνάρτηση που χειριζόμαστε βάσει ενός αριθμού (χειριστής), και βλέπουμε ότι το πατρικό έχει, αλλά έχουν δικές τους στατικές μεταβλητές και δικό τους σωρό τιμών.

1) δημιουργία

ΝΗΜΑ {

Στατική Κ=5

.....

} ΩΣ Κ \ \ εδώ μπορούμε να προσθέσουμε το Κάθε

2) χειρισμός

Νήμα Κ Καθε 200

ορίζει και ξεκινάει το νήμα Κ να τρέχει κάθε 200 χιλιοστά του δευτερολέπτου (με ελάχιστο 10 χιλιοστά), αυτό λέγεται σκανδαλισμός

Νήμα Κ Κρατά

σταματάει ο σκανδαλισμός του νήματος

Νήμα Κ Ξεκινά

ξεκινάει ο σκανδαλισμός

Νήμα Κ Σβησε

διαγράφεται το νήμα, ο αριθμός χειρισμού δεν θα ξαναχρησιμοποιηθεί, δεν θα δοθεί το ίδιο νούμερο σε νέο νήμα

Νήμα Κ Εκτέλεση Κ=10

Εκτελεί εντολή (οι εντολές) στο νήμα από άλλο νήμα

Μπορούμε να χρησιμοποιήσουμε αυτές τις εντολές μέσα σε ένα νήμα και να απευθυνθούμε σε αυτό με την ΑΥΤΟ. Τα τμήματα και οι συναρτήσεις δεν έχουν ιδιότητες νήματος και αν απευθυνθούμε στο ΑΥΤΟ θα βγει λάθος.

Νήμα ΑΥΤΟ σβησε

3) Αναφορά στην οθόνη για τα νήματα που έχουν εγγραφεί στη λίστα του διαχειριστή νημάτων.

Νήματα

Το Νήμα είναι ένα τμήμα εντολών το οποίο χειριζόμαστε με έναν αριθμό χειρισμού σε μια μεταβλητή και μπορούμε να ξεκινάμε, να σταματάμε, να ορίζουμε κάθε πότε θα τρέχει, να το σβήσουμε, και όλα αυτά παράλληλα με τις εντολές του τμήματος που τρέχει στο προσκήνιο. Μπορούν να τρέχουν περισσότερα από ένα νήματα

Γενικά μέσα σε συναρτήσεις μπορούμε να βάζουμε τμήματα και νήματα, και το ίδιο συμβαίνει και στα τμήματα και στα νήματα. Οι συναρτήσεις έχουν όνομα όπως και τα τμήματα ενώ τα νήματα έχουν αριθμό χειρισμού (γράφεται σε μια μεταβλητή) και χρησιμοποιούν το όνομα του τμήματος που τα δημιούργησε. Αυτό σημαίνει ότι μπορούν να δουν και να αλλάξουν τις μεταβλητές του τμήματος που τα δημιούργησε.

Μέσα σε ένα νήμα δεν μπορούμε να χρησιμοποιήσουμε τον αριθμό χειρισμού του, γιατί απλά όταν τρέχει το νήμα έχει βγει από μια ομάδα νημάτων και ο αριθμός που θα αναζητείται εκεί δεν θα βρεθεί, για να μην μπορεί να κληθεί δεύτερη φορά ενώ τρέχει ήδη. Για το λόγο αυτό αντί του αριθμού θα βάλουμε την λέξη Αυτό ή This
Νημα Αυτό κάθε 500 ' αλλάζουμε τον σκανδαλισμό του (το πότε θα ξαναπάρει εντολή να τρέξει)
Thread This Interval 500

ή

Νημα Αυτό Κρατά ' Οπότε με το πέρας της εκτέλεσης του νήματος σταματάει και ο σκανδαλισμός του

Είδη νημάτων:

Η έκτη έκδοση εγκαινιάζει τα νήματα. Αρχικά φτιάχτηκαν τα νήματα για την μουσική, όπου κανείς μπορεί να δώσει Φωνες με σκοπό και τέμπο καθώς και να ορίσει όργανο ανα φωνή και να ακούσει στο περιθώριο μουσική. Μετά σχεδιάστηκε ένα σύστημα αυλών όπου με νήμα πάλι ελέγχεται η εισαγωγή τιμών στον αυλό (ο δημιουργός διαβάζει....είναι ο εξυπηρετητής, ενώ οι πελάτες του στέλνουν δεδομένα), και τέλος δημιουργήθηκαν και τα νήματα εντός τμημάτων και συναρτήσεων.

Άρα έχουμε τρία είδη, τα νήματα μουσικής, αυλών και τμημάτων.

Όταν αυτά τρέχουν φαίνονται με την εντολή Νηματα, ως αριθμοί χειρισμού. Μπορούμε να δώσουμε την εντολή Νημα με τον ανάλογο αριθμό χειρισμού και να τα χειριστούμε.

Το ενδιαφέρον εδώ είναι το πότε αυτόματα παύει να τρέχει ένα νήμα!

Κάθε νήμα λοιπόν τρέχει όσο το τρέχει και το τμήμα ή η συνάρτηση που το δημιούργησε. Οι αυλοί και η μουσική τρέχουν και πέρα από αυτά. Η μουσική τερματίζει στο προκαθορισμένο πέρας της (κάθε Φωνη παίζει το δικό της σκοπό και τερματίζει ανεξάρτητα από το τι κάνει κάθε άλλη Φωνη). Οι αυλοί δέχονται τιμές εφόσον πάρουμε ότι έχουν ήδη στη μεταβλητή εισαγωγής, και παύουν να λειτουργούν αν σβήσουμε το νημα τους ή αν τερματίσει ο διερμηνευτής.

ΤΥΠΩΣΕ ΕΠΙΛΟΓΕΣ.ΦΑΝΕΡΕΣ

Φ=1

ΝΗΜΑ {

Φ++

```
ΑΝ ΕΠΙΛΟΓΕΣ.ΦΑΝΕΡΕΣ ΤΟΤΕ {  
  ΤΥΠΩΣΕ Φ, ΕΠΙΛΟΓΗ$(ΕΠΙΛΟΓΗ)
```

```
} ΑΛΛΙΩΣ {  
  ΤΥΠΩΣΕ Φ  
}
```

```
} ΩΣ Ω  
ΕΠΙΛΟΓΗ ΤΙΤΛΟΣ "Ο ΤΙΤΛΟΣ ΜΟΥ"  
  ΝΗΜΑ Ω ΚΑΘΕ 100  
ΑΝΟΙΓΜΑ.ΑΡΧΕΙΟΥ  
  ΝΗΜΑ Ω ΚΡΑΤΑ  
  ΜΕΤΑ 100 {  
    ΝΗΜΑ Ω ΞΕΚΙΝΑ  
  }
```

'χρειάζεται η εκκίνηση μετά από λίγο γιατί χρησιμοποιούμε τίτλο και αυτός "ανακατασκευάζει" τη λίστα. Αν "παρενοχλήσουμε" την κατασκευή τότε δεν θα πετύχει. Εδώ βλέπουμε τη χρήση της ΚΡΑΤΑ και της ΞΕΚΙΝΑ

```
ΕΠΙΛΟΓΗ "1η Γραμμή", "2η Γραμμή"  
ΤΥΠΩΣΕ ΕΠΙΛΟΓΗ
```

Αναγνωριστικό: ΝΗΜΑΤΑ [THREADS]
ΝΗΜΑΤΑ

Δίνει τους αριθμούς των νημάτων που βρίσκονται στη λίστα νημάτων, ανεξάρτητα αν είναι κρατημένα ή όχι. Δεν συμπεριλαμβάνεται εκείνο το νήμα που την δεδομένη στιγμή τρέχει (αφού έχει βγει εκτός λίστας...δεν μπορεί δηλαδή να ξανατρέξει όσο εκτελείται)

ΝΗΜΑΤΑ ΣΒΗΣΕ

Σβήνει όλα τα νήματα από τη λίστα νημάτων (και αυτών σε αναμονή και εκείνων που τρέχουν). Αυτό δεν σημαίνει ότι σταματούν όλα τα νήματα άμεσα, αλλά σίγουρα σύντομα γιατί δεν θα έχουν άλλη ευκαιρία να εκκινήσουν ξανά.

Αναγνωριστικό: ΡΟΥΤΙΝΑ [SUB]
ΡΟΥΤΙΝΑ ΑΛΦΑ(X)
\\ ΕΝΤΟΛΕΣ
ΤΕΛΟΣ ΡΟΥΤΙΝΑΣ

Οι ρουτίνες γράφονται στο τέλος των τμημάτων ή συναρτήσεων. Καλούμε τις ρουτίνες με την

ΔΙΑΜΕΣΟΥ. Οι ρουτίνες έχουν καθορισμένες παραμέτρους. Κάθε νέα μεταβλητή εντός της ρουτίνας θα σβηστεί με το πέρας αυτής. Μπορούμε να φύγουμε με την Έξοδος Ρουτίνας. Η M2000 χρησιμοποιεί τόνους στα Ελληνικά αλλά στον ορισμό της ρουτίνας η λέξη Ρουτινα δεν πρέπει να έχει τόνο (για λόγους ταχύτητας). Η γλώσσα αναζητεί πάντα από το τέλος τις ρουτίνες. Αν θέλουμε να έχουμε τοπική μεταβλητή μπορούμε με την εντολή Τοπική. Μπορεί μια ρουτίνα να καλέσει τον εαυτό της, μέχρι το ΟΡΙΟ.ΑΝΑΔΡΟΜΗΣ που αρχικά είναι 1000 Όταν βρει την εντολή ΡΟΥΤΙΝΑ ο μεταφραστής καθώς εκτελεί εντολές τότε κάνει έξοδο (δεν συνεχίζει)

Υ=100

Κ=30 \\ θα διαβαστεί στην ρουτίνα γιατί δεν είναι τοπική

ΔΙΑΜΕΣΟΥ ΑΛΦΑ(3)

ΔΙΑΜΕΣΟΥ ΑΛΦΑ(13)

ΔΙΑΜΕΣΟΥ ΑΛΦΑ(5)

ΤΥΠΩΣΕ Υ

ΡΟΥΤΙΝΑ ΑΛΦΑ(Χ)

ΤΟΠΙΚΗ Υ=10

ΑΝ Χ>10 ΤΟΤΕ ΕΞΟΔΟΣ ΡΟΥΤΙΝΑΣ

ΤΥΠΩΣΕ Χ*Υ, Κ

ΤΕΛΟΣ ΡΟΥΤΙΝΑΣ

Αναγνωριστικό: ΣΥΝΑΡΤΗΣΗ [FUNCTION]

1) Κανονικές Συναρτήσεις

Ορίζονται πριν την χρήση τους. Ο ορισμός της συνάρτησης πρέπει να εκτελεστεί για να υπάρχει η συνάρτηση. Αυτό σημαίνει ότι μπορούμε να επιλέγουμε ορισμούς πριν την χρήση βάζοντας τις επιλογές σε μια δομή ΑΝ ή σε Επίλεξε (ή Επέλεξε είναι το ίδιο για τη M2000)

Ό,τι μεταβλητές ορίζουμε είναι εξ ορισμού τοπικές. Δεν μπορούμε να δούμε μεταβλητές έξω από τη συνάρτηση εκτός αν είναι γενικές και δεν υπάρχει τοπική με ίδιο όνομα (σκιάζει την γενική).

Δείτε το παράδειγμα (σε ένα τμήμα έστω Α)

Αδειασε

Βαλε 1,2,3

Συναρτηση αλφα {

Σωρος ' μας δείχνει τα στοιχεία του

διαβασε κ, δ

= κ*δ

}

τυπωσε αλφα(10,5)

δες ολακαλα {

τυπωσε αλφα()

}

τυπωσε ολακαλα
σωρος

Οι συναρτήσεις διαφέρουν από τα τμήματα γιατί χρησιμοποιούνται σε αριθμητικές και αλφαριθμητικές παραστάσεις. Π.χ. εδώ μπορεί κάποιος να προσθέσει ένα 20 στο αλφα(10,5)
Τυπωσε αλφα(10,5)+20

Μπορούν να συνυπάρχουν οι αλφα(), αλφα\$() και αλφα%()

Το σύμβολο % λέει ότι θα γυρίσει η συνάρτηση ακέραια τιμή, το σύμβολο \$ λέει ότι θα γυρίσει η συνάρτηση αλφαριθμητική τιμή.

Κάθε συνάρτηση έχει δικό της σωρό τιμών και όταν την καλούμε βάζουμε τιμές στο δικό της σωρό ενώ όταν τερματίζει αυτός ο σωρός χάνεται με όποιες τιμές και να έχει μέσα.

Μια συνάρτηση μπορεί να καλέσει τον εαυτό της, με το όνομά της ή με το Λάμδα() ή Λάμδα\$() τα οποία λειτουργούν για οποιαδήποτε συνάρτηση και σημαίνει τη τρέχουσα συνάρτηση.

Υπάρχει όπως για το τμήμα και εδώ η ιδέα της στατικής μεταβλητής, δηλαδή μια μεταβλητή που υπάρχει μέσα σε μια συνάρτηση και κρατάει μια τιμή που μπορεί να αλλάζει, και σε κάθε κλήση σε αυτή την συνάρτηση θα υπάρχει εκεί για να χρησιμοποιηθεί. Ακόμα και σε αναδρομή αυτή η μεταβλητή θα μένει εκεί! Αν καλούμε με αναδρομή, η στατική είναι θεατή σε κάθε κλήση, κατά την αναδρομή. Δες Στατική

Οι συναρτήσεις μπορούν να περασθούν με αναφορά σε κλήση.

```
Συνάρτηση A(X) {  
    =X**2+3  
}  
Συνάρτηση B(X) {  
    =X**3+30  
}
```

```
Τμήμα Διάγραμμα (Αρχ, Τελ, Βημ, &Φ()) {  
    Για i=Αρχ έως Τελ ανα Βημ  
        Τύπωσε Φ(i),  
        Επόμενο  
        Τύπωσε  
}
```

Διάγραμμα 10, 50, 10, &A()
Διάγραμμα 10, 50, 10, &B()

Συνάρτηση μιας γραμμής κάνει η Κάνε πχ Κάνε $A(x)=x^2$ και είναι πάντα τοπική
Γενικές φτιάχνουμε με την Γενική

```
    Συνάρτηση Γενική α {  
        Διάβασε X  
        =X**2  
    }
```

Μέσα σε ομάδες (θα περάσουμε με αναφορά μια συνάπτηση που αλλάζει και μια μεταβλητή στην ομάδα ότην την καλούμε):

```
    Ομάδα Αλφα {  
        K=100  
        Συνάρτηση A(x) {  
            =x*.K  
            .K++  
        }  
    }
```

Βήτα=Αλφα

π->Αλφα 'δείκτης π δείχνει το Αλφα το οποίο θα υπάρχει μέχρι να σβηστεί το τμήμα που το δημιούργησε.

π1->(Αλφα) 'δείκτης π1 δείχνει σε ένα αντίγραφο του Αλφα (πτητική ομάδα, ο αριθμός δεικτών κρατάει την ομάδα)

Αλφα.K++

Τύπωσε π=>A(10), π1=>A(10)

π1=>K++

Τύπωσε π=>A(10)= π1=>A(10)

Τμήμα Εσωτερικό (&φ()) {

Τύπωσε φ(10)

}

Εσωτερικό &Αλφα.A()

Τύπωσε Αλφα.K=104, π=>K=104 ' (το π=>K είναι το Αλφα.K)

π->Βήτα

Τύπωσε Αλφα.K=104, π=>K=100 ' (το π=>K είναι το Βήτα.K)

Για π1 {

Εσωτερικό &.A()

}

Τύπωσε π1=>K=104

π=π1 ' δουλεύει και το -> ο π δείχνει αυτό που δείχνει ο π1

Τύπωσε π=>K=104 ' τώρα το π=>K δείχνει ότι το π1=>K

Γενικά μέσα σε συναρτήσεις μπορούμε να βάζουμε τμήματα και νήματα, και το ίδιο συμβαίνει και στα τμήματα και στα νήματα. Οι συναρτήσεις έχουν όνομα όπως και τα τμήματα ενώ τα νήματα

έχουν αριθμό χειρισμού (γράφεται σε μια μεταβλητή) και χρησιμοποιούν το όνομα του τμήματος που τα δημιούργησε. Αυτό σημαίνει ότι μπορούν να δουν και να αλλάξουν τις μεταβλητές του τμήματος που τα δημιούργησε.

Δες ΚΛΗΣΗ ΣΥΝΑΡΤΗΣΗ(και ΣΥΝΑΡΤΗΣΗ\$(

Υπάρχουν δυο αντικείμενα που χειρίζονται συναρτήσεις. Η λάμδα συνάρτηση, και το Γεγονός. Η λάμδα συνάρτηση συνδυάζει κλεισίματα με μια συνάρτηση. Το γεγονός έχει μια λίστα συναρτήσεων και μπορούμε να προσθέτουμε ή να αφαιρούμε από τη λίστα συναρτήσεις. (όχι τις απλές συναρτήσεις).

Η λάμδα συνάρτηση είναι μεταβλητή και συνάρτηση ταυτόχρονα. Το γεγονός καλείται με μια σειρά παραμέτρων οι οποίες πάνε σε όλες τις συναρτήσεις του γεγονότος, ως μια πολλαπλή κλίση. Το ενδιαφέρον με το γεγονός είναι ότι είναι μεταβλητή, και αφενός μπορεί να πάρει άλλη τιμή, και αφετέρου μπορεί να περαστεί είτε με τιμή είτε με αναφορά. Οι κλήσεις συναρτήσεων στο Γεγονός γίνονται σαν κλήσεις τμημάτων.

Μια συνάρτηση μπορεί να κληθεί με τη Κάλεσε. Δες ΚΑΛΕΣΕ

2) Απλές Συναρτήσεις

Οι απλές συναρτήσεις γράφονται στο τέλος ενός τμήματος ή μιας συνάρτησης. Δεν έχουν δικό τους σωρό τιμών, δεν μπορούν να περαστούν με αναφορά, δεν μπορούν να κληθούν με την κάλεσε. Οι απλές συναρτήσεις βλέπουν όλες τις μεταβλητές και ότι έχει το τμήμα ή η συνάρτηση που τις καλεί. Μπορούν να βρίσκονται εκεί που τις καλούμε, στο τέλος του κώδικα, ή στο κώδικα που αρχικά υπήρχε ο κώδικας που τρέχουμε, αδιάφανα για το πρόγραμμά μας. Για τοπικές μεταβλητές χρησιμοποιούμε την Τοπική ή Τοπικές (ορίζει και πίνακες) Ό,τι δημιουργούμε μέσα στη Συνάρτηση διαγράφεται όπως και στη Ρουτίνα. Ενώ οι ρουτίνες έχουν μεγάλο βάθος αναδρομής, οι απλές συναρτήσεις έχουν λίγο μεγαλύτερο από τις κανονικές συναρτήσεις. Δεν ορίζουμε απλές συναρτήσεις ή ρουτίνες εντός μιας απλής Συνάρτησης (δεν υπάρχει έλεγχος αν όντωςβάλουμε μια άλλη συνάρτηση στην ίδια συνάρτηση αλλά το όνομα της συναρτησης λειτουργεί σαν Τέλος)

Υποχρεωτικά το όνομα έχει παρενθέσεις ακόμα και αν δεν έχει παράμετρο.

Τύπωσε @A()

Συνάρτηση A()

=100

Τέλος Συνάρτησης

Μπορούμε να βγούμε πιο νωρίς με την Έξοδος Συνάρτησης. Μπορούμε να έχουμε ετικέτες εντός και να εκτελούμε διακλαδώσεις ακόμα και έξω από το κώδικα της συνάρτησης αρκεί να γίνει η αναμενόμενη επιστροφή. Το Τέλος Συνάρτησης είναι ίδιο με το Τέλος Ρουτίνας'ίδιο με την Επιστροφή χωρίς παραμέτρους.

Ονόματα μπορούν να είναι το A(), A%(), A\$() όπου A ότι όνομα αρχίζει από γράμμα και μετά μπορεί να υπάρχουν γράμματα, αριθμοί, η κάτω παύλα _ και η τελεία. Για κλήση χρειάζεται η @ πριν

Αν ένα όνομα υπάρχει στο βασικό λεξιλόγιο της M2000 δεν μπορεί να γίνει απλή συνάρτηση, αλλά μπορεί να αλλαχθεί προσωρινά σε ένα τμήματα με μια κανονική συνάρτηση. Στις συναρτήσεις με όνομα από το βασικό λεξιλόγιο το @ στην αρχή καλεί την αυθεντική συνάρτηση, είτε υπάρχει άλλη που την άλλαξε προσωρινά είτε όχι. Για το λόγο αυτό ακόμα και αν φτιάξουμε την Συνάρτηση Απολ() ... Τέλος Συνάρτησης, με @Απολ() θα καλέσουμε την αυθεντική Απολ(). Είναι εύκολο να δούμε ποιο όνομα δεν χρησιμοποιούμε στον διορθωτή, γιατί οι λέξεις του βασικού λεξιλογίου χρωματίζονται διαφορετικά.

Αναγνωριστικό: ΤΜΗΜΑ [MODULE]

Δημιουργία τμήματος ή αλλαγή ονόματος σε υπάρχον τμήμα κατά την εκτέλεση!

Μπορούμε να δημιουργήσουμε τμήματα μέσα σε άλλα τμήματα τα οποία θα έχουν ζωή μέχρι να τερματίσει το τμήμα που τα δημιούργησε.

```
Τμήμα αλφα {  
  διαβασε α  
  Αν α Τότε {  
    Τμήμα βητα {  
      Τύπωσε "ένα"  
    }  
  } ΑΛΛΙΩΣ {  
    Τμήμα βητα {  
      Τύπωσε "one"  
    }  
  }  
  βητα  
}  
Το τρέχουμε:  
>αλφα 1  
ένα  
>αλφα 0  
one
```

2) Αν θέλουμε να έχουμε δυο ή περισσότερα σύνολα μεταβλητών μπορούμε να αλλάζουμε το όνομα του τμήματος (προσωρινά κατά την εκτέλεση), δες το παράδειγμα

```

Τμήμα Α {
    α=45
    Τμήμα βητα
    α=23
    λιστα
}

```

Το τρέχον όνομα του τμήματος μπορούμε να το διαβάσουμε με την μεταβλητή μόνο για ανάγνωση Τμήμα\$

3) Μπορούμε να περάσουμε αναφορά σε ένα τμήμα χρησιμοποιώντας ως ισχνή αναφορά την επιστροφή από το Τμήμα\$

```

Τμήμα Β {
    Διάβασε Α, Β
    Τύπωσε Α*2, Β**2
}

```

```

Τμήμα Γ {
    Διάβασε Α$
    Α$.Β 20
}

```

Γ Τμημα\$, 20

Αναγνωριστικό: ΤΜΗΜΑΤΑ [MODULES]

ΤΜΗΜΑΤΑ

Δείνει στην έξοδο τα τμήματα στην μνήμη και στο τωρινό φάκελο στο δίσκο

ΤΜΗΜΑΤΑ !

ότι και το απλό αλλά τα τμήματα του δίσκου τα ταξινομεί κατ' όνομα

ΤΜΗΜΑΤΑ ?

Δείνει μόνο τα τμήματα της μνήμης

ΤΜΗΜΑΤΑ ?, "alfa"

Δείνει μόνο τα τμήματα της μνήμης που ξεκινούν από alfa

ΤΜΗΜΑΤΑ ?, "*alfa"

Δείνει μόνο τα τμήματα της μνήμης που περιέχουν το alfa στο όνομα

ΤΜΗΜΑΤΑ ?, "", "ΠΙΝΑΚΑΣ "

Δείνει μόνο τα τμήματα της μνήμης που έχουν στο κώδικα το Πίνακας ή με οποιοδήποτε συνδυασμ πεζών και κεφαλαίων,

ΤΜΗΜΑΤΑ ?, "alfa", "ΠΙΝΑΚΑΣ "

Δείνει μόνο τα τμήματα της μνήμης που περιέχουν το alfa στο όνομα και έχουν στο κώδικα το Πίνακας ή με οποιοδήποτε συνδυασμ πεζών και κεφαλαίων

\\ για να βρούμε ονόματα αρχείων και εκείνα τα αρχεία που έχουν συγκεκριμένες λέξεις

χρησιμοποιούμε το APXEA

APXEIA "GSB", "PRINT|DIM" ' ψάχνει σε αρχεία GSB εκείνα που έχουν το PRINT και το DIM

Μπορούμε να σώνουμε και να φορτώνουμε τμήματα από οπουδήποτε στο δίσκο. Αλλά υπάρχει ένας κατάλογος, ο κατάλογος χρήστη ο οποίος είναι εξ ορισμού ο φάκελος με το που ξεκινάει το περιβάλλον της M2000. Σε περίπτωση που εκκινήσουμε ένα τμήμα από το δίσκο, τρέχοντας ένα αρχείο gsb, τότε το περιβάλλον θα έχει για αρχικό κατάλογο το κατάλογο του αρχείου. Αν θέλουμε να χρησιμοποιήσουμε ένα τμήμα από το δίσκο, ως μια βιβλιοθήκη τμημάτων, τότε μπορούμε να την έχουμε στο κατάλογο χρήστη

ΚΑΤΑΛΟΓΟΣ ΧΡΗΣΤΗ

win KAT\$

ανοίγουμε τον explorer στο φάκελο χρήστη. Από εκεί μπορούμε να διαγράψουμε ή να μεταφέρουμε τμήματα.

Αναγνωριστικό: ΧΡΗΣΗ [USE]

1) Χρηση HELLO.GSB 1,"AAAA", 3

2) Χρηση "HELLO" 1,"AAAA", 3

3) A\$="HELLO" : Χρηση A\$ 1,"AAAA", 3

4) Χρηση "προγραμμα" "αυτο","εκεινο" Στο α\$ Ως Λ
Νημα Λ Καθε 200

περιμένουμε στο α\$ να πάρουμε στοιχεία τα οποία θα βάλει όποιος γνωρίζει το αναγνωριστικό του αυλού. Εδώ στέλνοντας στο "προγραμμα" τα δυο αλφαριθμητικά πάει και το αναγνωριστικό του αυλού.

4.1) α\$="ονομααυλου": Χρηση προγραμμα1 στο A\$

ξεκινάει άμεσα ο αυλός με ένα νήμα κάθε 100 χιλιοστά του δευτερολέπτου με όνομα αυλού το α\$. Μετά τη εντολή χρήση το α\$ έχει το πλήρες όνομα του αυλού και εφόσον σβήσουμε το περιεχόμενο ξεκινάμε να διαβάζουμε (αν μας στέλνει κάποια άλλη εφαρμογή)

5) Χρηση Αυλου Στο α\$ Ως Λ : Νημα Λ Καθε 200

φτιάχνουμε ένα αυλο χωρίς να εκινήσουμε κάποια εφαρμογή. Αυτό μπορεί να το κάνει η εφαρμογή που ξεκίνησε με το 4 και να δώσει στο αρχικό πρόγραμμα το νούμερο του δικού του αυλού ώστε και τα δυο προγράμματα θα περιμένουν να λάβουν στοιχεία βάσει νήματος (ανά προκαθορισμένο χρονικό διάστημα) και θα μπορούν να στέλνουν (με δοκιμή σφάλματος) όταν ο αντίστοιχος αυλός είναι καθαρός, δεν έχει δηλαδή δεδομένα για να διαβαστούν.

5.1) Και εδώ μπορούμε να ορίσουμε όνομα αυλού γράφοντάς το στην αλφαριθμητική μεταβλητή

Ομάδα: ΡΟΗ ΠΡΟΓΡΑΜΜΑΤΟΣ - FLOW CONTROL

Αναγνωριστικό: ΑΛΛΙΩΣ [ELSE]
ΔΕΣ ΑΝ, ΕΠΕΛΕΞΕ

Αναγνωριστικό: ΑΛΛΙΩΣ.ΑΝ [ELSE.IF]
δες Αν

Αναγνωριστικό: ΑΝ [IF]

1)

Αν συνθήκη Τότε { εντολές σε μια γραμμή }
για μια εντολή μπορεί να δοθεί το παρακάτω
Αν $I > K$ Τότε Τύπωσε I : Τύπωσε "και κάτι ακόμα" \ \ και ότι συνεχίζει στην ίδια γραμμή θα είναι η
συνέχεια του Τότε
Αν υπάρχει μια Αλλιώς τότε πρέπει να υπάρχει μπλοκ στην Τότε (ή να μην υπάρχει η Τότε)
Αν $I > K$ Τότε { Τύπωσε I } Αλλιώς Τύπωσε K
Από 9.5 μπορεί να γίνει και έτσι (χωρίς μπλοκ)
Αν $I > K$ Τότε Τύπωσε I Αλλιώς Τύπωσε K

Αν $I > 10$ Αλλιώς Τύπωσε "Μικρό ή ίσο με 10"

Αν Σημαία Τότε 400 \ \ είναι το ίδιο με Αν Σημαία=Αληθής Τότε Προς 400

Αν Σημαία Αλλιώς 400 \ \ είναι το ίδιο με Αν Σημαία=Αληθής Τότε Προς 400

Αν Σημαία Τότε Προς Εκεί

\ \ το Εκεί είναι ετικέτα

Εκεί:

2)

Αν συνθήκη Τότε {
 εντολές πολλών γραμμών
}

3)

Αν συνθήκη Τότε { εντολές μιας γραμμής } Αλλιώς { εντολές μιας γραμμής }

4)

Αν συνθήκη Τότε {

```
    εντολές πολλών γραμμών
} Αλλιώς {
    εντολές πολλών γραμμών
}
```

5) Προσθήκη από 6.4
Αν συνθήκη Τότε {
 εντολές πολλών γραμμών
} Αλλιώς.Αν συνθήκη Τότε {
 εντολές πολλών γραμμών
} Αλλιώς {
 εντολές πολλών γραμμών
}

6) Απλοποιημένη σύνταξη (από έκδοση 8.9 - αναθεώρηση 12)

Αν συνθήκη Τότε εντολές πολλών γραμμών
Αλλιώς.Αν συνθήκη Τότε εντολές πολλών γραμμών
Αλλιώς εντολές πολλών γραμμών

ή και αυτό ή όπως θέλουμε με κενές γραμμές
Αν συνθήκη Τότε {εντολές πολλών γραμμών }
Αλλιώς.Αν συνθήκη Τότε {εντολές πολλών γραμμών}
Αλλιώς {εντολές πολλών γραμμών}

7) Από την 9.5 έκδοση προστέθηκε η χωρίς μπλοκ ανάπτυξη σε μια γραμμή
Αν I>K Τότε Τύπωσε I Αλλιώς.Αν I>Δ Τότε Τύπωσε Δ Αλλιώς Τύπωσε K

8) Από την 9.6 έκδοση προστέθηκε και η ανάπτυξη Αν Τέλος Αν
Αν συνθήκη Τότε

Αλλιώς.Αν Τότε

Αλλιώς.Αν Αλλιώς

Αλλιώς.Αν Τότε

Αλλιώς

Τέλος Αν

Αν συνθήκη Αλλιώς

Τέλος Αν

Μπορούμε να βάζουμε όσα ΑΛΛΙΩΣ.ΑΝ θέλουμε! Μπορούμε στο τελευταίο μέρος (εδώ είναι το Αλλιώς) να βάλουμε στην ίδια γραμμή σειρά με εντολές που θα χωρίζουν με άνω και κάτω τελεία.

Όπως είναι το Όχι είναι και το Δεν

Η συνθήκη μπορεί να περιλαμβάνει τα λογικά ΟΧΙ, ΚΑΙ, Η, ΑΠΟ, συγκρίσεις > < = >= <= παρενθέσεις, συγκρίσεις αλφαριθμητικών, αριθμητικών κ.λ.π. Αν το αποτέλεσμα είναι ΨΕΥΔΕΣ δηλαδή 0 τότε πάμε στην Αλλιώς (Αν έχει δηλωθεί) όλα τα άλλα τα λαμβάνει ως αληθές (κανονικά τιμή -1)

Τύπωσε ΑΛΗΘΕΣ, ΨΕΥΔΕΣ, ΟΧΙ ΑΛΗΘΕΣ, ΑΛΗΘΕΣ Η ΨΕΥΔΕΣ, ΑΛΗΘΕΣ ΚΑΙ ΑΛΗΘΕΣ, ΑΛΗΘΕΣ ΑΠΟ ΨΕΥΔΕΣ

δίνει

-1 0 0 -1 -1 -1

Τύπωσε 2<4 Και "Α"="Α" αυτό από την έκδοση 6.4 είναι σωστό! (παλιά ήταν ΛΑΘΟΣ)

Μπορούμε να φωλιάζουμε τα Αν (μέσα να βάλουμε ότι εντολές θέλουμε όπως άλλα Αν, ΓΙΑ, ΕΩΣ κλπ).

Μπορούμε να κάνουμε έξοδο από ένα μπλοκ μέσα στην Αν αλλά Αν θέλουμε την Αν να την χρησιμοποιήσουμε για να καλέσουμε κάτι που θα αλλάξει τη ροή στο μπλοκ εντολών που βρίσκεται η Αν τότε δεν πρέπει να χρησιμοποιήσουμε τους τύπους της Αν με τις αγκύλες. παίζει και η Αν συνθήκη Αλλιώς όπως και το παρακάτω που δίνει "σωστά"

με τα παρακάτω Αν μια συνθήκη δεν είναι Οκ τότε οι άλλες δεν θα υπολογιστούν...ο μεταφραστής θα αλλάξει γραμμή!

Αν ψευδες αλλιως.αν ψευδες αλλιως.αν ψευδες Αλλιώς Τύπωσε "σωστά"

οι συνθήκες δεν χρειάζονται αγκύλες!

Αν αληθες Τότε Αν αληθες Τότε Αν αληθες Τότε Τύπωσε "σωστά"

ΠΡΟΣΟΧΗ στις συνθήκες δεν μπορούμε να χρησιμοποιήσουμε την μορφή των αλφαριθμητικών με αγκύλες.

Αναγνωριστικό: ΑΝΑΛΥΤΗΣ [PROFILER]

Κατά την ροή του προγράμματος μπορούμε να κάνουμε αναλύσεις ταχύτητας. Ο Αναλυτής ξεκινάει όπου εμείς θέλουμε και κατόπιν μετράμε όπου θέλουμε τον φόρτο, το πόσα χιλιοστά δευτερολέπτου "κόστησε".

Αναλυτής
εντολές.....
Τύπωσε Φορτος

Ουσιαστικά κάθε σειρά εντολών είναι κοστοβόρα σε επεξεργαστικό χρόνο. Έτσι η αναζήτηση του χρόνου είναι πολλές φορές ζωτικής σημασίας (ειδικά όταν έχουμε επαναλήψεις σε μεγάλο αριθμό). Η διαδικασία του αναλυτή φανερώνει το φορτίο του επεξεργαστή. Έτσι φόρτος εδώ είναι ο χρόνος εργασίας. Ο αναλυτής βάζει μπροστά τον μετρητή....(η μέτρηση γίνεται από το λειτουργικό σύστημα). Η μεταβλητή Φορτος δεν σταματάει τον μετρητή. Πρέπει να δώσουμε πάλι την εντολή ΑΝΑΛΥΤΗΣ για να ξεκινήσουμε νέα μέτρηση, διαφορετικά η επόμενη Φορτος θα μας δώσει την μέτρηση από την εντολή ΑΝΑΛΥΤΗΣ.

Αναγνωριστικό: ANAMONH [WAIT]
ANAMONH 100
Wait 100

Περιμένει το πρόγραμμα για 100 χιλιοστά του δευτερολέπτου, δίνοντας χρόνο στο λειτουργικό να κάνει κάτι άλλο.

Μετα 200 {
 Τύπωσε "Είμαι εδώ"
}
Αναμονη 500

Η δομή Μετα φτιάχνει ένα νήμα που θα τρέξει μια φορά σε 200 χιλιοστά του δευτερολέπτου. Αν δεν βάλουμε κάτι να κάνει το τμήμα να περιμένει τότε θα τερματίσει με συνέπεια να εξαφανίσει το νήμα, να μην τρέξει δηλαδή.

Άλλες εντολές που κάνουν αναμονή χωρίς όμως να βγαίνουν άμεσα από αυτήν είναι η ΚΑΘΕ και η ΚΥΡΙΟ.ΕΡΓΟ οι οποίες είναι η πρώτη δομή επανάληψης (κάθε διάστημα σε χιλιοστά που δίνουμε) χωρίς συνθήκη τερματισμού (μπορούμε να εκτελέσουμε την Έξοδος για να βγούμε από αυτή) και η δεύτερη ένα νήμα σαν την Μετα, που όμως όταν τερματίσει τερματίζει όλα τα νήματα, αλλά και αυτή δεν εκτελείται με κάποια συνθήκη..(ή μάλλον τερματίζει αν δεν υπάρχει άλλο νήμα εκτός από το δικό της).

Αναγνωριστικό: ΑΠΟ [ON]

ΑΠΟ μεταβλητή ΠΡΟΣ ετικετα1, ετικετα2, ετικετα3

Πρέπει να υπάρχουν στο ίδιο τμήμα οι ετικέτες

Προσοχή γιατί μπορεί να γίνει και άλμα προς τα πίσω και αν δεν αλλάξει στο μεταξύ η τιμή της μεταβλητής μπορεί να κολλήσει ο μεταφραστής (στις δοκιμές βάλτε πριν από την εντολή ΑΠΟ μια εντολή ΑΝΑΝΕΩΣΗ η οποία δίνει τον έλεγχο στιγμιαία στο σύστημα και μπορούμε να διακόψουμε ομαλά το πρόγραμμα με το ESC)

Αναγνωριστικό: ΓΙΑ [FOR]

Με χρήση του Θέσε Διακόπτες "+FOR" αλλάζουμε το τρόπο εκτέλεσης της ΓΙΑ να ταιριάζει με της BASIC, δηλαδή το πρόσημο του βήματος να μετράει, ώστε να υπάρχει περίπτωση να μην εκτελεστεί η ΓΙΑ. Με το κανονικό "-FOR", πάντα η Για εκτελείται μία τουλάχιστον φορά, η δε φορά αλλαγής απαρίθμησης εξαρτιέται από την αρχική και τελική τιμή. Μόνο όταν αυτές οι δυο τιμές είναι ίσες, μπορεί να καθορίσει το πρόσημο του βήματος (αν το έχουμε δώσει και μόνο τότε), την τελική τιμή μετά αυτήν τη δομή.

1) Με βήμα 1

```
ΓΙΑ μεταβλητή= αρχικη_τιμη ΕΩΣ τελικη_τιμη {  
    εντολές  
}
```

2) Με βήμα που θέλουμε

```
ΓΙΑ μεταβλητή= αρχικη_τιμη ΕΩΣ τελικη_τιμη ΑΝΑ βημα {  
    εντολές  
}
```

Από αναθεώρηση 51, έκδοση 8:

3) Το <= βάζει τιμή σε γενική μεταβλητή

4) Χρήση αρνητικού βήματος για περίπτωση που αρχή και τέλος είναι ίδιο.

Το Για v=1 εως 1 ανα -1 {} δίνει 0 μετά το πέρας στο v

Το Για v=1 εως 1 ανα 1 {} δίνει 2 μετά το πέρας στο v

Το Για v=1 εως 1 {} δίνει 1 μετά το πέρας στο v όπως ήταν πάντα.

Το βήμα είναι μια απόλυτη τιμή (δεν έχει σημασία το πρόσημο της) μπορεί η αρχική τιμή να είναι μεγαλύτερη της τελικής αλλά η μέτρηση θα γίνει πάντα προς τη τελική τιμή.

Η μεταβλητή μπορεί να είναι ακέραια η πραγματική (π.χ. A% ή B ή ειδικού τύπου Μακρύς) και είναι πάντα τοπική.

Μπορούμε να έχουμε πολλαπλά ΓΙΑ το ένα μέσα από το άλλο (Φωλιασμένα)

ΠΡΟΣΟΧΗ 1

Μπορούμε να αλλάξουμε την τιμή της I μέσα στο Για αλλά δεν θα αλλάξει ο αριθμός των επαναλήψεων και σε κάθε νέα επανάληψη η μεταβλητή θα έχει το σωστό νούμερο.

ΠΡΟΣΟΧΗ 2

Αν η αρχική τιμή είναι ίδια με την τελική ΔΕΝ θα αλλάξει μετά το ΓΙΑ (δες το 4 πάνω) αν δεν έχουμε θέσει βήμα.

5) Για Επόμενο

Για i=1 έως 100 ανά 2

Τύπωσε i

Επόμενο i

Για i=1 έως 100 ανά 2

Τύπωσε i

Επόμενο ' χωρίς τη μεταβλητή i

Αν θέλουμε να δημιουργήσουμε μια επανάληψη με επιλογή μέσα από την επανάληψη ενός διαφορετικού βήματος ή ενός τερματισμού πριν την ολοκλήρωση του αριθμού επαναλήψεων τότε πρέπει να χρησιμοποιήσουμε μια απλή επανάληψη με το ΚΥΚΛΙΚΑ όπου θα δώσουμε εσωτερικά στην ομάδα εντολών την εντολή αύξησης ή μείωσης του βήματος

Για I=1 Έως 100 Ανά 2 {

Αν I > 50 Τότε Έξοδος

Αν I > 30 Και I < 45 Τότε Συνέχισε

Τύπωσε I

}

Τύπωσε ">>>", I

\\ χωρίς χρήση Για

I=10

A=Τυχαίος(1,3)

{

Αν I < 30 Τότε Κυκλικά \\ απλά βάζει μια σημαία στο μπλοκ

Αν A > 1 και I > 25 Τότε Έξοδος

Τύπωσε I

I++

}

\\ χωρίς χρήση μπλοκ

```

I=10
A=Τυχαίος(1,3)
090 Σημαία=Ψευδές
100 Αν I<30 Τότε Σημαία=Αληθές
110 Αν A>1 και I>25 Τότε 150
120 Τύπωσε I
130 I++
140 Αν Σημαία Τότε 90
150 Τύπωσε "ΟΚ"

```

```

\\ Με χρήση νήματος
Οθόνη
I=10
A=Τυχαίος(1,3)
Εξ1=Ψευδές
Νήμα {
    I++
    Αν I>30 ή (A>1 και I>25) Τότε Εξ1=Αληθές : Νήμα Αυτό Σβήσε
    Τύπωσε I
} ως Ν κάθε 10
\\ πάντα το πλήκτρο του ποντικιού για να τερματίσει
Κύριο.Έργο 100 {
    Αν Δείκτης<>0 ή Εξ1 Τότε Έξοδος
}

```

```

Αναγνωριστικό: ΔΕΣ [TRY]
[ΕΝΤΟΛΕΣ ΡΟΗΣ ΤΜΗΜΑΤΩΝ] ' έλεγχος λαθών
' την μεταβλητή Β θα την δημιουργήσει η ΔΕΣ αν δεν την έχουμε ήδη δημιουργήσει
ΔΕΣ Β {
    ΤΥΠΩΣΕ Β ' 0
}
ΤΥΠΩΣΕ Β ' -1 δεν υπάρχει λάθος
ΔΕΣ Β {
    ΛΑΘΟΣ "Αυτό είναι ένα λάθος" ' δημιουργώ λάθος
}
ΤΥΠΩΣΕ ΛΑΘΟΣ$, Β ' το Β είναι 0
ΤΥΠΩΣΕ "Δεν σταμάτησε η ροή" ' δεν σταματάει η ροή εντολών.

```

```

ΔΕΣ Ι {
' ΕΝΤΟΛΕΣ
}

```

```
Δες {  
    ' και χωρίς μεταβλητή  
}
```

ΑΝ ΟΙ ΕΝΤΟΛΕΣ ΒΓΑΛΟΥΝ ΛΑΘΟΣ ΤΟΤΕ Η ΜΕΤΑΒΛΗΤΗ Ι ΓΙΝΕΤΑΙ 0 ΑΛΛΙΩΣ -1

Αν το λάθος βρίσκεται σε τμήμα που καλούμε μέσα από το μπλοκ της ΔΕΣ, τότε δεν επιστρέφει στη μεταβλητή που ορίζουμε το Ψευδές (το 0) και πρέπει να κοιτάμε την ΛΑΘΟΣ ή την ΛΑΘΟΣ\$.

Αν κοιτάξουμε μια φορά την ΛΑΘΟΣ\$ τότε αδειάζεται το λάθος, καθαρίζει δηλαδή εσωτερικά η μεταβλητή ΛΑΘΟΣ\$.

Υπάρχει και η Άδειασε Λάθος που μπορούμε να χρησιμοποιήσουμε χωρίς καν να κοιτάξουμε την ΛΑΘΟΣ\$

Αναγνωριστικό: ΔΙΑΜΕΣΟΥ [GOSUB]

1) Διαμέσου ετικέτα ή αριθμός

Απλή κλήση υπορουτίνας

Κάνουμε κλήση υπορουτίνας με αριθμό ή όνομα. Δεν είναι το ίδιο με τις Ρουτίνες με παρενθέσεις (Δες ΡΟΥΤΙΝΑ)

2) Διαμέσου Ρουτινα1()

\\ χωρίς πέρασμα παραμέτρων

2.1) Ρουτινα1()

Η εντολή Διαμέσου δεν είναι απαραίτητη για να κληθεί η ρουτίνα, αρκεί να μην υπάρχει πίνακας με ίδιο όνομα

Κλήση ρουτίνας που υπάρχει στο τέλος του τμήματος

Ρουτινα Ρουτινα1()

Τύπωσε "οκ"

Τέλος Ρουτίνας

3) Διαμέσου Ρουτινα1(100, 300, &A(), B) \\ με πέρασμα παραμέτρων

3.1) Ρουτινα1(100, 300, &A(), B)

Η εντολή Διαμέσου δεν είναι απαραίτητη για να κληθεί η ρουτίνα, αρκεί να μην υπάρχει πίνακας με ίδιο όνομα

Κλήση ρουτίνας που υπάρχει στο τέλος του τμήματος

Ρουτινα Ρουτινα1(X,Y, &Αλφα(), B)

Τύπωσε "οκ", X, Y, Αλφα(B)
Τέλος Ρουτίνας

Οι ρουτίνες με παρενθέσεις είναι κάτι παραπάνω από απλές ρουτίνες. Ότι νέο φτιάχνεται σε ρουτίνα με παρένθεση θα πάψει να υπάρχει στο πέρας της εκτέλεσης (όπως στα τμήματα και στις συναρτήσεις) και επιπλέον κάθε ρουτίνα μπορεί να καλεί τον εαυτό της (αναδρομή). Διαφέρει από τα τμήματα και τις συναρτήσεις για δυο λόγους: Ένας είναι ότι ανήκει στο τμήμα ή συνάρτηση και δεν δημιουργείται, άρα δεν μπορεί να αλλάξει ορισμό, και συνάμα έχει πρόσβαση όπως όλος ο κώδικας εντός τμήματος, άρα και στις μεταβλητές του. Δεύτερον δεν μπορεί να κληθεί από άλλο τμήμα παρά μόνο από αυτό που τρέχει - και από αυτό που υπάρχει στον αρχικό κώδικα και είναι εντός του τμήματος (για να μην γράφουμε δυο φορές την ίδια ρουτίνα...είναι ένας τρόπος του μεταφραστή για οικονομία, αλλά δεν έχει συνέπειες στο τι μεταβλητές βλέπει η ρουτίνα).

```
\ \ Χρήση έμμεσης κλήσης σε ρουτίνα
Διαμέσου αλφα(100,200) ' κανονικό
b$="(100, 200)"
Διαμέσου "αλφα"+b$
β$="αλφα(100,200)"
Διαμέσου β$
Πίνακας β$(10)
β$(3)="αλφα(100,200)"
Διαμέσου β$(3)
```

Ρουτίνα αλφα(χ, ψ)
Τύπωσε χ, ψ
Τέλος Ρουτίνας

\ \ Απλό Διαμέσου

```
Διαμέσου 100 : Διαμέσου 110
Διαμέσου 100 : Διαμέσου 110
Διαμέσου όνομα : Διαμέσου Όνομα \ \ προσοχή οι τόνοι έχουν σημασία στις ετικέτες.
Έξοδος \ \ Μπορούμε να γράψουμε Τέλος
100 Τύπωσε "ΟΚ" : Επιστροφή
110 Τύπωσε "ΛΑΘΟΣ" : Επιστροφή
ΌΝΟΜΑ: \ \ μόνο υπενθυμίσεις εδώ
Τύπωσε "ΌΝΟΜΑ" : Επιστροφή
Όνομα:
Τύπωσε "Όνομα" : Επιστροφή
```

Σύνθετο Διαμέσου

\\ χρησιμοποιούμε πέρασμα με αναφορά για να πάρουμε στην A\$ το αποτέλεσμα από την N\$
ΚΑΘΑΡΟ A\$

Διαμέσου ENA_ONOMA("Δώσε Όνομα", &A\$)

Τύπωσε A\$

ΡΟΥΤΙΝΑ ENA_ONOMA(T\$, &N\$)

ΕΝΩ N\$="" {

Τύπωσε T\$;

ΕΙΣΑΓΩΓΗ ":",N\$

}

Τύπωσε N\$

Τέλος Ρουτίνας

Για έξοδο πριν το τέλος της ρουτίνας εκτελούμε το Τέλος Ρουτίνας, ενώ στις απλές ρουτίνες αρκεί το Επιστροφή.

Προσοχή!

Επειδή οι Ρουτίνες βλέπουν όλες τις μεταβλητές, αν θέλουμε να σκιάσουμε μια υπάρχουσα θα την κάνουμε Τοπική

\\ Παράδειγμα

K=100

M=50

Άλφα(30)

Άλφα(60) \\ τίποτα

Δες {

αλφα(10) \\ αν ξεχάσουμε τον τόνο θα βγει λάθος

}

Τύπωσε M

Τέλος

Ρουτίνα Άλφα(X)

Τοπική K=Τυχαίος(1,10)

Αν X>50 Τότε Έξοδος Ρουτίνας

Τύπωσε X*K

M++

Τέλος Ρουτίνας

Αναγνωριστικό: ΔΙΕΚΟΨΕ [BREAK]

Έχει δυο χρήσεις

Μία για την ΕΠΙΛΕΞΕ ΜΕ ή SELECT CASE

και μια γενικότερη όπου διακόπτει την ροή ενός τμήματος χωρίς πρόβλημα!

```

1)
a=5
b=10
select case A
case 1, 5 to 20
    { print "ok"
      if b=10 then break
    }
case 2,4
    {print "one ok"
     continue
    }
case 3
    { print "that"
      print "and that"
    }
else
    print "??"
end select

```

```

2)
τμήμα Α {
    τυπωσε $(,4)
    για i=1 έως 20 {
        για κ=1 έως 10 {
            τυπωσε "κατι", i, κ
            αν i=2 και κ=2 τότε διεκοψε
        }
    }
    τυπωσε "δεν τυπώνει αυτή η γραμμή"
}
καλεσε Α
τυπωσε " αλλά τυπώνει αυτή"

```

Αναγνωριστικό: ΕΝΩ [WHILE]
 ΕΝΩ συνθήκη {
 εντολές

```
}
```

```
Ενώ αντικείμενο_επαναλήπτης {  
    εντολές  
}
```

Αυτή είναι μια ανακύκλωση που συμβαίνει αφού έστω και μια φορά ελεγχθεί η συνθήκη

Στην πρώτη περίπτωση μέσα στην ΕΝΩ θα πρέπει να υπάρχει ένας τρόπος που θα αλλάζει έτσι την συνθήκη ώστε να τερματίζει.

Στη δεύτερη θα τερματίσει στο πέρας της επανάληψης.

\\Παράδειγμα πρώτης περίπτωσης

A=10

```
Ενώ A>0 {  
    Τύπωσε A  
    A-- ' εδω αλλάζει η A άρα και η συνθήκη A>0  
}
```

\\ Παράδειγμα δεύτερης περίπτωσης

K=(1,2,3,4,5)

K1=Κάθε(K, 1, 3)

Τύπωσε "K1"

```
Ενώ K1 {  
    ' θα γίνει τρεις φορές η επανάληψη  
    Τύπωσε Πίνακας(K1)
```

```
}
```

K1=Κάθε(K, 1, 3) ' από την αρχή στο 3

K2=Κάθε(K, -1, 3) ' από το τέλος στο 3

Τύπωσε "K1, K2"

```
Ενώ K1, K2 {  
    ' θα γίνει τρεις φορές η επανάληψη  
    Τύπωσε Πίνακας(K1)*Πίνακας(K2)
```

```
}
```

K1=Κάθε(K, 1, 3)

K2=Κάθε(K, -1, 3)

Τύπωσε "K1*K2"

```
Ενώ K1 {  
    Ενώ K2 {  
        ' θα γίνει 3X3 φορές η επανάληψη  
        Τύπωσε Πίνακας(K1)*Πίνακας(K2)  
    }  
}
```

Υπάρχει και χωρίς μπλοκ

Ενώ συνθήκη

Τέλος Ενώ

Αναγνωριστικό: ΕΞΟΔΟΣ [EXIT]

εξοδος

Άμεση έξοδος από τμήμα ή από δομή με αγκύλες

Για $i=1$ έως 100

Αν $i>5$ Τότε Έξοδος Για

τυπωσε i

επόμενο i

Η ΕΞΟΔΟΣ από μπλοκ { } που έχουμε με εντολή ΚΥΡΙΟ.ΕΡΓΟ διαγράφει όλα τα νήματα του τμήματος.

Εξοδος Ρουτίνας

Εξοδος Για

Εξοδος Για ετικέτα

Για $i=1$ έως 10

αν $i=5$ τότε εξοδος για 10

Επόμενο i

5 Τυπωσε "όχι αυτό"

10 Τύπωσε "αυτό"

Αναγνωριστικό: ΕΠΑΝΑΛΑΒΕ [REPEAT]

ΕΠΑΝΕΛΑΒΕ & ΕΠΑΝΑΛΑΒΕ μπορούν να χρησιμοποιηθούν ως ίδια εντολή.

ΕΠΑΝΑΛΑΒΕ {

ΑΝ $B=3$ ΤΟΤΕ ΕΞΟΔΟΣ

ΑΝ $K=1$ ΤΟΤΕ ΣΥΝΕΧΙΣΕ

.....

} ΟΣΟ $A<>1$

```

ΕΠΑΝΑΛΑΒΕ {
    ΑΝ Β=3 ΤΟΤΕ ΕΞΟΔΟΣ
    ΑΝ Κ=1 ΤΟΤΕ ΣΥΝΕΧΙΣΕ
    .....
} ΜΕΧΡΙ Α=1

```

```

ΕΠΑΝΕΛΑΒΕ {
    ΑΝ Β=3 ΤΟΤΕ ΕΞΟΔΟΣ
    ΑΝ Κ=1 ΤΟΤΕ ΣΥΝΕΧΙΣΕ
} ΠΑΝΤΑ

```

...δεν σταματάει ποτέ...εκτός από το ESC πλήκτρο ή με μια EXIT μέσα από την επανάληψη.

Υπάρχει και χωρίς μπλοκ (σαρώνει για να βρει το Μεχρι ή το Πάντα οπότε είναι λίγο πιο αργή από την έκδοση με μπλοκ)

ΕΠΑΝΑΛΑΒΕ

ΟΣΟ συνθήκη

ΕΠΑΝΑΛΑΒΕ

ΜΕΧΡΙ συνθήκη

ΕΠΑΝΑΛΑΒΕ

ΠΑΝΤΑ

Από την έκδοση 6 μπήκε και η Repeat Until, μια δομή που θα μπορούσε να δουλέψει και με την Κυκλικά και ένα απλό {}

```

{
ΚΥΚΛΙΚΑ ' αναγκάζει τον διερμηνευτή να ξανατρέξει το μπλοκ όταν όμως φθάσει στο τέρμα του

```

```

αν Α=1 τότε εξοδος ' μπορούμε όμως να βγούμε με την έξοδο, ή με διεκοψε (βγαίνουμε άμεσα και από τμήμα)
}

```

Αναγνωριστικό: ΕΠΑΝΕΛΑΒΕ [DO]
ΔΕΣ ΤΟ ΕΠΑΝΑΛΑΒΕ

Αναγνωριστικό: ΕΠΙΛΕΞΕ [SELECT]

Μπορείς να χρησιμοποιήσεις το Διεκοψε και το Συνεχισε. δες τις εντολές αυτές.

Μπορείς να χρησιμοποιήσεις την Προς για να βγεις από μπλοκ εντολών προς μια ετικέτα (αριθμό ή γράμματα)

Σε κάθε Με μπορείς να γράψεις μια σειρά εντολών με διαχωριστικό :

αλλιώς για πολλές γραμμές χρειάζεται ένα μπλοκ { }

Το ίδιο ισχύει στη Αλλιώς ή Αλλιώς Με (αν χρησιμοποιούμε την Αν με Τέλος Αν τότε το Αλλιώς μόνο του θα το βρει η Αν και θα βγει λάθος)

ΕΠΙΛΕΞΕ ΜΕ αλφαριθμητικο\$ ή αριθμητικό

ΜΕ >=τιμη, τιμη1 ΕΩΣ τιμη2, τιμη3

{

εντολές

}

ΜΕ τιμη5, τιμη6, =τιμη7

εντολή μιας γραμμής

ΜΕ

...

ΑΛΛΙΩΣ ΜΕ

{

}

ΤΕΛΟΣ ΕΠΙΛΟΓΗΣ

Αναγνωριστικό: ΚΑΘΕ [EVERY]

ΚΑΘΕ 100 {

.....

ΑΝ Α=1 ΤΟΤΕ ΕΞΟΔΟΣ

}

Every 100 {

.....

if a=1 then exit

}

Εκτέλεση εντολών ανά τακτά χρονικά διαστήματα (χιλιοστά του δευτερολέπτου) και το οποίο δεν τερματίζει εκτός και αν δώσουμε κάπου την εντολή Εξοδος.

Επειδή τα νήματα είναι μέρη τμημάτων που τρέχουν παράλληλα ανά τακτά χρονικά διαστήματα πρέπει να δώσουμε χρόνο στο σύστημα για να τα τρέξει, διαφορετικά με το πέρασ του τμήματος που τα δημιούργησε θα σβηστούν. Έτσι με αυτή την δομή αφήνουμε τα νήματα να λειτουργήσουν και ελέγχουμε κάποια συνθήκη για να δούμε πότε θα τερματίσουμε.

Όμως από την έκδοση 6.2 μπήκε η νέα έκδοχή της η ΚΥΡΙΟ.ΕΡΓΟ που δημιουργεί ένα νήμα για να τρέξει το μπλοκ. Εδώ η εντολή ΕΞΟΔΟΣ διαγράφει όλα τα νήματα του τμήματος. Η διαφορά τους είναι σημαντική γιατί γίνεται διαφορετική κατανομή χρόνου επεξεργασίας. Στην περίπτωση της ΚΑΘΕ ένα μεγάλο τμήμα μπορεί να καθυστερήσει τόσο ώστε η ΚΑΘΕ να μην έχει χρόνο να τρέξει. Αν θέλουμε όμως με την ΚΥΡΙΟ.ΕΡΓΟ κάνουμε το μπλοκ της δομής να είναι ένα νήμα! Οπότε ο χρόνος των νημάτων μοιράζεται!**

Το αναγνωριστικό ΚΑΘΕ χρησιμοποιείται και στην εντολή ΝΗΜΑ

Καθε ή Interval

**Εξήγηση. Τα νήματα τα διαχειρίζεται ένα σύστημα με ένα κεντρικό ρολόι, σε μια εσωτερική λειτουργία την οποία καλεί εξωτερικά το λειτουργικό. Ενώ τρέχει το περιβάλλον της Μ2000 όταν υπάρχουν νήματα ξεκινάει ο εξωτερικός σκανδαλισμός. Σε κάθε σκανδαλισμό η εσωτερική λειτουργία κοιτάει σε μια λίστα νημάτων να δει ποιο έχει σειρά να το καλέσει. Στη κλήση το νήμα κοιτάει αν είναι ο χρόνος του να τρέξει και αν είναι οκ τότε θα τρέξει. Ο σκανδαλισμός όμως συνεχίζει να καλεί την εσωτερική ρουτίνα. Δεν γίνεται να καλέσουμε ένα νήμα δυο φορές γιατί σε κάθε περίπτωση το νήμα που πήρε το σκανδαλισμό βγαίνει εκτός λίστας και θα ξαναμπεί όταν τελειώσει ότι έχει να κάνει. Κάποια στιγμή η λίστα νημάτων θα αδειάσει. Όμως σε κάθε περίπτωση κρατάει το σύστημα μια λίστα με τα νήματα που τρέχουν. Έτσι το σύστημα δεν σταματάει τον σκανδαλισμό αν έχει νήματα που ήδη τρέχουν ενώ δεν έχει άλλα να τρέξει. Αν λοιπόν χρησιμοποιήσουμε την ΚΥΡΙΟ.ΕΡΓΟ τότε μόνο νήματα τρέχουν ενώ στην κύρια εφαρμογή υπάρχει ένας βρόχος που περιμένει πότε θα τερματίσει το κύριο έργο (το νήμα του). Ενώ η ΚΑΘΕ δεν ξεκινάει νήμα και απλά κοιτάει ένα εσωτερικό ρολόι πόσο χρόνο χρειάζεται μετά το πέρασ των εντολών που έχει εκτελέσει για να συμπληρώσει μέχρι την επόμενη γύρα

που θα ξανατρέξει τις εντολές του μπλοκ. Αν μάλιστα ο χρόνος περάσει τον προβλεπόμενο τότε απλά αμέσως έρχεται ο νέρος γύρος εκτέλεσης. Μοιάζει με το Νημα αλλά δεν είναι ίδιος ο μηχανισμός!

δες το παράδειγμα στο EVERY

Αναγνωριστικό: ΚΑΛΕΣΕ [CALL]

Με την Κάλεσε καλούμε τμήματα και συναρτήσεις, δίνοντας τον τρέχον σωρό.

1) Κάλεσε [Κενή] [Συνάρτηση] ΟΝΟΜΑ [ΠΑΡΑΜΕΤΡΟΙ]

Με το Κενή λέμε στην Κάλεσε να αγνοήσει τυχόν επιστροφή από τη συνάρτηση.

Μπορούμε σε μια συνάρτηση να μην έχουμε επιστροφή ως εντολή αλλά πάντα γυρνάει το 0 ή το "" αν είναι αλφαριθμητική.

Με το Συνάρτηση απλά δεν δίνουμε τις παρενθέσεις, και παραθέτουμε τις παραμέτρους όπως στην Κάλεσε με τμήμα

Καθαρό \ \ καθαρίζει και τις στατικές

Αδειασε \ \ αδειάζει το σωρό

Συνάρτηση Δικήμας {

 Διάβασε A,B,Γ,Δ

 Τύπωσε B,Γ,Δ

 Αν A=1 Τότε {

 =0

 } Αλλιώς {

 =1

 }

}

\ \ με την λέξη Κενή αδιαφορεί για την επιστροφή

Κάλεσε Κενή Συνάρτηση Δικήμας(0,2,3,4)

Κάλεσε Κενή Συνάρτηση Δικήμας 0,2,3,4

Κάλεσε Κενή Συνάρτηση Δικήμας, 0,2,3,4

Δες ok {

 Κάλεσε Δικήμας(0,2,3,4)

}

Αν Όχι Ok Τότε Τύπωσε Λάθος\$

A\$="Δικήμας("

```

Κάλεσε Α$, 1,2,3,4
Κάλεσε Δικήμας(, 1,2,3,4
Βάλε 4,3,2,1 ' το ένα θα γίνει κορυφή του σωρού
Κάλεσε Δικήμας(
Γενική Χ=10
Τμήμα Αλφα {
    Τύπωσε "οκ", Χ
    \ \ υπάρχει τρόπος για αναδρομή
    Αν Χ>1 τότε Χ-- : Κάλεσε Αλφα
}
Αλφα

```

2) Παράδειγμα με την Κάλεσε Τοπικά

```

Χ=10
Συνάρτηση Αλφα {
    Χ++
    Τύπωσε Χ
}
Κάλεσε Τοπικά Αλφα() \ \ 11
Κάλεσε Τοπικά "Αλφα()" \ \ 12
Συνάρτηση Βήτα {
    Τοπική Χ=1000
    Κάλεσε Τοπικά Αλφα()
}
Κάλεσε Τοπικά Βήτα() \ \ 1001
Κάλεσε Τοπικά Αλφα() \ \ 13
Τύπωσε Χ \ \ 13
Συνάρτηση Δέλτα {
    Διάβασε Νέο &Χ
    Κάλεσε Τοπικά Αλφα()
}
Μ=1233
Κάλεσε Τοπικά Δέλτα(&Μ) \ \ 1234
Τύπωσε Μ \ \ 1234

```

3) Ειδική περίπτωση όταν το όνομα τμήματος είναι ίδιο με μια εντολή:

3.1) Όταν το τμήμα είναι τοπικό

```

Τμήμα Τύπωσε {
    Ενώ Όχι Κενό {
        Διάβασε Α
        Τύπωσε $(("\A\I\η\θ\έ\ς;\A\I\η\θ\έ\ς;\Psi\ε\I\υ\δ\έ\ς"), Α,$(""))
    }
}

```

```

    Αν Πλάτος-Θέση<Στήλη Τότε @Τυπωσε
    \ \ το @ μπροστά από εντολή καλεί οπωσδήποτε την εντολή!
  }
}
Κάλεσε Τύπωσε -1,0,3
.Τύπωσε -1,0,3
Τύπωσε
Τύπωσε -1,0,3
3.2) Όταν το τμήμα είναι γενικό
Τμήμα Γενικό Τύπωσε {
  Ενώ Όχι Κενό {
    Διάβασε A
    @Τύπωσε $("\A\l\η\θ\έ\ς;\A\l\η\θ\έ\ς;\Ψ\ε\l\υ\δ\ι\έ\ς"), A,$("")
    Αν Πλάτος-Θέση<Στήλη Τότε @Τυπωσε
  }
}
Τύπωσε -1,0,3
@Τύπωσε -1,0,3

```

4) Όταν καλούμε Γεγονός, δηλαδή να εκτελεστούν οι όποιες συναρτήσεις του, τότε χρησιμοποιούμε την Κάλεσε Γεγονός, δίνοντας το όνομα του γεγονότος και ακολουθούν οι όποιες παράμετροι που θα πάνε σε όλες τις συναρτήσεις (δέχεται και πέρασμα με αναφορά) Δείτε το παράδειγμα στην εντολή Γεγονός όπου ορίζουμε ένα γεγονός.

Αναγνωριστικό: ΚΥΚΛΙΚΑ [LOOP]

1) Επανάληψη (LOOP) με έλεγχο στο τέλος

```

{
εντολές

```

ΑΝ συνθήκη ΤΟΤΕ ΕΞΟΔΟΣ

```

ΚΥΚΛΙΚΑ ' Προκαλεί την επανεκτέλεση αυτού του τμήματος (μεταξύ των δυο αγκυλών)
}

```

2) Επανάληψη (LOOP) με έλεγχο στην αρχή (όπως η ΕΝΩ), εδώ όμως μπορούμε να έχουμε πολλαπλές εξόδους (και άλλα Αν με ΕΞΟΔΟ)

```

{

```

```
ΑΝ συνθήκη ΤΟΤΕ ΕΞΟΔΟΣ
εντολές
ΚΥΚΛΙΚΑ
}
```

3) Και σε ένα τμήμα μπορούμε να δώσουμε την εντολή ΚΥΚΛΙΚΑ. Και ένα τμήμα έχει και αυτό αγκύλες (αν και από τον διορθωτή της M2000 (δες ΣΥΓΓΡΑΦΗ) δεν εμφανίζονται αυτές)

```
ΤΜΗΜΑ τάδε {
....
ΚΥΚΛΙΚΑ
....
}
```

Χρησιμοποιούμε την ΚΥΚΛΙΚΑ αντί για την ΓΙΑ όταν θέλουμε το βήμα να είναι μεταβλητό ή όταν θέλουμε έξοδο με συνθήκη. Ουσιαστικά και στις ΕΝΩ και ΕΠΑΝΑΛΑΒΕ μπορούμε να έχουμε έξοδο με συνθήκη κάπου μέσα, αλλά η ΚΥΚΛΙΚΑ δεν κάνει κάτι περισσότερο γιατί όπως και να έχει θα γίνει ο έλεγχος της συνθήκης τόσο στην ΕΠΑΝΑΛΑΒΕ όσο και στην ΕΝΩ ενώ οι εντολές μετά την ΚΥΚΛΙΚΑ θα συνεχιστούν (απλή η ΚΥΚΛΙΚΑ δίνει ένα σύνθημα στο πέρας του τμήματος εντολών να επαναληφθεί, δεν λειτουργεί άμεσα)

```
Π.Χ.
Ε
Ι=0
ΕΠΑΝΑΛΑΒΕ {
ΞΑΝΑ:
Ι=Ι+1
Α$=ΕΝΚΟΜ$
ΚΥΚΛΙΚΑ ' Σβήσε την εντολή αυτή δεν κάνει κάτι εδώ!
ΑΝ Α$="" ΤΟΤΕ ΠΡΟΣ ΞΑΝΑ
ΑΝ Α$=" " ΤΟΤΕ ΕΞΟΔΟΣ
ΤΥΠΩΣΕ Ι
} ΜΕΧΡΙ Ι=1000
```

```
Αναγνωριστικό: ΚΥΡΙΟ.ΕΡΓΟ [MAIN.TASK]
ΚΥΡΙΟ.ΕΡΓΟ 100 {

}
```

Είναι εντολή όπως η ΚΑΘΕ αλλά με την διαφορά ότι αποτελεί νήμα, και έτσι διαχειρίζεται το χρόνο ισότιμα με τα νήματα που τρέχουν. Μπορούμε να βγούμε με την εντολή ΕΞΟΔΟΣ ή με ESC. Δεν μπορούμε να έχουμε φωλιασμένα κύρια έργα στο ίδιο τμήμα

Αναγνωριστικό: ME [CASE]
ΔΕΣ ΕΠΙΛΕΞΕ ME

Αναγνωριστικό: ΜΕΡΟΣ [PART]
αν το M είναι Αληθές τότε το Μέρος {} Ως M δεν εκτελείται
αν δεν υπάρχει M τότε το φτιάχνει και όταν τρέχει το μπλοκ το M γίνεται Αληθές.
Θα αλλάξει ακριβώς στο τέλος της εκτέλεσης σε ψευδές
Μερος {
 Τύπωσε "οκ", M
} Ως M

χρησιμοποιείται όπου δεν θέλουμε κώδικας να εκτελεστεί ταυτόχρονα με άλλο κώδικα.
μπορούμε να καλούμε ρουτίνες ή τμήματα από το μπλοκ κανονικά.

Αναγνωριστικό: META [AFTER]

1) META 100

Όπως η ANAMONH 100

2)META 1000 {

ΕΝΤΟΛΕΣ

....

}

Εδώ έχουμε ένα ΝΗΜΑ που θα εκτελεστεί μια φορά σε 1 δευτερόλεπτο. Αν τερματίσει το τμήμα που την κάλεσε και δεν έχει εκτελεστεί τότε δεν θα εκτελεστεί, όπως όλα τα ΝΗΜΑΤΑ έτσι και αυτό δεν θα υπάρχει με το πέρας του τμήματος που το δημιουργήσε.

Μικρό παράδειγμα:

Αντέγραψε το παρακάτω σε ένα τμήμα:

K=100

μετα 1000 {

τυπωσε "οκ", K

}

τμήμα ΑΛΛΟ {

μετα 500 {τυπωσε "Είμαι στο ΑΛΛΟ"}

αναμονη 2000

τυπωσε "Είμαι ακόμα στο ΑΛΛΟ"

}

ΑΛΛΟ

Εδώ βλέπουμε ότι όταν τρέχει το τμήμα ΑΛΛΟ πρώτα έρχεται το σύντομο ΝΗΜΑ του ΑΛΛΟ και μετά κάπου στο χρόνο που δίνει η ΑΝΑΜΟΝΗ εμφανίζεται το ΝΗΜΑ του τμήματος που κάλεσε το ΑΛΛΟ και εμφανίζει και την μεταβλητή K. Γιατί τα νήματα διαβάζουν και φτιάχνουν αν θέλουμε μεταβλητές στο τμήμα που ανήκουν.

Ακόμα και να χαθεί το ΝΗΜΑ, δεν χάνονται οι μεταβλητές που φτιάχνει, παρά μόνο αν χαθεί το ΤΜΗΜΑ (επειδή είναι μέρος του τμήματος και όχι άλλο τμήμα).

Μπορούμε να ξέρουμε αν μια μεταβλητή υπάρχει πριν την χρησιμοποιήσουμε με την ΕΓΚΥΡΟ()

Μπορούμε να δοκιμάζουμε κώδικα αν έχει λάθος με την ΔΕΣ μεταβλητή { ΕΝΤΟΛΕΣ } και μετά ελέγχουμε την μεταβλητή.

Μεγαλύτερο παράδειγμα:

Εδώ γίνεται χρήση του ΑΝΑΛΥΤΗ φόρτου. Φόρτος είναι χρόνος που κάνει να εκτελέσει μια ομάδα εντολών της M2000 ο υπολογιστής. Ο υπολογιστής δεν κουράζεται και η απόδοσή του μετριέται με το πόσο χρόνο κάνει να εκτελέσει συγκεκριμένη εργασία.

ΟΘΟΝΗ 0, 0 \ \ το πρώτο 0 είναι το μαύρο χρώμα για φόντο, μετά ακυρώνουμε με το δεύτερο 0 την χωριστή οθόνη

ΠΕΝΑ 14 \ \ είναι το κίτρινο. από 0 έως 15 είναι τα παλιά χρώματα της QBasic.

\ \ μπορεί να βάλει κανείς χρώμα όπως στην Html, απαιτεί 6 δεκαεξαδικά ψηφία και το # στην αρχή

\ \ ΠΕΝΑ #FF77FF

K=ΨΕΥΔΕΣ

' ΕΣΩΤΕΡΙΚΟ ΤΜΗΜΑ

ΤΜΗΜΑ ΑΣΤΕΡΙΑ {

ΟΘΟΝΗ

ΓΙΑ I=1 ΕΩΣ 200 {

X=ΤΥΧΑΙΟΣ(0,ΠΛΑΤΟΣ-1)

```

        Y=ΤΥΧΑΙΟΣ(0,ΥΨΟΣ-2)
        ΤΥΠΩΣΕ @(X,Y),"*";
    }
}
' ΠΑΡΑΚΑΤΩ ΞΕΚΙΝΑΜΕ ΤΟΝ ΑΝΑΛΥΤΗ
' Ο ΟΠΟΙΟΣ ΜΗΔΕΝΙΖΕΙ ΤΟΝ ΜΕΤΡΗΤΗ ΧΡΟΝΟΥ Ή ΦΟΡΤΟ
' ΜΕΤΡΑΜΕ ΤΗ ΔΙΑΔΙΚΑΣΙΑ ΩΣ ΦΟΡΤΙΟ ΧΡΟΝΟΥ
' ΘΕΛΟΥΜΕ ΤΟ ΤΜΗΜΑ ΚΑΙ ΤΗΝ ΕΝΤΟΛΗ ΕΞΟΔΟΥ
ΑΝΑΛΥΤΗΣ
ΑΣΤΕΡΙΑ
ΑΝ ΨΕΥΔΕΣ ΤΟΤΕ {}
Λ=ΦΟΡΤΟΣ+10
' ΤΩΡΑ ΞΕΡΟΥΜΕ ΤΟ ΦΟΡΤΟ (ΒΑΖΟΥΜΕ ΚΑΙ ΛΙΓΟ ΠΑΡΑΠΑΝΩ)
' ΚΑΙ ΣΤΕΛΝΟΥΜΕ ΕΝΑ ΜΕΡΟΣ ΤΜΗΜΑΤΟΣ ΝΑ ΤΡΕΞΕΙ
' ΜΕΤΑ ΑΠΟ 10*Λ ΧΙΛΙΟΣΤΑ ΤΟΥ ΔΕΥΤΕΡΟΛΕΠΤΟΥ
ΜΕΤΑ 10*Λ {Κ=ΑΛΗΘΕΣ}
ΚΥΡΙΟ.ΕΡΓΟ Λ {
    ΑΣΤΕΡΙΑ
    ΑΝ Κ ΤΟΤΕ ΕΞΟΔΟΣ
}
'ΤΙΣ ΠΕΡΙΣΣΟΤΕΡΕΣ ΦΟΡΕΣ ΘΑ ΔΟΥΜΕ ΔΕΚΑ ΔΙΑΦΟΡΕΤΙΚΕΣ
'ΣΥΣΤΑΔΕΣ ΑΣΤΕΡΙΩΝ ΚΑΙ ΑΥΤΟ ΓΙΑΤΙ ΚΑΜΙΑ ΦΟΡΑ
'Ο ΥΠΟΛΟΓΙΣΤΗΣ ΤΡΩΕΙ ΧΡΟΝΟ..ΑΠΟ ΤΗΝ ΔΙΑΔΙΑΚΑΣΙΑ
ΟΘΟΝΗ 7
ΠΕΝΑ 1

```

Αναγνωριστικό: ΜΕΧΡΙ [UNTIL]
 Μέρος της δομής
 Επαναλαβε {

} Μεχρι A>2

Αναγνωριστικό: ΞΕΚΙΝΑ [RESTART]

Σε μια ομάδα εντολών η ΞΕΚΙΝΑ κάνει επανάληψη από την αρχή χωρίς να ολοκληρώσει την ομάδα (όπως κάνει το Κυκλικά)

Παράδειγμα

```
i=0
{
i++
αν τυχαίος(10)>5 τότε ξεκίνα
τυπώσε "κανονικά", i
αν i<10 τότε ξεκίνα
τυπώσε "τελευταία εντολή"
}
```

Αν αλλάξουμε το ξεκίνα με το κυκλικά τότε θα έχουμε άλλη απόκριση, θα τυπωθούν όλα τα i μέχρι και το 10 καθώς και το "τελευταία εντολή" θα τυπωθεί δέκα φορές!

Δες ΝΗΜΑ και ΤΑΙΝΙΑ όπου το αναγνωριστικό ΞΕΚΙΝΑ είναι διαφορετικό από αυτό εδώ.

Αναγνωριστικό: ΟΣΟ [WHEN]

Δες ΕΠΑΝΑΛΑΒΕ

i=0

Επανάλαβε ' ή Επανάλαβε

i++

Τύπωσε i

Όσο i<10

Αναγνωριστικό: ΠΡΟΣ [GOTO]

Αλλαγή ροής (άλμα) προς οποιαδήποτε διεύθυνση, σε ένα αυτοτελές τμήμα (να έχει αγκύλες)

ΠΡΟΣ ετικέτα

κάπου μέσα στο τμήμα και πάντα σε αρχή γραμμής θα μπει η ετικέτα

ΕΤΙΚΕΤΑ:

ή ένας αριθμός (μέχρι πέντε ψηφία και όχι 0 μπροστά)

προσέξτε την άνω και κάτω τελεία, μετά από αυτή πρέπει να ακολουθεί αλλαγή γραμμής. Κάθε ετικέτα πρέπει να έχει το πρώτο γράμμα ως γράμμα και όχι αριθμό, όπως οι μεταβλητές.

Στο παρακάτω πρόγραμμα χρησιμοποιούμε την ΠΡΟΣ για να επαναλάβουμε ένα τμήμα σε ένα τμήμα!

```
I=0
ΕΠΑΝΑΛΑΒΕ {
ΞΑΝΑ:
  I=I+1
  A$=ENKOM$
  AN A$="" ΤΟΤΕ ΠΡΟΣ ΞΑΝΑ
  AN A$=" " ΤΟΤΕ ΕΞΟΔΟΣ
  ΤΥΠΩΣΕ I
} ΜΕΧΡΙ I=1000
```

Μπορούμε να καλέσουμε εδώ την Συνεχεια (στη θέση της ΠΡΟΣ ΞΑΝΑ) και είναι το ίδιο. Αλλά μπορεί να θέλουμε διαφορετικό σημείο εισόδου και εκεί είναι χρήσιμη η ΠΡΟΣ

Δες την ΑΠΟ Μεταβλητή ΠΡΟΣ Ετικετα1, Ετικετα2...

Αναγνωριστικό: ΣΥΝΕΧΙΣΕ [CONTINUE]

1. Σε ένα μπλοκ εντολών πηδάει στο τέλος και αν υπάρχει επανάληψη συνεχίζει από την αρχή της. Σε αντίθεση με την ΕΞΟΔΟΣ η οποία τερματίζει την επανάληψη άμεσα.
2. Σε ένα μπλοκ σε μια ΜΕ (δες ΕΠΙΛΕΞΕ ΜΕ) συνεχίζει την ροή μετά το ΤΕΛΟΣ ΕΠΙΛΟΓΗΣ (συνδυάζεται με το ΔΙΕΚΟΨΕ). Μια ΕΞΟΔΟΣ απλά τερματίζει τι μπλοκ (θα πετάξει τις υπόλοιπες εντολές) αλλά δεν διώξει την ΔΙΕΚΟΨΕ (οπότε αν συνεχίζουν τα ΜΕ θα εκτελεστούν και αυτά, χωρίς να εκτελεστούν οι έλεγχοι συνθηκών).

```
a=5
b=10
select case A
case 1, 5 to 20
  { print "ok"
    if b=10 then break
  }
}
```

```

case 2,4
    {print "and ok"
     continue
    }
case 3
    { print "that"
      print "and that"
    }
else
    print "??"
end select

```

```

for i = 1 to 1000 {
    if i> 10 then exit
    print i
}
print i ' 11

```

```

for i = 1 to 1000 {
    if i> 10 then continue
    print i
}
print i ' 1001

```

Αναγνωριστικό: ΣΧΕΔΙΟ.ΝΗΜΑΤΩΝ [THREAD.PLAN]

1) ΣΧΕΔΙΟ.ΝΗΜΑΤΩΝ ΤΑΥΤΟΧΡΟΝΟ

Εκτέλεση μια εντολή από κάθε νήμα. Αν οι εντολές είναι σε μπλοκ {} τότε εκτελείται όλο το μπλοκ. Η εντολή ΑΝ καλό είναι να μπει σε παρενθέσεις αν έχουμε και την Αλλιώς.

2) ΣΧΕΔΙΟ.ΝΗΜΑΤΩΝ ΔΙΑΔΟΧΙΚΟ \ \ ΕΞ ΟΡΙΣΜΟΥ

Εξ ορισμού αυτό είναι το σχέδιο των νημάτων. Κάθε φορά ένα νήμα ολοκληρώνει τις εντολές του και δίνει τη θέση του σε άλλο.

Δεν μπορούμε να αλλάξουμε σψέδιο αν έχουμε νήματα στη λίστα νημάτων (threading pool)

Αναγνωριστικό: ΤΟΤΕ [THEN]

ΔΕΣ ΑΝ

Ομάδα: ΕΝΤΟΛΕΣ ΣΩΡΟΥ - STACK COMMANDS

Αναγνωριστικό: ΑΔΕΙΑΣΕ [FLUSH]

ΑΔΕΙΑΣΕ ' Αδειάζει ο σωρός

Υπάρχουν δυο παραλλαγές:

ΑΔΕΙΑΣΕ ΛΑΘΟΣ ' καθαρίζει το τελευταίο μήνυμα λάθους
και

\\ Δεν χρησιμοποιείται από την 9.4 ΑΔΕΙΑΣΕ ΣΚΟΥΠΙΔΙΑ ' ενεργοποιεί άμεσα ον συλλέκτη σκουπιδιών

Ο σωρός δεν σβήνει παρά μόνο με δυο τρόπους:

- 1) Τερματισμός του μεταφραστή (γιατί απλά δεν υπάρχει μετά)
- 2) Με την ΑΔΕΙΑΣΕ

Μπορούμε να δούμε αν είναι ΚΕΝΟΣ με την μεταβλητή ΚΕΝΟ (EMPTY) που παίρνει τιμές -1 και 0 (ΑΛΗΘΕΣ, ΨΕΥΔΕΣ)

Επίσης αν δεν είναι ΚΕΝΟ δηλαδή έχει τιμές μπορούμε να δούμε αν είναι αριθμός ή γράμματα με τις ΕΙΝΑΡ (ISNUM) και ΕΙΝΓΡ (ISLET)

Οι τιμές του σωρού διαβάζονται με δυο τρόπους: με την εντολή ΔΙΑΒΑΣΕ αλλά και με τις μεταβλητές ΤΙΜΗ και ΓΡΑΜΜΑ\$ (αν όμως έχουμε αριθμό και γράψουμε Β\$=ΓΡΑΜΜΑ\$ τότε θα βγει ΛΑΘΟΣ)

Από την έκδοση 6 έχουμε και άλλο τρόπο να δούμε το σωρο:

Βαλε 1,2,"αα"

Τυπωσε Φακελος\$()

μας δίνει

SNN

Που σημαίνει αλφαριθμητικό, αριθμός, αριθμός

Αναγνωριστικό: ΑΝΑΘΕΣΕ [COMMIT]

η εντολή Ανάθεσε λειτουργεί σαν την Διάβασε αλλά για ειδικό σκοπό

Ενώ η Διάβασε δεν βάζει δεύτερη αναφορά σε υπάρχουσα αναφορά, η Ανάθεσε το κάνει αν το στοιχείο είναι μέλος ομάδας.

```

Ομάδα άλφα {
    Πίνακας α()
    Τμήμα κ {
        Ανέθεσε .α()
    }
}
Πίνακες μ(20)=1, μ1(30)=100
Άλφα.κ &μ()
Τύπωσε Άλφα.α(19)=1
Άλφα.κ &μ1()
Τύπωσε Άλφα.α(19)=100

```

Αναγνωριστικό: ΑΠΕΔΩΣΕ [REFER]

Η απέδωσε είναι μια παραλλαγή της Διάβασε, που φτιάχνει αναφορές χωρίς να χρειάζεται το & στα ονόματα. Επίσης χρησιμοποιείται για να φτιάχνουμε αναφορές σε μέλη ομάδων. Οι αναφορές είναι μιας χρήσης, δεν μπορούμε να δώσουμε άλλη αναφορά, εκτός και αν το κάνουμε αυτό σε ένα μπλοκ προσωρινών ορισμών (Για αντικείμενο { }) όπου κάθε φορά που βγαίνουμε από το μπλοκ οι αναφορές σβήνονται, και έτσι ο διερμηνευτής μπορεί να τις φτιάξει με νέα είσοδο στο μπλοκ.

```

Ομάδα Άλφα {
Ιδιωτικό:
    μ=1, λ$="ok"
Δημόσιο:
    Τμήμα Τύπωσε {
        Τύπωσε .μ, .λ$
    }
}
Άλφα.Τύπωσε
\\ τώρα θα απδώσουμε μεταβλητές στις
\\ ιδιωτικές που δεν έχουμε κανονικά θέαση
Απέδωσε από Άλφα, μ1, λ1$
Τύπωσε μ1, λ1$
μ1++
λ1$+="...."
Άλφα.Τύπωσε

```

Αναγνωριστικό: ΒΑΛΕ [PUSH]

Στο σωρό βάζουμε τιμές αριθμητικές και αλφαριθμητικές όπως θέλουμε.

Από το σωρό μπορούμε να διαβάσουμε τιμές με την ΔΙΑΒΑΣΕ

```
B$="AAAA"
```

```
ΒΑΛΕ 12,B$,34+45
```

```
ΣΩΡΟΣ
```

Το τελευταίο θα βγει πρώτο: Ουρά τύπου LIFO

Εκτός από την ΒΑΛΕ μπορούμε να βάλουμε και με άλλο τρόπο τιμές στο Σωρό!

Με την ΣΕΙΡΑ όπου το πρώτο που θα μπει θα είναι το πρώτο που θα βγεί (FIFO τύπος)

Με την ΣΩΡΟΣ ΑλφαριθμητικήΜεταβλητή\$ όπου έχουμε βάλει σε ένα Αλφαριθμητικό μια δομή ΣΩΡΟΥ

Σημαντικό:

Πέρασμα μεταβλητών στο Σωρό με αναφορά στη τιμή!

Μέχρι και την έκδοση 5 οι μεταβλητές άφηναν τις τιμές τους στον σωρό και με ΔΙΑΒΑΣΕ μπορούσαμε να πάρουμε αυτές τις τιμές σε άλλες μεταβλητές. Αυτό ήταν χρήσιμο για να βάλουμε παράμετρους σε ένα τμήμα ή συνάρτηση. Δεν γινόταν να περάσει πίνακας άμεσα, αφού θα έπρεπε να περάσουν όλες οι τιμές και μετά να αντιγραφούν από το σωρό όλες οι τιμές στο πίνακα πάλι από εκεί που καλέσαμε το τμήμα ή τη συνάρτηση.

Στην έκτη έκδοση το θέμα λύθηκε. Μπορούμε να περνάμε πίνακες και μεταβλητές γενικότερα με δείκτες στο σωρό. Ο δείκτης λέει στη ΔΙΑΒΑΣΕ να φτιάξει μια μεταβλητή χωρίς να της δώσει νέο χώρο, αλλά να πάρει αυτόν που δείχνει ο δείκτης.

Χρησιμοποιούμε το χαρακτήρα & μόνο στις ΒΑΛΕ, ΣΕΙΡΑ και ΔΙΑΒΑΣΕ καθώς και στις κλήσεις των Τμημάτων και των Συναρτήσεων

```
ΠΙΝΑΚΑΣ ππ(10)
```

```
ππ(4)=1234
```

```
ΒΑΛΕ &ππ()
```

```
ΔΙΑΒΑΣΕ &κλ() ' Εδώ φτιάξαμε τον πίνακα κλ() που δείχνει στον ίδιο χώρο, αυτό του ππ()
```

```
ΤΥΠΩΣΕ κλ(4)
```

```
ονομα$="Κολόμβος"
```

```
ΤΜΗΜΑ ΑΑ {
```

```
    ΔΙΑΒΑΣΕ &κκ(), &κ$
```

```
    κ$="Ωραίος ο "+κ$ 'Αλλάζουμε τις τιμές για να δούμε ότι θα αλλάξουν
```

```
    κκ(4)=κκ(4)*2      'και έξω από το τμήμα!
```

```
}
```

```
ΑΑ &ππ(), &ονομα$
```

```
print ονομα$  
print ππ(4)
```

Επίσης με την προσθήκη αυτή φάνηκε και μια άλλη ιδιότητα που δεν είχε προβλεφθεί. Οι πίνακες και οι μεταβλητές εσωτερικά έχουν τιμές τύπου variant (ειδικός τύπος που υποστηρίζουν τα Windows). Δηλαδή μπορεί κάθε στοιχείο να είναι ή αριθμός ή αλφαριθμητικό. Όμως επειδή ο διερμηνευτής διαβάζει τον τύπο του πίνακα από την παρουσία των % και \$ (και εξ ορισμού χωρίς αυτά είναι αριθμητική τιμή), δίνει τον τύπο που ζητάμε. Με την μεταβλητή που δημιουργούμε ως αναφορά στη τιμή μπορούμε να αλλάξουμε τύπο:

```
πίνακας α(20) 'ο πίνακας πέρνει αριθμητικές τιμές  
βαλε &α() 'βάζουμε το δείκτη του στο σωρό  
διαβασε &γ$() 'δημιουργούμε έναν πίνακα με ίδιες τιμές με τον α() αλλά με άλλο τύπο  
γ$(4)="1023" 'βάζουμε αλφαριθμητικό και εδώ βάζουμε αριθμό (θα δούμε γιατί)  
τυπωσε α(2), γ$(4), α(4)
```

Τι μας τυπώνει τώρα:

το α(2) είναι 0

το γ\$(4) είναι το στοιχείο α(4) αλλά τυπώνεται ως αλφαριθμητικό

το α(4) έχει τον αριθμό 1023 γιατί εσωτερικά ο διερμηνευτής διάβασε την τιμή του α(4) ως αριθμό

αν είχαμε δώσει γ\$(4)="αααα" θα μας έδινε τιμή στο α(4) το μηδέν

Άλλο παράδειγμα με μεταβλητές:

```
α=1  
βαλε &α  
διαβασε &γ$  
γ$="1023"  
τυπωσε α, γ$
```

```
β$="1234"  
βαλε &β$  
διαβασε &δ  
τυπωσε δ
```

Αναγνωριστικό: ΔΙΑΒΑΣΕ [READ]

ΔΙΑΒΑΣΕ Α\$,Β\$,Κ\$

Διαβάζει τις τιμές και τις καταχωρεί σε κάθε μεταβλητή και αν η μεταβλητή δεν υπάρχει την δημιουργεί (εκτός από πίνακες που πρέπει να δημιουργηθούν με την ΠΙΝΑΚΑΣ)

ΔΙΑΒΑΣΕ I(1),I(2),I(3)

Δεν διαβάζει μονομιάς έναν πίνακα αλλά ένα στοιχείο

ΠΙΝΑΚΑΣ A(I)

ΒΑΛΕ 1,2,3,4,5,6,7,8,9,10

ΓΙΑ I=0 ΕΩΣ 9 { ΔΙΑΒΑΣΕ A(I) }

Η ΔΙΑΒΑΣΕ έχει και μια άλλη χρήση:

Καλούμε ένα τμήμα με το παρακάτω:

ΑΛΦΑ 12, "αασαα", Γ\$

Στον ορισμό του ΑΛΦΑ

ΤΜΗΜΑ ΑΛΦΑ {

ΔΙΑΒΑΣΕ K, A\$, B\$

.....

}

Αν υπάρχουν και άλλα στοιχεία στο σωρό το τμήμα ΑΛΦΑ τα βλέπει, δεξ το ΝΗΜΑ όμως όπου το ΝΗΜΑ βλέπει τις μεταβλητές του τμήματος που το δημιούργησε αλλά όχι το σωρό του, γιατί έχει δικό του σωρό.

Επίσης και οι συναρτήσεις που φτιάχνουμε έχουν το δικό τους σωρό.

ΤΜΗΜΑ ΑΛΦΑ {

ΔΙΑΒΑΣΕ X, Y

}

ΒΑΛΕ 1,2

ΑΛΦΑ 'είναι τμήμα

' θα διαβάσει το 2 θα το βάλει στο X (αν δεν υπάρχει το δημιουργεί), θα διαβάσει το 1 θα το βάλει στο Y (ομοίως δημιουργεί αν δεν το βρίσκει)

ΣΕΙΡΑ 1,2

' αν δεν υπάρχει από πριν τίποτα στο σωρό τότε το 1 θα γίνει κορυφή και το 2 θα ακολουθεί ΑΛΦΑ

' θα διαβάσει το 1 και μετά το 2

ΑΛΦΑ 1,2

' έτσι δεν μας νοιάζει τι έχει ο σωρός, όταν κληθεί θα υπάρχει μια ΣΕΙΡΑ στην κορυφή.

Επειδή μπορούμε να περνάμε μεταβλητές με αναφορά, μπορούμε να περάσουμε ολόκληρο πίνακα και μάλιστα να του αλλάξουμε την διάσταση και να τον επιστρέψουμε αλλαγμένο

ΠΙΝΑΚΑΣ α(1)

ΤΜΗΜΑ ΑΛΦΑ {

ΔΙΑΒΑΣΕ &K()

ΠΙΝΑΚΑΣ K(10,10)

K(4,4)=2
}
ΑΛΦΑ &α()
ΤΥΠΩΣΕ Α(4,4)

Αναγνωριστικό: ΠΑΝΩ [OVER]

1) ΠΑΝΩ 5

2) ΠΑΝΩ

3) ΠΑΝΩ 5,5

αντιγράφει στη κορυφή του σωρού το στοιχείο στη 5η θέση

η δεύτερη περίπτωση αντιγράφει την κορυφή, οπότε έχουμε δυο στοιχεία ίδια στις δυο πρώτες θέσεις

η τρίτη περίπτωση βγάζει αντίγραφα στα πέντε πρώτα στοιχεία (εκτελεί πέντε φορές το ΠΑΝΩ 5)

Δες ΦΕΡΕ, ΠΑΝΩ, ΠΕΤΑ, ΤΙΜΗΣΩΡΟΥ(),ΤΙΜΗΣΩΡΟΥ\$(),
ΑΡΙΘΜΟΣ, ΓΡΑΜΜΑ\$, ΕΙΝΑΡ, ΕΙΝΓΡ, ΚΕΝΟ, ΑΔΕΙΑΣΕ, ΣΩΡΟΣ

Αναγνωριστικό: ΠΕΤΑ [DROP]

ΠΕΤΑ 4

αν υπάρχουν 4 στοιχεία στο σωρό τα πετάει αλλιώς βγαίνει λάθος

Δες ΜΕΓΕΘΟΣ.ΣΩΡΟΥ (μεταβλητή που δίνει τον αριθμό των στοιχείων στο σωρό)

ΠΕΤΑ 0

δεν βγαίνει λάθος, και δεν κάνει τίποτα.

Αν δώσουμε μικρότερο από το 0 τότε βγαίνει λάθος

Μπορούμε με την ΠΕΤΑ, το ΜΕΓΕΘΟΣ.ΣΩΡΟΥ και τα ΤΙΜΗΣΩΡΟΥ() και ΤΙΜΗΣΩΡΟΥ\$() να φτιάξουμε πλαίσια παραμέτρων στο σωρό. Έτσι με αυτόν τον τρόπο κρατάμε τιμές που δεν αποδίδουμε σε μεταβλητές και καλούμε τμήματα χωρίς να πειράζουμε το σωρό. Απλά χρησιμοποιούμε μια μεταβλητή Σ για να μαρκάρουμε το μέγεθος του σωρού στην είσοδο ενός τμήματος και στο τέλος του να βάλουμε μια ΠΕΤΑ ΜΕΓΕΘΟΣ.ΣΩΡΟΥ-Σ, για να διορθώσουμε το σωρό σε αυτόν που ήταν όταν μπήκαμε στο τμήμα!

Αναγνωριστικό: ΣΕΙΡΑ [DATA]
Όπως το ΒΑΛΕ αλλά
Το πρώτο θα βγει πρώτο: Ουρά τύπου FIFO
(πριν από τις αλλαγές χρήσεως καλό είναι να αδειάζει ο σωρός)

ΑΔΕΙΑΣΕ
ΠΙΝΑΚΑΣ Α(Ι)
ΣΕΙΡΑ 1,2,3,4,5,6,7,8,9,10
ΓΙΑ Ι=0 ΕΩΣ 9 { ΔΙΑΒΑΣΕ Α(Ι) }

Βλέπουμε τα περιεχόμενα του σωρού με την ΣΩΡΟΣ
επίσης με την ΔΟΚΙΜΗ ανοίγει ένα πλαίσιο διαλόγου όπου φαίνεται άμεσα ο σωρός

Αναγνωριστικό: ΣΩΡΟΣ [STACK]

1) ΣΩΡΟΣ

Μας εμφανίζει τα στοιχεία του σωρού χωρίς να ανασύρει κάποιο.

2) ΣΩΡΟΣ Α\$

Καταχωρεί στο σωρό το περιεχόμενο σωρού που έχουμε φτιάξει στο Α\$ δεξ ΣΩΡΟΣ\$(
και αδειάζει το Α\$

3) ΣΩΡΟΣ Α\$, "NN"

δίνουμε εντολή να βγουν από το σωρό Α\$ δυο αριθμοί και να μπουν στο Σωρό του τμήματος ή
της συνάρτησης (Τα τμήματα έχουν κοινό σωρό ενώ οι συναρτήσεις και τα νήματα έχουν το δικό
τους)

Ουσιαστικά το "NN" είναι ίδιο με το Φακελος\$(Σωρος\$(2, 20.5))

Ο Σωρός αδειάζει ή διαβάζοντας τις τιμές ή με την εντολή ΑΔΕΙΑΣΕ FLUSH

Μπορείς να αντιγράψεις αυτές τις εντολές σε ένα τμήμα που θα ανοίξεις με την Σ Α (έστω Α το
τμήμα), το Σ είναι συντομογραφία της εντολής ΣΥΓΓΡΑΦΗ, ή μπορείς να δώσεις μια προς μια τις
εντολές για να δεις το αποτέλεσμα στον διερμηνευτή γραμμής εντολών

ΠΙΝΑΚΑΣ Α\$(10)

Α\$(2)=ΣΩΡΟΣ\$(1,2,"OK")

ΣΩΡΟΣ Α\$(2)

ΣΩΡΟΣ

ΤΥΠΩΣΕ ΜΗΚΟΣ(Α\$(2))

4) ΣΩΡΟΣ ΕΚΦΡΑΣΗ_ΑΛΦΑΡΙΘΜΗΤΙΚΗ

Αν δεν ξεκινάει η έκφραση από υπάρχουσα αλφαριθμητική μεταβλητή ή πίνακα, τότε
επεξεργάζεται ο διερμηνευτής την έκφραση και αφού κάνει έλεγχο ότι είναι λίστα σωρού τότε το
τοποθετεί στο σωρό.

5) ΣΩΡΟΣ ΝΕΟΣ

ΑΔΕΙΑΣΕ

```

ΣΕΙΡΑ 1,2,3,4
ΣΩΡΟΣ ΝΕΟΣ {
    ΒΑΛΕ 10,20
    ΣΕΙΡΑ 1,2
    ΣΩΡΟΣ
}
ΣΩΡΟΣ

```

6) Μεταβλητή Τύπου Σωρού - Μπορούν να μπουν οπουδήποτε, σε πίνακες, σε καταστάσεις, σε άλλους σωρούς.

```

α=ΣΩΡΟΣ
α=ΣΩΡΟΣ:=1,2,3 ' ότι θέλουμε ακόμα και σωρό με (Σωρός:=1,2,3)
β=ΣΩΡΟΣ ΑΝΩ Α, 2 ' θα βάλει τα 1,2 στο β και θα αφήσει το 3 στο α
α=ΣΩΡΟΣ:=1,2,3
β=ΣΩΡΟΣ ΚΑΤΩ Α, 2 ' θα βάλει τα 2,3 στο β και θα αφήσει το 1 στο α
Τύπωσε β ' 2,3
α=ΣΩΡΟΣ:=!α,!β
Τύπωσε α ' 1,2,3
\\ Χρήση του ! στο όρισμα για να περάσει το αντικείμενο σωρός ως ορίσματα στη συνάρτηση
Συνάρτηση β (α,β) {
    =α**β
}
σ1=Σωρός:=3,2
σ2=Σωρός:=100,2
Τύπωσε β(2,4), β(!σ1) , β(!Σωρός(σ2))
Τύπωσε Μήκος(σ1), Μήκος(σ2)

```

\\ Παράδειγμα

```

Α=ΣΩΡΟΣ \\ άδειος σωρός
ΣΩΡΟΣ Α { ΒΑΛΕ 1,2,3 } ' 3 2 1
ΣΩΡΟΣ Α { ΣΕΙΡΑ 1,2,3 } ' 3 2 1 1 2 3
ΑΔΕΙΑΣΕ ' Αδειάζουμε το σωρό του τμήματος
ΣΕΙΡΑ 1,2,3 ' βάζουμε στο σωρό του τμήματος 1 2 3
ΣΩΡΟΣ Α ' Αδειάζουμε τον Α στο σωρό τμήματος 1 2 3 3 2 1 1 2 3
ΣΩΡΟΣ ' Μας δείχνει το σωρό
Α=[] ' νέος σωρός στον Α με ότι έχει ο σωρός τμήματος. Ο σωρός τμήματος άδειασε
Τύπωσε Α
ΣΩΡΟΣ ΣΩΡΟΣ(Α,3) ' δίνουμε ένα αντίγραφο από τα τρία πρώτα
ΣΩΡΟΣ
ΤΥΠΩΣΕ Α ' ο Α δεν άδειασε
ΑΔΕΙΑΣΕ
ΣΕΙΡΑ 1,2,3,4
Α=ΣΩΡΟΣ:=20,10,1,2
ΣΩΡΟΣ Α {

```

```

    ΣΩΡΟΣ
    ΠΕΤΑ 2
    ΣΩΡΟΣ
}
ΣΩΡΟΣ
ΤΥΠΩΣΕ ΤΙΜΗΣΩΡΟΥ(A), ΜΗΚΟΣ(A) \ 1 2
ΣΩΡΟΣ A {
    ΔΙΑΒΑΣΕ X, Y
    ΓΙΑ ΑΥΤΟ { \ ΠΡΟΣΩΡΙΝΟΙ ΟΡΙΣΜΟΙ
        ΠΙΝΑΚΑΣ M(10)=5
        ΒΑΛΕ M()
    }
}
N=ΣΩΡΟΣ(A) \ ΑΝΤΙΓΡΑΦΟ
ΣΩΡΟΣ N {
    ΓΙΑ ΑΥΤΟ {
        ΔΙΑΒΑΣΕ K \ ΠΙΝΑΚΑΣ ΣΕ ΜΕΤΑΒΛΗΤΗ
        K+=100
        ΒΑΛΕ K
        ΤΥΠΩΣΕ K \ ΟΛΑ 105
        ΣΩΡΟΣ A {
            ΔΙΑΒΑΣΕ M()
            ΤΥΠΩΣΕ M() \ ΟΛΑ 5
        }
        ΤΥΠΩΣΕ ΜΗΚΟΣ(A) \ 0
    }
}
A=ΣΩΡΟΣ:="ΑΛΦΑ","ΒΗΤΑ", 100
ΤΥΠΩΣΕ ΜΗΚΟΣ(A), ΜΗΚΟΣ(N) \ 3 & 1
N=ΣΩΡΟΣ(A,N) \ ΝΕΟΣ ΣΩΡΟΣ ΩΣ A+B
ΣΩΡΟΣ N {
    ΣΩΡΟΣ \ ΜΑΣ ΔΕΙΧΝΕΙ ΤΙ ΣΤΟΙΧΕΙΑ ΕΧΕΙ
}
ΣΩΡΟΣ A {
    ΣΩΡΟΣ
}
A=ΣΩΡΟΣ(A,-3) \ ΑΝΑΣΤΡΟΦΗ, ΤΟ ΤΕΛΕΥΤΑΙΟ ΓΙΝΕΤΑΙ ΠΡΩΤΟ
ΣΩΡΟΣ A {
    ΣΩΡΟΣ \ ΜΑΣ ΔΕΙΧΝΕΙ ΤΙ ΣΤΟΙΧΕΙΑ ΕΧΕΙ
}
ΣΩΡΟΣ ΣΩΡΟΣ(A,-2) { \ ΒΓΑΖΕΙ ΑΝΤΙΓΡΑΦΟ ΤΟΥ A
\ ΤΑ ΔΥΟ ΤΕΛΕΥΤΑΙΑ ΣΤΟΙΧΕΙΑ ΕΙΝΑΙ ΣΕ ΑΝΑΣΤΡΟΦΗ
\ Ο Α ΠΑΡΑΜΕΝΕΙ ΩΣ ΕΧΕ

```

```

        ΣΩΡΟΣ
    }
    ΣΩΡΟΣ A {ΠΕΤΑ} \ ΠΕΤΑΕΙ ΤΗ ΚΟΡΥΦΗ
    A=ΣΩΡΟΣ(A,-ΜΗΚΟΣ(A))
    NN=ΚΑΘΕ(A)
    ΕΝΩ NN {
        ΤΥΠΩΣΕ ΤΙΜΗΣΩΡΟΥ$(NN)
    }

```

Αναγνωριστικό: ΦΕΡΕ [SHIFT]

- 1) ΦΕΡΕ 5
- 2) ΦΕΡΕ
- 3) ΦΕΡΕ N, M

Φέρνουμε το 5ο στοιχείο του σωρού στη κορυφή (αν δεν υπάρχει 5ο στοιχείο τότε βγαίνει λάθος)

Η δεύτερη περίπτωση είναι ισοδύναμη με το ΦΕΡΕ ΜΕΓΕΘΟΣ.ΣΩΡΟΥ

Στην τρίτη περίπτωση ζητάμε από το N να φέρει M στοιχεία.

Υπάρχει και το αντίστροφο ΦΕΡΕΠΙΣΩ

ΑΔΕΙΑΣΕ ' καθαρίζει το σωρό

ΣΕΙΡΑ 1,2,3,4,5 ' βάζει τη σειρά στο τέλος του σωρού με το τελευταίο στοιχείο...τελευταίο

ΦΕΡΕ 3

ΣΩΡΟΣ

3 1 2 4 5

' μπορούμε να αλλάξουμε το μεταφερόμενο στοιχείο και να το βάλουμε στη θέση του

ΒΑΛΕ ΑΡΙΘΜΟΣ+100 ' εδω διαβάζουμε τη κορυφή - άρα πετάμε το στοιχείο - και

ξαναβάζουμε το αποτέλεσμα

ΦΕΡΕΠΙΣΩ 3

ΣΩΡΟΣ

1 2 103 4 5

Αναγνωριστικό: ΦΕΡΕΠΙΣΩ [SHIFTBACK]

- 1) ΦΕΡΕΠΙΣΩ 5
- 2) ΦΕΡΕΠΙΣΩ
- 3) ΦΕΡΕΠΙΣΩ N, M

φέρνει τη κορυφή του σωρού στη 5η θέση

η δεύτερη περίπτωση στέλνει τη κορυφή στο πάτο.

Τρίτη περίπτωση: Στέλνουμε M στοιχεία στη N θέση.

Δες ΦΕΡΕ, ΠΑΝΩ, ΠΕΤΑ, ΣΩΡΟΥΤΙΜΗ(), ΣΩΡΟΥΤΙΜΗ\$(),
ΑΡΙΘΜΟΣ, ΓΡΑΜΜΑ\$, ΕΙΝΑΡ, ΕΙΝΓΡ, ΑΔΕΙΟ, ΑΔΕΙΑΣΕ, ΣΩΡΟΣ

Ομάδα: ΟΡΙΣΜΟΙ - DEFINITIONS

Αναγνωριστικό: ΑΛΛΑΞΕ [SWAP]

ΑΛΛΑΞΕ Α\$, Β\$

ΑΛΛΑΞΕ Α%,Β%

ΑΛΛΑΞΕ Α(3), Α(5)

ΑΛΛΑΞΕ Α, Α(5)

ΑΛΛΑΞΕ Α, Β

η ΑΛΛΑΞΕ είναι δυο φορές και παραπάνω πιο γρήγορη από την απλή μετακίνηση με τρίτη μεταβλητή.

Εσωτερικά όλες οι μεταβλητές και οι πίνακες είναι τύπου Variant, με 16 byte. Η swap αλλάζει αυτά τα 16 Byte.

Αναγνωριστικό: ΑΠΑΡ [ENUM]

\\ μπορούμε να βάζουμε ότι αριθμό θέλουμε ακόμα και πραγματικούς

\\ Εξ ορισμού Σκύλος=1 και Γάτα=2

\\ Μπορούμε να δώσουμε τ ο Σκύλος=0, Γάτα και το Γάτα θα γίνει 1

\\ ή Σκύλος=100, Γάτα=200

\\ αντί για κόμμα μπορούμε να αλλάζουμε γραμμή

\\ ή μπορούμε να βάζουμε αλλαγή γραμμής μετά το κόμμα

\\ δεν μπορούμε να αφήσουμε ένα κόμμα στο τέλος

Απαρ Κατοικίδια {Σκύλος, Γάτα}

α=Σκύλος

Τύπωσε α=1 ' αληθές

α++

Τύπωσε Εκφρ\$(α)="Γάτα", α=2

κ=Κάθε(Κατοικίδια)

Ενώ κ {

 Τύπωσε Εκφρ\$(κ), Εκφρ(κ)

 α=Εκφρ(κ)

 Τύπωσε α<Γάτα

 Αλφα(α)

}

α=Σκύλος

ΑλφαΜεΑναφορά(&α)

Τύπωσε α=Γάτα

Ρουτίνα Αλφα(β ως Κατοικίδια)

Τύπωσε β

Τέλος Ρουτίνας

Ρουτίνα ΑλφαΜεΑναφορά(&β ως Κατοικίδια)

Τύπωσε β

β++

Τέλος Ρουτίνας

Δες Απαρίθμηση

—

Αναγνωριστικό: ΑΠΑΡΙΘΜΗΣΗ [ENUMERATION]

\| μπορούμε να βάζουμε ότι αριθμό θέλουμε ακόμα και πραγματικούς

\| Εξ ορισμού Σκύλος=1 και Γάτα=2

\| Μπορούμε να δώσουμε τ ο Σκύλος=0, Γάτα και το Γάτα θα γίνει 1

\| ή Σκύλος=100, Γάτα=200

\| αντί για κόμμα μπορούμε να αλλάζουμε γραμμή

\| ή μπορούμε να βάζουμε αλλαγή γραμμής μετά το κόμμα

\| δεν μπορούμε να αφήσουμε ένα κόμμα στο τέλος

Απαρίθμηση Κατοικίδια {

Σκύλος

Γάτα

}

α=Σκύλος

Τύπωσε α=1 ' αληθές

α++

Τύπωσε Εκφρ\$(α)="Γάτα", α=2

κ=Κάθε(Κατοικίδια)

Ενώ κ {

Τύπωσε Εκφρ\$(κ), Εκφρ(κ)

α=Εκφρ(κ)

Τύπωσε α<Γάτα

Αλφα(α)

}

α=Σκύλος

ΑλφαΜεΑναφορά(&α)

Τύπωσε α=Γάτα

Ρουτίνα Αλφα(β ως Κατοικίδια)

Τύπωσε β

Τέλος Ρουτίνας
Ρουτίνα ΑλφαΜεΑναφορά(&β ως Κατοικίδια)
Τύπωσε β
β++
Τέλος Ρουτίνας

Δες Απαρ

Αναγνωριστικό: ΑΡΙΘΜΟΣ [DECIMAL]
ΑΡΙΘΜΟΣ A=1201290190192090291@
' Τύπος DECIMAL 27 ψηφίων (μπορούμε να βάλουμε και υποδιαστολή)

Αναγνωριστικό: ΑΥΤΟΜΑΤΟΙ_ΠΙΝΑΚΕΣ [AUTO_ARRAYS]
Αυτόματοι Πίνακες είναι οι πίνακες που δεν χρειάζεται να οριστούν με την Πίνακας και δίνουν έναν δείκτη σε αυτούς.

α=(,) μήκος(α)=0
α=(1,) μήκος(α)=1

\\ χρήση επάλληλων συναρτήσεων
\\ κάθε #συνάρτηση() παίρνει το αποτέλεσμα της προηγούμενης
A=(1,2,3,4,5,6,7,8)
\\ #αθρ() δίνει το άθροισμα
Τύπωσε A#αθρ(α)=36
\\ συνδυάζονται ορισμένες έτοιμες συναρτήσεις με λάμδα συναρτήσεις
Ζυγοί=Λάμδα ->Αριθμός υπολ 2 = 0
\\ γράφεται και με μεταβλητή
\\ σε πολλές γραμμές
Ζυγοί=Λάμδα ->{
 Διάβασε χ ' το Διάβασε χ μπορεί να γίνει (χ) όπως παρακάτω
 =χ υπολ 2 = 0
}
\\ ή σε μία γραμμή
Ζυγοί=Λάμδα (χ)->χ υπολ 2 = 0
\\ Στην αντιστοίχιση η λάμδα καλείται ως τμήμα
\\ οπότε η εξαγωγή γίνεται στον ίδιο σωρό που γίνεται η εισαγωγή.
Τετράγωνο=Λάμδα -> {
 Βάλε Αριθμός**2

```

}
\\ #Φίλτρο(λάμδα συνάρτηση [, εναλλακτική πηγή])
\\ αν το φίλτρο δεν δώσει πίνακα τότε δεν βγαίνει λάθος και θα γίνει η σύγκριση
Σημ : Ζυγοί=Λάμδα (χ)->χ>5000 ' θα τυπώσει μονο το "οκ"
Τύπωσε Α#Φίλτρο(Ζυγοί)#Αντ(Τετράγωνο)#Αθρ(=120, "οκ"
\\ μπορούμε να δώσουμε άμεσα τη λάμδα (αν δε έχει πολλές γραμμές)
\\ αλλά αν ακολουθεί παράμετρος τότε πρέπει να την βάλουμε μέσα σε παρενθέσεις.
Τύπωσε Α#Φίλτρο(λάμδα->αριθμός υπολ 2=0)#Αντ(Τετράγωνο)#Αθρ(=120, "οκ"
Τύπωσε Α#Αντ((λάμδα ->{Βάλε Αριθμός**2}))#Αθρ(=204
Τύπωσε Α#Πακ((λάμδα ->{Βάλε Αριθμός+Αριθμός}), 0)=Α#αθρ(
\\ το 1033 λέει να έχουν οι αριθμοί τη τελεία (όπως στα αγγλικά)
\\ δοκίμασε με 1032. Υπάρχει η εντολή Τοπικό που βάζει το νούμερο που θέλουμε για κάθε
εμφάνιση
Τύπωσε Α#Πακ$((λάμδα (α,β$) ->{Βάλε β$+" "+Γραφή$(α, 1033)}), "Δεδομένα:")="Δεδομένα: 1
2 3 4 5 6 7 8"

```

Αναγνωριστικό: ΓΕΓΟΝΟΣ [EVENT]

Ένα γεγονός είναι αντικείμενο που μπορεί να μπει στο σωρό και το βλέπουμε με Ταύτιση("Ε") (Ε ελληνική ή Ε αγγλικό).

Γενικό Γεγονός Άλφα { }

Γεγονός Άλφα {

Διάβασε μήνυμα\$

Διάβασε &δώσε_πίσω

Συνάρτηση {

\\ μπορώ και να σπάσω τις Διάβασε σε σειρά

\\ και εδώ πρέπει να τις αφήσω εκτός

\\ από αυτή τη συνάρτηση

\\ γιατί θα βάλει μία ο διερμηνευτής

Τύπωσε μήνυμα\$

}

}

Γεγονός Άλφα Κράτησε

Γεγονός Άλφα Καθαρό \\ (κάνει και κράτηση συνάμα)

Γεγονός Άλφα Άφησε

Γεγονός Άλφα Πέτα αααα() [, ββββ()]

Γεγονός Άλφα Νέο αααα() [, ββββ()]

Παράδειγμα:

```

Βηματιστής = λάμδα -> {
  \ \ τιμή εκκίνησης, τιμή τερματισμού (δίνει ψευδές)
  Διάβασε χ,ζ
  =Λάμδα χ, ζ->{
    Διάβασε γ
    Αν χ>=ζ Τότε Έξοδος
    Κάλεσε Γεγονός γ χ
    χ++
    =Αληθής
  }
}

```

```

α_από_10_μέχρι_14=Βηματιστής(10,15)
β=λάμδα->{Τύπωσε Αριθμός} \ \ ίδιο με { Διάβασε X : Τύπωσε X}
Γεγονός γεγ {Διάβασε χ}
\ \ Πολλαπλής εκπομπής (multicast)
Γεγονός γεγ Νέο β(), β(), β()
Γεγονός γεγ Πέτα β()
\ \ άρα έχουμε δυο τώρα!
\ \ κλήση γεγονότος
Κάλεσε Γεγονός γεγ 5
\ \ Με το βηματιστή, όπου περνάμε το γεγονός ως παράμετρο:
\ \ η παράμετρος εδώ είναι με τιμή. Δεν υπάρχει πρόβλημα!
\ \ Με αναφορά θα μπορούσαμε να χειριστούμε το γεγονός και αυτό έχει
\ \ αλλαγές μετά την επιστροφή
Ενώ α_από_10_μέχρι_14(γεγ) {}
\ \ ο διακόπτης Καθαρό κάνει και Κράτησε
Γεγονός γεγ Καθαρό
Συνάρτηση κ { Τυπωσε "οκ", αριθμός}
Γεγονός γεγ Νέο κ()
\ \ πρέπει να αφήσουμε το γεγονός να λειτουργεί! (το είχαμε κρατήσει πριν)
Γεγονός γεγ Άφησε
Κάλεσε Γεγονός γεγ 1
Γεγονός γεγ Κράτησε
\ \ η παρακάτω δεν θα εκτελεστεί ως γεγονός, αλλά η β θα τυπώσει το 2
\ \ γιατί ο έλεγχος γίνεται μετά την συγκέντρωση των παραμέτρων που δίνουμε στη γεγονός.
Κάλεσε Γεγονός γεγ 5+β(2)

```

ΓΕΝΙΚΗ A=10, A(20,30)

Μπορούμε να χρησιμοποιήσουμε και τα ΓΕΝΙΚΟ και ΓΕΝΙΚΕΣ

Μπορούμε να φτιάχνουμε και πίνακες. Αν υπάρχει μια ίδια γενική μεταβλητή ή πίνακας τότε ξαναφτιάχνεται ένας άλλος με το ίδιο όνομα, οπότε δεν πρέπει να μπαίνει σε ένα μπλοκ επανάληψης. Αν θέλουμε να καλέσουμε ένα τμήμα και θέλουμε να βλέπει μια στατική μεταβλητή τότε μπορούμε πριν το καλέσουμε να έχουμε μια γενική ως στατική. Μια γενική θα χαθεί μόλις τερματίσει ο δημιουργός της, ένα τμήμα ή μια συνάρτηση ή μια ρουτίνα (δες ΡΟΥΤΙΝΑ) ή μια ΓΙΑ ΑΥΤΟ {}. Αν υπάρχει ήδη μια άλλη γενική με το ίδιο όνομα πριν την εκκίνηση του δημιουργού της γενικής τότε αυτή επανέρχεται στο προσκήνιο. Ουσιαστικά μια γενική εντός ενός τμήματος είναι μια τοπική με εμβέλεια όλα τα παρακάτω τμήματα εκτός αυτών που έχει οριστεί τοπική με το ίδιο όνομα ή άλλη γενική.

Μια γενική αλλάζει τιμές με το <= ή χωρίς αυτό αν χρησιμοποιήσουμε τη ΣΤΗ

Γενική A=10

Στη A=30 \ \ δουλεύει και για γενικές.

A+=20 \ \ δουλεύει και για γενικές

A<=1000 \ \ διαφορετικά φτιάχνεται τοπική A

Οι πίνακες που ορίζουμε με την Γενική μπορούν να αλλάξουν αριθμό στοιχείων με την θέσε Πίνακας

Αν θέλουμε να δώσουμε αρχική τιμή σε πίνακα τότε χρησιμοποιούμε την πίνακας και για γενικές την θέσε πίνακας

όμως η Πίνακας δεν φτιάχνει νέους πίνακες με το ίδιο όνομα, αλλά τους ορίζει ξανά προσαρμόζοντας τα στοιχεία τους. Μπορούμε να αντιγράψουμε ένα πίνακα σε έναν πίνακα Π.χ. έχουμε έναν γενικό πίνακα A()

και θέλουμε να πάρει τιμές από έναν τοπικό γενικό α()

για αυτό {πίνακας β(30)=1 : α()=β()} \ \ ο β() θα σβήσει με το πέρας της για Τυπωσε α(12)

λιστα

Αναγνωριστικό: ΔΙΑΘΡΩΣΗ [BUFFER]

1) φτιάχνουμε περιοχές στη μνήμη

Διάρθρωση κενή αλφα ως ακέραιος * 100

A\$=Πεδίο\$("Γιώργος", 100)

Τύπωσε Μήκος(A\$), αλφα(0) \ \ 200 (σε ψηφία ή byte), η πραγματική διεύθυνση μνήμης της βάσης (0)

\ \ γράφουμε πάντα με την εντολή Επιστροφή

Επιστροφή αλφα, 0:=A\$

\ \ ανά ψηφίο

Επιστροφή αλφα, 2!:=255

\ \ βλέπουμε ότι διαβάζουμε ακέραιο 2 byte, δηλαδή χαρακτήρα)

Για i=0 έως 6

Τύπωσε εκφρ(αλφα, i), χαρκωδ(Μεσ\$(A\$, i+1,1))

Επόμενο i

Τύπωσε Μήκος(αλφα) \ \ 200 (2*100)

μ\$=εκφρ\$(αλφα,0,7*2)

Τύπωσε μήκος(μ\$), μ\$

Επιστροφή αλφα, 30:=65000 \ \ 0 .. 65535

\ \ διαβάζουμε από την θέση 30 (2*30 αν ήταν ψηφίο)

\ \ μπορούμε όμως να διαβάσουμε και ανά ψηφίο (byte)

\ \ 253*256+232=65000

\ \ στη θέση 60 είναι το 232 (Low Byte) και στη θέση 61 είναι το 253 (High Byte)

Τύπωσε εκφρ(αλφα,30), εκφρ(αλφα, 60 ως ψηφίο), εκφρ(αλφα, 61 ως ψηφίο)

2) Χρήση δομής ή χρήση άμεσου: ψηφίο (byte), ακέραιος (2 byte), μακρύς (4 byte)

Δομή Αλφα {

α ως ψηφίο*8

β ως ακέραιος*20 'κάνει για αλφαριθμητικό

γ ως μακρύς*4

}

Τύπωσε Αλφα("α") \ \ θέση 0

Τύπωσε Αλφα("β") \ \ θέση 8 (+8 από την αρχή)

Τύπωσε Αλφα("γ") \ \ θέση 48 (+48 από την αρχή)

Τύπωσε Μήκος(Αλφα) \ \ 64 bytes

Διάρθρωση Κάπα ως Αλφα*20

Τύπωσε Μήκος(Κάπα) \ \ 1280 ή 64*20 bytes

Επιστροφή Κάπα, 5 ! β :="Γιώργος"

Τύπωσε Εκφρ\$(Κάπα, 5 ! β, 7*2) \ \ 2 byte το γράμμα

Επιστροφή Κάπα, 3 ! β :="Γιώργος"+χαρ\$(0)) \ \ γράφει 8 bytes

Τύπωσε χαρ\$(Εκφρ\$(Κάπα, 3 ! β, 8)) \ \ 1 byte (ψηφίο) το γράμμα

3) Χρήση Δομής σε Δομή.

Δομή γράμματα_ονόματος {

γρα ως ακέραιος*20

}

Δομή η_δομή_μου {

άλφα ως ψηφίο

για_στοίχιση ως ψηφίο*3

Κάπα ως μακρύς*20

όνομα ως γράμματα_ονόματος*10

Κάπα2 ως διπλός

```

}
Διάρθρωση κενή άλφα ως η_δομή_μου*100
Τύπωσε Μήκος(άλφα), Μήκος.Εμφ(άλφα)
Τύπωσε 5*Μήκος(η_δομή_μου)+ η_δομή_μου("Κάπα"), η_δομή_μου("Κάπα")
Επιστροφή άλφα, 50!Κάπα!10:=50
sLong=4
\\Επιστροφή άλφα, 5*Μήκος(η_δομή_μου)+ η_δομή_μου("Κάπα")+10*sLong! :=50 ως μακρύς
\\Τύπωσε εκφρ(άλφα, 5*Μήκος(η_δομή_μου)+ η_δομή_μου("Κάπα")+40 ως μακρύς)
Τύπωσε εκφρ(άλφα, 50!Κάπα!10)
Επιστροφή άλφα, 50!όνομα!5:="George", 50!όνομα!6:="Hello"
Τύπωσε εκφρ$(άλφα, 50!όνομα!5, 20)
Τύπωσε εκφρ$(άλφα, 50!όνομα!6, 20)
Τύπωσε Μήκος(άλφα), τύπος$(η_δομή_μου)

```

4) Αριθμοί με πρόσημο και χωρίς

```

\\ σε θέσεις για ψηφίο (1 byte), ακέραιο (2 byte) και μακρύ (4 byte) μπαίνουν αριθμοί χωρίς
πρόσημο
\\ σε θέση διπλού μπαίνει κανονικά ο διπλός (8 byte)
Διάρθρωση αλφα ως μακρύς * 50
Επιστροφή αλφα, 5:=240,6:=δυναδικό.ακέραιο(-5),30:=3405435354
Τύπωσε εκφρ(αλφα,5), ακέραιο.δυναδικό(αλφα,6), ακέραιο.δυναδικό(αλφα,30)

```

\\ μετατροπή ενός

```

A=δυναδικό.ακέραιο(-500)
\\ ανάποδη μετατροπή
Τύπωσε ακέραιο.δυναδικό(A)
\\ μπορούμε να κρατήσουμε ακέραιο χωρίς πρόσημο σε μακρύ.
μακρύς B=ακέραιο.δυναδικό(3405435354)
Τύπωσε δυναδικό.ακέραιο(B)
\\ μπορούμε να χρησιμοποιήσουμε τις κανονικές μεταβλητές
\\ ως 8 ψηφία (byte) στη μνήμη
AA=3405435354 \\
A%=3405435354 \\ εσωτερικά είναι διπλός (κινητής υποδιαστολής με 8 ψηφία)
Τύπωσε AA, A%
Επιστροφή αλφα, 10 :=AA+1 ως διπλός \\ 8 bytes - offset 10*4=+40
Τύπωσε εκφρ(αλφα, 40 ως διπλός) \\ offset 40 byte
Διάρθρωση alfa1 ως ακέραιος * 100
Επιστροφή alfa1, 10 :=AA ως διπλός \\ 8 bytes - offset 10*2=+20
Τύπωσε εκφρ(alfa1, 20 ως διπλός) \\ offset 20 byte

```

5) Εκτέλεση Κώδικα γλώσσας μηχανής

Δείτε παράδειγμα στην Εκτέλεση για παράδειγμα της Διάρθρωση Κώδικα

6) Χρήση με την βιβλιοθήκη Μαθηματικά

Δομή Διάνυσμα {

 x Ως Διπλός

 y Ως Διπλός

 z Ως Διπλός

}

Δομή Διανύσματα {

 Vec1 Ως Διάνυσμα

 Vec2 Ως Διάνυσμα

}

Τύπωσε Μήκος(Διάνυσμα), Μήκος(Διανύσματα)

K=Κάθε(Διάνυσμα)

Ενώ K {

 Τύπωσε Εκφρ\$(K, K^)+ " θέση στη "+Εκφρ\$(K)

}

Διάρθρωση Κενή Άλφα ως Διανύσματα*10

Τύπωσε Άλφα(0), Άλφα(0,"Vec1"), Άλφα(0,"Vec2")

Τύπωσε Άλφα(0, "Vec1", Διάνυσμα("X")), Άλφα(0,"Vec1", Διάνυσμα("Y")), Άλφα(0,"Vec1", Διάνυσμα("Z"))

Τύπωσε Άλφα(0, "Vec1", Διάνυσμα("X")!), Άλφα(0,"Vec1", Διάνυσμα("Y")!), Άλφα(0,"Vec1", Διάνυσμα("Z")!)

Τύπωσε Άλφα(0, "Vec1")

Τύπωσε Άλφα(0, "Vec1!") ' αφαιρούμε τη τιμή Άλφα(0), δηλαδή δίνει μόνο η θέση σε ψηφία από την αρχή.

Τύπωσε Άλφα(0!) ' προσθέτουμε ψηφία (bytes) στο Άλφα(0), πχ Άλφα(10!) δίνει Άλφα(0)+10 ως θέση μνήμης

Διαν=Λάμδα Άλφα, Διάνυσμα (ν, α\$, β\$) -> {

 =Εκφρ(Άλφα, Άλφα(0, α\$, Διάνυσμα(β\$)!) Ως Διπλός)

}

Όρισε Μαθ Μαθηματικά

Τύπωσε Άλφα(1)-Άλφα(0)

x_θέση=Διάνυσμα("x")

For v=0 to 9 {

 Μέθοδος Μαθ, "Vector", Άλφα(v,"Vec1"), 1200,2500,1500

 Μέθοδος Μαθ, "Vector", Άλφα(v,"Vec2"), 3000,2000,1000

 Τύπωσε Διαν(v,"Vec1", "x")

 Τύπωσε Διαν(v,"Vec1", "y")

 Τύπωσε Διαν(v,"Vec1", "z")

 Τύπωσε Έκφρ(Άλφα, Άλφα(v, "Vec1", x_θέση!) Ως Διπλός)

 Τύπωσε Διαν(v,"Vec2", "y")

 Τύπωσε Διαν(v,"Vec2", "z")

```

Μέθοδος Μαθ, "VecString", Άλφα(v, "Vec1") Ως Αποτέλεσμα$
Τύπωσε Αποτέλεσμα$
Μέθοδος Μαθ, "VecString", Άλφα(v, "Vec2") Ως Αποτέλεσμα$
Τύπωσε Αποτέλεσμα$
}
Όρισε Μαθ Τίποτα

```

```

7)
Δομή alfa {
    a ως μακρύς
    c ως γράμμα*5 ' πέντε δείκτες σε BSTR (4X5=20byte)
    z ως ακέραιος*20
    zero ως μακρύς
}

```

```

Τύπωσε Μήκος(alfa) , Τύπος$(alfa("c"))
Διάρθρωση Κενή Mem ως alfa*10
Επιστροφή Mem, 2!c!4:="hello there"
Δεκαεξ Εκφρ(Mem, 2!c!4) ' string address in Hex
Επιστροφή Mem, 2!z:=Πεδίο$("Fix String",20) ' no zero
Τύπωσε Εκφρ$(Mem, 2!z, 10*2) ' need to pass byte number for fix string

```

```

Αναγνωριστικό: ΔΥΑΔΙΚΟ [BINARY]
ΔΥΑΔΙΚΟ {
    kwOVA5kDkQMgAKcDkQOhA5ED
} ΩΣ A$
ΤΥΠΩΣΕ A$

```

Στις αγκύλες πρέπει να είναι δυαδικά στοιχεία κωδικοποιημένα σε BASE64. Μπορούμε να εισάγουμε στοιχεία από το μενού στο διορθωτή, ως εισαγωγή πόρου., από αρχείο Η μεταβλητή μπορεί να είναι αριθμητική ή αλφαριθμητική ή στοχείο πίνακα αριθμητικού ή αλφαριθμητικού. Στα αριθμητικά έχουμε αυτόματα δείκτη σε διάρθρωση με τα στοιχεία, ενώ στα αλφαριθμητικά έχουμε αλφαριθμητικό με τα στοιχεία αποκωδικοποιημένα.

Μετατροπή σε BASE64 γίνεται επίσης με την ΕΠΑΝ\$(), η οποία εκτός από επανάληψη ενός αλφαριθμητικού κάνει "επαναμορφωποίηση".

```

Αναγνωριστικό: ΕΓΓΡΑΦΟ [DOCUMENT]
&H εντολή ΕΓΓΡΑΦΟ φτιάχνει νέες μεταβλητές αλφαριθμητικές που διαφέρουν από τις

```

κανονικές στο τρόπο που καταχωρούν τα δεδομένα. Ορατή διαφορά είναι ότι το = δεν κάνει αντικατάσταση αλλά προσθήκη στο τέλος.

ΕΓΓΡΑΦΟ A\$, B\$

Μπορούμε και στοιχείο πίνακα να φτιάξουμε ωε ΕΓΓΡΑΦΟ, αλλά όχι όλο το πίνακα με μια εντολή.

A\$={ΓΕΙΑ ΧΑΡΑ

ΟΛΑ ΚΑΛΑ

}

ΕΓΓΡΑΦΟ A\$, B\$

A\$="ΤΕΣΤ1"

B\$="ΤΕΣΤ2"

ΑΝΑΦΟΡΑ A\$

ΑΝΑΦΟΡΑ B\$

ΤΥΠΩΣΕ ΕΓΓΡΑΦΟΥ.ΜΗΚΟΣ(A\$), ΕΓΓΡΑΦΟΥ.ΜΗΚΟΣ(B\$)

' Το = δεν κάνει αντικατάσταση αλλά προσθέτει στη τελευταία γραμμή ότι βάλουμε...

'Με ΚΑΘΑΡΟ A\$ σβήνουμε τις παραγράφους

Χωρίς την χρήση της ΕΓΓΡΑΦΟ

A\$={ΓΕΙΑ ΧΑΡΑ

ΟΛΑ ΚΑΛΑ

}

A\$=A\$+"ΤΕΣΤ1" ' πρέπει να δημιουργούμε αντίγραφα

B\$="ΤΕΣΤ2"

ΑΝΑΦΟΡΑ A\$

ΑΝΑΦΟΡΑ B\$

ΤΥΠΩΣΕ ΕΓΓΡΑΦΟΥ.ΜΗΚΟΣ(A\$), ΕΓΓΡΑΦΟΥ.ΜΗΚΟΣ(B\$) ' η συνάρτηση δουλεύει και για απλά αλφαριθμητικά.

Κερδίζουμε όταν έχουμε μεγάλα κείμενα και η προσθήκη ενός μικρού μέρους στο ενδιάμεσο ή στο τέλος απαιτεί χρονοβόρα διαδικασία ή και διαθέσιμο χώρο για αντίγραφα.

Μπορούμε να χρησιμοποιήσουμε το <= για να μπορούμε να διαμορφώνουμε και σφαιρικά έγγραφα (καθολικά, που τα βλέπουν όλα τα τμήματα). Για έγγραφα σε πίνακα (πρέπει να καθορίζουμε ξεχωριστά κάθε στοιχείο του πίνακα αλλιώς θα μένουν ότι δεν έχουμε αλλάξει σε απλά αλφαριθμητικά)

Αναγνωριστικό: ΘΕΣΕ [SET]

Θεσε A\$="AAAAAAAAA"

Σε ένα τμήμα έστω ΤΜΗΜΑ ΑΛΦΑ αν δώσουμε το παρακάτω:

A\$="BBBBBBBBBB"

Τότε θα έχει φτιαχτεί μια μεταβλητή ΑΛΦΑ.Α\$ που μέσα στο τμήμα θα διαβάζεται σαν Α\$.

Μπορούμε όμως να διαβάζουμε και μεταβλητές που έχουμε φτιάξει με την ΘΕΣΕ.

Μάλιστα η εντολή ΘΕΣΕ δεν ορίζει μόνο σφαιρικές μεταβλητές (καθολικές) αλλά κάνει και ενέργειες σαν να τις πληκτρολογούσαμε στον μεταφραστή γραμμής (από εκεί που καλούμε το πρώτο τμήμα)

Δες το παρακάτω παράδειγμα (το γράφουμε σε ένα τμήμα)

ΘΕΣΕ Α\$="111111"

A\$="22222222"

ΤΥΠΩΣΕ Α\$

ΘΕΣΕ ΤΥΠΩΣΕ Α\$

Θα μας δώσει τα

22222222

1111111

Δοκίμασε χωρίς το Α\$="22222222"

1111111

1111111

Οι σφαιρικές μεταβλητές διαβάζονται από όλα τα τμήματα εφόσον το τμήμα όπου δημιουργήθηκαν δεν έχει τερματίσει (με τον τερματισμό ενός τμήματος πίνακες και μεταβλητές σβήνουν, δεν σβήνει όμως ο σωρός τιμών ΔΕΣ ΣΩΡΟΣ)

Αν θέλουμε να αλλάξουμε τιμή σε μια σφαιρική μεταβλητή σε ένα τμήμα...δεν είναι τόσο εύκολο. Με την ΘΕΣΕ μπορούμε να δώσουμε αρχική τιμή αλλά αν αυτή η τιμή βγαίνει από παράσταση τότε πρέπει να κάνουμε αυτό:

ΒΑΛΕ 12*X ' βάζουμε στο σωρό την τιμή της παράστασης ενώ είμαστε στο τμήμα ΑΛΦΑ

ΘΕΣΕ ΣΦΑΙΡΙΚΗ=ΑΡΙΘΜΟΣ ' Εδώ το τμήμα θα αλλάξει όνομα (θα είναι κενό) επειδή εδώ είναι σαν να είμαστε στο πρώτο επίπεδο, στον μεταφραστή γραμμής.

Αν δεν βάλω την ΘΕΣΕ τότε θα φτιαχτεί τοπική μεταβλητή και από εκεί και πέρα θα διαβάζει την τοπική ο μεταφραστής,εκτός και αν πάλι την διαβάσουμε με την ΘΕΣΕ

' Τα τμήματα μοιράζονται τον ίδιο σωρό

' Εκτός από αυτά που ξεκίνησαν σε μια συνάρτηση δες παρακάτω

ΘΕΣΕ ΒΑΛΕ ΣΦΑΙΡΙΚΗ ' η θεσε βάζει πάντα στην κορυφή του σωρού

ΔΙΑΒΑΣΕ ΠΑΛΙΑσφαιρική

Δες ένα παράδειγμα που δείχνει πως φαίνεται μια ΣΦΑΙΡΙΚΗ μεταβλητή

Αυτό μπορείς να το αντιγράψεις σε ένα τμήμα έστω α

Θεσε ΣΦΑΙΡΙΚΗ=5
Τυπωσε ΣΦΑΙΡΙΚΗ ' 5
X=20
Βαλε 20*X
Θεσε ΣΦΑΙΡΙΚΗ=Αριθμος
ΤΥΠΩΣΕ ΣΦΑΙΡΙΚΗ '400

Συναρτηση α {
Τμημα ενα {
Σειρα 3,2,1 'η Σειρα βάζει στο σωρό από το τέλος με πάτο το 1
}
ενα
διαβασε α,β,γ ' τον οποίο διαβάζει εδώ
=α+β+γ ' εδώ επιστρέφει τιμή η συνάρτηση
}

τυπωσε α() '6
' Μπορούμε να βάλουμε στη συνάρτηση τιμές
' Θα μπουν όλες στο σωρό μπροστά από αυτές στο τμήμα ενα
τυπωσε α(5,5,5), α(5,5), α(5) ' τυπώνει 15 13 10

ΣΦΑΙΡΙΚΗ=100
Τυπωσε ΣΦΑΙΡΙΚΗ '100 σκίαση της ΣΦΑΙΡΙΚΗΣ με τοπική με το ίδιο όνομα

Συναρτηση β {
=ΣΦΑΙΡΙΚΗ 'Εδώ η β δεν βλέπει την τοπική παραπάνω αλλά την σφαιρική
}
τυπωσε β() '400

Αναγνωριστικό: ΙΔΙΟΤΗΤΑ [PROPERTY]

#1) Οι ιδιότητες είναι εσωτερικά ομάδες με κάποια Αξία, αν ορίσουμε ότι θα έχει μόνο Αξία τότε δεν θα αλλάζει τιμή απ' έξω

Ομάδα Άλφα {
X=10
Ιδιότητα Όνομα\$ {Αξία} ="Γιάννης"
Ιδιότητα Κάτι=100
Τμήμα ΘέσεΝέοΌνομα {
Διάβασε .[Όνομα]\$
}
}

Τύπωσε Άλφα.Όνομα\$

Δες { Άλφα.Όνομα\$="Ανδρέας"}

\ \ δεν αλλάζει - δίνει λάθος - η Δες { } αφαιρεί το λάθος

Τύπωσε Άλφα.Όνομα\$

Άλφα.ΘέσεΝέοΌνομα "Ανδρέας"

Τύπωσε Άλφα.Όνομα\$

2) Η ιδιότητα Άλφα έχει δυο μέλη την Αξία και την Θέσε, η δεύτερη χρησιμοποιείται στο = (αντικατάσταση ή νέα τιμή). Για να δούμε από μια ιδιότητα κάτι στη γονική ομάδα πρέπει να το ενώσουμε με ένα όνομα που διαβάζεται στην ομάδα της ιδιότητας, και αυτό το κάνει η Ένωσε. Αυτό συμβαίνει γιατί οι ομάδες βλέπουν το δικό τους περιεχόμενο και όχι κάτι έξω από αυτό. Μια εσωτερική ομάδα δεν έχει πρόσβαση σε αυτήν που περιέχεται και αυτό το λύνει η Ένωσε. Έχει σημασία γιατί αν αντιγράψουμε την ομάδα (που χρησιμοποιούμε ως ιδιότητα) τότε η ένωση δεν θα πετύχει. Η τιμή της ιδιότητας δεν ανήκει στην ομάδα της ιδιότητας αλλά στη γονική ομάδα!

Ομάδα Κάτι {

Γεγονός Γ {

Διάβασε Τι\$

Συνάρτηση {

Τύπωσε Τι\$

}

}

Ιδιότητα Άλφα {

Αξία {

Ένωσε Γονικό Γ στη Γ

Αν Άλφα>300 Τότε Κάλεσε Γεγονός Γ, "Υψηλή Τιμή"

\ \ η μεταβλητή Αξία δίνει την επιστροφή, οπότε αν την αλλάξουμε εδώ δεν θα

αλλάξουμε την μεταβλητή πίσω από την ιδιότητα

}

Θέσε {

Αν Αξία>400 Τότε Αξία=400

}

} = 10

}

Τύπωσε Κάτι.Άλφα

Κάτι.Άλφα=500

M=Κάτι.Άλφα \ \ Συμβαίνει το γεγονός Γ

Τύπωσε M \ \ η M έχει πάρει τιμή την αξία της Κάτι.Άλφα

3) Οι ιδιότητες μπορούν να έχουν και άλλες ιδιότητες. Στο όνομα της ιδιότητας υπάρχει μεταβλητή με [], όπως εδώ η Μετρητής είναι η [Μετρητής] και όταν θέλουμε να έχουμε πρόσβαση από μια συνάρτηση (και η αξία εσωτερικά συνάρτηση είναι), τότε βάζουμε την τελεία μπροστά.

Ομάδα Κάτι {

Ιδιότητα Άλφα {

```

    Αξία {
        .[Μετρητής]++
    }
    Θέσε
} = 10
Ομάδα Άλφα {
    Ιδιότητα Μετρητής {Αξία}=0
}

```

```

}
Κάτι.Άλφα=500
Για i=1 έως 100 : N=Κάτι.Άλφα : Επόμενο i
Τύπωσε Κάτι.Άλφα.μετρητής

```

4) Σε περίπτωση που το όνομα της ιδιότητας είναι αναγνωριστικό μόνο ανάγνωσης της M2000 πρέπει να φτιάξουμε την ιδιότητα απ' ευθείας ως ομάδα (η εντολή Ιδιότητα είναι μια μακροεντολή που βάζει τις σωστές εντολές)

```

Ομάδα Κάτι {
    Ιδιότητα ΜιαΤιμή {
        Αξία
        Θέσε {
            .[Διπλός]<=Αξία*2
        }
    }
}

```

```

}
Ομάδα ΜιαΤιμή {
    \ Επειδή υπάρχει το αναγνωριστικό Διπλός στο λεξικό της M2000
    \ Η Μακροεντολή Ιδιότητα χρησιμοποιεί το =Διπλός και φέρνει λάθος το 7
    \ Μπορούμε όμως να φτιάξουμε την ιδιότητα απ' ευθείας ως Ομάδα
Ιδιωτικό:

```

[Διπλός]

Δημόσιο:

```

    Ομάδα Διπλός {
        Αξία {
            Ένωσε γονικό [Διπλός] στη Δπλ
            =Δπλ
        }
    }
}

```

```

}
Τελεστής "+=" {
    Διάβασε N
    Ένωσε γονικό ΜιαΤιμή στο A
    A=A+N
}

```

```

Συνάρτηση Ρίζα {
    Ένωσε γονικό ΜιαΤιμή στο A
}

```

```

        =Ρίζα(A)
    }
}

```

```

Κάτι.ΜιαΤιμή=100
Τύπωσε Κάτι.ΜιαΤιμή
Τύπωσε Κάτι.ΜιαΤιμή.Διπλός
Τύπωσε Κάτι.ΜιαΤιμή.Ρίζα()
Κάτι.ΜιαΤιμή+=1000
Τύπωσε Κάτι.ΜιαΤιμή.Διπλός

```

5) Δείκτες σε ιδιότητες

```

Κλάση Δες_Ιδιότητα {

```

```

    Ιδιωτικό:

```

```

        Πίνακας Βάση 1, A(3)

```

```

    Δημόσιο:

```

```

        Ιδιότητα Όνομα$ {Αξία} = "Δες_Ιδιότητα"

```

```

        Ομάδα Άλφα {

```

```

            Ιδιότητα Όνομα$ {Αξία}= "Άλφα(δείκτης) - Μόνο για ανάγνωση"

```

```

        }

```

```

        Ιδιότητα Άλφα {

```

```

            Αξία (δείκτης1) {

```

```

                Ένωσε γονικό A() στο A()

```

```

                Δες ok {

```

```

                    Αξία=A(δείκτης1)

```

```

                }

```

```

                Άδειασε Λάθος

```

```

                Αν όχι Ok τότε Λάθος "Δείκτης εκτός ορίων"

```

```

            }

```

```

        }

```

```

Κλάση:

```

```

    Τμήμα Δες_Ιδιότητα {

```

```

        .A():=100,450,1890

```

```

    }

```

```

}

```

```

M=Δες_Ιδιότητα()

```

```

Τύπωσε M.Άλφα(1),M.Άλφα(2),M.Άλφα(3)

```

```

Τύπωσε M.Άλφα.Όνομα$

```

```

Δες Ok {

```

```

    Τύπωσε M.Άλφα(10)

```

```

}

```

```

Τύπωσε "Το όνομά μου είναι: ";M.Όνομα$

```

```

Αν όχι Ok τότε Τύπωσε "Λάθος"+Λάθος$

```

Αναγνωριστικό: KANE [DEF]

1) Κάνε Αριθμό A, B, Γ

Κάνε Γράμμα A\$, B\$, Γ\$

φτιάχνουμε μόνο τοπικές μεταβλητές και δεν πρέπει να υπάρχουν άλλες τοπικές με το ίδιο όνομα, αλλιώς βγάζει λάθος

(να αποφεύγουμε ορισμό σε ρουτίνες, εκτός αν έχουμε καλό σχεδιασμό)

γράμματα και αριθμοί (όλοι οι τύποι) μπορούν να έχουν το ίδιο όνομα (όχι όμως το ίδιο όνομα σε δυο αριθμητικούς τύπους)

Αριθμητικοί τύποι: Αριθμός, Διπλός, Απλός, Λογιστικό, Μακρύς, Λογικός, Ακέραιος

Δίνουμε τα ονόματα ως Αριθμό, Διπλό, Απλό, Λογιστικό, Μακρύ, Λογικό, Ακέραιο

Αντιστοιχούν στα Decimal, Double, Single, Currency, Long, Boolean, Integer

Όλες οι μεταβλητές αριθμητικών τύπων κάνουν αυτόματη προσαρμογή στο τύπο τους όταν τους δίνουμε τιμή. Μπορεί να βγεί λάθος αν δεν χωράει ο αριθμός κατά την προσαρμογή.

Κάνε Τοπικά A, B, Γ ' θα φτιάξει νέες τοπικές ακόμα και αν υπάρχουν (για ρουτίνες)

μπορούμε να δίνουμε τύπους και τιμές (χωρίς σύμβολο τύπου), ή μόνο τιμές (και με σύμβολο τύπου)

Κάνε A=100~, B ως Λογιστικό=12323.213, Γ ως Μακρύς=100

\\ με την Κάνε Μακρύ φτιάχνουμε 32bit μεταβλητές

\\ μπορούμε όμως ορισμένες να τις δηλώσουμε με την ως ή με τύπο στο νούμερο

\\ με χρήση των & ~ @ # %

Κάνε Μακρύ α=10, β=20, γ ως Διπλός=30, δ=40#

Τύπωσε Τύπος\$(α), Τύπος\$(β), Τύπος\$(γ), Τύπος\$(δ)

2) KANE A(X)=X**2 : KANE A\$(X\$)=X\$+X\$

ΤΥΠΩΣΕ A(100), A\$("00")

Μπορούμε να φτιάχνουμε συναρτήσεις σε μια εντολή

Εδώ ουσιαστικά έχουμε δυο κανονικές συναρτήσεις όπως αυτές

Συνάρτηση A {

Διάβασε X

=X**2

}

Συνάρτηση A\$ {

Διάβασε X\$

=X\$+X\$

}

Αναγνωριστικό: ΚΑΤΑΣΤΑΣΗ [INVENTORY]

\\Λίστα ειδών

```

\\εδώ μόνο με μοναδικά κλειδιά
Κατάσταση Αλφα = 1,2,3,4,5
νέα προσθήκη: Αλφα=Λίστα:=1,2,3,4,5
\\εδώ με όμοια κλειδιά (στην τύπου ουρά δεν διαγράφουμε κλειδιά οπουδήποτε, παρά μόνο
από το τέλος)
Βήτα=Ουρά:=1,2,3,1,2,3
Βήτα=(Ουρά:=1,2,3,1:="α",2:="β",3:="γ") ' βάζουμε παρενθέσεις αν μπει σε συνάρτηση λόγω
του ":" χαρακτήρα.
Πέτα Βήτα 3 ' θα πετάξει 3 στοιχεία από το τέλος
Τύπωσε Βήτα ' 1 2 3
Βήτα=Ουρά:=1,2,3,1,2,3
Πέτα Βήτα Εως 2 ' θα κρατήσει τα 2 πρώτα
Τύπωσε Βήτα ' 1 2
για i=0 έως μήκος(Βήτα)-1
\\ τιμή ανά θέση και κλειδί ανά θέση
Τύπωσε Βήτα$(i!), Εκφρ$(Βήτα, i)
Επόμενο i
\\ Εύρεση κλειδιού, δίνει την τιμή και τη θέση
Αν Υπάρχει(Βήτα, 2) Τότε Τύπωσε Εκφρ(Βήτα)=200, Εκφρ(Βήτα!)=1
Επιστροφή Βήτα, 2:="Αλφαριθμητικό"
Προσθήκη Βήτα, "κάτι":="Αυτό"
Αν Υπάρχει(Βήτα, 2) Τότε Τύπωσε Εκφρ$(Βήτα)="Αλφαριθμητικό", Εκφρ(Βήτα!)=1
Τύπωσε Βήτα$("κάτι")="Αυτό"

```

```

επιστροφή Αλφα, 1:="οκ", 2:=-500
\\ η συνάρτηση Υπάρχει σε Ουρά θα δείξει στα όμοια κλειδιά το τελευταίο που μπήκε στην ουρά.
Τύπωσε Υπάρχει(Αλφα, 1)
Τύπωσε Αλφα$(1), Αλφα(2)
Τύπωσε Μήκος(Αλφα)
Προσθήκη Αλφα, "κάτι άλλο":="αυτό"
Αφαίρεση Αλφα, 3,4,5

```

```

\\ Παράδειγμα:
Συνάρτηση Πίνακας_2D { Πίνακας α(Αριθμός, Αριθμός) : =α()}
Συνάρτηση Πίνακας_1D {=0,1,2,3,4,5,6,7,8,9}
Κατάσταση α=0:=Πίνακας_2D(5,5), 1:=Πίνακας_1D(),2,3:=100,4
α(0)(2,2)=α(1)(5)*10
Τύπωσε α(1)(5), α(0)(2,2)
Πίνακας β(5)
Για i=0 έως 1 {

```

```

    β(i)= α(i)()
}
Για i=2 έως 4 {
    β(i)=α(i)
}
β(0)(2,2)=1000
Τύπωσε β(0)(2,2), α(0)(2,2)
Τύπωσε β(1)(3), β(2), β(3), β(4)
\\ η επιστροφή στη κατάταξη γίνεται με ειδική εντολή
Επιστροφή α, 4:=400
\\ όχι όμως για πίνακες σε αυτήν:
α(1)(5)+=100
Τύπωσε β(4), α(4), α(1)(5), β(1)(5) ' 4, 400, 105, 5
\\ τα νούμερα στη κατάταξη είναι κλειδιά
\\ μπορεί να είναι αριθμοί ή αλφαριθμητικά
Προσθήκη α, "2X2":=Πίνακας_2D(2,2)
α("2X2")(0,0)=100
Τύπωσε α("2X2")(0,0) ' 100
\\ μπορούμε να αφαιρούμε στοιχεία
\\ δεν αλλάζει η "αρίθμηση" γιατί στην ουσία είναι κλειδιά..μοναδικά!
\\ η αφαίρεση δεν λειτουργεί για Ουρά
Αφαίρεση α, 1,4
Τύπωσε Μήκος(α) ' 4, δίνει το μήκος σε στοιχεία!
Τύπωσε Υπάρχει(α, 0) ' -1 ναι υπάρχει
Τύπωσε Υπάρχει(α, 1) ' 0 δεν υπάρχει

```

```

\\ μπορούμε να δούμε τα κλειδιά
\\ με χρήση ιδιοτήτων.
Με α, "done", αληθές, "keytostring" ως κλειδί$
Για i=0 έως μήκος(α)-1
Με α, "index", i
Τύπωσε Κλειδί$
Επόμενο i

```

```

\\ ή με απλό τρόπο:
Για i=0 έως μήκος(α)-1
ββ$=α$(i!) \\ απλά εξάγουμε μια τιμή αλφαριθμητική ή το ""
\\ όμως πάμε το index στο i και κάνουμε το done =true
\\ το "done" είναι σημαία που λέει ότι το index είναι έγκυρο.
\\ οπότε εδώ δεν δίνουμε καθόλου νούμερο, αλλά χρησιμοποιεί
\\ η κατάταξη το τρέχον (είναι στην ιδιότητα index)
\\ Η συνάρτηση Εκφρ$ με το ! έχει σκοπό να δώσει το κλειδί
Τύπωσε Εκφρ$(α!)

```

Επόμενο ι

Η εκφρ\$(α, 0) δίνει πάντα το κλειδί του στοιχείου 0 (το πρώτο στη κατάταξη)

\ \ Μπορούμε να βρούμε τη θέση (προσοχή γιατί οι θέσεις αλλάζουν αν ταξινομήσουμε, ή διαγράψουμε)

Αν υπάρχει(α, "2X2") τότε κ=εκφρ(α!)

Τύπωσε "θέση=";κ

Τύπωσε α(κ!)(0,0)

Τύπωσε Τύπος\$(α(κ!))

Πίνακας αα()

αα()=α(κ!)

Τύπωσε αα(0,0)

Αναγνωριστικό: ΚΛΑΣΗ [CLASS]

Η κλάση μπορεί να έχει τρία διαφορετικά μέρη (όπως και η ομάδα) βάσει των Ιδιωτικό:, Δημόσιο: και Κλάση:. Μπορούμε να έχουμε πολλά τμήματα από το καθένα, δεν παίζει ρόλο η σειρά.

Ένα τμήμα με ίδιο όνομα με τη κλάση είναι αυτό που θα κληθεί στη συνάρτηση

Όνομα_Κλάσης()

Οι κλάσεις είναι πάντα γενικές. Όμως μενουν ως έχουν μέχρι μια άλλη γενική με ίδιο όνομα τις αλλάζουν.

1)

Κλαση αλφα {

Ιδιωτικό:

ζ=10

Πίνακας αλφα(10)=5

Δημόσιο:

χ, υ

τμήμα αλφα {

διαβασε .X, .υ, .ζ

.χ/= .ζ

}

συνάρτηση γέμισε_πίνακα {

=.αλφα()

}

}

2)

Ομαδα1=αλφα(10,20,5)

```

για Ομάδα1 {
    τυπωσε .X, .Y
    Δες { τύπωσε .ζ}
    Δες {τύπωσε .αλφα(5)}
}

```

```

Πίνακας κ()
κ()=Ομάδα1.γέμισε_πίνακα()
Τύπωσε κ(5)
3)

```

```

Κλάση Άλφα {

```

```

    Ιδιωτικό:

```

```

        X, Y

```

```

    Δημόσιο:

```

```

        Αξία {

```

```

            = .X/ .Y

```

```

        }

```

```

Κλάση:

```

```

    Τμήμα Άλφα {

```

```

        Διάβασε .X, .Y

```

```

    }

```

```

}

```

```

N=Άλφα(100,200)

```

```

Τύπωσε N

```

```

M=Ομάδα(N) \ \ γυρίζει το αντικείμενο μόνο, δεν υπάρχει το τμήμα ΆΛΦΑ.

```

```

3) Ειδική συνάρτηση Διαγραφή {}

```

\ \ Η συνάρτηση αυτή θα κληθεί από την Καθαρό εφόσον δεν υπάρχει άλλος δείκτης να δείχνει την ομάδα που θέλουμε να διαγράψουμε. Μετρητής_Αλφα=0

```

Κλάση Άλφα {

```

```

    Ένας_Αριθμός

```

```

    Διαγραφή {

```

```

        Τύπωσε "Διαγραφή"

```

```

        Μετρητής_Αλφα--

```

```

    }

```

```

Κλάση:

```

```

    Τμήμα Άλφα (.Ένας_Αριθμός){

```

```

        Μετρητής_Αλφα++

```

```

    }

```

```

}

```

```

α=Άλφα(12345)

```

```

Τύπωσε Μετρητής_Αλφα=1

```

```

\ \ φτιάχνουμε έναν δείκτη στο α

```

κ->α

Τύπωσε κ=>Ένας_Αριθμός=12345

\| βάζουμε στο κ τη κενή ομάδα (θα το κάνει ο διερμηνευτής αυτό)

κ->0&

\| Τώρα μειώθηκε ο αριθμός

\| Μόνο η Καθαρό καλεί τη Διαγραφή

Καθαρό α ' καλεί την διαγραφή αν είναι ο τελευταίος δείκτης στο αντικείμενο

Τύπωσε Μετρητής_Αλφα=0

Αναγνωριστικό: ΛΑΜΔΑ [LAMBDA]

A=λάμδα ->100

Τύπωσε A() \| 100

B=10

\| η B συλλαμβάνεται από την λάμδα A, ως αντίγραφο

A=λάμδα B -> {

=B : B--

}

M=A

Τύπωσε A(), A(), B, M()

A=M \| ξαναπαίρνει τον ορισμό της M() με το B σε 9

Τύπωσε A(), A(), B

Οι συναρτήσεις λάμδα είναι συναρτήσεις σε μεταβλητές. Μπορούμε να τις αλλάξουμε. Αλλάζουν όμως με αντικατάσταση.

Μπορούν να πάρουν παραμέτρους και μπορούν να κληθούν με ή χωρίς επιστροφή τιμής.

Μπορεί να επιστρέφει οτιδήποτε ακόμα και μια άλλη συνάρτηση λάμδα. Μπορούμε να τις περνάμε ως παραμέτρους.

Αν μας ενδιαφέρει αποτέλεσμα αλφαριθμητικό τότε:

B\$="ΓΕΙΑ ΧΑΡΑ"

Θ\$="!"

Αλφα\$=λάμδα\$ B\$, Θ\$ ->B\$+" "+γράμμα\$+Θ\$

Τύπωσε Αλφα\$("ΓΙΩΡΓΟ")

Πίνακας A\$(10)

A\$(2)=Αλφα\$

Τύπωσε A\$(2)("ΑΝΔΡΕΑ")

Κατάσταση M = "μία λάμδα":=Αλφα\$, 200:="ΓΕΙΑ"

Τύπωσε M\$("μία λάμδα")("ΠΕΤΡΟ"), M\$(200)

Για να έχουμε αναδρομή στη λάμδα και επειδή μπορεί να αλλάζει όνομα (ανάλογα με το ποιο

όνομα την κρατάει), έχουμε τα Λάμδα() και Λάμδα\$() που λειτουργούν εντός της λάμδα για να την ξανατρέξουν.

Τύπωσε λάμδα (α)->{α**2}(3) ' δίνει 9

\ \ μπορεί να γραφτεί και έτσι

Τύπωσε λάμδα (α)->{

=α**2

}(3) ' δίνει 9

Αναγνωριστικό: ΜΑΚΡΥΣ [LONG]

μακρυσ α,β, γ=34.5 , κ[100] ' 32 bit (4byte)

μακρυσ μακρύς α1,β1, γ1=34.5, κ1[100] ' 64 bit (8 byte)

Πίνακας π1(10, 10) ως μακρύς=1, π2(10, 10) ως μακρύς μακρύς=1

α=4

? τυπος\$(β)

ΛΙΣΤΑ

φτιάχνουμε ακέραιους των 4bytes

Περιοχή για μακρύ Α -2,147,483,648 έως 2,147,483,647

Αναγνωριστικό: ΜΕ.ΑΝΤΙΚΕΙΜΕΝΟ [WITH]

Πολύπλοκη εντολή για να αναλυθεί εδώ.

Χρησιμοποιείται για να αλλάξουμε τιμές σε ιδιότητες αντικειμένων, να διαβάσουμε, και να αντιστοιχίσουμε μια μεταβλητή στην ιδιότητα. Η αντιστοίχιση μπορεί να γίνει και σε όνομα πίνακα, δηλαδή με παρενθέσεις, για να δίνουμε ορίσματα όταν διαβάζουμε ή όταν γράφουμε στην ιδιότητα.

Ένα απλό παράδειγμα με δυο ιδιότητες, την ReadyState και την ResponseText

η <έτοιμο> δεν είναι απλά μια μεταβλητή αλλά ένα αντικείμενο που συνδέεται με μια ιδιότητα ενός άλλου αντικειμένου. Η σύνδεση είναι έμμεση. Αν διαγραφεί το αντικείμενο και χρησιμοποιήσουμε την <έτοιμο> τότε θα βγει λάθος!

Αντικείμενα που ορίζουμε με την Όρισε και τις ιδιότητες τους δεν μπορούμε να τις δώσουμε ως επιστροφή από το τμήμα που τα δημιουργούμε, γιατί το πραγματικό αντικείμενο θα διαγραφεί και αυτό που θα δώσουμε θα δείχνει το τίποτα!

Μπορούμε να περνάμε αυτές τις μεταβλητές με αναφορά όπως τις κανονικές μεταβλητές σε κλήσεις με τη χρήση του &.

```

Όρισε Κείμενοhtml "Microsoft.XMLHTTP"
ΔοκιμήURL$ = "http://httpbin.org/"
Με Κείμενοhtml, "readyState" ως έτοιμο
Μέθοδος Κείμενοhtml "open", "GET", ΔοκιμήURL$, Αληθές
Μέθοδος Κείμενοhtml "send"
Ενώ έτοιμο<>4 {
    Τύπωσε Πάνω έτοιμο
    Αν Δείκτης Τότε Έξοδο
    Αναμονή 1
    Ανανέωση
}
Τύπωσε
Έγγραφο ββ$
Αν έτοιμο=4 Τότε {
    Με Κείμενοhtml, "responseText" ως έτοιμο$
    ββ$=έτοιμο$
}
Αναφορά ββ$
Σημ : Πρόχειρο ββ$
Όρισε Κείμενοhtml Τίποτα

```

Αναγνωριστικό: ΜΕΘΟΔΟΣ [METHOD]

Καλούμε μέθοδο σε αντικείμενο τύπου Com (όπως και οι φόρμες χρήστη) τα οποία ορίζουμε με την εντολή Όρισε

Οι μέθοδοι μπορεί να έχουν προαιρετικές παραμέτρους, οπότε η απουσία τους δεν θα δημιουργήσει πρόβλημα!

Επίσης μπορεί να γυρνούν τιμή ή όχι

Αν γυρνούν τιμή αυτή μπορεί να είναι αντικείμενο. Μπορούμε να επιστρέφουμε τιμή σε μεταβλητή, ή σε νέα μεταβλητή (να σκιάζουμε υπάρχουσα). Επίσης μπορούμε να αν το δέχεται το αντικείμενο, να χρησιμοποιήσουμε τη ΜεΓεγονότα παραλλαγή, ώστε να πάρουμε γεγονότα κατά την διάρκεια ζωής του αντικειμένου

Ειδικά για γεγονότα που δεν έχουν σχέση με τις φόρμες και τα στοιχεία της M2000, οι συναρτήσεις θα έχουν το όνομα της μεταβλητής μαζί με μια κάτω παύλα _ και μετά το όνομα του γεγονότος. Σε περίπτωση που κάποιες παράμετροι του γεγονότος είναι με πέρασμα με αναφορά τότε χρησιμοποιούμε στην συνάρτηση του γεγονότος μια Διάβασε Νέο &παρ1, &παρ2 ώστε στο τέλος της εκτέλεσης οι τιμές των παρ1 και παρ2 να αντιγραφούν πίσω στο αντικείμενο που παρήγαγε το γεγονός. Σε κάθε περίπτωση υπάρχουν δυο τελευταίοι παράμετροι στο σωρό της

κλήσης (ο οποίος όπως και να έχει καταστρέφεται μετά τη κλήση), ένα αλφαριθμητικό με πληροφορίες (αν υπάρχουν αλλιώς κενό αλφαριθμητικό) και το αντικείμενο

Στο παράδειγμα φτιάχνουμε μια φόρμα

```
Όρισε Φόρμα1 Φόρμα
Με Φόρμα1, "Title","η φόρμα μου"
Μέθοδος Φόρμα1,"Show", 1
Όρισε Φόρμα1 Τίποτα
```

Στο παράδειγμα θα φτιάξουμε ένα αρχείο Xml και θα το διαβάσουμε μέσω του αντικειμένου Msxml2.DOMDocument.6.0 που υπάρχει στα Windows και μπορεί να φορτωθεί στη M2000

```
Έγγραφο α$={<?xml version="1.0"?>
```

```
<Ονόματα>
```

```
  <στοιχείο αα="69">
```

```
    <όνομακωδ>VB6</όνομακωδ>
```

```
    <ποσότητα>2</ποσότητα>
```

```
  </στοιχείο>
```

```
  <στοιχείο αα="70">
```

```
    <όνομακωδ>C++</όνομακωδ>
```

```
    <ποσότητα>3</ποσότητα>
```

```
  </στοιχείο>
```

```
  <στοιχείο αα="71">
```

```
    <όνομακωδ>M2000</όνομακωδ>
```

```
    <ποσότητα>10</ποσότητα>
```

```
  </στοιχείο>
```

```
</Ονόματα>
```

```
}
```

```
Σώσε Έγγραφο α$,"άλφα.xml", -2 ' χωρίς BOM
```

```
Αναμονή 100
```

```
Όρισε ΑντικείμενοXml "Msxml2.DOMDocument.6.0"
```

```
Μέθοδος ΑντικείμενοXml, "Load", κατ$+"άλφα.xml" ως ok
```

```
Αν ok τότε {
```

```
  Όρισε ΣτοιχείοXML με ΑντικείμενοXml, "documentElement"
```

```
  Μέθοδος ΣτοιχείοXML, "selectSingleNode", "στοιχείο[@αα=70]" ως ΕπιλεγμένοΣτοιχείο
```

```
  Μέθοδος ΕπιλεγμένοΣτοιχείο, "selectSingleNode", "όνομακωδ" ως ΕπιλεγμένοΠεδίο
```

```
  Με ΕπιλεγμένοΠεδίο, "text" ως η_τιμή_μου$
```

```
  Τύπωσε η_τιμή_μου$
```

```
  η_τιμή_μου$="George"
```

```
  Μέθοδος ΑντικείμενοXml, "Save", κατ$+"αλφα.xml" ως Ok
```

```
  Μέθοδος ΣτοιχείοXML, "getElementsByTagName", "στοιχείο" ως Αντικείμενο_NodeList
```

```

Με Αντικείμενο_NodeList , "length" ως Αντικείμενο_NodeList_μήκος
Τύπωσε Αντικείμενο_NodeList_μήκος
Για i=0 έως Αντικείμενο_NodeList_μήκος-1 {
    Μέθοδος Αντικείμενο_NodeList , "item", i ως Αντικείμενο_NodeList_στοιχείο
    Μέθοδος Αντικείμενο_NodeList_στοιχείο, "getElementsByTagName", "όνομακωδ" ως
Αντικείμενο_NodeList2
    Μέθοδος Αντικείμενο_NodeList2 , "item", 0 ως Αντικείμενο_NodeList_στοιχείο2
    Με Αντικείμενο_NodeList_στοιχείο2 , "text" ως α12$
    Πένα 12 {
        Τύπωσε α12$
    }
    Όρισε Αντικείμενο_NodeList_στοιχείο2 Τίποτα
    Όρισε Αντικείμενο_NodeList2 Τίποτα
    Όρισε Αντικείμενο_NodeList_στοιχείο Τίποτα
}
Όρισε Αντικείμενο_NodeList Τίποτα
Όρισε ΕπιλεγμένοΠεδίο Τίποτα
Όρισε ΕπιλεγμένοΣτοιχείο Τίποτα
Όρισε ΣτοιχείοXML Τίποτα
}
Όρισε ΑντικείμενοXml Τίποτα
Εγγραφο ααα$
Φόρτωσε.Έγγραφο ααα$, "alfa.xml"
Αναφορά ααα$

```

Και τα αντικείμενα Κατάσταση και Έγγραφο είναι Com αντικείμενα και μπορούμε να χρησιμοποιήσουμε μεθόδους

Όπως στο παράδειγμα παρακάτω:

```

Έγγραφο α$={réché
    sin
    peach
    pêche
}

```

Τύπωσε "Με 1036 Γαλλικά"

Μέθοδος α\$, "SetLocaleCompare", 1036

Ταξινόμηση Αύξουσα α\$, 1,4

Αναφορά α\$

Τύπωσε "Με 1033 Αγγλικά"

Μέθοδος α\$, "SetLocaleCompare", 1033

Ταξινόμηση Αύξουσα α\$, 1,4

Αναφορά α\$

Δες παράδειγμα στην συνάρτηση Ιδιότητα\$()

Αναγνωριστικό: ΜΕΤΑΒΛΗΤΕΣ [VARIABLES]
ΔΕΣ ΜΕΤΑΒΛΗΤΗ

Αναγνωριστικό: ΜΕΤΑΒΛΗΤΗ [VARIABLE]
ΜΕΤΑΒΛΗΤΗ Α=10, Β=100, Δ(1 ΕΩΣ 30,-5 ΕΩΣ 5)=1
ΜΕΤΑΒΛΗΤΕΣ Α ΩΣ ΑΚΕΡΑΙΟΣ=10, Β="ΓΡΑΜΑ", Γ ΩΣ ΜΑΚΡΥΣ=100
Δημιουργεί μεταβλητές τοπικές όπως οι ΤΟΠΙΚΗ / ΤΟΠΙΚΕΣ, αλλά όχι αν υπάρχει ήδη τοπική με το ίδιο όνομα (αλλάζει όμως τη τιμή έχουμε δώσει και είναι ίδιου τύπου)

Αναγνωριστικό: ΜΕΤΑΒΛΗΤΗ_ [VAR]
Δες ΜΕΤΑΒΛΗΤΗ

Αναγνωριστικό: ΟΜΑΔΑ [GROUP]
ΟΜΑΔΑ

1)

ομαδα αλφα { α, β, Γ=5, δ\$ }

τύπωσε αλφα.γ

ομαδα βητα {

 πινακας α(20)=20

 α(1)=10

 κ=0

 συναρτηση ενα {

 αυτο.κ++

 =αυτο.α(αυτο.κ)

 }

}

τύπωσε βητα.ενα()

τμημα αλλο {

 διαβασε &αλληλομαδα

```
    ? αλληλομαδα.ενα()  
}
```

αλλο &βητα

2) Ιδιωτικό και δημόσιο μέρος στον ορισμό της ομάδας

Ομάδα Άλφα {

Ιδιωτικό:

 X, Y=100, Z

Δημόσιο:

 Τμήμα Δείξε_μου {

 Τύπωσε .X, .Y, .Z

 }

}

Άλφα.Δείξε_μου

Τύπωσε Έγκυρο(Άλφα.X) \ 0 - δεν είναι έγκυρο

3) Προσωρινό μέρος στον ορισμό ομάδας

Ομάδα Άλφα {

Ιδιωτικό:

 X, Y=100, Z

Δημόσιο:

 Τμήμα Δείξε_μου {

 Τύπωσε .X, .Y, .Z

 }

Κλάση:

 Τμήμα Πάρε_Τιμές {

 Διάβασε .X, .Y, .Z

 }

}

Άλφα.Δείξε_μου

Άλφα.Πάρε_Τιμές 1,2,100

Άλφα.Δείξε_μου

M=Άλφα

Δες ok {

 M.Πάρε_Τιμές 1,2,100

}

Αν όχι ok τότε Τύπωσε "Λάθος:" + Λάθος\$

' η M δεν έχει το τμήμα που φορτώνει τις μεταβλητές!

3) Τελεστές - Δες το Τελεστής

4) Ιδιότητες - Δες το Ιδιότητα

5) Παλιός τρόπος δημιουργίας ομάδας με αλφαριθμητικό

A\$=" X,Y,Z"

Ομάδα Άλφα Τύπος A\$

Άλφα.X+=100

Τύπωσε Άλφα.X

6) Εξαγωγή σειριακή σε μεταβλητές με αναφορά

Ομάδα Άλφα {X, Y, Z}

Διάβασε από Άλφα, X1, Y1, Z1

X1+=100

Y1=1

Z1=200

Για Άλφα {

 Τύπωσε .X, .Y, .Z

}

7)Μέλη ομάδας

Ομάδα Άλφα {

 X, Y, Z

 Πίνακας A1()

 A=Λάμδα X=1 -> {=X : X++}

}

Σ=Ομάδα.Σύνολο(Άλφα)

Για Ι=1 έως Σ {

 Τύπωσε Μέλος\$(Άλφα, Ι), Μέλους.Τύπος\$(Άλφα, Ι)

}

Τύπωσε Άλφα.A(), Άλφα.A()

8) Συγχώνευση Ομάδας

Ομάδα Άλφα {X=10, Y=20}

Ομάδα Βήτα {Y=30, Z=100}

Άλφα=Βήτα

Τύπωσε Άλφα.Z, Άλφα.Y

9) Ομάδα σε πίνακα ως κοινή ομάδα

οι συναρτήσεις, τα τμήματα και οι τελεστές γράφονται μια φορά για όλα τα στοιχεία πίνακα

Στους πίνακες δεν γίνεται συγχώνευση αλλά αντικατάσταση

Ομάδα Άλφα {

 X=10,Y=20

 Συνάρτηση Γινόμενο {

 =.X, .Y

 }

}

Πίνακας Βάση 1, A(20)=Άλφα

A(2).X+=10

Τύπωσε A(2).Γινόμενο()

10)Ομάδα σε πίνακα χωρίς κοινή ομάδα (Ισχύει και για Κατάσταση)

Μπορούμε να αλλάξουμε μια συνάρτηση σε συγκεκριμένο αντικείμενο στο πίνακα. Ο πίνακας μπορεί να πάρει διαφορετικές ομάδες.

Ομάδα Άλφα {

 X=10,Y=20

```

        Συνάρτηση Γινόμενο {
            =.X* .Y
        }
    }
    Πίνακας Βάση 1, A(20)
    A(2)=Άλφα
    A(2).X+=10
    Τύπωσε A(2).Γινόμενο()
    Για A(2) {
        Ομάδα Προσωρινή {
            Συνάρτηση Γινόμενο {
                =.X*.Y*100
            }
        }
        \\ κάνουμε συγχώνευση
        Αυτό=Προσωρινή
    }
    Τύπωσε A(2).Γινόμενο()
    Κατάσταση Βήτα="Θέση A":=Άλφα
    Τύπωσε Βήτα("Θέση A").Γινόμενο()
    Βήτα("Θέση A").X+=20
    Τύπωσε Βήτα("Θέση A").Γινόμενο()
    Για Βήτα("Θέση A") {
        Ομάδα Προσωρινή {
            Συνάρτηση Γινόμενο {
                =.X*.Y*100
            }
        }
        \\ κάνουμε συγχώνευση
        Αυτό=Προσωρινή
    }
    Τύπωσε Βήτα("Θέση A").Γινόμενο()
    11) Αλλαγή τιμής μεταβλητές ομάδας
    Ομάδα Άλφα {
        X=10
        Πίνακας Βάση 1, M(10)
        Τμήμα ΠάρεΤιμές {
            Διάβασε K, Λ
            \\ εδω θέλει <=
            .X<=K*Λ
            \\ οι πίνακες ως αντικείμενα δεν θέλουν <=
            Αν Λ<>0 Τότε .M(2)=K/Λ
        }
    }

```

}

12) κατασκευής ομάδας με Κλάση - δεξ Κλάση

Αλφα.ΠάρεΤιμές 100,50

Τύπωσε Άλφα.Χ, Άλφα.Μ(2)

Αναγνωριστικό: ΟΡΙΣΕ [DECLARE]

ΟΡΙΣΕ Μηνυμα ΑΠΟ "user32.MessageBoxW" {Μακρυσ Α, κειμενο\$, κειμενο2\$, Μακρυσ τυπος}

Τύπωσε Μηνυμα(Παραθυρο, "Γεια χαρά", "Γιώργος", 2)

δεξ DECLARE

mybuf\$=επαν\$(χαρ\$(0), 1000)

\\ Αν ορίσουμε συνάρτηση με το C πριν, τότε λέμε ότι είναι κλήση τύπου C

\\ το σύμβολο ... δηλώνει λίστα παραμέτρων. Ισχύει μόνο για το τελευταίο στοιχείο.

\\ μόνο Unicode κλήσεις για C.

\\ Το Γενικό μπορούμε να το αφήσουμε αν θέλουμε την συνάρτηση τοπική.

Όρισε Γενικό MyPrint Από C "msvcrt.swprintf" { &sBuf\$, sFmt\$, ... }

\\ το σύμβολο ! πριν από μια αριθμητική έκφραση για παράμετρο...

\\.....λέει στον διερμηνευτή ότι θα καταχωρηθεί ως Μακρύς ή Long.

A=MyPrint(&myBuf\$, "Γειά P1=%s, P2=%d, P3=%.4f, P4=%s", "ABC", !123456, 1.23456, "xyz")

Τύπωσε Αρις\$(myBuf\$,A)

Η όρισε κάνει αρκετά πράγματα, ένα επίσης από αυτά είναι ότι φτιάχνει πίνακες αντικειμένων, και αν ζητήσουμε με γεγονότα, τότε θα φτιάξει και τα ανάλογα αντικείμενα (είναι κρυφά) τα οποία ανακατευθύνουν τα γεγονότα σε συναρτήσεις στο τμήμα που δημιουργούμε τα αντικείμενα. Για τα στοιχεία της διεπαφής δεν βάζουμε το ΜεΓεγονότα.

Όρισε Α() πάνω ' αυξάνει κατά ένα τα αντικείμενα και φτιάχνει και για το νέο αντικείμενο για γεγονότα

Όρισε Α() τίποτα ' διαγράφει τα αντικείμενα

Χρησιμοποιούμε την Μέθοδος και την Με, με την πρώτη για να καλέσουμε μεθόδους, με ή χωρίς επιστροφή τιμής, και τη Με για να βάζουμε τιμές ή να βγάζουμε σε μεταβλητές. Οι μεταβλητές στην Με συνδέονται με τα αντικείμενα, ώστε μέσω αυτών να διαβάζουμε ή να γράφουμε άμεσα τιμές.

Γενικά καλύτερα να δει κανείς παραδείγματα πάνω σε αυτά.

Αναγνωριστικό: ΠΙΝΑΚΑΣ [DIM]

Εντολή που ορίζει πίνακες με έως 10 διαστάσεις, αριθμητικούς και αλφαριθμητικούς.

Νέες προσθήκες από την έκδοση 8
Επιστροφή πίνακα από συνάρτηση:

```
Συναρτηση ΕναςΠινακας {  
    Διαβασε τοσα  
    Πινακας α(τοσα) : για i=0 εως τοσα-1 {α(i)=i+1}  
    =α()  
}  
Πινακας κ()  
κ()=ΕναςΠινακας(10)  
Για i=0 εως 9 {τυπωσε κ(i)}
```

Όλοι οι πίνακες σε κάθε διάσταση με όρισμα N ξεκινούν από το 0 και έχουν N στοιχεία ή άνω όριο N-1

1) Τοπικοί πίνακες

ΠΙΝΑΚΑΣ A(20)=5, A\$(1000) , A%(20000)=1

Πινακας A() ' χωρίς στοιχεία

Πίνακας Βάση 1, A(),B(10) ' με βάση το 1 (παίρνει τιμές 1 ή 0)

\\ ο B() έχει στοιχεία από το 1 έως το 10

Τύπωσε διάσταση(B(),0) ' μας δίνει την βάση

\\ εδώ αλλάζουμε βάση

Πίνακας Βάση 0 στο A(), B() ' αλλαγή βάσης στο A()

2) Σφαιρικός - Καθολικός πίνακας (αλλαγή από έκδοση 8)

ΓΕΝΙΚΟ K(300)=20

3) Τοπικο K(20) - τοπικός πίνακας (όλοι τοπικοί είναι εξ ορισμού) αλλά δες την ΓΙΑ ΑΥΤΟ { }

4) Διαγραφή στοιχείων πίνακα (το όνομα παραμένει, αλλά όλα τα στοιχεία σβήνουν, και τυχόν αντικείμενα τύπου ΕΓΓΡΑΦΟ)

ΠΙΝΑΚΑΣ A(0)

5) Αλλαγή διαστάσεων με λιγότερα, ίδια ή περισσότερα στοιχεία, χωρίς να αλλαχθούν όσα απομείνουν.

ΠΙΝΑΚΑΣ A(2,2)=5 ' 2X2= 4 στοιχεία και βάζουμε αρχική τιμή το 5

ΠΙΝΑΚΑΣ A(2) ' διαγράφουμε τα τελευταία 2

ΤΥΠΩΣΕ A(1) ' 5

ΠΙΝΑΚΑΣ A(2,2) ' προσθέτουμε 2 στοιχεία με 0 τιμή

ΤΥΠΩΣΕ A(0,1) '5

ΤΥΠΩΣΕ A(1,0) '0

A(1,0)=5,6 ' βάζουμε στο A(1,0) το 5 και στο επόμενο στη μνήμη που είναι εδώ

το A(0,1) το 6

ΤΥΠΩΣΕ A(1,1) '5

ΠΙΝΑΚΑΣ A(4) ' Τώρα έχουμε μονοδιάστατο πίνακα με τα ίδια στοιχεία

ΤΥΠΩΣΕ A(3) ' 6

Παρατηρήσεις: Μπορούμε να προσθέτουμε στοιχεία αυξάνοντας μόνο την πρώτη διάσταση αν δεν θέλουμε να μετακινηθούν τα υπάρχοντα στοιχεία σε άλλες θέσεις.

6) Μόνο για αλφαριθμητικούς πίνακες απόδοση αριθμητικών τιμών (στην έκδοση 8 υπάρχει και το ΣΤΟΚ)

ΓΡ=1

ΠΙΝΑΚΑΣ Α\$(ΓΡ,5) ' 1 γραμμή με 5 πεδία

Α\$(ΓΡ-1,0) := "ΓΙΩΡΓΟΣ", 100, 20, "ΗΠΕΙΡΟΣ", 12.5

ΓΡ++

ΠΙΝΑΚΑΣ Α\$(ΓΡ,5) ' δίνουμε μια γραμμή ακόμα

Α\$(ΓΡ-1,0) := "ΑΛΕΞΑΝΔΡΟΣ", 101, 120, "ΜΑΚΕΔΟΝΙΑ", 14.5

ΒΑΛΕ &Α\$() : ΔΙΑΒΑΣΕ &Α() ' φτιάχνουμε μια αναφορά πίνακα που δίνει όμως αριθμητικό αποτέλεσμα

Χ=0 ' επιλέγουμε τη πρώτη γραμμή

ΤΥΠΩΣΕ Α\$(Χ,0),Α(Χ,1),Α(Χ,2),Α\$(Χ,3),Α(Χ,4)

Παρατηρήσεις: Μια αριθμητική τιμή θα διαβαστεί και σαν αλφαριθμητική. Το ίδιο και μια αλφαριθμητική θα διαβαστεί ως αριθμητική στον αριθμητικό πίνακα που έχει αναφορά στον αλφαριθμητικό.

7) Μετατροπή ενός στοιχείου αλφαριθμητικού πίνακα σε ΕΓΓΡΑΦΟ

ΠΙΝΑΚΑΣ Α\$(10)

ΕΓΓΡΑΦΟ Α\$(4) = {Πρώτη γραμμή
Δεύτερη με αλλαγή γραμμής
}

ΑΝΑΦΟΡΑ Α\$(4)

τα έγγραφα μπορούν να χρησιμοποιηθούν όπως και τα αλφαριθμητικά αλλά βρίσκονται στη μνήμη χωρισμένα σε παραγράφους. Μπορούν να χρησιμοποιηθούν ως δυναμικά αναπτυσσόμενες λίστες.

8) Σκίαση πίνακα

Πίνακας Νέο Α(10)

Εδώ θα σκιάσουμε τον Α() αν υπάρχει - αυτό σε μια Ρουτίνα ή σε Για { } όπου όλα τα νέα (μεταβλητές και τμήματα και συναρτήσεις) είναι προσωρινά.

Η εντολή ΠΙΝΑΚΑΣ φτιάχνει πίνακες αριθμητικούς, αλφαριθμητικούς και ακέραιους ενώ μπορεί και να τους γεμίσει με μια τιμή.

Με το πέρας του τμήματος που δημιουργήσε τους πίνακες, παύουν οι πίνακες να υπάρχουν.

Από την έκδοση 6.1 οι πίνακες μπορούν να ξαναοριστούν χωρίς να σβήσουν τα περιεχόμενα τους (αν θέλουμε τα σβήνουμε με το =0 ή =""). Μπορούμε να μεγαλώσουμε ή να μικρύνουμε τον πίνακα ή να αλλάξουμε διάσταση!

Δηλαδή μπορούμε τον Α(2,2) με τιμές από (0..1, 0..1) να τον κάνουμε Α(4) (0..3)

Μπορούμε να περνάμε με αναφορά τιμής πίνακες σε συναρτήσεις και τμήματα όπου εκεί μπορούν να μεταβληθούν οι τιμές των στοιχείων τους και στην επιστροφή να φανεί άμεσα! Δείτε

την ΒΑΛΕ

ΠΙΝΑΚΑΣ Α(100)

A(24)=1,2,3

Το στοιχείο 26 θα έχει τιμή 3

Ειδικά για τους πίνακες αλφαριθμητικών υπάρχει το ":"=" ως χειριστής όπου διαβάζει νούμερα και αλφαριθμητικά ταυτόχρονα.

ΠΙΝΑΚΑΣ Α\$(100)

A\$(10):="",34>Α, Β, ΜΗΚΟΣ("ΑΛΦΑ")

τα στοιχεία 11 και 12 θα πάρουν αριθμητικές τιμές ως αλφαριθμητικά.

Επίσης μπορούμε να βάλουμε αλφαριθμητικά με αγκύλες..(αλλά δεν μπορούμε να τις βάλουμε σε συνθήκες)

π.χ. μια διατύπωση συνθήκης είναι αυτή "ΑΛΦΑ">Β\$ ΚΑΙ Κ<3*Λ η οποία επιστρέφει τιμή 0 ή -1

για ταξινόμηση μπορεί να χρησιμοποιηθεί μια "άμεση" βάση δεδομένων!

Εδώ η ιδιομορφία δεν είναι στη χρήση της βάσης αλλά στο ότι έχουμε καθορισμένο μέγεθος πεδίου.

π.χ.

βαλε "αααααα", "λλλλλλ", "ββββββ"

βάση "τεστ" 'σβήνει αν υπάρχει και δημιουργεί νέα βάση

αρχείο "τεστ","π1","ονομα",κειμενο, 20

ταξη "τεστ", "π1","ονομα", αυξουσα

για i=1 εως 3 {

προσθηκη "τεστ","π1", γραμμα\$

}

για i=1 εως 3 {

ανακτηση "τεστ","π1",i,"", ""

διαβασε ποσα, ονομα\$

τυπωσε ονομα\$

}

αφαιρεση "τεστ"

Προσθήκη 8.2 αναθεώρηση 16

Πίνακας Βάση 1, Α(10)

φτιάχνει άμεσα ένα πίνακα από 1 έως 10, και δεν αλλάζει την τρέχουσα βάση (από πιο νούμερο δηλαδή ξεκινάει ο πίνακας), μπορεί να είναι 0 ή 1.

Μετά τη λέξη βάση μπορούμε να βάλουμε το 1 ή το 0 αλλά όχι αριθμητική έκφραση π.χ. 3-2

Δες την εντολή Βάση (κάνει και άλλες δουλειές, με τις βάσεις δεδομένων)

Όταν αντιγράφουμε έναν πίνακα σε έναν άλλο, πχ Α()=Β() πρέπει να έχουμε ορίσει τον Α() ακόμα και χωρίς στοιχεία. Η βάση του Α() χάνεται και παίρνει αυτή που έχει ο Β().

Μπορούμε να βάλουμε πίνακα σε θέση πίνακα

\\ Παράδειγμα:

Πίνακας Α(3), Β(10)=3

```

A(0)=B()
A(0)(9)=100
Τύπωσε A(0)(9)/5
Για Αυτό {
    Πίνακας N()
    N(0)=A(0)
    Πίνακας N(2,5)
    A(0)=N()
}
Τύπωσε A(0)(0,4), A(0)(1,4)

```

Γέμισμα στοιχείων πίνακα με παραγωγή τιμών από Λάμδα συνάρτηση:

```

K=Λάμδα N=1 -> {=N:N++}
Πίνακας M(3,4)<<K()
Τύπωσε M()

```

Αναγνωριστικό: ΣΤΑΘΕΡΗ [CONST]
Δημιουργία μεταβλητών μόνο για ανάγνωση.

```

1)
ΣΤΑΘΕΡΗ α=10, β$="αλφα"
2)
ΣΤΑΘΕΡΕΣ α=10, β$="αλφα", γ%=12121
3)
ΓΕΝΙΚΗ ΣΤΑΘΕΡΗ α=10
4)
Μέσα σε τμήμα αν θέλουμε να πάρουμε την τιμή γενικής σταθερής σε σταθερή
ΣΤΑΘΕΡΗ Α

```

```

5)
ΓΕΝΙΚΗ ΣΤΑΘΕΡΗ Α
Μέσα σε τμήμα αν θέλουμε να πάρουμε την τιμή γενικής σταθερής ως γενική Σταθερή

```

Μια τοπική Α σκιάζει μια γενική σταθερή, και δεν είναι σταθερή, δηλαδή μπορούμε να αλλάξουμε την τιμή της.

Με την εντολή λίστα βλέπουμε τις σταθερές με τιμές σε []

Αναγνωριστικό: ΣΤΑΤΙΚΗ [STATIC]

Στατική ή Στατικές
ΣΤΑΤΙΚΕΣ A=0, B=10

Το παρακάτω παράδειγμα γράψτε το σε αρχείο με Σ "στ.gsb" ανοίγει ο διορθωτής για το αρχείο, αντιγράφουμε, πατάμε το Esc, και το φορτώνουμε και τρέχει άμεσα με το Φόρτωσε στ Με την εντολή Καθαρό καθαρίζουμε τις στατικές

```
Τμήμα άλφα {  
  Αν Ταύτιση("A") Τότε Πέτα : Καθαρό  
  \\ γενικό μόνο στο Τμήμα άλφα και σε ότι καλεί.  
  Τμήμα Γενικό M {  
    Στατικές α=0, β$="όνομα", ζ=(1,2,3,4,5)  
    ζ+=20  
    Τύπωσε ζ  
  }  
  Τμήμα Δοκιμή1 {  
    M  
    M  
    M  
  }  
  Τύπωσε "3 φορές καλούμε το M από το Δοκιμή1"  
  Δοκιμή1  
  Τύπωσε "2 φορές καλούμε το M από Τμήμα αυτό"; Τμήμα$  
  M  
  M  
  Τύπωσε "3 φορές καλούμε ξανά το M από το Δοκιμή1"  
  Δοκιμή1  
}  
Τμήμα βήτα {  
  κ=100  
  άλφα 1 ' το βήτα έχει δικό του αντικείμενο εκτέλεσης, και κρατάει δικές του στατικές για ότι  
  καλεί.  
  Τύπωσε κ  
}  
άλφα  
βήτα
```

Δεν μπορούμε να έχουμε στατική με ίδιο όνομα με τοπική στο τμήμα.

Δεν μπορούμε να έχουμε στατικές ως πίνακες με παρενθέσεις (μπορούμε να έχουμε δείκτες σε πίνακες)

Οι στατικές δεν μπορούν να περαστούν με αναφορά.

Οι στατικές σβήνουν με την Καθαρό μαζί με τις μεταβλητές του τμήματος ή της συνάρτησης που

δίνουμε την εντολή Καθαρό

Πάντα το πατρικό αντικείμενο εκτέλεσης κρατάει τις στατικές των τμημάτων που καλεί.

Αναγνωριστικό: ΣΤΗ [LET]

ΣΤΗ X=10, Y=30, K%=15, K\$="όλα καλά"

Δημιουργούμε ή και βάζουμε τιμές σε μεταβλητές. Δεν χρειάζεται εξ ορισμού αλλά. Η ΣΤΗ έχει το γνώρισμα ότι εκτελεί πρώτα το δεξιό μέρος και μετά το αριστερό έτσι στην εντολή ΣΤΗ

A(X+Y(3))=K(10) πρώτα θα εκτελεστεί το δεξιό μέρος K(10) και μετά το αριστερό X+Y(3).

Επίσης η ΣΤΗ δεν σκεπάζει ποτέ μια γενική μεταβλητή δημιουργώντας μια τοπική. Αν υπάρχει η γενική Λ τότε η ΣΤΗ Λ=5 βάζει τιμή 5 στη γενική μεταβλητή Λ. Χωρίς την ΣΤΗ πρέπει να κάνουμε αυτό Λ<=5 διαφορετικά η Λ=5 θα φτιάξει μια νέα Λ και θα προηγείται της γενικής.

Αναγνωριστικό: ΣΤΟΚ [STOCK]

Κάθε πίνακας μπορεί να χρησιμοποιηθεί ως στοκ τιμών.

1) ΣΤΟΚ πίνακ(δείκτης) απο τιμη1, τιμη2, τημη3\$

2) ΣΤΟΚ πίνακ(δείκτης) σε μεταβλ1, μεταβλ2, μεταβλήτη\$

3.1) ΣΤΟΚ πίνακ(δείκτης) για 20, πίνακ(δείκτης+100)

3.2) ΣΤΟΚ πίνακ(δείκτης) για 20, πίνακ2(δείκτης+100)

4.1) ΣΤΟΚ πίνακ(δείκτης) θεσε 100 (αδειάζει όλα τα στοιχεία)

4.2 ΣΤΟΚ πίνακ(δείκτης) θεσε 100. κλαση1() σε ΚΛΑΣΗ

Αναγνωριστικό: ΤΟΠΙΚΗ [LOCAL]

1) Τοπική Α

Ισχύει και το Τοπικές

Η εντολή αυτή δημιουργεί πάντα νέες μεταβλητές ακόμα και αν υπάρχουν τοπικές με το ίδιο όνομα

Για όσο υπάρχει η τοπική που φτιάχνουμε, άλλες τοπικές με το ίδιο όνομα δεν θα φαίνονται.

Το μπλοκ Για Αυτό { } διαγράφει ότι νέο φτιάχνουμε, οπότε οι τοπικές X και Y στο παράδειγμα θα διαγραφούν και θα αφήσουν τις παλιές για χρήση.

Παράδειγμα:

Στη X=4, Y=5

Για Αυτό {

Τοπικές X=10, Y=30

αλφα()

}

αλφα()
Τέλος
Ρουτίνα αλφα()
 Τύπωσε X, Y
Τέλος Ρουτίνας

2) Σε ορισμό Ομάδας έχει άλλη έννοια.

\ \ Στο παρακάτω παράδειγμα (έχει δυο μέρη), βλέπουμε τη λειτουργία της τοπικής (εδώ η λέξη Τοπικές έχει ίδια ισχύ. Προσοχή η λέξη Τοπικά ή Τοπικό δεν έχει ίδια ισχύ και εντός του ορισμού της ομάδας είναι λάθος το οποίο δεν εμφανίζεται άμεσα, απλά κάνει την ομάδα να μην διαβάσει όλο τον ορισμό.

Τοπική μπορεί να είναι μια συνάρτηση ή και ένα τμήμα στην Ομάδα. Και αυτά δεν μεταφέρονται σε πέρασμα με αναφορά, και δεν αλλάζουν ορισμούς σε κλειστές ομάδες (σε πίνακες δηλαδή).

Ομάδα Άλφα {
 Τοπικές X=10, Πίνακας M(3)=1
 Z=3
}
Άλφα.M(2)=1000
Άλφα.X++
Τύπωσε Άλφα.X, Άλφα.M(2) \ \ 11 1000
Βάλε &Άλφα \ \ βάζουμε στο σωρό την ισχνή αναφορά στο Άλφα
Διάβασε &Βήτα \ \ η Διάβασε συνδέει την ισχνή αναφορά στο Βήτα, αλλά παραβλέπει ότι έχει δηλωθεί με τις εντολές Τοπική/Τοπικές
Τύπωσε Έγκυρο(Βήτα.X), Έγκυρο(Βήτα.M(2)) \ \ οι τοπικές δεν περιλαμβάνονται σε πέρασμα με αναφορά
\ \ Όμως σε ισχνή αναφορά γίνεται, επειδή απευθείας χρησιμοποιούμε το αρχικό
A\$=ισχνή\$(Άλφα.X)
Τύπωσε Εκφρ(A\$.) \ \ θα μπορούσαμε εδώ να μην βάλουμε την τελεία. Όμως δείχνει καλύτερο!
\ \ με χρήση ισχνής αναφοράς διαβάζουμε τοπικές σε εσωτερικό τμήμα
Τμήμα Εσωτερικό {
 Διάβασε A\$
 Τύπωσε Μορφή\$("Εσωτερικό {0}", Έκφραση(A\$.X)) \ \ και Εκφρ() είναι το ίδιο
}
Εσωτερικό Ισχνή\$(Άλφα)
Θήτα=Άλφα \ \ Στην αντιγραφή μπαίνουν οι αρχικές τιμές στις τοπικές της ομάδας
Τύπωσε Έγκυρο(Θήτα.X), Θήτα.X, Θήτα.M(2)
\ \ Μέρος 2ο. Δείχνει πώς λειτουργή η τοπική μεταβλητή σε κλειστή ομάδα (σε πίνακα)
Πίνακας A(10)
A(0)=Άλφα \ \ στην αντιγραφή περνάει η αρχική τιμή του X το 10
Άλφα.X+=20 \ \ η X είναι όμως 11 στην Άλφα
Τύπωσε Άλφα.X \ \ 31
A(0).X+=20 \ \ γίνεται η πρόσθεση αλλά η τιμή στο επόμενο διάβασμα...

```

Τύπωσε A(0).X \ 10 ..γυρνάει στην αρχική, όπως ορίστηκε στην δήλωση της Ομάδας
Για A(0) {
    .X+=20
    Τύπωσε .X \ 30  εδώ φαίνεται ακόμα η νέα τιμή γιατί δεν έχει κλείσει η ομάδα
    \ αν είχαμε συναρτήσεις εντός της ομάδας που έβλεπαν την X τότε θα έβλεπαν και την νέα
    τιμή
    \ και αυτό δηλώνει ότι η συνάρτηση τρέχει σε ανοικτή ομάδα.
    \ ένας λόγος ύπαρξης αυτών των τοπικών είναι για να μπορούμε να καταλαβαίνουμε μέσω
    κώδικα
    \ την κατάσταση της ομάδας.
    \ παρακάτω καλούμε το τμήμα Εσωτερικό με αναφορά στο Αυτό, που εδώ είναι η ανοικτή
    ομάδα του A(0)
    Εσωτερικό &Αυτό
}
Τύπωσε A(0).X \ δίνει 10 γιατί άνοιξε ξανά, άρα πήρε την αρχική τιμή.

```

Αναγνωριστικό: ΥΠΕΡΚΛΑΣΗ [SUPERCLASS]

Μπήκε από στην έκδοση 8.9 αναθεώρηση 38.

Η Υπερκλάση είναι ομάδα και όχι συνάρτηση που φτιάχνει ομάδες. Όμως μπορεί να φτιάχνει ομάδες όπως όλες οι ομάδες που αντιγράφουν τον εαυτό τους. Εδώ αντιγράφει και έναν δείκτη στο ειδικό αντικείμενο της Υπερκλάσης. Αυτό μπορεί να έχει μοναδικές μεταβλητές, τις οποίες τις βλέπουν όλα τα αντικείμενα που φέρουν τον δείκτη στην Υπερκλάση. Για να το δουν πρέπει να ανοίξουν μια Για Υπερκλάση {}

Ως ομάδα η ΑΛΦΑ δεν έχει προσβάσιμα μέλη. Δηλαδή δεν μπορούμε να χειριστούμε τα μέλη της πχ ΑΛΦΑ.X

Μπορούμε να βάλουμε κατασκευαστή αλλά δεν παίρνει παραμέτρους. Δεν μπορούμε να βάλουμε μια υπερκλάση πάνω από μια υπερκλάση. Πάντα μια ομάδα μπορεί να έχει ή να μην έχει μια υπερκλάση, αλλά μόνο μία. Βέβαια μια ομάδα μπορεί να συσταθεί με "αθρώματα" τμηματικά. αλλά θα έχει δικά της όλα τα στοιχεία. Εδώ με την Υπερκλάση όλα τα παραγόμενα από το ΑΛΦΑ μπορούν να παράξουν ομάδες με κοινές μεταβλητές. Οι μεταβλητές της Υπερκλάσης μπορεί να είναι ότι θέλουμε αλλά μόνο οι δημόσιες και οι μοναδικές μπορούν να λογιστούν ως πεδία της κλάσης. Έξω από την κλάση δεν υπάρχει πρόσβαση ακόμα και στα δημόσια, γιατί στην ουσία τα δημόσια γυρίζουν το πεδίο του αντικειμένου και όχι της κλάσης. Πρόσβαση στα πεδία αυτά υπάρχει μόνο μέσα από τα αντικείμενα που "γνωρίζουν" την υπερκλάση τους.

Το πεδίο Μοναδικό (Unique) μπήκε για να μην δημιουργούνται πεδία/μεταβλητές σε αυτά που φτιάχνουμε από την Υπερκλάση. Έτσι αυτά τα πεδία είναι "κοινά" σε όλα τα αντικείμενα που κληρονομούν το δείκτη στην Υπερκλάση.

Μπορούμε να την κάνουμε Γενική με το Υπερκλάση Γενική Άλφα { }

```

Υπερκλάση ΑΛΦΑ {
  Μοναδικό:
    Χ, Υ
    Πίνακας Α(20)
  Δημόσιο:
    \ η Κ θα υπάρχει και ως Κ της υπερκλάσης και ως Κ του αντικειμένου
    Κ=10
    Τμήμα Εμφάνισε {
      Για Υπερκλάση, Αυτό {
        \ εμφανίζουμε το Κ της κλάσης και του αντικειμένου
        Τύπωσε .Κ, ..Κ
        \ Τα .Χ και .Υ είναι μόνο για τη κλάση
        .Χ++
        Τύπωσε .Χ, .Υ
        \ Και πίνακες μπορούμε να βάλουμε ως μοναδικά στοιχεία.
        Τύπωσε .Α()
      }
    }
}
Κ=ΑΛΦΑ
Κ.Εμφάνισε

```

```

Κλάση Δέλτα {
  Ιδιότητα ΜόνοΑνάγνωση {Αξία} = 1234321
  Τμήμα Δέλτα (Αυτό) {}
}
Μ=Δέλτα(ΑΛΦΑ)
Μ.Εμφάνισε
Τύπωσε Μ.ΜόνοΑνάγνωση
\ εδώ ο Α() έχει τα τμήματα κοινά για όλα τα στοιχεία.
\ και επιπλέον όλα τα στοιχεία έχουν σύνδεση με την κλάση του ΑΛΦΑ
\ ακόμα και να "χαθεί" το ΑΛΦΑ, η σύνδεση διατηρεί το αντικείμενο με τα στοιχεία.
Πίνακας Α(100)=Δέλτα(Αλφα)
Α(4).Εμφάνισε

```

Εδώ είναι το μεγαλύτερο παράδειγμα, με διπλή υπερκλάση, σε ένα αντικείμενο με δυο ομάδες. Κάθε ομάδα μπορεί να πάρει μια υπερκλάση, αλλά κάθε ομάδα μπορεί να έχει οποιοδήποτε αριθμό εμφωλευμένων ομάδων.

```

ΥπερΚλάση Κάπα1 {
  χ=10

```

```

Τμήμα Εμφάνισε {
    Για ΥπερΚλάση {
        .χ++
        Τύπωσε .χ, Αυτό.χ
    }
}
}
ΥπερΚλάση Κάπα2 {
    χ=50
    Τμήμα Εμφάνισε {
        Για ΥπερΚλάση {
            .χ++
            Τύπωσε .χ, Αυτό.χ
        }
    }
}
}
Ομάδα Άλφα {
    Ομάδα Βήτα {
        \ \ δεν θα βάλουμε εδώ ορισμούς γιατί έχει η ΥπερΚλάση
        \ \ αλλά θα μπορούσαμε να ξαναπαίξουμε τον ορισμό και να αλλάξουμε
        \ \ ακόμα και ένα τμήμα που ορίστηκε στην υπερκλάση για τη συγκεκριμένη ομάδα.
    }
}
\ \ Πρώτα θα συγχωνεύσουμε τη ΥπερΚλάση (θα φτιάξει νέα μέλη και θα δώσει και ένα δείκτη)
Άλφα.Βήτα=Κάπα2
Άλφα=Κάπα1
\ \ Τώρα έχουμε δυο ΥπερΚλάσεις, μια για την εξωτερική Ομάδα, και μια για την εσωτερική
Ομάδα
Άλφα.Εμφάνισε          \ \ 11 10
Άλφα.Βήτα.Εμφάνισε    \ \ 51 50

Ζ=Άλφα

Ζ.Εμφάνισε              \ \ 12 10
Ζ.Βήτα.Εμφάνισε        \ \ 52 50
Άλφα.Εμφάνισε          \ \ 13 10
Άλφα.Βήτα.Εμφάνισε    \ \ 53 50

Πίνακας Α(10)
Α(3)=Ζ
Α(3).Εμφάνισε          \ \ 14 10
Α(3).Βήτα.Εμφάνισε    \ \ 54 50

```

Κατάσταση Κατ1= 100:=Z

\\ τα κλειδιά είναι αλφαριθμητικά, απλά φαίνονται και ως αριθμοί!

Κατ1("100").Εμφάνισε \\ 15 10

Κατ1(100).Βήτα.Εμφάνισε \\ 55 50

Ομάδα: ΕΓΓΡΑΦΑ - DOCUMENTS

Αναγνωριστικό: ΔΙΟΡΘΩΣΕ [EDIT.DOC]

Διόρθωσε [κώδικα] Έγγραφο\$, [θέση δρομέα],[τίτλος εγγράφου], [χρώμα φόντου]

[] σημαίνει προαιρετικό

Παράμετροι

1. το έγγραφο πρέπει να είναι μεταβλητή τύπου [έγγραφο] ή στοιχείο πίνακα αλφαριθμητικού που έχει δηλωθεί ως έγγραφο.
2. Αν θέλουμε δίνουμε δρομέα

αν βάλουμε χρώμα φόντου αυτό της πέννας τότε έχουμε διαφάνεια 100%

Αναγνωριστικό: ΕΥΡΕΣΗ [FIND]

ΕΥΡΕΣΗ μεταβλητή_τύπου_εγγράφου\$, αρχική θέση

Έγγραφο Αλφα\$={ΑΑΑΓΓΓ

BBB

ΓΓΓ

ΔΔΔ

}

N=1

Εύρεση Αλφα\$, "ΓΓΓ", N

Διάβασε N

Ενώ N {

 Διάβασε παράγραφος, θέση_στη_παράγραφο

 Τύπωσε μορφή\$("Βρέθηκε στη θέση {0}, στη παράγραφο {1}", θέση_στη_παράγραφο, παράγραφος)

 Εύρεση Αλφα\$, "ΓΓΓ", N+1

 Διάβασε N

}

Αναγνωριστικό: ΚΑΤΑΧΩΡΗΣΗ [OVERWRITE]

ΚΑΤΑΧΩΡΗΣΗ

1)\ για έγγραφο και για αλφαριθμητικό

\ δουλεύει ανάποδα από την Μέρος\$(), εδώ καταχωρούμε σε θέση μεταξύ των σημαδιών, εδώ το "."

α\$="ααα.βββ"

Καταχώρηση α\$, ".", 1:="Γεια",5:="ένα ακόμα"

Τύπωσε α\$

Τύπωσε Τύπος\$(α\$)

Πίνακας κ\$(5)

κ\$(3)="ααα.βββ"

Καταχώρηση κ\$(3), ".", 1:="Γεια",5:="ένα ακόμα"

Τύπωσε κ\$(3)

Τύπωσε Τύπος\$(κ\$(3))

Καθαρό α\$

Έγγραφο α\$="ααα.βββ"

Καταχώρηση α\$, ".", 1:="Γεια",5:="ένα ακόμα"

Τύπωσε α\$

Τύπωσε Τύπος\$(α\$)

Πίνακας κ\$(5)

Καθαρό κ\$(3)

Έγγραφο κ\$(3)="ααα.βββ"

Καταχώρηση κ\$(3), ".", 1:="Γεια",5:="ένα ακόμα"

Τύπωσε κ\$(3)

Τύπωσε Τύπος\$(κ\$(3))

1.1) Χρήση με μεταβλητές τύπου γράμμα (String), απλές, πίνακες πολλαπλών τύπων και πίνακες αλφαριθμητικών. Οι πίνακες αλφαριθμητικών (Πίνακας κ(10) ως γράμμα) έχουν για κάθε στοιχείο μόνο 4 Bytes. Ενώ αυτοί τύπου Variant (άτυπος, παίρνει οποιοδήποτε τύπο), έχουν για κάθε στοιχείο 12Bytes (και παίρνουν διάφορους αριθμητικούς τύπους, καθώς και τύπο γράμμα και αντικείμενα)

γράμμα α="ααα.βββ"

Καταχώρηση &α, ".", 1:="Γεια",5:="ένα ακόμα"

Τύπωσε α="ααα.Γεια....ένα ακόμα"

Τύπωσε Τύπος\$(α)="String"

Πίνακας κ(5) ως γράμμα

κ(3)="ααα.βββ"

Καταχώρηση &κ(3), ".", 1:="Γεια",5:="ένα ακόμα"

Τύπωσε κ(3)="ααα.Γεια....ένα ακόμα"

Τύπωσε Τύπος\$(κ(3))="String", Τύπος\$(κ(2))="String"

Πίνακας $\mu(5)$
 $\mu(3) = \text{"ααα.βββ"}$
 $\mu(2) = 102020$
Καταχώρηση & $\mu(3)$, ".", 1:="Γεια", 5:="ένα ακόμα"
Τύπωσε $\mu(3) = \text{"ααα.Γεια....ένα ακόμα"}$
Τύπωσε Τύπος\$($\mu(3)$)="String", Τύπος\$($\mu(2)$)="Double"

2)\ για έγγραφο
έγγραφο α\$, β\$
α\$={aaaaaaaaaaaa
12345678901234567890
}
β\$=α\$
\ ανοίγει χώρο και γράφει
παρεμβολή στο 2, 10 α\$="γεια"
Αναφορά β\$
Αναφορά α\$
Καθαρό α\$
Έγγραφο α\$=β\$
\ γράφει πάνω στο κείμενο
παρεμβολή στο 2, 10 α\$="γεια"
Αναφορά β\$
Αναφορά α\$
Καθαρό α\$
α\$=β\$
καταχώρηση 2,10 α\$="γεια"
Αναφορά β\$
Αναφορά α\$
Καθαρό α\$
α\$=β\$
καταχώρηση 2 α\$="γεια" // αλλάζει όλη τη παράγραφο
Αναφορά β\$
Αναφορά α\$

Αναγνωριστικό: ΛΕΞΕΙΣ [WORDS]

ΛΕΞΕΙΣ έγγραφο\$

έχει δυο παραλλαγές ανάλογα με το ποια συνάρτηση έχουμε χρησιμοποιήσει πριν!
την Εγγράφου.Λέξεις() ή την Εγγράφου.Μοναδικές.Λέξεις()

Σε κάθε περίπτωση έχουμε ταξινομημένη λίστα. Στην δεύτερη οι όμοιες λέξεις είναι
ομαδοποιημένες και σε παρένθεση φαίνεται ο αριθμός τους στο κείμενο.

```

Έγγραφο άλφα$={ ένα, δυο, τρία
    και περισσότερες, όπως αυτή και η άλλη, ή ένα και δυο
}
Τύπωσε Εγγράφου.Λέξεις(άλφα$)
μ=Μέγεθος.Σωρού+1
Λέξεις άλφα$
Αν Μέγεθος.Σωρού >= μ Τότε {
    Τύπωσε Μέγεθος.Σωρού - μ
    Για ι=Μέγεθος.Σωρού έως μ {
        Τύπωσε Γράμμα$
    }
}
}
Τύπωσε Εγγράφου.Μοναδικές.Λέξεις(άλφα$)
μ=Μέγεθος.Σωρού+1
Λέξεις άλφα$
Αν Μέγεθος.Σωρού >= μ Τότε {
    Για ι=Μέγεθος.Σωρού έως μ {
        Τύπωσε Γράμμα$
    }
}
}

```

Αναγνωριστικό: ΠΑΡΕΜΒΟΛΗ [INSERT]

ΠΑΡΕΜΒΟΛΗ

\\ για αλφαριθμητικά

α\$="12345678901234567890"

β\$=α\$

\\ ανοίγει χώρο και γράφει

παρεμβολή 10 α\$="γεια"

Τύπωσε β\$

Τύπωσε α\$

α\$="12345678901234567890"

\\ γράφει πάνω στο κείμενο

παρεμβολή 10,2 α\$="γεια"

Τύπωσε β\$

Τύπωσε α\$

// ΔΟΥΛΕΥΕΙ για αλφαριθμητικά χωρίς \$ στο τέλος:

\\ για αλφαριθμητικά

α="12345678901234567890"

β=α

```
\ \ ανοίγει χώρο και γράφει  
παρεμβολή 10 α="γεια"  
Τύπωσε β  
Τύπωσε α  
α="12345678901234567890"  
\ \ γράφει πάνω στο κείμενο  
παρεμβολή 10,2 α="γεια"  
Τύπωσε β  
Τύπωσε α
```

```
// Και για πίνακες  
Πίνακας α(10) ως γράμμα // και χωρίς να ορίσουμε ότι όλος ο πίνακας θα είναι  
αλφαριθμητικών.  
α(3)="12345678901234567890"  
β=α(3)  
\ \ ανοίγει χώρο και γράφει  
παρεμβολή 10 α(3)="γεια"  
Τύπωσε β  
Τύπωσε α(3)  
α(1)="12345678901234567890"  
\ \ γράφει πάνω στο κείμενο  
παρεμβολή 10,2 α(1)="γεια"  
Τύπωσε β  
Τύπωσε α(1)
```

// και για πίνακες αλφαριθμητικών (αλλά στην ουσία είναι όπως ο α() με τη διαφορά ότι όταν του δίνουμε τιμές τις κάνει αλφαριθμητικά). Επίσης αυτός ο πίνακας μπορεί να "σηκώσει" και έγγραφο (που είναι αντικείμενο ως μια λίστα παραγράφων), και πρέπει να το ορίσουμε χωριστά για κάθε στοιχείο του.

```
Πίνακας α$(10)  
α$(3)="12345678901234567890"  
β=α$(3)  
\ \ ανοίγει χώρο και γράφει  
παρεμβολή 10 α$(3)="γεια"  
Τύπωσε β  
Τύπωσε α$(3)  
α$(1)="12345678901234567890"  
\ \ γράφει πάνω στο κείμενο  
παρεμβολή 10,2 α$(1)="γεια"  
Τύπωσε β  
Τύπωσε α$(1)
```

```

\\Με έγγραφο σε στοιχεία πίνακα.
\\ για έγγραφο
Πίνακες α$(10), β$(10)
έγγραφο α$(3), β$(2)
α$(3)={aaaaaaaaaaaa
12345678901234567890
}
β$(2)=α$(3)
\\ ανοίγει χώρο και γράφει, το πρώτο νούμερο είναι θ θέση χαρακτήρα (όχι η γραμμή)
παρεμβολή στο 2, 10 α$(3)="γεια"
Αναφορά β$(2)
Αναφορά α$(3)
Καθαρό α$(3)
Έγγραφο α$(3)=β$(2)
\\ γράφει στο κείμενο στη 2η γραμμή (υπάρχει και δυνατότητα τρίτης παραμέτρου για να
ορίσουμε τι θα διαγραφεί πρώτα)
παρεμβολή στο 2, 10 α$(3)="γεια"
Αναφορά β$(2)
Αναφορά α$(3)
Καθαρό α$(3)
α$(3)=β$(2)
καταχώρηση 2,10 α$(3)="γεια"
Αναφορά β$(2)
Αναφορά α$(3)
Καθαρό α$(3)
α$(3)=β$(2)
καταχώρηση 2 α$(3)="γεια" \\ αλλάζει την παράγραφο με την εισαγωγή του "γεια"
Αναφορά β$(2)
Αναφορά α$(3)

```

```

\\ Το ίδιο αλλά με απλά μεταβλητή για έγγραφο (πρέπει να έχει το $)
έγγραφο α$, β$
α$={aaaaaaaaaaaa
12345678901234567890
}
β$=α$
\\ ανοίγει χώρο και γράφει, το πρώτο νούμερο είναι θ θέση χαρακτήρα (όχι η γραμμή)
παρεμβολή στο 2, 10 α$="γεια"
Αναφορά β$

```

Αναφορά α\$
Καθαρό α\$
Έγγραφο α\$=β\$
\\ γράφει στο κείμενο στη 2η γραμμή (υπάρχει και δυνατότητα τρίτης παραμέτρου για να ορίσουμε τι θα διαγραφεί πρώτα)
παρεμβολή στο 2, 10 α\$="γεια"
Αναφορά β\$
Αναφορά α\$
Καθαρό α\$
α\$=β\$
καταχώρηση 2,10 α\$="γεια"
Αναφορά β\$
Αναφορά α\$
Καθαρό α\$
α\$=β\$
καταχώρηση 2 α\$="γεια" \\ αλλάζει την παράγραφο με την εισαγωγή του "γεια"
Αναφορά β\$
Αναφορά α\$

Αναγνωριστικό: ΠΡΟΣΘΕΣΕ.ΕΓΓΡΑΦΟ [APPEND.DOC]
ΠΡΟΣΘΕΣΕ.ΕΓΓΡΑΦΟ ονομα\$, τυπος_αρχείου
τύπος αρχείου:
0 για UTF-16LE
1 για UTF-16BE
2 UTF-8
3 ANSI

εξ ορισμού είναι Utf-8
\\ εδω αν υπάρχει αρχείο name1.txt θα "πατηθεί" από νέο αρχείο
\\ με ίδιο όνομα αλλά με κενό περιεχόμενο
Έγγραφο αλφα\$={Μια μόνο παράγραφος
}
Σώσε.Έγγραφο αλφα\$,"name1.txt"
\\ έστω μια άλλη φορά ή σε άλλο πρόγραμμα:
\\ θέλουμε να προσθέσουμε
Έγγραφο βήτα\$={Μια άλλη παράγραφος
}
Αν Υπάρχει("name1.txt") τότε {
Πρόσθεσε.Έγγραφο βήτα\$,"name1.txt"
} αλλιώς {
Σώσε.Έγγραφο βήτα\$,"name1.txt"

```
}  
\\ τώρα διαβάζουμε όλο μαζί  
Έγγραφο καππα$  
Φόρτωσε.Έγγραφο καππα$, "name1.txt"  
Αναφορά καππα$
```

Αναγνωριστικό: ΣΥΓΧΩΝΕΥΣΕ.ΕΓΓΡΑΦΟ [MERGE.DOC]
ΣΥΓΧΩΝΕΥΣΕ.ΕΓΓΡΑΦΟ έγγραφο\$

Συγχωνεύει ένα κείμενο από αρχείο σε ένα έγγραφο στην μνήμη.

```
\\ γράψε κάτι και πάτα esc  
Συγγραφή "alfa1.txt"  
\\ γράψε κάτι και πάτα esc  
Συγγραφή "alfa2.txt"  
Έγγραφο α$  
Φόρτωσε.Έγγραφο α$, "alfa1.txt"  
Συγχώνευσε.Έγγραφο α$, "alfa2.txt"  
Αναφορά α$
```

Αναγνωριστικό: ΣΩΣΕ.ΕΓΓΡΑΦΟ [SAVE.DOC]
ΣΩΣΕ.ΕΓΓΡΑΦΟ έγγραφο\$, όνομα_αρχείου\$, τυπος_αρχείου
// μετατροπή στο τρόπο που έχει φορτωθεί (encoding + newline type)
ΣΩΣΕ.ΕΓΓΡΑΦΟ έγγραφο\$ ως μεταβλητη\$
// μετατροπή με το τρόπο που θέλουμε
ΣΩΣΕ.ΕΓΓΡΑΦΟ έγγραφο\$, τύπος_αρχείου ως μεταβλητη\$

τύπος αρχείου:

```
0 UTF-16LE -4 UTF-16LE No BOM 10 lf -10 lf NoBom, 20 cr -20 cr NoBom  
1 UTF-16BE -1 NoBom 11 lf -11 lf NoBom 21 cr -21 cr NoBom  
2 UTF-8 -2 NoBom 12 lf -11 lf NoBom 22 cr -22 cr NoBom  
3 ANSI  
1032 - ANSI Greek Locale  
Με έγγραφο$, "ListLoadedType", 10 ' lf θέτουμε νέα τιμή  
ή  
Με έγγραφο$, "ListLoadedType" ως ΤύποςΦόρτωσης
```

Τύπωσε ΤύποςΦόρτωσης ' διαβάζουμε τον τύπο
ΤύποςΦόρτωσης=10 ' αλλάζουμε το τύπο

α=-12

Τύπωσε " Κωδικοποίηση "+αν\$(απολ(α υπολ 10) υπολ 4+1->"UTF-16LE", "UTF-16BE",
"UTF-8", "ANSI");

Τύπωσε " | Αλλαγή γραμμής:"+αν\$(απολ(α δια 10) +1->"crlf", "lf", "cr");

Τύπωσε " | BOM: "+αν\$(α<0->"No ", "")

Το BOM είναι δυο χαρακτήρες στην αρχή του κειμένου που ορίζουν το τύπο (μόνο για UTF-16LE, UTF-16BE, UTF-8).

Στο διάβασμα αρχείου η Φόρτωση Έγγραφο μπορεί να βρει αυτόματα:

1. το τύπο αν δεν υπάρχει BOM (εξετάζει μέρος του κειμένου), ξεχωρίζει και το ANSI αλλά δεν γνωρίζει τη κωδικοσελίδα, χρησιμοποιεί ότι έχει οριστεί στο Τοπικό (Locale)
2. Το τρόπο αλλαγής γραμμής.

εξ ορισμού ο τύπος είναι Utf-8 με BOM και crlf (Windows)

\\ εδώ αν υπάρχει αρχείο name1.txt θα "πατηθεί" από νέο αρχείο

\\ με ίδιο όνομα αλλά με κενό περιεχόμενο

Έγγραφο αλφα\$={Μια μόνο παράγραφος

}

Σώσε.Έγγραφο αλφα\$,"name1.txt"

\\ έστω μια άλλη φορά ή σε άλλο πρόγραμμα:

\\ θέλουμε να προσθέσουμε

Έγγραφο βήτα\$={Μια άλλη παράγραφος

}

Αν Υπάρχει("name1.txt") τότε {

Πρόσθεσε.Έγγραφο βήτα\$,"name1.txt"

} αλλιώς {

Σώσε.Έγγραφο βήτα\$,"name1.txt"

}

\\ τώρα διαβάζουμε όλο μαζί

Έγγραφο καππα\$

Φόρτωσε.Έγγραφο καππα\$, "name1.txt"

Αναφορά καππα\$

Αναγνωριστικό: ΤΑΞΙΝΟΜΗΣΗ(ΕΓΓΡΑΦΟΥ) [SORT(DOCUMENT)]

Δες ΤΑΞΙΝΟΜΗΣΗ

Αναγνωριστικό: ΦΟΡΤΩΣΕ.ΕΓΓΡΑΦΟ [LOAD.DOC]
ΦΟΡΤΩΣΕ.ΕΓΓΡΑΦΟ έγγραφο\$, όνομα_αρχείου\$ [, LocaleID]

```
ονομα$="Γιώργος"  
Κείμενο utf-8 alfa.txt { Γραμμές Κειμένου  
    Εδώ βάζουμε μια τιμή "##ονομα$##"  
    Άλλες γραμμές  
}
```

Έγγραφο α\$
Φόρτωσε.Εγγραφο α\$, Προσωρινό\$+"alfa.txt"
Αναφορά α\$

Ομάδα: ΧΕΙΡΙΣΜΟΣ ΑΡΧΕΙΩΝ - FILE OPERATIONS

Αναγνωριστικό: ΑΝΟΙΞΕ [OPEN]
ΕΝΑ BYTE ανά χαρακτήρα (χρησιμοποιεί την κωδικοσελίδα του συστήματος)
Η εντολή μπορεί να ορίσει μεταβλητή χειρισμού αν δεν έχει οριστεί.
Το ΑΠΟΚΛΕΙΣΤΙΚΑ είναι προαιρετικό και σημαίνει για αποκλειστική χρήση
Το # είναι προαιρετικό
ΑΝΟΙΞΕ ονομα_αρχείου\$ ΓΙΑ ΕΙΣΑΓΩΓΗ ΑΠΟΚΛΕΙΣΤΙΚΑ ΩΣ #μεταβλητη_χειρισμου_αρχείου
ΑΝΟΙΞΕ ονομα_αρχείου\$ ΓΙΑ ΕΞΑΓΩΓΗ ΑΠΟΚΛΕΙΣΤΙΚΑ ΩΣ #μεταβλητη_χειρισμου_αρχείου
\\ αν στην ΕΞΑΓΩΓΗ δώσουμε όνομα αρχείου το κενό αλφαριθμητικό τότε η Τύπωσε # θα δίνει
στην οθόνη
\\ αυτό το θέλουμε για να δούμε σε δοκιμές τι δίνουμε!
ΑΝΟΙΞΕ ονομα_αρχείου\$ ΓΙΑ ΣΥΜΠΛΗΡΩΣΗ ΑΠΟΚΛΕΙΣΤΙΚΑ ΩΣ
#μεταβλητη_χειρισμου_αρχείου
ΑΝΟΙΞΕ ονομα_αρχείου\$ ΓΙΑ ΠΕΔΙΑ ΑΠΟΚΛΕΙΣΤΙΚΑ ΩΣ #μεταβλητη_χειρισμου_αρχείου
ΜΗΚΟΣ=συνολικο_μηκος_γραμμης_πεδίων_σε_χαρακτήρες
Μέγιστο μήκος για πεδία 32766

ΔΥΟ BYTE ανά χαρακτήρα (Unicode WIDE 16bit)
Η εντολή μπορεί να ορίσει μεταβλητή χειρισμού αν δεν έχει οριστεί.
Το ΑΠΟΚΛΕΙΣΤΙΚΑ είναι προαιρετικό και σημαίνει για αποκλειστική χρήση
Το # είναι προαιρετικό
ΑΝΟΙΞΕ ονομα_αρχείου\$ ΓΙΑ ΕΥΡΙΑ ΕΙΣΑΓΩΓΗ ΑΠΟΚΛΕΙΣΤΙΚΑ ΩΣ
#μεταβλητη_χειρισμου_αρχείου
ΑΝΟΙΞΕ ονομα_αρχείου\$ ΓΙΑ ΕΥΡΙΑ ΕΞΑΓΩΓΗ ΑΠΟΚΛΕΙΣΤΙΚΑ ΩΣ
#μεταβλητη_χειρισμου_αρχείου
\\ αν στην ΕΞΑΓΩΓΗ δώσουμε όνομα αρχείου το κενό αλφαριθμητικό τότε η Τύπωσε # θα δίνει

στην οθόνη

\\ αυτό το θέλουμε για να δούμε σε δοκιμές τι δίνουμε!

ΑΝΟΙΞΕ ονομα_αρχείου\$ ΓΙΑ ΕΥΡΙΑ ΣΥΜΠΛΗΡΩΣΗ ΑΠΟΚΛΕΙΣΤΙΚΑ ΩΣ

#μεταβλητη_χειρισμου_αρχείου

ΑΝΟΙΞΕ ονομα_αρχείου\$ ΓΙΑ ΕΥΡΙΑ ΠΕΔΙΑ ΑΠΟΚΛΕΙΣΤΙΚΑ ΩΣ

#μεταβλητη_χειρισμου_αρχείου ΜΗΚΟΣ=συνολικο_μηκος_γραμμης_πεδιων_σε_χαρακτήρες

Μέγιστο μήκος για πεδία 32766/2 (σε χαρακτήρες είναι οι μισοί από τους 32766)

Συναρτήσεις

ΤΕΛΟΣ() ' δίνει αληθές αν φθάσαμε στο τέλος του

ΕΓΓΡΑΦΕΣ() ' δίνει το μήκος σε χαρακτήρες ή τον αριθμό πεδίων αν ανοίξουμε ΓΙΑ ΠΕΔΙΑ ένα αρχείο

Εντολέ για ΠΕΔΙΑ μόνο

ΠΑΡΕ 'διαβάζουμε μια εγγραφή από μια θέση

ΔΩΣΕ ' γράφουμε μια εγγραφή σε μια θέση (στη θέση Εγγραφες()+1 προσθέτουμε εγγραφή

Μια εγγραφή είναι ένα αλφαριθμητικό το οποίο δεν το έχουμε "τεμαχισμένο" με σύμβολα αλλά μπορούμε να το χωρίσουμε σε πεδία

απλά με την ΜΕΣ\$(και για να τη δώσουμε πίσω να την συννενώσουμε με το +

(Πεδιο1\$+Πεδιο2\$) αλλά πρέπει να έχουμε συνολικό μήκος το μήκος της εγγραφής. Δες εντολές

ΠΕΔΙΟ και ΠΕΔΙΟ()

Εντολές μετακίνησης (Εκτός τύπου ΠΕΔΙΩΝ)

ΕΔΩ=ΜΕΤΑΘΕΣΗ(#Α) ' Διαβάζουμε πριν γράψουμε το σημείο που θα γράψουμε

ΜΕΤΑΘΕΣΗ #Α, ΕΔΩ ' Μεταθέτουμε τον δρομέα του αρχείου στο σημείο που θέλουμε να διαβάσουμε

Στα αρχεία τύπου ΕΙΣΑΓΩΓΗ, ΕΞΑΓΩΓΗ και ΣΥΜΠΛΗΡΩΣΗ (όπως η ΕΙΣΑΓΩΓΗ αλλά δεν σβήνει το αρχείο και μεταθέτει το δρομέα στο τέλος), έχουμε τους παρακάτω τρόπους να διαβάζουμε και να γράφουμε

Διαβάζουμε με τα ΕΙΣΑΓΩΓΗ, ΓΡΑΜΜΗ ΕΙΣΑΓΩΓΗΣ και ΕΙΣΑΓΩΓΗ\$()

Γράφουμε με την ΤΥΠΩΣΕ και την ΓΡΑΨΕ

οτιδηποτε\$=ΕΙΣΑΓΩΓΗ\$(#Α, 20) ' Διαβάζουμε 20 χαρακτήρες ANSI ή UNICODE ανάλογα με το δηλώσαμε στην ΑΝΟΙΞΕ την ΕΥΡΙΑ μορφή χαρακτήρων (2 Bytes για κάθε γράμμα).

ΤΥΠΩΣΕ #Α, γρ\$ ' τυπώνει μια γραμμή κειμένου στο αρχείο. Με σύμβολο ; τυπώνουμε χωρίς την αλλαγή γραμμής.

' Ουσιαστικά φτιάχνουμε γραμμές που θα διαβαστούν αργότερα με την ΓΡΑΜΜΗ ΕΙΣΑΓΩΓΗΣ ή την ΕΙΣΑΓΩΓΗ\$()

ΓΡΑΜΗ ΕΙΣΑΓΩΓΗΣ #Α, γρ\$ ' διαβάζει μια γραμμή από αρχείο κειμένου μέχρι να βρει χαρακτήρα που να δηλώνει αλλαγή γραμμής, και αφήνει τον δρομέα στην αρχή της επόμενης γραμμής.

ΓΡΑΨΕ #Α, α\$, Β ' Βάζει γραμμές με δεδομένα έτσι: "λέξη", 1213 μπορεί να γράψει μέσα στο

αλφαριθμητικό και εισαγωγικά και αλλαγές γραμμών. Όμως ο κατάλληλος τρόπος να διαβαστούν τα δεδομένα είναι η ΕΙΣΑΓΩΓΗ
ΕΙΣΑΓΩΓΗ #A, A\$, B ' Εισάγει στις μεταβλητές δεδομένα που έχουμε γράψει πρώτα με τη ΓΡΑΨΕ

Μπορούμε να χρησιμοποιούμε ότι θέλουμε αρκεί να γνωρίζουμε τι κάνουμε!

Γράψε σε ένα ΤΜΗΜΑ το παρακάτω...δοκίμασέ το χωρίς το ΕΥΡΙΑ (στα δυο ΑΝΟΙΞΕ)

```
A$={1η γραμμή
    "περιεχόμενο σε εισαγωγικά" και "εδώ"
    τρίτη γραμμή
} ' το περιθώριο αριστερά δεν θα μπει στην A$
K=0
ΑΝΟΙΞΕ ΚΑΤ$+"αυτο.ααα" ΓΙΑ ΕΥΡΙΑ ΕΞΑΓΩΓΗ ΩΣ K
ΓΡΑΨΕ #K, 100,1,2,A$
ΓΡΑΨΕ #K, 20
ΓΙΑ I=1 ΕΩΣ 20 {
    ΓΡΑΨΕ #K, 2*I+1
}
ΚΛΕΙΣΕ #K
ΑΝΟΙΞΕ ΚΑΤ$+"αυτο.ααα" ΓΙΑ ΕΥΡΙΑ ΕΙΣΑΓΩΓΗ ΩΣ K
ΕΙΣΑΓΩΓΗ #K, A,B,M, A$
ΤΥΠΩΣΕ A, B, M
ΑΝΑΦΟΡΑ A$
ΕΙΣΑΓΩΓΗ #K, ΤΟΣΑ
ΠΙΝΑΚΑΣ Λ(ΤΟΣΑ)
ΓΙΑ I=0 ΕΩΣ ΤΟΣΑ-1 {
    ΕΙΣΑΓΩΓΗ #K, Λ(I)
    ΤΥΠΩΣΕ Λ(I)
}
ΚΛΕΙΣΕ #K
```

Επίσης φτιάξε ένα ακόμα τμήμα (πρόσεξε και εδώ τη χρήση της μορφής ΕΥΡΙΑ να είναι ίδια και στο ΑΝΟΙΞΕ που έφτιαξε το αρχείο

```
K=0
ΑΝΟΙΞΕ ΚΑΤ$+"αυτο.ααα" ΓΙΑ ΕΥΡΙΑ ΕΙΣΑΓΩΓΗ ΩΣ K
ΕΝΩ ΟΧΙ ΤΕΛΟΣ(#K) {
    ΓΡΑΜΜΗ ΕΙΣΑΓΩΓΗΣ #K, A$
    ΤΥΠΩΣΕ A$
}
ΚΛΕΙΣΕ #K
```

τώρα διαβάζουμε το αρχείο, γραμμή προς γραμμή χωρίς να τοποθετούμε τα δεδομένα σε ειδικές μεταβλητές.

Αναγνωριστικό: ΑΡΧΕΙΑ [FILES]

ΑΡΧΕΙΑ (δίνει όλα τα txt αρχεία)

ΑΡΧΕΙΑ "mdb" (δίνει όλα τα αρχεία με επέκταση mdb)

ΑΡΧΕΙΑ + "jpg" \ Με τις δυο αυτές εντολές εμφανίζουμε λίστα με όλα τα αρχεία jpg και
ΕΠΙΛΟΓΗ ! / επιλέγουμε με το ποντίκι ή τα βελάκια και το enter

ΑΡΧΕΙΑ "gsb", "ΤΥΠΩΣΕ " \βρίσκει όλα τα αρχεία GSB που περιέχουν την ΤΥΠΩΣΕ

ΑΡΧΕΙΑ "gsb", "ΤΥΠΩΣΕ | ΠΙΝΑΚΑΣ" \βρίσκει όλα τα αρχεία GSB που περιέχουν την ΤΥΠΩΣΕ
και την ΠΙΝΑΚΑΣ

ΑΡΧΕΙΑ !! "*" ' δίνει όλα τα αρχεία ταξινομημένα κατά τύπο.

Αναγνωριστικό: ΓΡΑΜΜΗ ΕΙΣΑΓΩΓΗΣ [LINE INPUT]

ΓΡΑΜΜΗ ΕΙΣΑΓΩΓΗΣ #μεταβλητή_χειρισμού_αρχείου, αλφαριθμητική_μεταβλητή\$

ΓΡΑΜΜΗ ΕΙΣΑΓΩΓΗΣ #μεταβλητή_χειρισμού_αρχείου, αλφαριθμητική_μεταβλητή\$(δείκτης)

Θα φορτώσει χαρακτήρες μέχρι να βρει τους χαρακτήρες αλλαγής γραμμής 13 ή 13+10

το

νεαγραμμή\$={

}

βάζει κατευθείαν στο νεαγραμμή\$ μια αλλαγή γραμμής και μπορούμε αν θέλουμε να
ξαναφτιάξουμε το αρχείο, από όπου με την ΓΡΑΜΜΗ ΕΙΣΑΓΩΓΗΣ διαβάζουμε μια προς μια τις
γραμμές του και αν προσθέσουμε το νεαγραμμή\$ τότε μπορούμε να περάσουμε όποια γραμμή
θέλουμε σε ένα άλλο αρχείο.

Για να μην βγει λάθος πρέπει κάθε φορά πριν διαβάσουμε με την ΓΡΑΜΜΗ ΕΙΣΑΓΩΓΗΣ να
βλέπουμε αν υπάρχει κάτι για να εισάγουμε, με την ΤΕΛΟΣ(μεταβλητή_χειρισμού)

Αλλαγή γραμμής βρίσκει αν δει ένα μόνο CR (13) ή LF (10) ή CRLF ή LF CR

ΔΕΣ ΤΗΝ ΕΙΣΑΓΩΓΗ ΚΑΙ ΤΟ ΕΙΣΑΓΩΓΗ\$()

Αναγνωριστικό: ΓΡΑΨΕ [WRITE]

Αν χρησιμοποιήσουμε όμοια αρχείου στην Άνοιξε το "" για εξαγωγή τότε η έξοδος γίνεται στην

οθόνη.

Μπορούμε να ορίσουμε το Tab ως χαρακτήρα διαχωρισμού πεδίων και το κόμμα για διαχωρισμό δεκαδικών στα ελληνικά.

ΓΡΑΨΕ ΜΕ ΧΑΡ\$(9), ","

με ΓΡΑΨΕ ΜΕ ",";" επαναφέρουμε τη ΓΡΑΨΕ στο κανονικό.

Υπάρχει και τρίτη παράμετρος στην Γράψε Με προαιρετικά. Με Αληθής τα αλφαριθμητικά μετατρέπονται όπου χρειάζεται σε ακολουθίες escape (όπως κάνει το Επαν\$(αλφαριθμητικό))

ΓΡΑΨΕ #μεταβλητη_χειρισμου_αρχιου, μεταβλητη, μεταβλητη\$,

Γράφει σε ένα αρχείο (ΓΙΑ ΕΞΑΓΩΓΗ ή ΓΙΑ ΣΥΜΠΛΗΡΩΣΗ) σε μια γραμμή με διαχωριστικά αριθμούς ή αλφαριθμητικά ή και τα δύο, έτσι ώστε να διαβαστούν με την εντολή ΕΙΣΑΓΩΓΗ. Επίσης σε ένα αρχείο που το έχουμε συμπληρώσει μόνο με την ΓΡΑΨΕ, αν έχουμε δώσει κατάληξη CSV τότε μπορούμε να το ανοίξουμε με το Microsoft Excel.

Μπορούμε χωρίς πρόβλημα στην ανάγνωση (με INPUT #) να γράφουμε αλφαριθμητικά με πολλές γραμμές

Αναγνωριστικό: ΔΩΣΕ [PUT]

Η εντολή αυτή δουλεύει για τα αρχεία τύπου πεδίου ως εξής:

ΔΩΣΕ μεταβλητη_χειρισμου_αρχιου, μεταβλητη_πεδιο\$, θεση_στο_αρχαιο

Για να προσθέσουμε (ΣΥΜΠΛΗΡΩΣΗ) ένα αρχείο τύπου ΠΕΔΙΑ

εδω=ΕΓΓΡΑΦΕΣ(αλφα)+1

ΔΩΣΕ αλφα, α1\$+α2\$+α3\$+α4\$, εδω

όπου τα α1\$, α2\$, α3\$, α4\$ έχουν το καθένα συγκεκριμένο μήκος λέξης

Αν το σύνολο των μηκών των επιμέρους λέξεων είναι μικρότερο ή μεγαλύτερο του μήκους όλης της εγγραφής (δες μήκος στην ΑΝΟΙΞΕ ... ΓΙΑ ΠΕΔΙΑ ... ΜΗΚΟΣ=...) τότε ανάλογα προσαρμόζεται (ο μεταφραστής ή κόβει το πλεόνασμα ή προσθέτει διαστήματα)

ΛΙΓΑ ΛΟΓΙΑ ΓΙΑ ΤΑ ΑΡΧΕΙΑ ΤΥΠΟΥ ΠΕΔΙΑ / RANDOM

Αυτά τα αρχεία λέγονται και αρχεία σταθερού μήκους εγγραφής. Πρώτα επιλέγουμε το μήκος της εγγραφής μας και κατόπιν μπορούμε να γράφουμε, να διαβάζουμε, να αλλάζουμε οποιαδήποτε εγγραφή. Για να διαγράψουμε όμως μια εγγραφή χρειάζεται λίγο κόπος παραπάνω και αυτό γιατί η μνήμη που έχει παραχωρηθεί δεν μπορεί να αφαιρεθεί (άρα να διαγραφεί) αλλά με χειρισμό μπορούμε να μην προσμετράμε κάποια εγγραφή και με μια αντιγραφή σε νέο αρχείο των έγκυρων εγγραφών μπορούμε να συμπίεσουμε το αρχείο μας διαγράφοντας τα λογικώς διαγραμμένα. Και τα λογικώς διαγραμμένα είναι αυτά που τα έχουμε μαρκάρει ως διαγραμμένα! Π.Χ. ο πρώτος χαρακτήρας της εγγραφής μπορεί να μας δείχνει αν είναι η εγγραφή διαγραμμένη ή όχι.

Πρέπει να βλέπουμε ένα αρχείο τύπου ΠΕΔΙΑ / RANDOM ως ένας πίνακας αλφαριθμητικών σταθερού μήκους.

Δες την εντολή ΠΑΡΕ

Αναγνωριστικό: ΕΙΚΟΝΕΣ [BITMAPS]

ΕΙΚΟΝΕΣ

Μας δίνει όλα τα αρχεία τύπου bmp στον τρέχον κατάλογο

Ίδιο με το ΑΡΧΕΙΑ "BMP|JPG|GIF|DIB|ICO|CUR|PNG|TIF"

ΚΑΤΑΛΟΓΟΣ "C:"

ΕΙΚΟΝΕΣ

ΕΙΚΟΝΕΣ ! ' αλλάζει το τρόπο ταξινόμησης...το κάνει εδώ βάση ονόματος.

ΑΡΧΕΙΑ "bmp"

Αναγνωριστικό: ΗΧΟΙ [SOUNDS]

ΗΧΟΙ

Μας δίνει όλα τα αρχεία τύπου wav στον τρέχον κατάλογο, ίδιο με το ΑΡΧΕΙΑ "wav"

ΚΑΤΑΛΟΓΟΣ "C:"

ΗΧΟΙ

ΑΡΧΕΙΑ "wav"

Αναγνωριστικό: ΚΛΕΙΣΕ [CLOSE]

ΚΛΕΙΣΕ μεταβλητη_χειρισμου_αρχειου

ΚΛΕΙΣΕ μεταβλητη_χειρισμου_αρχειου1, ..., μεταβλητη_χειρισμου_αρχειουN

Μπορούμε να βάζουμε # πριν την μεταβλητή χειρισμού (αν θέλουμε)

η ΚΛΕΙΣΕ χωρίς μεταβλητή χειρισμού κλείνει όλα τα αρχεία και τις συνδέσεις με βάσεις δεδομένων

ΚΛΕΙΣΕ ΒΑΣΗ ονομα_βασης1\$ [,ονομα_βασης2\$]

κλείνουμε επιλεκτικά τις συνδέσεις

Αναγνωριστικό: ΜΕΤΑΘΕΣΗ [SEEK]

ΜΕΤΑΘΕΣΗ #L, 10

σε ένα ανοιχτό αρχείο μεταθέτει τη θέση του δρομέα εγγραφής στο συγκεκριμένο Byte.
ελάχιστο 1 και μέγιστο όσο το μέγιστο ενός μακρυ ακεραίου (LONG)

Αν το μέγεθος του αρχείου είναι μικρότερο τότε τοποθετεί τον δρομέα μετά το τέλος, έτσι η
εγγραφή θα μεγαλώσει το αρχείο
μπορούμε να διαβάσουμε το με την SEEK(#L)

Αναγνωριστικό: ΟΝΟΜΑ [NAME]

ΟΝΟΜΑ αρχείο_ετσι ΩΣ αρχείο_αλλιως

Αλλαγή ονόματος αρχείου που βρίσκεται στον τρέχον κατάλογο

Αναγνωριστικό: ΠΑΡΕ [GET]

Το αντίστροφο της ΔΩΣΕ

ΠΑΡΕ μεταβλητη_χειρισμου_αρχειου, αλφαριθμητική_μεταβλητή\$, θεση_στο_αρχειο

ΠΑΡΕ μεταβλητη_χειρισμου_αρχειου, αλφαριθμητική_μεταβλητή\$(δείκτης), θεση_στο_αρχειο

π.χ. Λαμβάνουμε την τελευταία εγγραφή στην μεταβλητή α\$

οι εγγραφές είναι τμήματα στο αρχείο σειρών από bytes, με συγκεκριμένο μήκος,
το οποίο δηλώνουμε στο άνοιγμα. Δες ΑΝΟΙΞΕ, ΚΛΕΙΣΕ, ΔΩΣΕ

εδω=ΕΓΓΡΑΦΕΣ(αλφα)

ΠΑΡΕ αλφα, α\$, εδω

Αναγνωριστικό: ΣΧΕΔΙΑ [DRAWINGS]

ΣΧΕΔΙΑ

Μας δίνει όλα τα αρχεία τύπου wmf στον τρέχον κατάλογο

Ίδιο με το ΑΡΧΕΙΑ "wmf|emf"

ΚΑΤΑΛΟΓΟΣ "C:"

ΣΧΕΔΙΑ
ΑΡΧΕΙΑ "wmf"
ΣΧΕΔΙΑ ! ' ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΟΝΟΜΑ

Αναγνωριστικό: ΤΑΙΝΙΕΣ [MOVIES]
ΤΑΙΝΙΕΣ
Μας δίνει όλα τα αρχεία τύπου avi στον τρέχον κατάλογο
Ίδιο με το ΑΡΧΕΙΑ "avi"

ΚΑΤΑΛΟΓΟΣ "C:"
ΤΑΙΝΙΕΣ
ΑΡΧΕΙΑ "avi"

Ομάδα: ΕΝΤΟΛΕΣ ΚΟΝΣΟΛΑΣ - CONSOLE COMMANDS

Αναγνωριστικό: ΑΝΑΝΕΩΣΗ [REFRESH]
Από την τελευταία έκδοση δεν ισχύει αυτό "Όταν αλλάζει η γραμμή εκτύπωσης μετά από μια τύπωση τότε αυτόματα γίνεται ανανέωση οθόνης", εκτός και αν έχουμε εκτελέσει την καθολική εντολή ΘΕΣΕ ΑΡΓΑ
Μπορούμε να προκαλέσουμε όποτε θέλουμε ανανέωση της οθόνης με την εντολή ΑΝΑΝΕΩΣΗ και τις παραλλαγές της:

1) ΑΝΑΝΕΩΣΗ
Αυτή είναι η απλή μορφή που κάνει αυτόματη ανανέωση της οθόνης

2) ΑΝΑΝΕΩΣΗ 0
Μηδενίζει τον μετρητή ανανέωσης οθόνης

3) ΑΝΑΝΕΩΣΗ 100
Ορίζει τον χρόνο που θα γίνουν ανανεώσεις αυτόματα. Κάθε φορά που το σύστημα ζητάει ανανέωση ελέγχεται αν έχει περάσει το οριζόμενο χρονικό διάστημα και αν ναι τότε γίνεται ανανέωση. ΠΡΟΣΟΧΗ δεν σχετίζεται με την ΑΝΑΜΟΝΗ όπου η ροή στο συγκεκριμένο νήμα διακόπτεται για λίγα δευτερόλεπτα. Και οι δύο εντολές αποσκοπούν στο να δώσουν χρόνο στο σύστημα, αλλά η δεύτερη (η ΑΝΑΜΟΝΗ) γίνεται με καθυστέρηση ανεξάρτητα αν υπάρχει κάτι να γίνει στο παρασκήνιο ή όχι.

Αναγνωριστικό: ΑΝΑΦΟΡΑ [REPORT]

ΑΝΑΦΟΡΑ!4 ' βάζει το πλάτος του TAB (χαρακτήρας 9) στους 4 χαρακτήρες - για κάθε επίπεδο κρατάει διαφορετικό πλάτος.

ΑΝΑΦΟΡΑ "ΚΕΙΜΕΝΟ ΜΕ ΑΛΛΑΓΕΣ ΓΡΑΜΜΩΝ"

ΑΝΑΦΟΡΑ τροπος_αναφορας, "ΚΕΙΜΕΝΟ ΜΕ ΑΛΛΑΓΕΣ ΓΡΑΜΜΩΝ"

ΑΝΑΦΟΡΑ τροπος_αναφορας, "ΚΕΙΜΕΝΟ ΜΕ ΑΛΛΑΓΕΣ ΓΡΑΜΜΩΝ", πλατος_Δες_πως

ΑΝΑΦΟΡΑ τροπος_αναφορας, "ΚΕΙΜΕΝΟ ΜΕ ΑΛΛΑΓΕΣ ΓΡΑΜΜΩΝ", πλατος_Δες_πως,

Γραμμες_Για_Τυπωμα

ΑΝΑΦΟΡΑ τρ_αναφορας, "ΚΕΙ....Ν", πλατος_Δες_πως, Γραμμες_Για_Τυπωμα ΓΡΑΜΜΗ

αρχική_γραμμή

Νέες παραλλαγές:

ΑΝΑΦΟΡΑ τρ_αναφορας, "ΚΕΙ....Ν", πλατος_Δες_πως, Γραμμες_Για_Τυπωμα Ως Μεταβλητή\$

ΑΝΑΦΟΡΑ τρ_αναφορας, "ΚΕΙ....Ν", πλατος_Δες_πως, Γραμμες_Για_Τυπωμα ΓΡΑΜΜΗ

αρχική_γραμμή Ως Μεταβλητή\$

* οι νέες παραλλαγές δουλεύουν με αρνητικό νούμερο στις γραμμές_για_τύπωμα και δεν τυπώνουν αλλά βάζουν τις γραμμές στη μεταβλητή.

Η αναφορά ξεκινάει από τη στήλη που καθορίζει ο δρομέας. Το αριστερό διάστημα που ορίζει ο δρομέας θα μείνει κενό.

Τρόπος αναφοράς είναι:

0 Στοίχιση Αριστερά και Δεξιά

1 Στοίχιση Δεξιά

2 Στοίχιση Κέντρο

3 Στοίχιση Αριστερά

Μπορούμε λοιπόν πριν καλέσουμε την ΑΝΑΦΟΡΑ να ξέρουμε το αριστερό όριο (left margin), αλλά πρέπει να έχουμε και έναν τρόπο να ορίσουμε το δεξιό περιθώριο (right margin). Για το λόγο αυτό δίνουμε το πλατος_Δες_πως ως αριθμό, με προσοχή όμως γιατί αν ο αριθμός είναι μικρότερος από το Πλάτος (της οθόνης) τότε η εντολή τον λαμβάνει ως πλάτος σε χαρακτήρες (της Οθόνης), αν είναι μεγάλος τότε η εντολή τον θεωρεί ως twips (1440 twips είναι μια ιντσα).

Σε κάθε περίπτωση γίνεται μεταφορά λέξης στην επόμενη γραμμή αν δεν χωράει, όμως δεν είναι σίγουρο ότι θα τυπωθεί γιατί μπορεί να είναι την δηλώσουμε μη εκτυπωσιμη. Οι εκτυπώσιμες γραμμές είναι αυτές που ξεκινάνε από την αρχική_γραμμή (ή αν δεν δώσουμε αρχική τότε θεωρείται η πρώτη γραμμή, η 1), και συνολικά φθάνουν τις γραμμές_για_τύπωμα. Αν δώσουμε αρνητικό νούμερο τότε η εντολή δεν τυπώνει αλλά υπολογίζει πόσες γραμμές θα τυπωθούν (αν τυπώναμε). Την απάντηση την λαμβάνουμε σε μια μεταβλητή, ανάγνωσης μόνο, την ΓΡΑΜΜΕΣΑΝΑΦΟΡΑΣ ή REPORTLINES

Μπορούμε να χρησιμοποιήσουμε την Αναφορά στον ΕΚΤΥΠΩΤΗ για να τυπώσουμε κείμενο.

Αρκεί να την βάλουμε σε μια δομή ΕΚΤΥΠΩΤΗΣ {

Αναλογική γραφή μπορούμε να τυπώσουμε με την ΤΥΠΩΣΕ αλλά δεν έχει σχέση με την

ΑΝΑΦΟΡΑ

Δείτε το παράδειγμα:

```
FORM 48,32
```

```
FOR I=0 TO 8 {
```

```
PRINT $(I, 8),145355,34.23432,34.222
```

```
}
```

Η εντολή ΑΝΑΦΟΡΑ είναι μια σύνθετη εντολή για να φτιάξει κανείς στήλες κειμένου (με παράγραφους) στην οθόνη με αναλογική γραφή και με αυτόματο γύρισμα λέξης στην επόμενη γραμμή, προσαρμόζοντας κατάλληλα τα κενά των λέξεων στην κάθε γραμμή.

Η εντολή έχει δυο καταστάσεις λειτουργίας...μια που πράγματι τυπώνει και μια άλλη που υπολογίζει τις γραμμές που χρειάζεται και απλά ενημερώνει μια εσωτερική μεταβλητή. Στην πρώτη κατάσταση μπορούμε να ορίσουμε πόσες γραμμές θα προσπεράσει και πόσες θα τυπώσει. Έτσι αν ξέρουμε πόσες γραμμές είναι όλο το κείμενο που θέλουμε να τυπώσουμε με την εντολή αναφορά υπολογίζοντας τη μισή οθόνη και κάτι λιγότερο, τότε μπορούμε να φτιάξουμε δυο στήλες με δυο εντολές, μια για κάθε στήλη, όπου θα δώσουμε τον αριθμό των γραμμών που θα αφήσουμε και τον αριθμό των γραμμών που θα γράψουμε. Στη περίπτωση που δεν δώσουμε αυτούς τους αριθμούς τότε η αναφορά τρέχει σε όλη την οθόνη και σταματάει για να διαβάσουμε και πατώντας το διάστημα προχωράει στο επόμενο κομμάτι. Μπορούμε να δώσουμε πολλαπλές αναφορές αλλάζοντας σε ορισμένες τον τρόπο στοίχισης, και πάλι θα καταμετρηθούν οι τυπωμένες γραμμές ώστε να δουλεύει σωστά το κράτημα της αναφοράς για να πάρει χρόνο ο αναγνώστης και να διαβάσει τι γράφει!

Μπορούμε να δώσουμε ένα κείμενο σε ένα αλφαριθμητικό:

```
A$={Πρώτη γραμμή
```

```
Δεύτερη γραμμή
```

```
Τρίτη Γραμμή}
```

```
Αναφορά A$
```

```
Αναφορά 1, A$
```

```
Αναφορά 2, A$
```

```
Αναφορά 3, A$
```

Μια εξήγηση εδώ:

Κάθε γραμμή στο αλφαριθμητικό τερματίζει με δυο χαρακτήρες, εκτός από την τελευταία γραμμή, που όμως δεν αναφέρονται αυτά στις γραμμές εκτύπωσης (αυτές καθορίζονται από το πλάτος εκτύπωσης). Δηλαδή υπάρχουν στο αλφαριθμητικό σημάδια χαρακτήρες που δηλώνουν την αλλαγή παραγράφου (και όχι γραμμής). Μια παράγραφος είναι μια ομάδα λέξεων που τερματίζει σε μια συγκεκριμένη ακολουθία χαρακτήρων ή στο πέρας του αρχείου ή το πέρας του αλφαριθμητικού (το οποίο δουλεύει εδώ σαν αρχείο).

Παρακάτω έχουμε ένα παράδειγμα (μπορούμε να το αντιγράψουμε σε ένα τμήμα), όπου σε ένα αλφαριθμητικό βάζουμε ένα κείμενο. Μετά κάνουμε μια επανάληψη για 20 φορές όπου η μεταβλητή i χρησιμοποιείται στην εντολή ΑΝΑΦΟΡΑ ή REPORT ως το πλάτος σε χαρακτήρες οθόνης.

```

σ$={σ
Α
β.
.
..
. ....
}
για i=1 εως 20 {
οθονη
δρομεας 10,10
τυπωσε "*"1234567890";
αναφορα 2, σ$,i,-1000
δρομεας 10,11
τυπωσε "*";
κ= ΓΡΑΜΜΕΣΑΝΑΦΟΡΑΣ
τυπωσε @(θεση,γραμμη, θεση+i,θεση+κ, 5,8);
αναφορα 2, σ$,i,κ
τυπωσε i , κ
κ$=ενκομ$
}

```

Η ΓΡΑΜΜΕΣΑΝΑΦΟΡΑΣ δίνει το αποτέλεσμα από την ΑΝΑΦΟΡΑ με αρνητικό αριθμό γραμμών για εκτύπωση (εδώ -1000), η οποία δεν τυπώνει έτσι παρά μόνο υπολογίζει αυτό το νούμερο.

Αναγνωριστικό: ΑΝΑΨΕ [SHOW]

ΑΝΑΨΕ

Μας δείχνει το πλαίσιο γραφικών και κειμένου (την οθόνη). Ενεργοποιείται αυτόματα σε εντολές που θα γράψουμε χαρακτήρες με πληκτρολόγηση πάνω στο πλαίσιο αυτό.

Αντίθετη εντολή ΣΒΗΣΕ

Όταν κάνουμε διπλό κλικ (ή μονό σε κάποιες περιπτώσεις) πάνω σε ένα αρχείο gsb τότε θα εμφανιστεί το περιθώριο αλλά όχι η οθόνη. Αυτή θα εμφανιστεί αν δώσουμε ΑΝΑΨΕ ή μια εντολή που αυτόματα ανάβει την οθόνη.

Υπάρχει η μεταβλητή μόνο για ανάγνωση ΟΘΟΝΗ η οποία είναι -1 όταν η φόρμα είναι στο προσκήνιο

Αναγνωριστικό: ΑΦΗΣΕ [RELEASE]

ΑΦΗΣΕ

Αφήνει την προηγούμενα κρατημένη οθόνη (πρέπει να προηγηθεί η ΚΡΑΤΗΣΕ)

Αναγνωριστικό: ΓΡΑΜΜΑΤΟΣΕΙΡΑ [FONT]

1) ΓΡΑΜΜΑΤΟΣΕΙΡΑ "VERDANA"

Αλλάζουμε γραμματοσειρά. Οι γραμματοσειρές πριν την έκτη έκδοση άλλαζαν μόνο με την εντολή Αρχή για τα γράμματα που τυπώνουμε σε στήλες, ενώ μπορούσαμε με την Επιγραφή να τυπώσουμε γράμματα με οποιαδήποτε γραμματοσειρά. Αυτό που κάνει η εντολή Γραμματοσειρά είναι να αλλάξει την τρέχουσα γραμματοσειρά χωρίς όμως να επανακαθορίσει την οθόνη. Αν και δεν αλλάζει το μέγεθος γραμμάτων (αριθμητικά) μπορεί το τρέχον μέγεθος να διαφέρει σε άλλη γραμματοσειρά!

Ο σκοπός αυτής της εντολής είναι να τυπώσουμε σε θέσεις χαρακτήρων με την γραμματοσειρά που θέλουμε, και να ξαναγυρίσουμε στη βασική. Υπάρχει η Γραμματοσειρά\$ που κρατάει την τρέχουσα γραμματοσειρά. Θα την γράψουμε σε μια μεταβλητή, θα αλλάξουμε γραμματοσειρά και θα επανέλθουμε στην αρχική.

Τύπωσε Γραμματοσειρά\$

2) ΓΡΑΜΜΑΤΟΣΕΙΡΑ ΦΟΡΤΩΣΕ "path\name.ttf" [, "path\name.ttf"]

μπορούμε να δώσουμε όνομα αρχείου χωρίς μονοπάτι, θα χρησιμοποιήσει το τρέχον αυτή η εντολή φορτώνει γραμματοσειρές, δεν τις ορίζει (το όνομα του αρχείου είναι διαφορετικό από το όνομα της γραμματοσειράς)

Στο Επίλεξε.Γραμματοσειρά αν πατήσουμε το end μας πάει στη τελευταία γραμματοσειρά, αυτήν που βάλαμε!

3) ΓΡΑΜΜΑΤΟΣΕΙΡΑ ΔΙΑΓΡΑΦΗ "path\name.ttf" [, "path\name.ttf"]

για να διαγράψουμε μια γραμματοσειρά αφενός δεν πρέπει να την χρησιμοποιούμε και αφετέρου πρέπει να δώσουμε το όνομα όπως καταχωρήθηκε στην φόρτωση. Αν δεν είχαμε δώσει το μονοπάτι πρέπει να βάλουμε το ίδιο μονοπάτι. Αν δεν τις διαγράψουμε δεν πειράζει γιατί θα διαγραφούν αυτόματα στο τέλος εκτέλεσης του διερμηνευτή.

Αναγνωριστικό: ΔΙΑΣΤΙΧΟ [LINESPACE]

Τύπωσε διάστιχο

Διάστιχο 120 ' τιμές ανά δυο pixel (δηλαδή ανά 30 twips), το 120 είναι 8 pixels για

ΠΛΑΤΟΣ.ΣΗΜΕΙΟΥ=15

Φόρμα 80

Αναγνωριστικό: ΔΙΠΛΑ [DOUBLE]

ΔΙΠΛΑ

Κάνει τα γράμματα να εμφανίζονται με διπλό ύψος και πλάτος. Επίσης και η ολίσθηση γίνεται διπλής γραμμής.
Αντίθετη ΚΑΝΟΝΙΚΑ

Αναγνωριστικό: ΔΡΟΜΕΑΣ [CURSOR]
ΔΡΟΜΕΑΣ θέσηΧ, θέσηΨ
ΔΡΟΜΕΑΣ 0,0 ' η πάνω αριστερή γωνία
ΤΥΠΩΣΕ "ΓΕΙΑ"

ΤΥΠΩΣΕ @(10,10),"ΓΕΙΑ"
'Εδώ το κόμμα μετά την εσωτερική εντολή @() της ΤΥΠΩΣΕ δεν κάνει τίποτα άλλο από το να την ξεχωρίζει από τα υπόλοιπα.

ΤΥΠΩΣΕ "ΑΑ", ΘΕΣΗ, ΣΤΗΛΗ, ΓΡΑΜΜΗ
'Εδώ το κόμμα μετά το "ΑΑ" πάει τον δρομέα στην αρχή της επόμενης στήλης
'Η ΘΕΣΗ δηλώνει ότι αριστερά από τη θέση χαρακτήρα βρίσκονται τόσοι χαρακτήρες..
'Η ΣΤΗΛΗ δηλώνει τον αριθμό χαρακτήρων ως πλάτος στήλης. Επειδή ξεκινάμε από αριστερά από το 0 για στήλη 10 'χαρακτήρων η θέση της δεύτερης στήλης θα είναι στο 10 (από 0 έως 9 θα είναι οι δέκα χαρακτήρες που έχει η πρώτη στήλη).

ΤΥΠΩΣΕ "ΑΑ",,"ΒΒ",,"ΓΓ"

Αναγνωριστικό: ΕΙΚΟΝΙΔΙΟ [ICON]
Εάν έχουμε θέσει τίτλο (δες ΤΙΤΛΟΣ) τότε μπορούμε να αλλάξουμε εικονίδιο
ΕΙΚΟΝΙΔΙΟ "ταδε.ico"

Αναγνωριστικό: ΕΛΛΗΝΙΚΑ [GREEK]
ΕΛΛΗΝΙΚΑ
αλλάζει την κωδικοσελίδα αλλά και τη γλώσσα που εμφανίζονται τα μηνύματα λάθους
ΔΕΣ ΛΑΤΙΝΙΚΑ, ΧΑΡΑΚΤΗΡΕΣ

Αναγνωριστικό: ΕΠΙΓΡΑΦΗ [LEGEND]
1) Εγγραφή επιγραφών με οποιαδήποτε γραμματοσειρά

- ΕΠΙΓΡΑΦΗ φραση\$, γραμματοσειρα\$, μεγεθος_γραμματοσειρας\$, γωνια_σε_ακτινια
- 2) Όπως η (1) αλλά με γωνία
ΕΠΙΓΡΑΦΗ φραση\$, γραμματοσειρα\$, μεγεθος_γραμματοσειρας\$, γωνια_σε_ακτινια
- 3) Όπως η (2) αλλά με στοίχιση
ΕΠΙΓΡΑΦΗ φραση\$, γραμματοσειρα\$, μεγεθος_γραμματοσειρας\$, γωνια_σε_ακτινια, στοίχιση
στοίχιση 1- δεξιά 2-κέντρο 3 αριστερά
εδώ δεν έχουμε μεταφορά λέξης σε νέα γραμμή όπως στην Αναφορά (δεν υπάρχει ορισμός μέγιστου μήκους γραμμής)
- 4) Όπως η (3) αλλά με επιλογή ποιότητας
ΕΠΙΓΡΑΦΗ φραση\$, γραμματοσειρα\$, μεγεθος_γραμματοσειρας\$, γωνια_σε_ακτινια, στοίχιση, ποιότητα
Η ποιότητα μπορεί να είναι 0 ή μη μηδέν (ό,τι άλλο από μηδέν)
Η διαφορά είναι ότι στη 0 ποιότητα τυπώνεται στο φόντο ακριβώς τα γράμματα με το χρώμα που έχουμε επιλέξει. Σε κάθε άλλη (μη μηδενική) το χρώμα διαφοροποιείται στις άκρες των γραμμάτων για να δείχνουν πιο καθαροί στο τύπωμα.
- 5) Όπως η 4 αλλά έχει επιπλέον τον αριθμό σε twips (θετικό ή αρνητικό) που καθορίζει την πρόσθεση ή αφαίρεση χώρου μεταξύ των γραμμάτων
ΕΠΙΓΡΑΦΗ φραση\$, γραμματοσειρα\$, μεγεθος_γραμματοσειρας\$, γωνια_σε_ακτινια, στοίχιση, ποιότητα
- 6) ΕΠΙΓΡΑΦΗ ! κειμενο\$, πλατος_χαρακτήρων, υψος_χαρακτήρων
Δες ΕΙΣΑΓΩΓΗ !

Παράδειγμα:

Δείτε παρακάτω πως μπορούμε να τυπώνουμε φράσεις με περίγραμμα σε κάθε γράμμα (outline text)

```
ΟΘΟΝΗ 3,0
ΘΕΣΗ 6100,3300
ΚΥΚΛΟΣ ΓΕΜΙΣΜΑ 1,2700
ΚΥΚΛΟΣ ΓΕΜΙΣΜΑ 3,2000,,3
ΚΥΚΛΟΣ ΓΕΜΙΣΜΑ 4,1500,,4
ΘΕΣΗ 2000,5000
ΠΕΝΑ 12
ΠΑΧΟΣ 3 {
ΧΡΩΜΑ 4 {
ΕΠΙΓΡΑΦΗ "ΟΧΙ ΚΑΠΝΙΣΜΑ","ARIAL", 48 , ΠΙ/6
}}
ΠΕΝΑ 11
```

Στην περίπτωση (5) η εκτύπωση γίνεται σε πλαίσιο όπου η αριστερή γωνία είναι η θέση του δρομέα χαρακτήρων και εμείς δίνουμε πλάτος και ύψος. Στις άλλες περιπτώσεις η εκτύπωση

γίνεται με αρχή τον δρομέα γραφικών. Και στις δυο περιπτώσεις η γραφή είναι αναλογική. Πιο νέο παράδειγμα!

```
πένα #7700ff
ποιότητα = αληθής
κέντρο = 2
οθόνη #fccbb7, 0 \ \ ακύρωση τυχόν διαχωρισμού οθόνης
διπλα : Αναφορά κέντρο, "Παράδειγμα Επιγραφής" : κανονικά
οθόνη #efbbf7, 2 \ \ διαχωρισμός οθόνης από τη τρίτη γραμμή (0 και 1 είναι οι δυο πρώτες)
θέση 6000,6000
παλιά_τιμή_Φ = Φαρδιά
παλιά_τιμή_Π = Πλάγια
Φαρδιά 1 : Πλάγια 1
Πένα #224488 { \ \ χρώμα γύρω από τα γράμματα
  Πάχος 3 {
    Χρώμα #ff55aa { \ \ χρώμα μέσα στα γράμματα
      Επιγραφή {Γειά σου
        Το όνομά μου είναι Γιώργος
      }, "Arial Black", 48, pi/6, κέντρο, ποιότητα, -90
      \ \ 60 twips are 4 pixels in 15 dpi screens or 5 pixels in 12 dpi
    }
  }
}
Πλάγια παλιά_τιμή_Π : Φαρδιά παλιά_τιμή_Φ
θέση 12000, 6000
Επιγραφή {Γειά σου
  Το όνομά μου είναι Γιώργος}, "Arial Black", 40, pi/6, κέντρο, ποιότητα
α$=κομ$ \ \ περιμένει ένα πλήκτρο να πατηθεί, πριν τελειώσει το πρόγραμμα
```

```
Αναγνωριστικό: ΕΠΙΠΕΔΟ [LAYER]
ΕΠΙΠΕΔΟ 2 {
  εντολές που πάνε στον παίκτη 2
}
```

Εκτός από την Φορμα που δεν λειτουργεί στο ΕΠΙΠΕΔΟ όλες οι άλλες εντολές της οθόνης μπορούν να χρησιμοποιηθούν. Π.χ. η Τυπος για να αλλάξει το μέγεθος γραμμάτων αλλά και το μέγεθος του επιπέδου. Η εντολή Κίνηση μετακινεί το Επίπεδο.

Επειδή τα επιπεδα δημιουργούνται με εικόνα...όταν καθαρίζουμε το επίπεδο με την εντολή Οθονη τότε επαναφέρουμε την αρχική εικόνα. Αν έχουμε επιλέξει διαφάνεια στο Παικτη (στη δημιουργία του Επιπέδου) τότε θα συνεχίσει το Επίπεδο να έχει διαφάνεια. Αυτό δεν σημαίνει ότι δεν θα τυπώνει εκεί που υπάρχει διαφάνεια, απλά δεν θα εμφανίζεται. Επίσης όταν με το ποντίκι κάνουμε κλικ πάνω σε διάφανο σημείο ενός επιπέδου τότε το κλικ δεν πάει στο επίπεδο αλλά σε εκείνο το αντικείμενο που έχει αδιάφανο σημείο ακριβώς από κάτω.

Αναγνωριστικό: ΕΠΙΦΑΝΕΙΑ [DESKTOP]

ΕΠΙΦΑΝΕΙΑ

Εμφανίζει τα περιεχόμενα της οθόνης πίσω από το περιθώριο μέσα σε αυτό

ΕΠΙΦΑΝΕΙΑ 100

ορίζει διαφάνεια όλης της εφαρμογής στο 100/255 *100%

Με ΕΠΙΦΑΝΕΙΑ 255 επαναφέρουμε την "μη διαπερατότητα" ή την αδιαφάνεια στην εφαρμογή.

ΕΠΙΦΑΝΕΙΑ 255,1

ορίζουμε το χρώμα 1 (από τα 16 βασικά) αν είναι το διάφανο. Μπορούμε να φορτώσουμε μια εικόνα στο "βαθύ" φόντο. Υπάρχει δηλαδή ένα φόντο που χρησιμοποιείται όταν βάζουμε την διαφάνεια και καταλαμβάνει όλη την οθόνη. Σε αυτό το φόντο μπορούμε να βάλουμε μια εικόνα. Δες την παρακάτω εντολή:

ΕΠΙΦΑΝΕΙΑ ΕΙΚΟΝΑ Α\$

Το Α\$ μπορεί να είναι αρχείο εικόνας ή αλφαριθμητικό που περιέχει εικόνα.

Η εικόνα πάει στο "βαθύ" φόντο, το οποίο ετοιμάζεται όταν επιλέξουμε διαφάνεια. Για την ακρίβεια αλλάζει το σύστημα που δουλεύει η βασική φόρμα της εφαυτό επιβαρύνει το σύστημα.

ΕΠΙΦΑΝΕΙΑ ΚΡΥΨΕ

Εξαφανίζει το βαθύ φόντο. Αυτό σημαίνει ότι η φόρμα μας αν έχει διαφάνεια τότε θα εκθέτει οτιδήποτε πίσω από αυτήν ώστε να μπορεί να επιλεχθεί.

ΕΠΙΦΑΝΕΙΑ ΚΑΘΑΡΗ

Μαυρίζει το βαθύ φόντο και το κάνει αδιαπέραστο.

Η εντολή ΕΠΙΦΑΝΕΙΑ 255 επαναφέρει την αδιαφάνεια, αλλά δεν επαναφέρει το αρχικό σύστημα, άρα επιβαρύνει το σύστημα (στα ΧΡ μόνο υπάρχει θέμα)

Αναγνωριστικό: ΕΤΙΚΕΤΑ.ΦΟΡΜΑΣ [FORMLABEL]

ΕΤΙΚΕΤΑ.ΦΟΡΜΑΣ "ΕΠΙΓΡΑΦΗ", [ΓΡΑΜΜΑΤΟΣΕΙΡΑΣ], [ΤΥΠΟΣ], [ΣΤΟΙΧΙΣΗ], [ΔΙΑΣΤΗΜΑ ΓΡΑΜΜΑΤΩΝ]

Στα αγγλικά έχουμε πιο εύκολη ταύτιση. Στα ελληνικά θέλει δουλειά για να βρεθεί το κατάλληλο. Αυτό συμβαίνει γιατί το τετράγωνο που γράφεται ένας χαρακτήρας έχει υπολογιστεί από τους αγγλικούς χαρακτήρες.

ΠΑΡΑΘΥΡΟ 8, ΣΥΣΚΕΥΗ

ΓΡΑΜΜΑΤΟΣΕΙΡΑ "LUCIDA CONSOLE"

ΤΥΠΟΣ 22.5

ΤΥΠΩΣΕ ΓΡΑΜΜΑΤΟΣΕΙΡΑΣ, ΤΥΠΟΣ

ΔΡΟΜΕΑΣ 10, 9

ΤΥΠΩΣΕ "ΑΒΓΔΕ-----"

ΔΡΟΜΕΑΣ 10, 10

ΘΕΣΗ !

ΒΗΜΑ 0,ΥΨΟΣ.ΣΗΜΕΙΟΥ

ΕΤΙΚΕΤΑ.ΦΟΡΜΑΣ "ΑΒΓΔΕ-----",,ΤΥΠΟΣ*.94,,ΠΛΑΤΟΣ.ΣΗΜΕΙΟΥ

ΔΡΟΜΕΑΣ 10, 11

ΤΥΠΩΣΕ "ΑΒΓΔΕ-----"

ΣΗΜ:ΕΞΟΔΟΣ

ΔΡΟΜΕΑΣ 10, 10

ΠΕΝΑ 0 {

ΤΥΠΩΣΕ "ΑΒΓΔΕ----- ΑΠΟΔΕΙΞΗ"

}

Αναγνωριστικό: ΚΑΝΟΝΙΚΑ [NORMAL]

ΚΑΝΟΝΙΚΑ

Επιλέγουμε κανονικά γράμματα

(βάση του τύπου που έχουμε δώσει στην ΤΥΠΟΣ ή ΠΑΡΑΘΥΡΟ)

Αντίθετη ΔΙΠΛΑ

Αναγνωριστικό: ΚΙΝΗΣΗ [MOTION]

Μετακίνηση της οθόνης πάνω από το περιθώριο

ΚΙΝΗΣΗ ΚΕΝΤΡΟ

ΚΙΝΗΣΗ θέσηΧ, θέσηΥ

ΚΙΝΗΣΗ θέσηΧ

ΚΙΝΗΣΗ ,θέσηΥ

Μπορούμε να διαβάσουμε την θέση της ΟΘΟΝΗΣ με τα ΚΙΝΗΣΗ.Χ και ΚΙΝΗΣΗ.Υ

Αναγνωριστικό: ΚΙΝΗΣΗ.Π [MOTION.W]

Μετακίνηση του περιθωρίου πάνω στην επιφάνεια εργασίας

ΚΙΝΗΣΗ.Π;

ΚΙΝΗΣΗ.Π θέσηΧ, θέσηΥ

ΚΙΝΗΣΗ.Π θέσηΧ

ΚΙΝΗΣΗ.Π ,θέσηΥ

Παρατήρηση: εδώ το ; δηλώνει κέντρο.

Υπάρχουν οι μεταβλητές ΚΙΝΗΣΗ.ΠΧ και ΚΙΝΗΣΗ.ΠΥ που μας δείχνουν τα τωρινά Χ και Υ της άνω αριστερής γωνίας. Η εντολή ΕΠΙΦΑΝΕΙΑ αν είχε δοθεί πριν την μετακίνηση τότε δείχνει λάθος εικόνα, σε αυτήν την περίπτωση πρέπει να την ξανακαλέσουμε.

Αναγνωριστικό: ΚΡΑΤΗΣΕ [HOLD]

ΚΡΑΤΗΣΕ

Κρατάει προσωρινά τα περιεχόμενα της οθόνης σε εσωτερικό χώρο και τα αποδίδει με την ΑΦΗΣΕ

Μπορούμε να φτιάξουμε γρήγορα ένα εφε όπου θα αναβοσβήνει κάτι! Ή μπορούμε να μετακινούμε έναν δείκτη ως ένα γραφικό και κάθε φορά πριν μια ΑΝΑΝΕΩΣΗ 0 να αφήνουμε με την ΑΦΗΣΕ την καθαρή εικόνα και να βάζουμε το δείκτη έτσι δεν θα βλέπουμε ποτέ να λείπει ο δείκτης αλλά να μετακινείται. Μπορούμε να χρησιμοποιήσουμε αντί για γραφικό δείκτη μια εικόνα με την εντολή ΔΙΑΦΑΝΟ

HOLD και ΚΡΑΤΑ είναι αναγνωριστικά για το ΝΗΜΑ

Αναγνωριστικό: ΚΥΛΙΣΗ [SCROLL]

ΚΥΛΙΣΗ ΑΝΩ

ΚΥΛΙΣΗ ΚΑΤΩ

ΚΥΛΙΣΗ ΧΩΡΙΣΜΑ 5 ' δηλώνει την τελευταία ακίνητη σειρά του σταθερού άνω τμήματος

Με την εντολή ΟΘΟΝΗ , 5 πετυχαίνουμε το ίδιο αλλά καθαρίζουμε την οθόνη (το κάτω τμήμα)

Η εντολή ΑΝΑΦΟΡΑ που τυπώνει κείμενο σταματάει την ολίσθηση και περιμένει πάτημα πλήκτρου ή κλικ ποντικού στην οθόνη για να συνεχίσει. Το ίδιο κάνουν και άλλες εντολές που τυπώνουν πολλαπλές γραμμές, όπως η ΤΜΗΜΑΤΑ

Μπορούμε να γράψουμε απευθείας κείμενο σε μια μεταβλητή
Μεταβλητη\$={Πρώτη γραμμή
Δεύτερη γραμμή
Τρίτη γραμμή
} ' η θέση της αγγύλης στο τέλος ορίζει το μέγεθος του κενού που δεν θα
περιλαμβάνεται στη μεταβλητή
Αναφορά Μεταβλητη\$
Τυπώνει:
Πρώτη γραμμή
Δεύτερη γραμμή
Τρίτη γραμμή

Αναγνωριστικό: ΛΑΤΙΝΙΚΑ [LATIN]
ΛΑΤΙΝΙΚΑ
αλλάζει την κωδικοσελίδα αλλά και τη γλώσσα που εμφανίζονται τα μηνύματα λάθους
ΔΕΣ ΛΑΤΙΝΙΚΑ, ΧΑΡΑΚΤΗΡΕΣ

Αναγνωριστικό: ΟΘΟΝΗ [CLS]
ΟΘΟΝΗ χρωμα, υψος_διαχωρισμού
ΟΘΟΝΗ ,υψος_διαχωρισμού
Το υψος_διαχωρισμού θα είναι κάποια γραμμή με 0 την πρώτη από πάνω.
Από την έκδοση 3 μπορούμε να έχουμε και στο περιθώριο διαχωρισμό χωρίς να επηρεάζουμε
αυτόν της οθόνης.
Ο διαχωρισμός μας δείχνει το ύψος του τμήματος της οθόνης που θα ολισθαίνει προς τα πάνω.
Ολισθαίνει πάντα το κάτω τμήμα. Με ΟΘΟΝΗ ,0 ακυρώνουμε τον διαχωρισμό - ολόκληρη η
οθόνη μπορεί να ολισθαίνει.

Όταν εκτυπώνουμε στον εκτυπωτή τότε η εντολή ΟΘΟΝΗ δεν κάνει τίποτα.
Μπορούμε με την ΚΥΛΙΣΗ να αλλάξουμε το ύψος διαχωρισμού χωρίς να σβήσουμε την οθόνη,
όπως επίσης να δώσουμε εντολή για ΚΥΛΙΣΗ ΑΝΩ ή ΚΥΛΙΣΗ ΚΑΤΩ.

ΔΕΣ ΕΝΤΟΛΕΣ ΚΟΝΣΟΛΑΣ

Αναγνωριστικό: ΠΑΡΑΘΥΡΟ [WINDOW]
Αλλάζουμε το περιθώριο και την οθόνη μαζί ώστε η οθόνη να καλύπτει όλο το περιθώριο
1) Κεντράρουμε αυτόματα

ΠΑΡΑΘΥΡΟ τυπος_οθονης, πλατος, υψος;

2) Το παράθυρο (πρώην περιθώριο) παραμένει στην πάνω αριστερή γωνία 0

ΠΑΡΑΘΥΡΟ τυπος_οθονης, πλατος, υψος

3) Το παράθυρο κεντράρεται και το ύψος δίνεται αυτόματα από τον μεταφραστή

ΠΑΡΑΘΥΡΟ τυπος_οθονης, πλατος

4) Επαναφορά του παραθύρου στις διαστάσεις της επιφάνειας εργασίας.

ΠΑΡΑΘΥΡΟ τυπος_οθονης, 0 ' επιλέγουμε την συσκευή οθόνης 1

ΠΑΡΑΘΥΡΟ τυπος_οθονης, 1 ' επιλέγουμε την συσκευή οθόνης 2

η μεταβλητή ΣΥΣΚΕΥΗ μας δίνει τον αριθμό της συσκευής οθόνης (από 0)

(Υπάρχει μεταβλητή ΠΑΡΑΘΥΡΟ που δίνει τον χειριστή φόρμας ή hwnd)

Για να γράψουμε στο παράθυρο, δηλαδή πίσω από την οθόνη, θα χρησιμοποιήσουμε την εντολή Περιθωριο όπου ανάμεσα σε δυο αγκύλες θα δώσουμε εντολές που θα κατεθούν την εκτύπωση εκεί.

Μπορούμε να μετακινούμε το παραθυρο με την κινήση.π

Μπορούμε να φτιάξουμε έναν στόχο και να μετακινούμε το παράθυρο με το ποντίκι

Παρακάτω έχουμε ένα παράδειγμα που μας δείχνει το πως φτιάχνουμε τρεις στόχους, το έναν στο περιθώριο, τον άλλο στο βασικό επίπεδο και ένα στο επίπεδο 1 (ή παίκτης 1). Ένα νήμα μετακινεί κάθε 4 δευτερόλεπτα τον παίκτη 1. Όλοι οι στόχοι κάνουν απλά ΜΠΙΠ όταν πατηθούν. Στο τμήμα Β δημιουργούμε το Επίπεδο 1 το οποίο έχει αρχικό μέγεθος το μέγεθος του επιπέδου που το δημιούργησε!

Τμήμα Α {

Παραθυρο 12,10000, 8000;

ΕΣ=0

Τυπος 12, Χ.Σημεια-2000, 2000

Οθονη 1

Κινήση 2000,4000

Περιθωριο {

Οθονη 6

Δρομεας 13,4

Στοχος ΕΔ,"ΤΕΛΟΣ",12,1,13,5,5,"ΤΕΛΟΣ"

}

ΕΔ1=0

Δρομεας 13,3

Στοχος ΕΔ1,"ΜΠΙΠ",12,1,13,5,5,"ΠΑΤΑ ΕΔΩ"

Β

Επίπεδο 1 {

ΕΔ2=0

Δρομεας 13,3

Στοχος ΕΔ2,"ΜΠΙΠ",12,1,13,5,5,"ΠΑΤΑ ΕΔΩ"

}

```

Νημα {
    Παικτης 1, Τυχαιος(4000), Τυχαιος(4000)
} Ως Λ
Νημα Λ Καθε 3000
Καθε 100 {
    Σαρωσε 0.3
    Αν Ενκομ$<>" Τότε Εξοδος
}
Νημα Λ Σβησε
Παικτης 0
Στοχοι Νεοι

```

```

}
Τμημα Β {
    Επιπεδο 1 {
        Παραθυρο 12, 7000,1500
        Οθονη 2
        Τυπωσε "ΟΚ"
    }
    Παικτης 1, 100,2000
    Παικτης 1 Δειξε
}

```

```

Αναγνωριστικό: ΠΕΔΙΟ [FIELD]
Α$="ΓΙΩΡΓΟΣ"
ΠΕΔΙΟ 10,10,20 ΩΣ Α$
ΤΥΠΩΣΕ ΠΕΔΙΟ, Α$

```

Σημειώστε ότι το πλήκτρο Insert αλλάζει τον τρόπο παρεμβολής, τον κάνει να "πατάει" ή όχι τα γράμματα προς τα δεξιά. Επίσης με τα άνω και κάτω βελάκια μπορούμε να τερματίσουμε την εισαγωγή. Το αποτέλεσμα του τερματισμού, εκτός από την καταγραφή στην μεταβλητή που δίνουμε είναι και η καταγραφή της αιτίας τερματισμού στην εσωτερική μεταβλητή ΠΕΔΙΟ. Αν μας δώσει -1 σημαίνει έξοδος με το άνω βελάκι, 1 έξοδος με το κάτω βελάκι ή το Enter, 99 έξοδος με το Esc.

Αν θέλουμε δίνουμε την εντολή ΠΕΔΙΟ ΝΕΟ και σβήνουμε το αποτέλεσμα της μεταβλητής ΠΕΔΙΟ. Μεταξύ της εντολής και του ελέγχου της μεταβλητής μπορούμε να κάνουμε έλεγχο του όποιου αποτελέσματος και να αλλάξουμε την κατάσταση.

Έτσι ενώ κανονικά δίνει -1 (πάνω), 1 (κάτω) και 99 το Esc (άκυρο από απόφαση του χρήστη) με την ΠΕΔΙΟ ΝΕΟ δίνουμε τιμή στην ΠΕΔΙΟ το 0 που σημαίνει "άκυρο από έλεγχο"

```

ΠΕΔΙΟ ΣΥΝΘΗΜΑ 1,1, 10 ΩΣ ΣΥΝΘΗΜΑΤΙΚΟ$

```

Αν ζητήσουμε πεδίο από τμήμα έξω από την οθόνη, ή δεν φθάνει ο χώρος για να πάρει η γλώσσα ένα πεδίο, τότε ανοίγει παράθυρο για εισαγωγή τιμής όπως το παράδειγμα παρακάτω: (τότε δεν δουλεύει το PASSWORD, για τα PASSWORD καλύτερα να ετοιμάζουμε ΦΟΡΜΑ)

A\$=Πεδίο\$("1213", 10)

Πεδίο Πλατος+10, 1,10 Ως A\$

Τυπωσε @(0,11); A\$

Νέοι αριθμοί για έξοδο:

-20 Home

20 End

-10 pageup

10 pagedown

1 Tab

-1 Shift Tab

Αναγνωριστικό: ΠΕΝΑ [PEN]

ΠΕΝΑ χρώμα

Δίνει το χρώμα των γραμμάτων και των γραμμών

ΤΥΠΩΣΕ ΠΕΝΑ

μας δίνει ή ένα αρνητικό νούμερο ή αν υπάρχει αντιστοιχεί ένα θετικό από 0 ως 15.

Μπορούμε να δώσουμε 0,1,2 ... 15 για τα βασικά χρώματα των Windows, όπως και αρνητικές τιμές για τα 16 εκατομμύρια χρώματα που μπορούν να απεικονισθούν (24 bit).

ΤΥΠΩΣΕ ΧΡΩΜΑΤΑ

μας δίνει τον αριθμό χρωμάτων της οθόνης.

ΤΥΠΩΣΕ ΧΡΩΜΑ(128,0,255)

μας δίνει το χρώμα με κόκκινο 128, πράσινο 0 και μπλε 255

ΠΕΝΑ ΧΡΩΜΑ(128,0,255)

ΘΕΣΗ 3000,3000 : ΤΥΠΩΣΕ ΣΗΜΕΙΟ

Μας τυπώνει το χρώμα στην θέση 3000,3000 σε αρνητική τιμή.

τυπωσε -σημείο ' σε δεκαδικό σύστημα

τυπωσε β16\$(-σημείο,3) ' σε δεκαεξαδικό με 6 ψηφία, 2 για κάθε byte)

Δες την εντολή ΕΠΕΛΕΞΕ.ΧΡΩΜΑ η οποία σου δίνει το σπάντα τρόπο επιλογής χρωμάτων.

Αναγνωριστικό: ΠΕΡΙΘΩΡΙΟ [BACKGROUND]

Μπορούμε να γράψουμε και να σχεδιάσουμε στο περιθώριο (πλαίσιο πίσω από την ΟΘΟΝΗ). Το περιθώριο είναι η φόρμα που κρατάει όλα τα άλλα επίπεδα και λειτουργεί και αυτό ως επίπεδο, δηλαδή με την εντολή ΠΑΡΑΘΥΡΟ μπορεί να πάρει συγκεκριμένο πλάτος και ύψος και μέγεθος γραμμών, ενώ μπορούμε να χρησιμοποιήσουμε και την ΦΟΡΜΑ για να ορίσουμε ανάλυση οθόνης σε χαρακτήρες ώστε να υπολογιστούν αυτόματα το μέγεθος και οι διαστάσεις του περιθωρίου.

Το περιθώριο μπορεί να κοπεί από τη ΦΟΡΜΑ ή μπορεί να ξαναμεγαλώσει με αυτήν. Επίσης μπορεί να έχει αντίγραφο της εικόνας της επιφάνειας εργασίας με την εντολή ΕΠΙΦΑΝΕΙΑ Μπορούμε να γράψουμε και να σχεδιάσουμε στο περιθώριο (πλαίσιο πίσω από την ΟΘΟΝΗ).

1)

```
ΠΕΡΙΘΩΡΙΟ {  
    φοντο 1,5  
}
```

2) Μπορούμε να ορίσουμε το μέγεθος γραμμών εδώ 20

```
ΠΕΡΙΘΩΡΙΟ 20 {  
    φοντο 1,5  
}
```

Αναγνωριστικό: ΠΕΡΙΘΩΡΙΟ _ [BACK]

Μπορούμε να γράψουμε και να σχεδιάσουμε στο περιθώριο (πλαίσιο πίσω από την ΟΘΟΝΗ). Το περιθώριο είναι η φόρμα που κρατάει όλα τα άλλα επίπεδα και λειτουργεί και αυτό ως επίπεδο, δηλαδή με την εντολή ΠΑΡΑΘΥΡΟ μπορεί να πάρει συγκεκριμένο πλάτος και ύψος και μέγεθος γραμμών, ενώ μπορούμε να χρησιμοποιήσουμε και την ΦΟΡΜΑ για να ορίσουμε ανάλυση οθόνης σε χαρακτήρες ώστε να υπολογιστούν αυτόματα το μέγεθος και οι διαστάσεις του περιθωρίου.

Το περιθώριο μπορεί να κοπεί από τη ΦΟΡΜΑ ή μπορεί να ξαναμεγαλώσει με αυτήν. Επίσης μπορεί να έχει αντίγραφο της εικόνας της επιφάνειας εργασίας με την εντολή ΕΠΙΦΑΝΕΙΑ

1)

```
ΠΕΡΙΘΩΡΙΟ {  
    φοντο 1,5  
}
```

2) Μπορούμε να ορίσουμε το μέγεθος γραμμών εδώ 20

```
ΠΕΡΙΘΩΡΙΟ 20 {  
    φοντο 1,5  
}
```

Αναγνωριστικό: ΠΛΑΓΙΑ [ITALIC]

ΠΛΑΓΙΑ : ΤΥΠΩΣΕ "ΑΥΤΑ ΕΙΝΑΙ ΓΡΑΜΜΑΤΑ": ΠΛΑΓΙΑ

κάθε φορά που χρησιμοποιούμε την εντολή ΠΛΑΓΙΑ αυτή αλλάζει τα γράμματα από όρθια σε πλάγια και πάει λέγοντας...

ΑΛΛΟΣ ΤΡΟΠΟΣ

ΠΛΑΓΙΑ 1 - κάνει τα γράμματα πλάγια

ΠΛΑΓΙΑ 0 - κάνει τα γράμματα ορθά

Αναγνωριστικό: ΠΛΑΙΣΙΟ [FRAME]

Η εντολή αυτή μας δίνει έναν τρόπο για να σχεδιάσουμε πλαίσια με συντεταγμένες χαρακτήρων.

Η πάνω αριστερή γωνία του πλαισίου θα είναι αυτή του δρομέα πριν την εκτέλεση της εντολής.

1) Πλαίσια με γραμμή περιγράμματος και φόντο

ΠΛΑΙΣΙΟ πλάτος, ύψος, χρώμα_φοντου, χρώμα_πλαισιου.

Το πλάτος και το ύψος σε χαρακτήρες.

Ο δρομέας παραμένει στην θέση του.

Αν δεν δώσουμε χρώμα πλαισίου τότε αυτό θα είναι το χρώμα πένας.

2) Πλαίσια σε στυλ εντολής ΒΑΨΕ@ με απόλυτες συντεταγμένες

ΠΛΑΙΣΙΟ @ απόλυτηθεσηX, απόλυτηθεσηY, τυπος, παραμετρος1, παραμετρος2

3) Πλαίσια σε στυλ εντολής ΒΑΨΕ@ με σχετικές συντεταγμένες

ΠΛΑΙΣΙΟ @ (πλάτος), (ύψος), τυπος, παραμετρος1, παραμετρος2

Τα τρία τελευταία στοιχεία είναι ίδια με αυτά της ΒΑΨΕ @ (η διαφορά των δύο εντολών είναι ότι η ΒΑΨΕ@ λογαριάζει τις διαστάσεις σε twips και κάνει χρήση του δρομέα γραφικών)

Αναγνωριστικό: ΣΒΗΣΕ [HIDE]

ΣΒΗΣΕ

Κρύβει την ΟΘΟΝΗ (αλλά όχι το περιθώριο ή την επιφάνεια εργασίας)

Θα ανοίξει αυτόματα αν κάπου στο πρόγραμμά μας ζητάμε εισαγωγή από πληκτρολόγιο με παράλληλη εμφάνιση στην οθόνη.

Με τις ΑΝΑΨΕ & ΣΒΗΣΕ μπορούμε να δείχνουμε κάτι στο περιθώριο ενώ φτιάχνουμε την οθόνη μας κρυφά!

Σε συνδυασμό με την ΚΙΝΗΣΗ μπορούμε να δημιουργήσουμε ωραία εφέ παρουσιάσεων (π.χ. ολίσθηση από δεξιά ή από πάνω προς τα κάτω)

Αναγνωριστικό: ΣΗΜΑΔΙ [MARK]

ΣΗΜΑΔΙ πλάτος_σε_χαρακτηρες, υψος_σε_χαρακτηρες, χρώμα, ελλειψη_η_κυκλος

ελλειψη_η_κυκλος = 1 ή 0 (Η έλλειψη θα φανεί όταν δώσουμε πλάτος διάφορο από το ύψος)
Ο δρομέας χαρακτήρων δεν αλλάζει και έτσι μπορούμε να γράψουμε πάνω από το σημάδι γράμματα.

Παράδειγμα
ΤΥΠΩΣΕ " ";
ΣΗΜΑΔΙ 1,1,14
ΤΥΠΩΣΕ " ";
ΣΗΜΑΔΙ 8,1,5,1
ΤΥΠΩΣΕ " ΣΗΜΑΔΙ"

Αναγνωριστικό: ΤΟΠΙΚΟ [LOCALE]

1) ΤΟΠΙΚΟ κωδικός

Βάζει και το αλφαριθμητικό για τους λογικούς (γνωρίζει τα ελληνικά και τα αγγλικά)

2) ΤΟΠΙΚΟ ";\N\α\ί;\O\χ\ι" \\\ μπορούμε να δώσουμε δικά μας ονόματα

3) ΤΟΠΙΚΟ "" \\\ ακυρώνουμε όποια επιλογή και θέτουμε το βασικό αλφαριθμητικό (δίνει το αληθές ως True)

Το περιβάλλον δουλεύει με Unicode χαρακτήρες στην οθόνη, στα αλφαριθμητικά, στον διορθωτή, στις βάσεις δεδομένων και σε αρχεία που δηλώνονται με το αναγνωριστικό ΕΥΡΙΑ, αλλά μπορεί να χειριστεί αλφαριθμητικά όπου οι χαρακτήρες του ανήκουν σε κάποιο ΤΟΠΙΚΟ σύστημα, όπως το 1032 που είναι για την Ελλάδα.

κάθε Τοπικό αλλάζει την κωδικοσελίδα και το σύνολο χαρακτήρων, για τις συναρτήσεις που τις χρησιμοποιούν. Το περιβάλλον θα συνεχίσει να λειτουργεί με όλους τους χαρακτήρες του Unicode συνόλου. Πρόβλημα θα υπάρξει αν θέλουμε να εμφανίσουμε έναν χαρακτήρα που δεν περιέχεται στην γραμματοσειρά που χρησιμοποιούμε. Το οποίο θα εμφανίζει τον ζητούμενο χαρακτήρα ως ένα τετράγωνο

Για να χρησιμοποιήσουμε διάφορες γλώσσες δεν χρειαζόμαστε την εντολή ΤΟΠΙΚΟ. Αυτό που αλλάζει αυτή η εντολή και φαίνεται καθαρά στο παρακάτω παράδειγμα είναι οι χαρακτήρες που αντιπροσωπεύουν τα νούμερα από το 33 μέχρι το 255 αν και πολλά από αυτά είναι ίδια σε όλα τα σύνολα χαρακτήρων.

Για παράδειγμα μπορούμε να δούμε το σύνολο χαρακτήρων της Τουρκικής γλώσσας:

Τοπικό 1055 ' μπορούμε να βάλουμε τόνους στις εντολές που είναι στα Ελληνικά.

Για I=33 Εως 255 {

Τυπωσε Χαρ\$(I);

}

Τυπωσε ' αλλαγή γραμμής

Δοκιμάστε με το 2077 για Φιλανδία, 1037 για Ισραήλ.
Ουσιαστικά όλοι οι χαρακτήρες μπορούν να τυπωθούν και ανακατεμένοι
Τυπωσε χαρ\$(254,2077), χαρ\$(249,1037)

Αν προσθέσουμε γλώσσες στο πληκτρολόγιο μπορούμε να γράφουμε με οποιαδήποτε και με οποιοδήποτε πληκτρολόγιο.

Αναγνωριστικό: ΤΥΠΟΣ [MODE]

1) Ορίζουμε μέγεθος γραμμάτων, πλάτος σε twips και ύψος σε twips

ΤΥΠΟΣ τυπος_οθονης, πλάτος, υψος

2) Με αυτόματο υπολογισμό ύψους (θα έχει την αναλογία ύψους προς πλάτος με την επιφάνεια εργασίας)

ΤΥΠΟΣ τυπος_οθονης,2 πλάτος

3) Αλλάζουμε μόνο το μέγεθος των γραμμάτων (άρα αλλάζει και ο αριθμός γραμμών και πόσα γράμματα μπορούν να τυπωθούν)

ΤΥΠΟΣ τυπος_οθονης

Παρατήρηση

Ο τύπος οθονης είναι το μέγεθος χαρακτήρων. Π.χ. ΤΥΠΟΣ 16 σημαίνει γράμματα 16 στιγμών. Ανάλογα το μέγεθος γραμμάτων αλλάζει και το ΠΛΑΤΟΣ και ΥΨΟΣ που μας δείχνουν πόσους χαρακτήρες χωρούν στην γραμμή και στην στήλη αντίστοιχα.

Οι μεταβλητές μόμο ανάγνωσης κλιμαξ.χ και κλιμαξ.υ μας δείχνουν το μέγεθος της οθόνης μας σε twips

Μπορούμε να φτιάξουμε οθόνη μεγαλύτερη από το περιθώριο. Όμως δεν θα είναι όλη ορατή.

Μπορούμε να την μετακινούμε σε οποιαδήποτε διεύθυνση με την εντολή ΚΙΝΗΣΗ

Με την ΠΑΡΑΘΥΡΟ τυπος_οθονης,0 επαναφέρουμε την οθόνη στο πλήρες μέγεθος όπου όλη να είναι στο μέγιστο ορατή.

Στην έκδοση έξι μπορούμε να έχουμε 32 επίπεδα σαν οθόνες. Αυτά τα 32 επίπεδα είναι οι Παικτες. Κάθε παίκτης μπορεί να πάρει εικόνα και να δηλώσουμε ποιο χρώμα θα είναι το διάφανο καθώς και ποια άλλα κοντινά χρώματα θα είναι επίσης διάφανα, όπως επίσης και τι μέγεθος θα έχει η εικόνα (το μέγεθος του παίκτη σε σχέση με την αρχική εικόνα). Αν θέλουμε όμως να χρησιμοποιήσουμε τον Παικτη σαν Οθόνη, και δεν θέλουμε να έχει διάφανα σημεία και δεν θέλουμε εικόνα για φόντο, τότε καλούμε την ΕΠΙΠΕΔΟ με τον αριθμό του επιπέδου που θέλουμε και δημιουργούμε ένα επίπεδο όμοιο με την οθόνη ή το επίπεδο απ' όπου κλήθηκε η εντολή. Εδώ λοιπόν η Τυπος δεν αλλάζει το μέγεθος σε υψος και πλατος του επιπέδου (αν δώσουμε μέγεθος απλά θα το αγνοήσει) παρά μόνο το μέγεθος γραμμάτων. Προσοχή εδώ να μην δώσουμε τέτοιο μέγεθος που δεν θα χωράει ένα γράμμα στο επίπεδο! Μπορούμε να

δώσουμε την ΠΑΡΑΘΥΡΟ με μέγεθος γραμμάτων και διαστάσεις του επιπέδου. Μπορούμε να αλλάξουμε την γραμματοσειρά με την εντολή ΓΡΑΜΜΑΤΟΣΕΙΡΑ.

Η Τυπος και η εντολή Παραθυρο απενεργοποιούν τους στόχους στο επίπεδο όταν τις καλούμε. Οι στόχοι σβήνουν ή με την εντολή ΣΤΟΧΟΙ ΝΕΟΙ ή με την εντολή ΑΡΧΗ
Δες ΣΤΟΧΟΣ

Αναγνωριστικό: ΦΑΡΔΙΑ [BOLD]

ΦΑΡΔΙΑ : ΤΥΠΩΣΕ "ΑΥΤΑ ΕΙΝΑΙ ΓΡΑΜΜΑΤΑ": ΦΑΡΔΙΑ

κάθε φορά που χρησιμοποιούμε την εντολή ΦΑΡΔΙΑ αυτή αλλάζει το πάχος από φαρδύ σε λεπτό και πάει λέγοντας...

ΑΛΛΟΣ ΤΡΟΠΟΣ

ΦΑΡΔΙΑ 1 - κάνει τα γράμματα φαρδιά

ΦΑΡΔΙΑ 0 - κάνει τα γράμματα λεπτά

Αναγνωριστικό: ΦΟΝΤΟ [GRADIENT]

Μπορούμε να γεμίσουμε την οθόνη ή το περιθώριο ή την σελίδα του εκτυπωτή με ένα φόντο ενός ή δυο χρωμάτων.

ΦΟΝΤΟ χρωμα1

ΦΟΝΤΟ χρωμα1, χρωμα2

ΦΟΝΤΟ χρωμα1, χρωμα2, οριζόντια

οριζόντιο= 0 ή 1

ΠΑΡΑΔΕΙΓΜΑ

ΤΜΗΜΑ Α {

τυπος 12, 8000

περιθωριο {

επιφανεια

χρωμα ! {φοντο 1,2}

}

}

Αναγνωριστικό: ΦΟΡΜΑ [FORM]

1) ΦΟΡΜΑ 10

η εντολή αυτή κάνει και για τον εκτυπωτή και για τα επίπεδα.

2) ΦΟΡΜΑ 80,25

η εντολή αυτή κάνει και για τα επίπεδα (αλλά δεν αλλάζει την απόσταση γραμμής αυτόματα)

3) ΦΟΡΜΑ

4) ΦΟΡΜΑ ;

5) ΦΟΡΜΑ 10,25;

η εντολή ΦΟΡΜΑ αλλάζει τον τυπο των γραμμάτων (το μέγεθος) της Οθόνης ώστε να πετύχει το πλάτος σε αριθμό χαρακτήρων που θέλουμε ή και το ύψος σε αριθμό χαρακτήρων, στο παράθυρο που έχουμε ορίσει.

Το πρόβλημα εδώ είναι ότι αν δώσουμε πολύ μεγάλο νούμερο απλά δεν θα βλέπουμε γράμματα!

Αν δεν δώσουμε ύψος τότε δίνεται αυτόματα το μέγιστο που μπορεί να δώσει.

Μια φορμα 20 και μια φορμα 20,80 θα δώσουν δυο διαφορετικές οθόνες που θα έχουν όμως ίδιο πλάτος σε χαρακτήρες.

Η τρίτη περίπτωση κάνει το σύστημα να κόψει το περιθώριο γύρω από την οθόνη

Η τέταρτη περίπτωση κάνει το σύστημα να επαναφέρει το περιθώριο γύρω από την οθόνη (καλύπτει δηλαδή όλη την επιφάνεια εργασίας).

Η πέμπτη κάνει ότι η 2 και η 4 μαζί και επιπλέον βάζει Διαστιχο 0

Στις 1,2,5 το ! μετά το αναγνωριστικό Φόρμα αφήνει ένα μικρότερο πλαίσιο για να μην καλύπτεται σε οθόνες όπου ένα μέρος της οθόνης κρύβεται (αυτό γίνεται σσε τηλεοράσεις).

Δοκιμάστε το Φόρμα 60,32 και το Φόρμα ! 60,32

Αν θέλουμε να φτιάξουμε μια οθόνη σαν στήλη, με 20 γράμματα στη γραμμή και 60 γραμμές χαρακτήρων τότε:

Φορμα ;

Φορμα 20,60

Φορμα

' τώρα θα την στείλω αριστερά, θα κόψω είκοσι γραμμές και θα την τοποθετήσω πάνω αριστερά κίνηση.π 0

φορμα 20,40

κίνηση 0,0

φορμα

Αναγνωριστικό: ΧΑΡΑΚΤΗΡΕΣ [CHARSET]

ΧΑΡΑΚΤΗΡΕΣ 161 (όμοιο με εντολή ΕΛΛΗΝΙΚΑ)

ΧΑΡΑΚΤΗΡΕΣ 0 (όμοιο με εντολή ΛΑΤΙΝΙΚΑ)

ΤΥΠΩΣΕ ΧΑΡΑΚΤΗΡΕΣ

161

Μπορούμε να ορίσουμε κωδικοσελίδα για οθόνη, περιθώριο, επίπεδα και εκτυπωτή.

ΑΛΛΑΖΕΙ ΤΗΝ ΚΩΔΙΚΟΣΕΛΙΔΑ ΣΤΗΝ ΕΚΤΥΠΩΣΗ ΣΤΗΝ ΟΘΟΝΗ (Η ΣΤΟΝ ΕΚΤΥΠΩΤΗ ΑΝΑΛΟΓΑ)

ΔΕΣ ΕΛΛΗΝΙΚΑ, ΛΑΤΙΝΙΚΑ

Από τις ΡΥΘΜΙΣΕΙΣ μπορούμε να αλλάξουμε την από εκκίνηση τιμή της κωδικοσελίδας

Ομάδα: ΟΘΟΝΗ ΚΑΙ ΑΡΧΕΙΑ - SCREEN AND FILES

Αναγνωριστικό: ΔΕΚΑΕΞ [HEX]

ΔΕΚΑΕΞ 10, 0X10+0XAF00

δουλεύει όπως η ΤΥΠΩΣΕ αλλά για εμφανίζει δεκαεξαδικά νούμερα (0 ή θετικό αριθμό). Δίνει -??? ή +??? αν είναι εκτός ορίων

Αναγνωριστικό: ΕΙΣΑΓΩΓΗ [INPUT]

1)ΕΙΣΑΓΩΓΗ Α\$

2)ΕΙΣΑΓΩΓΗ "X=",A

με ; ανάμεσα σε δυο αριθμητικές μεταβλητές ζητάει κόμμα ή αλλαγή γραμμής και η επιγραφή εμφανίζεται μόνο την πρώτη φορά.

Στο τέλος ενα ";" δουλεύει όπως στην Τύπωσε, κολάει τον δρομέα στην άκρη (δεν αλλάζει γραμμή).

3)ΕΙΣΑΓΩΓΗ #A, A\$,B,K\$

εισάγει από αρχείο σε UNICODE ή ANSI ανάλογα με την εντολή ΑΝΟΙΞΕ που έχουμε δώσει.

Παράδειγμα εξαγωγής csv αρχείου. Βγάλτε το Ευρεία για εξαγωγή/εισαγωγή με ένα Ψηφίο (byte) ανά χαρακτήρα.

Γράψε με χαρ\$(9), ",", αληθές 'αληθές σημαίνει ότι τα αλφαριθμητικά θα κωδικοποιούνται κατά json

Γράψε με χαρ\$(9), ",", αληθές [, αληθές] ' το δεύτερο αληθές (μη μηδεν))σημαίνει να μην μπουν εισαγωγικά στο αλφαριθμητικά.

Άνοιξε "my.dat" για ευρεία εξαγωγή ως #κ

Γράψε #κ, "Row 1", -213231313132312.23123@, -52322323.32#, -1.2121e+10

Γράψε #κ, {Line1

Line2

Line3

}

Κλείσε #κ

\\ κατά json σημαίνει ότι ειδικοί χαρακτήρες κωδικοποιούνται ώστε να γράφεται

\\ το αλφαριθμητικό με πολλές γραμμές χωρίς να έχει μέσα του αλλαγές γραμμών

\\ και άλλοι χαρακτήρες κωδικοποιούνται αλλά εμάς μας ενδιαφέρει ότι βάζουμε

\\ ένα κείμενο σε μια γραμμή!

Εισαγωγή με char\$(9), ",", αληθές 'αληθές σημαίνει ότι τα αλφαριθμητικά θα διαβάζονται κατά json

Άνοιξε "my.dat" για ευρεία Εισαγωγή ως #κ

Εισαγωγή #κ, R\$, M1=0@, Z1=0#, check

Εισαγωγή #κ, Txt\$

Κλείσε #κ

Τύπωσε R\$, M1, Z1, check

Τύπωσε Τύπος\$(M1), Τύπος\$(Z1)

Τύπωσε M1

Τύπωσε Z1

Αναφορά txt\$ ' η αναφορά τυπώνει κείμενα

\\ αν το κείμενο έχει πολλές γραμμές σταματάει...

\\ για να μας δείξει μέρος και συνεχίζει με διάστημα, ή με πλήκτρο ποντικιού.

4) ΕΙΣΑΓΩΓΗ ! μεταβλητη\$, πλατος [Μήκος=20] εξ ορισμού το μέγιστο μήκος για τη μεταβλητή είναι το 50

4.1) ΕΙΣΑΓΩΓΗ ! μεταβλητη, πλατος \\ μόνο για αριθμό - όχι για τύπου Long

4.2) ΕΙΣΑΓΩΓΗ ! μεταβλητη%, πλατος \\ μόνο για αριθμό ακέραιο, όχι τύπου Long. Πέρνει μέχρι 16 ψηφία.

με Esc επιστρέφει η παλιά τιμή. Διαβάζει και στοιχεία πινάκων. Τι απλές μεταβλητές τις δημιουργεί αν δεν υπάρχουν

η μεταβλητή μόνο ανάγνωσης Πεδίο δίνει 99 για το Esc, -1 για το πάνω βελάκι και 1 για το κάτω και το enter.

με χρήση του Θέσε Διακόπτες "+INP" το Πεδίο γυρίζει 13 αν πατήσουμε το Enter (κανονικά γυρίζει 1)

5) ΕΙΣΑΓΩΓΗ ! μεταβλητη\$, πλατος, υψος [, Τίτλος\$]

5.1) ΕΙΣΑΓΩΓΗ ! μεταβλητη\$, πλατος, υψος, Τίτλος\$, θέση δρομέα.

Η θέση δρομέα όσο το μήκος συν ένα πάει το δρομέα στο τέλος.

Και οι δυο παραλλαγές μας δίνουν έναν τρόπο να εισάγουμε κείμενο σε μια μεταβλητή.

Επιπλέον από την έκδοση 3 έχει προστεθεί και η παρακάτω αντικατάσταση:

A\$={Πρώτη γραμμή κειμένου

Δεύτερη γραμμή κειμένου

...

...sdsdsds

}

Μια ΤΥΠΩΣΕ A\$ δεν τυπώνει σωστά το κείμενο, διότι δεν λογαριάζει τις αλλαγές γραμμών που έχουν καταχωρηθεί στο A\$. Πρέπει να χρησιμοποιηθεί η ΕΠΙΓΡΑΦΗ ! ή η Αναφορά.

Αναγνωριστικό: ΕΛ? [?]
Δες Τύπωσε

Αναγνωριστικό: ΤΥΠΩΣΕ [PRINT]

1) Μπορούμε για ευκολία να βάζουμε το ? αντί για Τυπωσε ή Print
? 2+3=5 και 2-1>0

2) Τύπωσε 1, 2+4, "Λέξη "+α\$ \ \ αν δεν χωράνε στη γραμμή συνεχίζουν από κάτω κατά στήλες

3) Τύπωσε "Όνομα: ";α\$; " ";β\$

4) Τύπωσε @(10,10),"Στη θέση 10, 10"

5) Τύπωσε \$(3,8),100,200,"Λέξη",300 \ \ ορίζουμε στήλη 8 χαρακτήρων και στοίχιση δεξιά

6) Τύπωσε \ \ σκέτο αλλάζει γραμμή, στη τελευταία κάτω γραμμή η αλλαγή προκαλεί ολίσθηση γραμμών προς τα πάνω

7) Δεκαεξ 0xFF00+0x0055 \ \ Η Τύπωσε σε εξαγωγή δεκαεξαδικού
0xFF55

8) Τύπωσε #A, "Μια γραμμή στο αρχείο με μεταβλητή χειρισμού το A"

Η Τύπωσε #A απλά βάζει δυο χαρακτήρες για αλλαγή γραμμής

Η Τύπωσε #A,123,456 βάζει το 123456

(δεν βάζει διαχωριστικό χαρακτήρα, μπορούμε να βάλουμε , " ",)

Η Γράψε βάζει διαχωριστικά

Ο χαρακτήρας ; μπορεί να μπει, και αν είναι τελευταίος τότε δεν μπαίνει αλλαγή γραμμής.

Οι αριθμοί γράφονται βάσει της επιλογής "+DEC" (με τελεία για δεκαδικά) ή "-DEC" (με ότι έχει το σύστημα ως χαρακτήρα δεκαδικών)

Μπορούμε να το αλλάξουμε με την Τοπικό 1032 (ελληνικά) ή άλλο νούμερο πχ Τοπικό 1033 (αγγλικά), μέχρι όμως να το δώσουμε ξανά ή να δώσουμε την Θέσε Διακόπτες "+DEC" (ή "-DEC")

9) Τύπωσε \$("#,###,###.00", 20), 112313.3788 \ \ 20 το πλάτος στήλης

\ \ το ίδιο δίνει και αυτό: Τύπωσε \$("#,###,###.00", 20), τιμη(μορφή\$("{0:2}", 112313.3788))

10) Τύπωσε μορφή\$("αριθμός με δυο δεκαδικά:{0:2} και εδώ μια λέξη:{1}", 10.678, "OK")

\ \ η μορφή\$() είναι γενική συνάρτηση, ο αριθμός μετά το :
δείχνει το τελευταίο δεκαδικό αν υπάρχει

νέες προσθήκες στην έκδοση 8

Όλες οι αλλαγές χρωμάτων και στηλών και τρόπου εκτύπωσης περιορίζονται στην εντολή

Εξ ορισμού οι "φόντο", "πάνω", "υπό", "μέρος" τυπώνουν αναλογικά, το \$(4)

11) Τύπωσε πάνω \$(2), "Στη μέση της οθόνης" \ \ εξ ορισμού σβήνει όλη τη γραμμή χαρακτήρων (φόντο) εκτός από μια λεπτή γραμμή

\ \ δεν αλλάζει γραμμή

\ \ από κάτω και τυπώνει σε μια στήλη

όσο το πλάτος της οθόνης

12) Τύπωσε υπό ~(#ff0066),"κάτω από μια γραμμή" \ \ εξ ορισμού γράφει μια γραμμή λεπτή και αλλάζει γραμμή χαρακτήρων

\\ εδώ με χρώμα γραμμάτων ~()

13) Τύπωσε μέρος "Χωρίς αλλαγή γραμμής" \\ δεν αλλάζει γραμμή, δεν σβήνει το φόντο.

14) Τύπωσε Φόντο 1,2,3,4,5 \\ τυπώνει αφού καθαρίσει τη γραμμή με το χρώμα φόντου (οθόνης), ίσως καθαρίσει και άλλες γραμμές αν χρειαστεί.

15) Τύπωσε #-2, μορφή\$("Hello\n\there\t100\t500\6000")

Με το πρόγραμμα παρακάτω δοκιμάζουμε τι θα πήγαινε σε αρχείο, στην οθόνη, χωρίς να δώσουμε όνομα

Άνοιξε "" Για Εξαγωγή ως #F ' το F θα γίνει -2

Τύπωσε #F, μορφή\$("Hello\n\there\t100\t500\6000")

Κλείσε #F

Δες τα @(), \$(), ~()

Πολύπλοκη εντολή. Μπορεί να γράφει νούμερα, γράμματα, πλαίσια, εικόνες, σχέδια, να αλλάζει πλάτος στηλών, να διαμορφώνει την εκτύπωση των αριθμών, των ημερομηνιών κ.λ.π. Προσοχή η Τυπωσε δεν σβήνει το χώρο που τυπώνει, δηλαδή εμφανίζει το γράμμα και το κενό χώρο του γράμματος μπορεί να φαίνεται το φόντο. Για το λόγο αυτό όταν θέλουμε να ξαναγράψουμε γράμματα σε μια περιοχή στην οθόνη θα πρέπει με την @(να βάψουμε το φόντο, ή να βάλουμε μια εικόνα. Η οθόνη που γράφουμε στη τελευταία γραμμή αν περάσουμε την κάτω δεξιά άκρη τυπώνοντας θα κάνει ολίσθηση και θα μας φέρει μια κενή γραμμή (εκεί δεν χρειάζεται να προετοιμάσουμε το φόντο). Μπορούμε με την εντολή Οθόνη να ορίσουμε το χρώμα του φόντου και την γραμμή από την οποία θα χάνονται οι γραμμές, με συνέπεια αυτό το τελευταίο να μας δίνει κάποιες γραμμές που δεν θα αλλάζει η ολίσθηση (αυτό λέγεται split screen)

Από την έκδοση 6 μπήκε και η αναλογική γραφή, δηλαδή με χρήση ως οδηγούς τις θέσεις των χαρακτήρων στην μη αναλογική γραφή (όπου το κάθε γράμμα είναι κάτω απο το άλλο, ενώ στην αναλογική δεν μπορεί να συμβεί αυτό αφού το κάθε γράμμα έχει άλλο πλάτος) μπορούμε να τοποθετούμε με την στοίχιση που θέλουμε επιγραφές. Στις παλαιότερες εκδόσεις έπρεπε να χρησιμοποιήσει κανείς την ΕΠΙΓΡΑΦΗ και δεν μπορούσε να στοιχίσει κατάλληλα.

Η αναλογική γραφή μπαίνει με τον κατάλληλο κωδικό σε μια \$() ή άμεσα με της Τύπωσε πάνω|υπό|μέρος

ΤΥΠΩΣΕ 12*3+4*5, "ΚΑΛΗΜΕΡΑ"

ΟΘΟΝΗ ΧΡΩΜΑ(128,100,50)

ΤΥΠΩΣΕ \$(,10),1,2,3

ΤΥΠΩΣΕ \$(,20),1,2,3

ΤΥΠΩΣΕ -(128+100*2**16+50*2**16)

ΤΥΠΩΣΕ 1,2,3,4,5,,6,7,8,9,10

ΤΥΠΩΣΕ "ΜΠΡΑΒΟ ΜΟΥ"

ΤΥΠΩΣΕ "3 στο τετράγωνο =" ; 3**2

A=13

ΤΥΠΩΣΕ A, A*10

ΤΥΠΩΣΕ @(10,10),"ΕΔΩ"

ΤΥΠΩΣΕ @(1,1,K-1,9,B\$+"para1031.jpg");

Δες και τους 9 διαφορετικούς τρόπους τυπώματος (παρατήρησε τις διαφορές αριθμών και αλφαριθμητικών)

```
FOR I=0 TO 8 {  
PRINT $(I,14),145355,"George Karras", "The Best"  
}  
print  
print  
print $(0,1), string$("1234567890",5)  
print $(4,8), 321321,"asdaddsd xc c xcv v xv 12 sdfdsfd 1 s srsr","aasasa"
```

Δες αναλυτικό πίνακα για τις ενσωματωμένες λειτουργίες @() και \$()

```
Οθονη  
Ανανεωση 1000 ' το 25 είναι το κανονικό.  
τυπωσε $(,4);  
Για i=1 εως 1000 {  
τυπωσε i,  
'το κόμμα λέει να τυπώσει μετά στο επόμενο πεδίο ή στήλη εκτύπωσης  
}
```

δοκιμάστε χωρίς το , και επίσης χωρίς το Οθονη (το σβήσιμο της οθονης)

Εκτός από την Τυπωσε η 6 έκδοση έφερε και την Αναφορα μια εντολή που τυπώνει αναλογικά για κείμενα, και με την οποία μπορούμε να φτιάχνουμε στείλες, να ορίζουμε τον τρόπο φορμαρίσματος, και να τυπώνουμε συγκεκριμένες γραμμές συγκεκριμένου πλάτους από ένα όριο αριστερά του κειμένου που μπορούμε να καθορίσουμε. Επίσης αυτή η εντολή διαχειρίζεται αυτόματα την κράτηση της οθόνης για να διαβάζουμε το κείμενο. Μπορεί να χρησιμοποιηθεί για να τυπώνει κείμενα σε εκτυπωτή με αυτόματο γύρισμα λέξεων word wrap (εκεί δεν λειτουργεί η κράτηση οθόνης).

Ομάδα: ΧΕΙΡΙΣΤΕΣ ΤΗΣ ΤΥΠΩΣΕ - OPERATORS IN PRINT

Αναγνωριστικό: ΕΛ\$ [\$(]

Η \$() είναι εσωτερική συνάρτηση της ΤΥΠΩΣΕ.

ΤΥΠΩΣΕ

\$(στοίχιση)

\$(στοίχιση, πλάτος στήλης)

\$(φόρμα\$)

\$(φόρμα\$, πλάτος στήλης)

στοίχιση αποτελέσματος αριθμητικών και λογικών παραστάσεων:

0 = δεξιά, 1 = αριστερά, 2 = κέντρο, 3 = δεξιά - με σταθερά διαστήματα (όπως οι παλιές γραφομηχανές)

4 = δεξιά, 5 = αριστερά, 6 = κέντρο, 7=δεξιά ,8 = δεξιά και αριστερά μαζί - με αναλογικά διαστήματα (όπως στις εφημερίδες)

στοίχιση αποτελέσματος αλφαριθμητικών παραστάσεων:

0 = αριστερά, 1 = αριστερά, 2 = κέντρο, 3 = δεξιά με σταθερά διαστήματα (όπως οι παλιές γραφομηχανές)

4 = αριστερά, 5 = αριστερά, 6 = κέντρο, 7 = δεξιά , 8 = αριστερά με αναλογικά διαστήματα (όπως στις εφημερίδες)

στα 0 και 4 τα αλφαριθμητικά γεμίζουν όσες στήλες χρειαστούν. Σε όλα τα άλλα γίνεται αποκοπή.

Παράδειγμα - σε αυτό η αλλαγή γραμμής γίνεται με την ΔΡΟΜΕΑΣ ή CURSOR όπου δεν γίνεται ανανέωση στην οθόνη, οπότε εμφανίζεται σε κάθε πάτημα πλήκτρου όλη η αλλαγή των στοιχείων στην οθόνη.

Γράφουμε edit a και κάνουμε επικόλληση το παρακάτω και μετά esc για να επιστρέψουμε στην γραμμή εντολών και δίνουμε a για να τρέξει το τμήμα

Flush

j=1

Form 80,32

Repeat {

 j=j+1

 Cls

 Print "Example for types of Print, Column:";j;" characters";

 Cursor 0,Row+1

 Data 0,1,2,3,4,5,6,7, 8

 Repeat {

 Read i

 Print \$(l,j),145355,"George Karras", "The Best",

 Cursor 0,Row+1

 } Until Empty

 Cursor 0,Row+1

 Print \$(0);"press any key"

 a\$=Key\$

} Until j=25

Print

φόρμα\$: αλφαριθμητικό δεξ παρακάτω.

πλάτος στήλης (μπορούμε να δώσουμε μέχρι το μέγιστο που είναι το πλάτος οθόνης)

Παρατήρηση: Όταν ένας αριθμός έχει πλάτος μεγαλύτερο της στήλης και δεν έχουμε ορίσει φόρμα τότε αν η στοίχιση είναι δεξιά ή αριστερά, ο αριθμός θα καταλάβει όσες στήλες χρειασθεί.

Ουσιαστικά αυτή η συνάρτηση είναι δυο! Μία για την στοίχιση και μία για το φορμάρισμα των εξαγομένων αριθμών ή και αλφαριθμητικών. Και οι δυο παραλλαγές ισχύουν μέχρι να ξαναδώσουμε μια ίδια εντολή αλλά με άλλες παραμέτρους. Η \$("") τερματίζει την χρήση φόρμας

ΔΕΣ "ΦΟΡΜΑΡΙΣΜΑ ΑΛΦΑΡΙΘΜΗΤΙΚΩΝ"

ΔΕΣ "ΦΟΡΜΑΡΙΣΜΑ ΑΡΙΘΜΩΝ"

ΔΕΣ "ΦΟΡΜΑΡΙΣΜΑ ΗΜΕΡΟΜΗΝΙΩΝ ΚΑΙ ΩΡΑΣ"

ΔΕΣ "ΦΟΡΜΑΡΙΣΜΑ ΓΙΑ ΚΑΘΕ ΤΥΠΟ"

Αναγνωριστικό: ΕΛ@ [@ (]

Η@ () είναι εσωτερική συνάρτηση της ΤΥΠΩΣΕ.

ΤΥΠΩΣΕ @(10,10);"ΓΕΙΑ ΧΑΡΑ"

@(X)

@(,Y)

@(X, Y)

@(X, Y, X1 ,Y1, XP)

@(X, Y, X1, Y1, XP, ΠΕΡ)

@(X, Y, X1, Y1, "όνομα αρχείου γραφικών")

@(X, Y, X1, Y1, "όνομα αρχείου γραφικών", 1)

Τα X, Y μεταθέτουν σε απόλυτη θέση τον δρομέα. Υπάρχει και εντολή που το κάνει χωρίς να δώσουμε το π.χ. ΤΥΠΩΣΕ @(10,5); αλλά το ΔΡΟΜΕΑΣ 10,5

τα X1 και Y1 ορίζουν ένα σημείο πιο δεξιά και πιο κάτω από τα X και Y όπου θα βάψουμε, ή και θα βάλουμε περίγραμμα τα χρώματα των οποίων θα είναι το XP και ΠΕΡ.

Επίσης αντί να βάψουμε μπορούμε να τοποθετήσουμε ένα γραφικό δίνοντας το όνομα του αρχείου μαζί με το μονοπάτι όπου βρίσκεται στον δίσκο (θα το δούμε αυτό αργότερα)

Επειδή ορίζουμε με τα X,Y και X1,Y1 δική μας αναλογία πλευρών X1-X προς Y1-Y μπορούμε να απλώσουμε ή όχι μια εικόνα. Αν την απλώσουμε θα καλύπτει όλο το τετράπλευρο, διαφορετικά αν δώσουμε την τελευταία παραλλαγή θα κρατήσει την αναλογία της εικόνας και θα

επεκτείνει ή θα μικρύνει ανάλογα την Υ διάσταση.

Άρα αν και δεν βάφει αυτόματα η M2000 πίσω από τους χαρακτήρες (στο φόντο), στην ΤΥΠΩΣΕ με την @() μπορεί να το κάνει.

Και κάτι τελευταίο, οι εικόνες μπορεί να είναι jpg, bmp αλλά και σχέδια wmf ή emf. Ειδικά για τα σχέδια wmf και emf αυτά αναπαράγονται χωρίς να αλλάζουν τον φόντο (διάφανα).

Αναγνωριστικό: ΕΛ~ [~(]
Τύπωσε ~(#12ff00), "Άλφα"

Ομάδα: ΣΤΟΧΟΙ ΚΑΙ ΕΠΙΛΟΓΗ - TARGET AND MENU

Αναγνωριστικό: ΑΛΛΑΓΗ [CHANGE]

Τα ορίσματα δίνονται με όνομα και τιμή. Μπορούν να δοθούν και ανακατωμένα:

ΠΕΝΑ, ΦΡΑΣΗ, ΦΟΝΤΟ, ΠΛΑΙΣΙΟ, ΟΔΗΓΙΑ

Η μεταβλητή ΕΔ κρατάει το νούμερο του ΣΤΟΧΟΥ (πρώτο νούμερο το 0)

Δώστε:

ΑΛΛΑΓΗ ΕΔ, ΦΡΑΣΗ "οκ"

ΑΛΛΑΓΗ ΕΔ, ΦΟΝΤΟ 2

ΑΛΛΑΓΗ ΕΔ, ΠΛΑΙΣΙΟ 15

ΑΛΛΑΓΗ ΕΔ, ΠΕΝΑ 15, ΠΛΑΙΣΙΟ 7, ΦΟΝΤΟ 3, ΦΡΑΣΗ "ΟΛΑ ΚΑΛΑ"

ΑΛΛΑΓΗ ΕΔ, ΟΔΗΓΙΑ "ΟΘΟΝΗ"

ΔΕΣ ΣΤΟΧΟΣ

Αναγνωριστικό: ΕΠΙΛΟΓΗ [MENU]

1) Σβήνει την λίστα επιλογών

ΕΠΙΛΟΓΗ

2) Άμεση εμφάνιση λίστας επιλογών (εδώ θα έχει τα "ΝΑΙ" και "ΟΧΙ"

ΕΠΙΛΟΓΗ "ΝΑΙ", "ΟΧΙ"

3) Εμφάνιση της κρατημένης λίστας επιλογών

ΕΠΙΛΟΓΗ !

4) Πρόσθεση στοιχείου στην λίστα επιλογών (αλλά όχι εμφάνιση)

ΕΠΙΛΟΓΗ + "ΔΕΝ ΑΠΑΝΤΩ"

Η επιλογή θα εμφανιστεί στον δρομέα χαρακτήρων αν χωράει δεξιά και κάτω διαφορετικά αλλάζει το πάνω και αριστερό σημείο γι να χωρέσει η λίστα. Αν έχει πολλά στοιχεία (μάξιμουμ 2000) τότε εμφανίζει και μπάρα ολίσθησης,

Μπορούμε να διαβάσουμε την επιλογή με τον αριθμό της από την μεταβλητή ΕΠΙΛΟΓΗ
Μπορούμε να τυπώσουμε το στοιχείο που επιλέχθηκε με την
ΤΥΠΩΣΕ ΕΠΙΛΟΓΗ\$(ΕΠΙΛΟΓΗ)
Ο πίνακας ΕΠΙΛΟΓΗ\$() περιλαμβάνει στοιχεία από το 1 έως το ΕΠΙΛΟΓΕΣ
Οι μεταβλητές ΕΠΙΛΟΓΗ, ΕΠΙΛΟΓΕΣ και ο πίνακας ΕΠΙΛΟΓΕΣ\$() αλλάζουν μετά από μια
εμφάνιση της εντολής ΕΠΙΛΟΓΗ, ΑΡΧΕΙΑ, ΔΕΙΞΕ

Αναγνωριστικό: ΣΑΡΩΣΕ [SCAN]

1) Με αναμονή ενεργοποίησης στόχου

ΣΑΡΩΣΕ

2) Με περιορισμένη αναμονή ενεργοποίησης στόχου

ΣΑΡΩΣΕ 0.5

Αναγνωριστικό: ΣΤΟΧΟΙ [TARGETS]

ΣΤΟΧΟΙ ΝΕΟΙ

Καθαρίζει την μνήμη που απασχολούσαν οι στόχοι. Δεν αλλάζει την οθόνη. Το ίδιο γίνεται με την
ΤΥΠΟΣ και ΠΑΡΑΘΥΡΟ αλλά έχουμε και καθαρισμό της οθόνης.

Αναγνωριστικό: ΣΤΟΧΟΣ [TARGET]

ΣΤΟΧΟΣ ΕΔ,"ΜΠΙΠ",60,3,7,5,5,"ΠΑΤΑ ΕΔΩ ΜΕ ΤΟ ΑΡΙΣΤΕΡΟ ΠΛΗΚΤΡΟ ΤΟΥ ΠΟΝΤΙΚΙΟΥ"

ΣΤΟΧΟΣ ΕΔ, 0

ΣΤΟΧΟΣ ΕΔ,1

ΔΕΣ ΕΝΤΟΛΗ ΑΛΛΑΓΗ

Παρατήρηση:

Δημιουργία - Πρώτα θέτουμε τον δρομέα χαρακτήρων (εντολή ΔΡΟΜΕΑΣ ή με την ΤΥΠΩΣΕ)
κατόπιν δίνουμε μια μεταβλητή (εδώ ΕΔ) όπου αυτή μετά την εκτέλεση της εντολής θα έχει έναν
αριθμό που θα δείχνει τον συγκεκριμένο στόχο - μεταβλητή χειρισμού στόχων. Μετά ακολουθεί η
οδηγία που θα εκτελεί ο στόχος. Αμέσως μετά ακολουθεί το πλάτος και το ύψος τα οποία δεν
αλλάζουν μέχρι να σβήσουν όλοι οι στόχοι με μια εντολή οθόνης ΤΥΠΟΣ. Ακολουθεί το χρώμα
πλαίσιου καθώς και το χρώμα φόντου. Τέλος δυο σημαντικοί παράμετροι ακολουθούν, ο τύπος
του στόχου και η φράση που θα εμφανίζει.

Ο τύπος του στόχου παίρνει τιμές:

Που ορίζουν θέση της φράσης και τρόπος εκτέλεσης οδηγίας

1 2 3 Θέση της φράσης σχετικά με το πλαίσιο

4 5 6 Το 5 σημαίνει στο κέντρο του πλαισίου

7 8 9 Αν προσθέσουμε το 10 τότε η οδηγία δεν εκτελείται αλλά στέλνεται σαν να την πληκτρολογήσαμε. Δοκιμάστε στο παράδειγμα με τύπο 5 και με 15.

Ενεργοποίηση.

Με ΣΤΟΧΟΣ μεταβλητή, 0 απενεργοποιούμε τον στόχο ενώ με τιμή 1 ενεργοποιούμε και επανασχεδιάζουμε. Κάθε εντολή ΑΛΛΑΓΗ επανασχεδιάζει τον στόχο.

Τέλος μπορούμε να δώσουμε σαν οδηγία το όνομα ενός τμήματος ώστε να εκτελέσουμε πολλαπλές εντολές.

Αν καθαρίσουμε την οθόνη με την εντολή ΟΘΟΝΗ οι στόχοι παραμένουν νοητά και αν είναι ενεργοί τότε με το αριστερό πλήκτρο του ποντικιού πάνω σε μια περιοχή στόχου θα έχουμε ενεργοποίηση αυτού. Δες εντολή ΣΤΟΧΟΙ ΝΕΟΙ που σβήνει από την μνήμη τους στόχους.

Ομάδα: ΓΡΑΦΙΚΑ 2D - DRAWING 2D

Αναγνωριστικό: ΒΑΨΕ [FILL]

1) ΒΑΨΕ πλατος.χ, υψος.ψ, χρώμα1, χρώμα2, διευθυνση, τμημα

ΒΑΨΕ πλατος.χ, υψος.ψ, χρώμα1, χρώμα2, διευθυνση

ΒΑΨΕ πλατος.χ, υψος.ψ, χρώμα1, χρώμα2

ΒΑΨΕ πλατος.χ, υψος.ψ, χρώμα1

ΒΑΨΕ πλατος.χ, υψος.ψ

Επεξήγηση:

Δίνουμε πάντα πλάτος και ύψος (θα βαφτεί ένα παραλληλόγραμμο από την τελευταία θέση του δρομέα γραφικών - δες ΘΕΣΗ)

Χωρίς να δώσουμε το χρώμα (0 ..15 βασικά χρώματα των Windows, αρνητικές τιμές -1 έως -16777216 RGB), χρησιμοποιείται το χρώμα πένας (δες ΠΕΝΑ)

Με δεύτερο χρώμα το βάψιμο γίνεται με διαβάθμιση (όπως στην ΦΟΝΤΟ)

Η διεύθυνση μπορεί να πάρει τιμές 0 ή 1

Το τμήμα παίρνει τιμές 0 ή 1

Ο δρομέας γραφικών μεταφέρεται στην νέα θέση (προηγούμενη.χ +πλάτος.χ, προηγούμενη.ψ +υψος.ψ)

2) ΒΑΨΕ @ πλατος.χ, υψος.ψ, τροπος, αριθμος , περιθωριο_σε_twips

ΒΑΨΕ @ πλατος.χ, υψος.ψ, , αλφαριθμητικό\$

Επεξήγηση:

Με αυτήν την παραλλαγή μπορούμε να εμφανίζουμε στοιχεία των Windows:

Τρόπος:

0 - Σχεδιάζει πλαίσια

1 - Σχεδιάζει φόντο για γράμματα

2 - Σχεδιάζει

3 - Πλαίσιο εστίασης (ΒΑΨΕ @1000,1000,3,1 με δεύτερη όμοια εντολή φεύγει το πλαίσιο)

4 - Διάφορα τετράπλευρα (όπως τα στοιχεία ελέγχου των windows)

5 - Κεντράρει και τυπώνει μια λέξη ή έναν αριθμό

6 - Όπως η (4) αλλά μπορούμε να τυπώνουμε κάποιες πλευρές (1+2+4+8 όλο το τετράπλευρο)

Ο δρομέας γραφικών δεν αλλάζει θέση.

3D πλαίσια - ΒΑΨΕ @ 2000,2000,4,1

Αναγνωριστικό: ΒΗΜΑ [STEP]

Οι τιμές των συντεταγμένων είναι σε twips

Η εντολή ΒΗΜΑ μεταφέρει τον δρομέα γραφικών σε σχετική θέση (ενώ η ΘΕΣΗ μετακινεί τον δρομέα σε απόλυτη θέση)

1) Καρτεσιανές συντεταγμένες

ΒΗΜΑ 1000,500 (μεταφορά του δρομέα γραφικών κατά 1000 στον οριζόντιο άξονα και κατά 500 στον κατακόρυφο - προς τα δεξιά και προς τα κάτω)

ΒΗΜΑ ,-500 (μεταφορά δρ. γρ. προς τα άνω κατά 500 twips)

ΒΗΜΑ -1000 (μεταφορά δρ. γρ. προς τα αριστερά κατά 1000 twips)

2) Πολικές Συντεταγμένες

ΒΗΜΑ ΓΩΝΙΑ $\pi/4$,1000 (μεταφορά του δρομέα γραφικών κατά 1000 twips απόσταση σε γωνία $\pi/4$ δηλ. 45 μοίρες από τον ορίζοντα)

Με πολικές συντεταγμένες δουλεύουν και οι εντολές ΧΑΡΑΞΕ, ΠΟΛΥΓΩΝΟ, ΚΑΜΠΥΛΗ. Οι πολικές συντεταγμένες είναι πάντα σχετικές.

Αναγνωριστικό: ΓΕΜΙΣΕ [FLOODFILL]

Η εντολή ΓΕΜΙΣΕ χρησιμοποιείται για να γεμίζουμε χρώμα ακανόνιστα σχήματα, που είτε έχουν ομοιόχρωμη περίμετρο (ένας τρόπος), είτε έχουν ένα χρώμα και θέλουμε να το αλλάξουμε.

1) ΓΕΜΙΣΕ Θεση.Χ, Θεση.Υ, χρώμα (δες ΒΑΨΕ για το χρώμα)

ΓΕΜΙΣΕ 1000,1000, 5

ΓΕΜΙΣΕ 1000,1000 (χρησιμοποιεί το χρώμα της πέννας)

2) ΓΕΜΙΣΕ ΧΡΩΜΑ Θεση.Χ, Θεση.Υ, χρώμα

ΓΕΜΙΣΕ ΧΡΩΜΑ 1000, 1000, 5

ΓΕΜΙΣΕ ΧΡΩΜΑ 1000, 1000

Παρατήρηση

αν θέλουμε να γεμίσουμε χρώμα στην θέση του δρομέα γραφικών τότε δίνουμε

ΓΕΜΙΣΕ „5

ΓΕΜΙΣΕ ΧΡΩΜΑ „5

Αναγνωριστικό: ΘΕΣΗ [MOVE]

ΘΕΣΗ 1000,1000

ΘΕΣΗ 3000

ΘΕΣΗ ,5000

Μεταφορά του δρομέα γραφικών σε απόλυτη θέση.

Μπορούμε να διαβάσουμε την τωρινή θέση από δυο μεταβλητές (μόνο για διάβασμα), τις

ΘΕΣΗ.X και ΘΕΣΗ.Y

Επίσης το 0,0 είναι πάνω αριστερά και το κάτω δεξιά σημείο είναι το ΚΛΙΜΑΞ.X, ΚΛΙΜΑΞ.Y (συνώνυμα Χ.ΣΗΜΕΙΑ και Υ.ΣΗΜΕΙΑ)

Αρνητικές τιμές μπορούμε να δώσουμε αλλά ο δρομέας θα πάει εκτός οθόνης, έτσι μια ΧΑΡΑΞΕ 3000,3000 θα χαράξει γραμμή μέσα στην οθόνη αλλά ένα τμήμα αυτής)

Αναγνωριστικό: ΚΑΜΠΥΛΗ [CURVE]

Η εντολή ΚΑΜΠΥΛΗ μοιάζει με την πολύγωνο με την διαφορά ότι χρησιμοποιεί τρία σημεία για κάθε καμπύλη (πρώτο, δεύτερο, τρίτο) κατά το πρότυπο BEZIER.

Η τελική θέση της καμπύλης μπορεί να βρεθεί από το άθροισμα των χ και ψ χωριστά.

Π.Χ

ΚΑΜΠΥΛΗ 1000,1000,0,2000,500,1000

συνολικό Χ μετατόπιση= $1000+0+500 = 1500$

συνολικό Ψ μετατόπιση= $1000+2000+1000= 4000$

ΚΑΜΠΥΛΗ ΓΩΝΙΑ γωνία1, απόσταση1, γωνία2, απόσταση2, γωνία3, απόσταση3

ΚΑΜΠΥΛΗ ΓΩΝΙΑ γωνία1, απόσταση1, γωνία2, απόσταση2, γωνία3, απόσταση3, ..., ..., ...,

γωνίαN, απόστασηN, γωνίαN, απόστασηN, γωνίαN, απόστασηN

παραδείγματα εδώ CURVE

Αναγνωριστικό: ΚΥΚΛΟΣ [CIRCLE]

ΚΥΚΛΟΣ ακτίνα, λόγος, χρωμα_περιφερειας, απο_ακτινια, εως_ακτινια

ΚΥΚΛΟΣ ΓΕΜΙΣΜΑ χρωμα1, ακτίνα, λόγος, χρωμα_περιφερειας, απο_ακτινια, εως_ακτινια

ΘΕΣΗ 1000,1000

ΚΥΚΛΟΣ 500

ΘΕΣΗ 5000,5000

ΠΑΧΟΣ 4 {

ΚΥΚΛΟΣ 3000,1/2,15, 0,ΠΙ/2

}

Δες ΠΟΛΥΓΩΝΟ, ΚΑΜΠΥΛΗ

Αναγνωριστικό: ΟΜΑΛΑ [SMOOTH]

ΟΜΑΛΑ ΝΑΙ

ΟΜΑΛΑ ΟΧΙ

Βάζει το GDI+ για γραμμές, καμπύλες, πολύγωνα και κύκλους, ώστε να βγαίνουν οι γραμμές των σχημάτων ομαλά.

Αναγνωριστικό: ΠΑΧΟΣ [WIDTH]

ΠΑΧΟΣ παχος [, επιλογή τύπου γραμμής] {

εντολές

}

Όταν έχουμε GDI+ τότε οι τύποι γραμμής από 2 έως 5 λειτουργούν για κάθε πάχος. Στο GDI λειτουργούν μόνο με πάχος 1 pixel

Μέσα στις αγκύλες οι εντολές γραφικών χρησιμοποιούν το επιλεγμένο πάχος γραμμής σε pixels.

Δείτε στο παρακάτω παράδειγμα φωλιασμένες εντολές τύπου ΠΑΧΟΣ

```
τμημα πα {  
παχος 2 {  
θεση 1000,1000  
χαραξε 1000,1000  
παχος 5 {  
θεση 3000,3000  
χαραξε 0,1000  
}  
θεση 3000,1000  
χαραξε 1000,-1000  
}  
θεση 4000,4000  
χαραξε 0,1000  
}
```

Αναγνωριστικό: ΠΟΛΥΓΩΝΟ [POLYGON]

Η εντολή πολύγωνο όπως και η καμπύλη έχουν σημεία (θέσεις του δρομέα γραφικών) σχετικές με την ακριβώς προηγούμενη.

Παράδειγμα

α) ΠΟΛΥΓΩΝΟ 4, 1000,0,0,1000,-1000,0,0,-1000

β) ΠΟΛΥΓΩΝΟ 4, 1000,0,0,1000,-1000,0,1000,-1000

Οι α και β είναι σχεδόν ίδιες. Η διαφορά θα φανεί αν δώσουμε δυο ή περισσότερες φορές την ίδια εντολή. Στην περίπτωση α όλα τα τετράγωνα γράφονται το ένα πάνω στο άλλο. Στην περίπτωση β το κάθε νέο τετράγωνο γράφεται δίπλα δεξιά.

Δηλαδή το τελευταίο σημείο δηλώνει την θέση που θα μετακινηθεί ο δρομέας, ενώ το σχήμα πάντα κλείνει στην θέση του πρώτου σημείου.

Υπάρχει και η παραλλαγή με την ΓΩΝΙΑ

ΘΕΣΗ 5000,2000

ΠΟΛΥΓΩΝΟ 4, ΓΩΝΙΑ ΠΙ/4,3000,-ΠΙ/4,3000,ΠΙ/4,-3000,-ΠΙ/4,-3000

Προσέξτε μια διαφορά με την Καμπύλη. Η εντολή ΠΟΛΥΓΩΝΟ θέλει το χρώμα γεμίσματος.

Αν δεν θέλουμε γέμισμα:

ΘΕΣΗ 5000,3000

ΧΡΩΜΑ { ΠΟΛΥΓΩΝΟ 4, ΓΩΝΙΑ ΠΙ/4,3000,-ΠΙ/4,3000,ΠΙ/4,-3000,-ΠΙ/4,-3000 }

Αναγνωριστικό: ΣΧΕΔΙΟ [DRAWING]

1) Χωρίς να ορίσουμε μέγεθος, το μέγεθος θα βγει από το πλαίσιο που περιέχει όλα τα στοιχεία αν θέλουμε να βάψουμε με την Φόντο, τότε θα βαφτεί με το μέγεθος της οθόνης στην οποία φτιάξαμε το σχέδιο

Το A είναι διάρθρωση μνήμης (buffer), το οποίο περιέχει το σχέδιο σαν αρχείο emf. Μπορούμε να σώσουμε το αρχείο με απλό τρόπο όπως φτιάχνουμε δυαδικά αρχεία, με την εντολή Άνοιξε "όνομα.emf" για έξαγωγή ως #Φ... και μετά βάζουμε μια ΔΩΣΕ #φ, A και μετά μια Κλείσε #Φ Μπορούμε να γράψουμε το σχέδιο στο πρόχειρο με την Πρόχειρο A.

```
Σχέδιο {  
εντολές....  
} ως A
```

2) Ορίζουμε ένα μέγεθος, πχ 3000 twips X 3000 twips.

```
Σχέδιο 3000, 3000 {  
εντολές....  
} ως A
```

Αναγνωριστικό: ΧΑΡΑΞΕ [DRAW]

1) Καρτεσιανές Συντεταγμένες

ΧΑΡΑΞΕ σχετικη_θεση_χ, σχετικη_θεση_ψ, χρωμα

2) Πολικές Συντεταγμένες

ΧΑΡΑΞΕ ΓΩΝΙΑ απολυτη_γωνια_χ, αποσταση, χρωμα

απόλυτη γωνία σημαίνει ότι το 0 θα δίνει πάντα κατεύθυνση δεξιά
οι γωνίες είναι σε ακτίνια (2*π είναι πάλι το 0, κατεύθυνση δεξιά)

Το χρώμα και στις δυο περιπτώσεις μπορεί να είναι:

α) Σταθερά χρώματα των WINDOWS από 0 έως 15

β) RGB (RED GREEN BLUE) με πρόσημο -

μπορεί κανείς να χρησιμοποιήσει την συνάρτηση ΧΡΩΜΑ(κόκκινο, πράσινο, μπλε)

τα βασικά χρώματα δέχονται τιμές 0 .. 255. Η συνάρτηση αυτή μπορεί να χρησιμοποιηθεί οπουδήποτε έχουμε να δώσουμε τιμή χρώματος και δίνει ένα αρνητικό νούμερο. Η έκκτη έκδοση περνάει και τιμές δεκαεξαδικές..π.χ. -0xfffff είναι το άσπρο (το - χρειάζεται γιατί τα θετικά νούμερα τα βλέπει ο διερμηνευτής ως χρώματα των Windows)

γ) 0X80000001 (δεκαεξαδικός αριθμός) δίνει χρώμα του συστήματος των Windows (για προχωρημένους)

3) Χαραξε εως 3000,4000

Χαραξε εως 3000,4000, 4

Χαράζει γραμμή από τη θέση του δρομέα γραφικών έως τις απόλυτες συντεταγμένες 3000,

4000 που θα γίνουν η νέα θέση του δρομέα.

ΠΑΡΑΔΕΙΓΜΑ

ΘΕΣΗ 3000,3000

ΧΑΡΑΞΕ 2000,0

ΧΑΡΑΞΕ 0,2000,15

ΧΑΡΑΞΕ -2000,0

ΧΑΡΑΞΕ 0,-2000,14

ΧΑΡΑΞΕ ΓΩΝΙΑ ΠΙ/4,2000,5

ΘΕΣΗ μετακινεί σε απόλυτη θέση τον δρομέα γραφικών

ΒΗΜΑ μετακινεί σε σχετική θέση τον δρομέα γραφικών (έχει και ΒΗΜΑ ΓΩΝΙΑ όπου η γωνία είναι απόλυτη και παρέχουμε και την απόσταση)

ΧΑΡΑΞΕ μετακινεί σε σχετική θέση τον δρομέα γραφικών (έχει και εδώ ΧΑΡΑΞΕ ΓΩΝΙΑ)

ΧΑΡΑΞΕ ΕΩΣ μετακινεί σε απόλυτη θέση

ΔΕΣ ΠΟΛΥΓΩΝΟ ΚΑΜΠΥΛΗ

ΔΕΣ ΡΑΤΗ (ΧΡΩΜΑ) και ΠΑΧΟΣ (γραμμών)

Μπορούμε να φτιάχνουμε πλαίσια σε όρια χαρακτήρων με την ΤΥΠΩΣΕ με την εσωτερική εντολή της @()

Αναγνωριστικό: ΧΡΩΜΑ [COLOR]

1) όλα τα γραφικά μέσα στις αγκύλες θα φτιάχνουν μια περιοχή με χρώμα

ΧΡΩΜΑ χρωμα [, τρόπος εμφάνισης χρώματος] {

εντολές

}

ο τρόπος εμφάνισης είναι από 0 έως 7, με το 1 να μην δείχνει καθόλου φόντο!

τα 2 3 4 5 6 7 είναι διαγραμμίσεις.

2) όλα τα γραφικά μέσα στις αγκύλες θα φτιάχνουν μια περιοχή με χρώμα που θα αναστρέφει τα υπάρχοντα χρώματα της οθόνης (δύο φορές η ίδια εντολή κάνει τα γραφικά να εξαφανίζονται

ΧΡΩΜΑ ! χρωμα [, τρόπος εμφάνισης χρώματος] {

εντολές

}

Επειδή χρησιμοποιεί το Χοr (ή Από όπως λέγεται στη Μ2000) δεν μπορεί να χρησιμοποιηθεί με το Ομαλά Ναί (GDI+), και αυτόματα και πρόσκαιρα αλλάζει σε Ομαλά Όχι (GDI)

Δες παράδειγμα στην εντολή ΚΑΜΠΥΛΗ

3) όλα τα γραφικά τυπώνονται με το χρώμα της πέννας χωρίς γέμισμα.

ΧΡΩΜΑ {

εντολές

}

Αυτό μάλιστα κάνει και για την ΤΥΠΩΣΕ, δες παράδειγμα παρακάτω:

' Εκτύπωση γραμμάτων με εξωτερική γραμμή - outline)

ΤΜΗΜΑ Α{

ΔΙΠΛΑ

ΠΕΝΑ 2

ΔΡΟΜΕΑΣ 5,5

ΠΑΧΟΣ 10 {ΧΡΩΜΑ ! {ΤΥΠΩΣΕ "ΓΙΩΡΓΟΣ"} }

ΔΡΟΜΕΑΣ 5,5

ΧΡΩΜΑ ! 4 {ΤΥΠΩΣΕ "ΓΙΩΡΓΟΣ"}

ΚΑΝΟΝΙΚΑ

ΠΕΝΑ 11

ΤΥΠΩΣΕ "ΟΚ"

}

4) Καθορισμός περιοχής τυπώματος

Και αυτή η επιλογή έχει να κάνει με το GDI και όχι το GDI+

Αν μετά την αγκύλη βάλουμε το ; τότε η περιοχή δεν θα χρωματιστεί αλλά όλα τα γραφικά από δω και πέρα θα εμφανίζονται μόνο σε αυτήν την περιοχή. Μάλιστα μπορεί η περιοχή να έχει πολλά τμήματα.

Τμήμα Α {

χρωμα {

θεση 5000,4000

ΠΟΛΥΓΩΝΟ 4, ΓΩΝΙΑ ΠΙ/4,3000,-ΠΙ/4,3000,ΠΙ/4,-3000,-ΠΙ/4,-3000

θεση 200,200

πολυγωνο 4, 5000,0,0,1000,-5000,0,0,-1000

};

φοντο 3,5

θεση 3000,3000

κυκλος 3000

' Το παρακάτω χρειάζεται για την επαναφορά της αρχικής περιοχής

θεση 0,0

χρωμα {πολυγωνο 0, χ.σημεια,0,0,υ.σημεια,-χ.σημεια,0,0,-υ.σημεια};

}

ή καλύτερα χρώμα {};

Η αρχική περιοχή ξαναεμφανίζεται πάντα μετά από μια εντολή ΟΘΟΝΗ.

Αναγνωριστικό: ΧΡΩΜΑΤΙΣΕ [PSET]
ΧΡΩΜΑΤΙΣΕ
ΧΡΩΜΑΤΙΣΕ 15
ΧΡΩΜΑΤΙΣΕ ΧΡΩΜΑ(100,100,500)
ΧΡΩΜΑΤΙΣΕ 15, 6000,6000
ΧΡΩΜΑΤΙΣΕ ΧΡΩΜΑ(100,100,500), 6000,6000

Ομάδα: ΕΝΤΟΛΕΣ ΕΙΚΟΝΩΝ - BITMAP COMMANDS

Αναγνωριστικό: ΑΝΤΙΓΡΑΨΕ [COPY]
Από το σημείο του δρομέα μπορούμε να αντιγράψουμε εικόνες
1) Άμεση αντιγραφή
ΑΝΤΙΓΡΑΨΕ Χ.προορισμού, Υ.προορισμού, Πλάτος, Ύψος
2) Αντιγραφή ολόκληρης της οθόνης στον δίσκο (δεν παίζει ρόλο ο δρομέας)
ΑΝΤΙΓΡΑΨΕ "όνομα αρχείου"
3) Αντιγραφή σε μεταβλητή
ΑΝΤΙΓΡΑΨΕ πλάτος, ύψος ΣΤΟ αλφαριθμητικό\$
4) Αντιγραφή από μεταβλητή
ΑΝΤΙΓΡΑΨΕ θέσηΧ, θέσηΥ ΜΕ αλφαριθμητικό\$
5) Αντιγραφή με περιστροφή και αυξομείωση μεγέθους (γίνεται και εξομάλυνση)
ΑΝΤΙΓΡΑΨΕ θέσηΧ, θέσηΥ ΜΕ αλφαριθμητικό\$, γωνία
ΑΝΤΙΓΡΑΨΕ θέσηΧ, θέσηΥ ΜΕ αλφαριθμητικό\$, γωνία, μέγεθος
6) Αντιγραφή όπως το 5 αλλά στις περιστροφές φαίνεται στις γωνίες η εικόνα από κάτω
ΑΝΤΙΓΡΑΨΕ θέσηΧ, θέσηΥ ΕΠΑΝΩ αλφαριθμητικό\$, γωνία
ΑΝΤΙΓΡΑΨΕ θέσηΧ, θέσηΥ ΕΠΑΝΩ αλφαριθμητικό\$, γωνία, μέγεθος

το αλφαριθμητικό μπορεί να είναι από πίνακα, π.χ. α\$(4)

Η εντολή δεν μετακινεί τον δρομέα γραφικών

ΦΟΡΜΑ 60, 32
ΟΘΟΝΗ 7
ΠΕΝΑ 15
ΤΥΠΩΣΕ "Αυτή είναι η πρώτη γραμμή"
ΤΥΠΩΣΕ "Αυτή είναι η δεύτερη γραμμή"
ΤΥΠΩΣΕ "Αυτή είναι η τρίτη γραμμή"
ΤΥΠΩΣΕ "Αυτή είναι η τέταρτη γραμμή"
ΘΕΣΗ 0,0
ΚΑΘΑΡΟ α\$
ΑΝΤΙΓΡΑΨΕ 3000,2000 ΣΤΟ α\$

ΦΟΝΤΟ 3, 5 ' βάζει χρώμα με διαβάθμιση στο φόντο.
ΑΝΤΙΓΡΑΨΕ 6000,4000 ΕΠΑΝΩ α\$, 30, 200
ΑΝΤΙΓΡΑΨΕ 6000,4000 ΕΠΑΝΩ α\$, 45
ΑΝΤΙΓΡΑΨΕ 2000,2000 ΜΕ α\$, 45

Αναγνωριστικό: ΔΙΑΦΑΝΟ [SPRITE]

ΕΝΤΟΛΕΣ ΓΡΑΦΙΚΩΝ ΕΙΚΟΝΩΝ, BITMAPS Με αυτήν την εντολή μπορούμε να βάζουμε στην θέση του δρομέα εικόνες με διαφάνεια. Προσοχή η θέση του δρομέα γραφικών αντιστοιχεί στο μέσον της εικόνας. Η διαφάνεια μπορεί έχει δυο μορφές: Το χρώμα μασκαρίσματος, το οποίο δηλώνει ποια σημεία τις εικόνες μας δεν θα τυπωθούν και έτσι θα δείχνουν τα χρώματα του φόντου. Η διαπερατότητα (0 έως 100) μας δείχνει τα άλλα σημεία της εικόνας που θα τυπωθούν, πόσο διάφανα θα είναι έτσι ώστε να βλέπουμε και το φόντο σαν να ήταν η εικόνα μας από διαφανές πλαστικό! Η περιστροφή είναι σε μοίρες.

Για κάθε σημείο της εικόνας (και κάθε χρώμα ξεχωριστά RGB) θα γίνει ο παρακάτω υπολογισμός:

Τελικό χρώμα=Χρώμα_Σημείου_Οθόνης * (100- Διαπερατότητα)/100 +
Χρώμα_Σημείου_Εικόνας * Διαπερατότητα/100

ΔΙΑΦΑΝΟ Α\$

ΔΙΑΦΑΝΟ Α\$, χρώμα μασκαρίσματος

ΔΙΑΦΑΝΟ Α\$, χρώμα μασκαρίσματος, Γωνία_Περιστροφής

ΔΙΑΦΑΝΟ Α\$,χρώμα μασκαρίσματος, Γωνία_Περιστροφής, Ποσοστό_Μεγέθους

ΔΙΑΦΑΝΟ Α\$, χρ_μασκ, Γωνία_Περιστροφής, Ποσοστό_Μεγέθους, Διαπερατότητα [, 1]
η τελευταία παράμετρος αν δεν είναι 0 σημαίνει ότι δεν θα κρατήσει το σύστημα το χώρο που "πατάει" το διάφανο.

ΔΙΑΦΑΝΟ Α\$, χρ_μασκ, Γωνία_Περιστροφής, Ποσοστό_Μεγέθους, Εικόνα_Μάσκας\$ [, 1]

Διάφανο Διάρθρωση_Μνήμης_με_Εικόνα [, χρ_μασκ[, Γωνία_Περιστροφής[,

Ποσοστό_Μεγέθους[, Διαπερατότητα [, 1]]]]]

Με το Διάρθρωση() φτιάχνουμε μια νέα διάρθρωση από αρχείο (πχ ένα png αρχείο) και το χρησιμοποιούμε ως διάφανο. Τα png αρχεία μπορεί να έχουν και αυτά μάσκα για μερική αδιαφάνεια, και δεν χάνεται στη διάρθρωση και έτσι χρησιμοποιείται πάντα στην εμφάνιση.

Σε κάθε εκτύπωση η μεταβλητή ΔΙΑΦΑΝΕΙΑ\$ λαμβάνει την περιοχή που θα καλυφθεί, έτσι μπορούμε να την σώνουμε σε άλλη μεταβλητή και να επαναφέρουμε την οθόνη στο προηγούμενη όψη (εκτός αν δεν το θέλουμε, οπότε βάζουμε το , 1 στο τέλος)

Όταν εργαζόμαστε με οθόνες με χρώμα μικρότερο από 24 bit (δηλαδή λιγότερα από 16 εκατομμύρια χρώματα) το χρώμα μασκαρίσματος πρέπει να δοθεί με επιλογή χρώματος από ένα σημείο της εικόνας, συντεταγμένες σε pixels:

ΤΥΠΩΣΕ ΧΡΩΜΑΤΑ
ΔΙΑΦΑΝΟ Α\$, ΣΗΜΕΙΟ(Α\$,0,0)

Στην έκτη έκδοση μπορούμε να χρησιμοποιήσουμε τα επίπεδα (οι παίκτες) σαν διαφάνειες. Η διαφορά τους είναι ότι δεν έχουν μερική διαφάνεια (διαπερατότητα) αλλά όμως είναι σε ξεχωριστό επίπεδο...που σημαίνει ότι μπορούμε να κάνουμε ότι θέλουμε στο βασικό επίπεδο ή στο περιθώριο χωρίς να τα πειράξουμε! Επίσης οι παίκτες έχουν και ειδική συνάρτηση ΣΥΓΚΡΟΥΣΗ(για να ελέγχουμε αν μεταξύ τους ή με συγκεκριμένο πλαίσιο έχουμε σύγκρουση.

Παρακάτω είναι ένα απλό πρόγραμμα που παίρνει ένα κομμάτι απο την οθόνη πλάτους 3000 twips και ύψους 1000, και βάζει στο 1000,1000.

```
ανανewση 100
α$=""
θεση 0,0
αντιγραψε 3000,1000 στο α$
θεση 1000,1000
διαφανεια α$,0 ' το τυπώνουμε εδώ για να πάρουμε το β$ μετά
για i=1 εως 100 {
ανανewση 0
β$=διαφανεια$
διαφανεια β$ ' τώρα σβήνουμε την παλιά διαφάνεια
βημα 30,30
διαφανεια α$,0
ανανewση ' τώρα εμφανίζουμε την οθόνη
}
```

Πιο γρήγορη μέθοδος για να αποδώσουμε το φόντο, είναι με τη χρήση HOLD και RELEASE. Εδώ δίνουμε και περιστροφή (σε μοίρες) και το μέγεθος (από 11 μέχρι 211 τοις εκατό, δηλαδή 100 για κανονικό μέγεθος).

```
ανανewση 10 ' αν το κάνουμε 100 θα γίνονται λιγότερες ανανεώσεις.
α$=""
θεση 0,0
αντιγραψε 3000,1000 στο α$
κρατησε
θεση 1000,1000
διαφανεια α$,0
```

```

i=11
καθε 10 {
    i++
    αφησε
    βημα 45,30
    διαφανεια α$,i,i+10
    ' ανανεωση ' αφήνουμε το σύστημα να χειριστεί αυτόματα την ανανέωση
    αν i>=200 τότε εξοδος
}

```

Αναγνωριστικό: EIKONA [IMAGE]

1) Εμφάνιση εικόνας στην οθόνη (bmp, wmf, emf, jpg, gif, ico)

EIKONA "αρχείο" ,X[,Y]

Εικόνα Διάθρωση_Μνήμης_με_Εικόνα, X [,Y]

Αν δώσουμε και Y τότε αλλάζουμε την αναλογία X/Y

2) Μεταφορά εικόνας σε άλλη εικόνα.

EIKONA εικόνα\$ ΣΤΟ άλλη_εικόνα\$ [(χρώμα_φόντου)] ' το χρώμα είναι για emf εικόνα\$

EIKONA εικόνα\$ [(χρώμα_φόντου)] ΣΤΟ άλλη_εικόνα\$ ' το χρώμα είναι για png εικόνα\$

EIKONA Διάθρωση_Μνήμης_με_Εικόνα [(χρώμα_φόντου)] ΣΤΟ άλλη_εικόνα\$

EIKONA Διάθρωση_Μνήμης_με_Εικόνα ΣΤΟ άλλη_εικόνα\$[(χρώμα_φόντου)]

3) Μεταφορά με αλλαγή μεγέθους

EIKONA εικόνα\$ ΣΤΟ άλλη_εικόνα\$ [(χρώμα_φόντου)], ποσοστό_μεγέθους (για 100 το ίδιο μέγεθος)

ομοίως όπως στο 2...

4) Μεταφορά με αλλαγή σχέσης πλάτους προς ύψος

EIKONA εικόνα\$ ΣΤΟ άλλη_εικόνα\$, ποσοστό_πλάτους, ποσοστό_ύψους

ομοίως όπως στο 2...

5) Εξαγωγή ως bmp

EIKONA εικόνα\$ ΕΞΑΓΩΓΗ αρχείο_εικόνας\$

6) Εξαγωγή ως jpg με συμπίεση

EIKONA εικόνα\$ ΕΞΑΓΩΓΗ αρχείο_εικόνας\$, ποσοστό_ποιότητας

στο 100% έχουμε τη καλύτερη ποιότητα αλλά με μικρότερη συμπίεση (λιγότερη απώλεια)

Αναγνωριστικό: ΠΑΙΚΤΗΣ [PLAYER]

Πάνω από την οθόνη υπάρχουν 32 παίκτες ή επίπεδα. Είναι τα γνωστά sprites μόνο που σε αυτή την έκδοση μπορούν να λειτουργούν ως 32 νέες οθόνες προβάλλοντας στοιχεία.

Για ιστορικούς λόγους έχει δοθεί το όνομα ΠΑΙΚΤΗΣ ή PLAYER...έτσι λέγονταν στα ATARI 8bit, υπολογιστές με παρόμοιες δυνατότητες τη δεκαετία του '80

1) ΠΑΙΚΤΗΣ 0

καταργεί όλους τους παίκτες

2) ΠΑΙΚΤΗΣ 1, 2000, 3000 ME AA\$, χρωμαδιαφάνειας, ένταση ΜΕΓΕΘΟΣ 1.4

2.1) ΠΑΙΚΤΗΣ 1, 2000, 3000 ME AA\$, ΜΑΣΚΑΣ\$, ένταση ΜΕΓΕΘΟΣ 1.4

player 1, 2000,3000 use AA\$, 1, 0, γωνία size 1.4

οι παίκτες έχουν αριθμό προτεραιότητας. Ο παίκτης 1 βρίσκεται πάντα πίσω από τον 2. Ο παίκτης 32 είναι πάνω απ' όλους.

η θέση 2000,3000 έχει να κάνει με το παράθυρο και όχι με την οθόνη.

η εικόνα AA\$ μπορεί να φορτωθεί με πολλούς τρόπους. Δεν υπάρχει περιορισμός στο μέγεθος, πέρα από τους περιορισμούς του συστήματος (περίπου διπλάσιο μέγεθος από την οθόνη είναι καλά). Μπορούμε να επιλέξουμε χρώμα διαφάνειας, θετικά νούμερα από το 0 ως το 15, τα βασικά χρώματα και αρνητικά νούμερα με την Χρωμα(0...255, 0...255, 0...255). Μπορούμε να δώσουμε μια ένταση που σημαίνει εδώ ότι όσο μεγαλύτερο νούμερο δώσουμε (1...255) τόσα περισσότερα χρώματα θα κάνουμε διάφανα, από το βασικό. Με -1 στην ένταση ακυρώνουμε την διαφάνεια.

Η γωνία είναι σε μοίρες.

Μετά τη γωνία μπαίνουν αν θέλουμε το σημείο hot spot, Αυτό το σημείο ορίζεται για τη συγκεκριμένη γωνία και μεγέθυνση. Εκτός από το να μεταθέτει κατά X και Y τον παίκτη, στη Συγκρουση() με τις δυο παραμέτρους λαμβάνεται υπόψιν για να γίνει ο έλεγχος βάσει της νέας μετάθεσης.

Τέλος έχουμε το μέγεθος όπου 1 είναι το 100%.

3) ΠΑΙΚΤΗΣ 1 ΔΕΙΞΕ

Player 1 show

Εμφανίζει το παίκτη. Η προηγούμενη εντολή δεν εμφανίζει τον παίκτη!

4) ΠΑΙΚΤΗΣ 1 ΚΡΥΨΕ

Player 1 hide

Εξαφανίζει το παίκτη. Ο παίκτης δεν χάνεται απλά περιμένει εντολές!

5) ΠΑΙΚΤΗΣ 1 ΑΛΛΑΞΕ 4

Player 1 swap 4

Αλλάζουμε προτεραιότητα στο παίκτη...Στην ουσία έχουμε μεταβλητές έστω A και B και κάνουμε αλλαγές στις προτεραιότητες και καταχωρούμε τις αλλαγές στα A και B και δουλεύουμε με τα A και B ως ξεχωριστές "οντότητες". Δες τη Συγκρουση(Προστέθηκαν δυο ακόμα παραλλαγές, η ΠΑΝΩ και η ΥΠΟ όπου αντί να αλλάξουμε δυο εικόνες μεταφέρουμε μια πάνω από μια άλλη ή κάτω από μια άλλη. Στην ουσία μεταφέρονται οι προτεραιότητες, πχ το Παίκτης 1 πάνω 3 κάνει την εικόνα με προτεραιότητα 1 να έχει προτεραιότητα 3. Αυτό που κάνουμε μετά είναι να κρατάμε λογαριασμό ώστε να ξέρουμε για κάποιο παίκτη πια προτεραιότητα έχει.

6) ΠΑΙΚΤΗΣ 1, 5000,6000

μετακινούμε τον παίκτη στο 5000,6000

Μπορούμε να χειριστούμε τον κάθε Παικτη ως οθονη με την εντολή Επιπεδο

Δες το παράδειγμα εδώ Player

Ομάδα: ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ - DATABASES

Αναγνωριστικό: ΑΝΑΖΗΤΗΣΗ [SEARCH]

ΑΝΑΖΗΤΗΣΗ ονομα_βασης\$, πινακας\$,1,"πεδιο","like","τιμη"

ΑΝΑΖΗΤΗΣΗ ονομα_βασης\$, πινακας\$,1,"πεδιο",">=",12044

δουλεύει όπως η ανάκτηση αλλά εδώ βάζουμε και τον τρόπο που θα συγκριθεί το πεδίο με την τιμή

ΠΡΟΣΟΧΗ στην LIKE σε ερωτήματα SQL δεν χρησιμοποιείται το * αλλά το % (λειτουργούσε παλιά λόγω του DAO, αλλά λόγω της χρήσης του αντικειμένου ADO θέλει αυτή την αλλαγή, διαφορετικά το * θα το βλέπει ως χαρακτήρα, οπότε δεν θα γυρνάει κάτι)

Αναγνωριστικό: ΑΝΑΚΤΗΣΗ [RETRIEVE]

ΑΝΑΚΤΗΣΗ ονομα_βασης\$, πινακας\$, εγγραφή1, ονομα_πεδίου\$, τιμη_πεδίου\$

Ο πίνακας μπορεί να είναι ερώτημα SQL

π.χ.

Πρώτη Εγγραφή

1) ΑΝΑΚΤΗΣΗ ονομα_βασης\$, πινακας\$,1,"",""

2) ΑΝΑΚΤΗΣΗ ΒΑΣ\$, "SELECT * FROM [ΕΓΓΡΑΦΗ] WHERE

(month(ΗΜΕΡΑ)="+ΓΡΑΦΗ\$(ΜΝ)+") AND (year(ΗΜΕΡΑ)="+ΓΡΑΦΗ\$(ΕΤ)+")",1,"",""

3) ΑΝΑΚΤΗΣΗ "HELP2000","COMMANDS",1,"DESCRIPTION","HELP"

Δίνει στην κορυφή του σωρού τον αριθμό εγγραφών, και μια εγγραφή (την 1 που ζητήσαμε), όπου το DESCRIPTION είναι το HELP

4) ΑΝΑΚΤΗΣΗ "HELP2000","GROUP",1,"GROUPNUM",20

Δίνει στην κορυφή του σωρού τον αριθμό εγγραφών, και μια εγγραφή (την 1 που ζητήσαμε), όπου το GROUPNUM ισούται με 20.

Η βάση HELP2000 είναι η βάση δεδομένων της βοήθειας της γλώσσας. Από εκεί ήρθε αυτό το κείμενο!

Η SQL μας δίνει πολλές δυνατότητες, παρακάτω μας φτιάχνει έναν πίνακα με μια μόνο εγγραφή με ένα μόνο πεδίο το TOTAL το οποίο είναι το άθροισμα όλων των πεδίων ΠΟΣΟ από τον πίνακα-αρχείο ΕΓΓΡΑΦΗ που σε συνδυασμό με τον πίνακα-αρχείο ΛΟΓΑΡΙΑΣΜΟΙ επιλέγει

μόνο αυτές τις εγγραφές που έχουν λογαριασμούς με θετικό πρόσημο αλλά και τις εγγραφές που έχουν κοινό μήνα και έτος, δηλαδή βγάζει το σύνολο των εσόδων τον συγκεκριμένο μήνα του έτους!

```
ΑΝΑΚΤΗΣΗ ΒΑΣ$, "SELECT SUM(ΠΟΣΟ) AS TOTAL FROM [ΕΓΓΡΑΦΗ] INNER JOIN  
[ΛΟΓΑΡΙΑΣΜΟΙ] ON ΛΟΓΑΡΙΑΣΜΟΙ.ΚΩΔ=ΕΓΓΡΑΦΗ.ΚΩΔ WHERE  
ΛΟΓΑΡΙΑΣΜΟΙ.ΠΡΟΣΗΜΟ='+' AND month(ΕΓΓΡΑΦΗ.ΗΜΕΡΑ)='"+ΓΡΑΦΗ$(ΜΝ)+" AND  
year(ΕΓΓΡΑΦΗ.ΗΜΕΡΑ)='"+ΓΡΑΦΗ$(ΕΤ),1,"", ""
```

Το αποτέλεσμα της ανάκτησης δίνεται στον σωρό

Δες ΣΩΡΟΣ

Στο όνομα βάσης μπορούμε να δώσουμε μονοπάτι (ο τόπος που βρίσκεται...π.χ. c:\ονομα) αλλά δεν χρειάζεται ο τύπος αρχείο (mdb), μπαίνει αυτόματα.

Οι βάσεις δεδομένων που διαχειρίζεται η M2000 έχουν κλείδωμα, όχι για να μην ανοίγουν αλλά για να ξέρει η M2000 ότι είναι δικές τις βάσεις.

ΠΡΟΣΟΧΗ στην LIKE σε ερωτήματα SQL δεν χρησιμοποιείται το * αλλά το % (λειτουργούσε παλιά λόγω του DAO, αλλά λόγω της χρήσης του αντικειμένου ADO θέλει αυτή την αλλαγή, διαφορετικά το * θα το βλέπει ως χαρακτήρα, οπότε δεν θα γυρνάει κάτι)

Αναγνωριστικό: ΑΡΧΕΙΟ [TABLE]

ΑΡΧΕΙΟ ονομα_βάσης\$, ονομα_πινακα\$, ονομα_πεδίου1\$, τυπος_πεδίου, μηκος_πεδίου, ...
άλλα πεδία

Δεν πάνε όλοι οι τύποι για όλους τους τύπους βάσεων δεδομένων.

π.χ. για αριθμούς βάλτε το ΛΟΓΙΣΤΙΚΟ.

Τύποι πεδίου: ΛΟΓΙΚΟΣ, ΨΗΦΙΟ, ΑΚΕΡΑΙΟΣ, ΜΑΚΡΥΣ

ΛΟΓΙΣΤΙΚΟ, ΑΠΛΟΣ, ΔΙΠΛΟΣ, ΗΜΕΡΟΜΗΝΙΑ

ΔΥΑΔΙΚΟ, ΚΕΙΜΕΝΟ, ΥΠΟΜΝΗΜΑ

Αγγλικά:

BOOLEAN, BYTE INTEGER, LONG

CURRENCY, SINGLE, DOUBLE, DATEFIELD

BINARY, TEXT, MEMO

Παράδειγμα:

κ\$="βαση1"

ΒΑΣΗ κ\$

ΑΡΧΕΙΟ κ\$, "ΛΟΓΑΡΙΑΣΜΟΙ", "ΚΩΔ", ΜΑΚΡΥΣ, 0, "ΤΙΤΛΟΣ", ΚΕΙΜΕΝΟ, 40, "ΠΡΟΣΗΜΟ",
ΚΕΙΜΕΝΟ, 1

ΤΑΞΗ κ\$, "ΛΟΓΑΡΙΑΣΜΟΙ", "ΤΙΤΛΟΣ", ΑΥΞΟΥΣΑ

ΑΔΕΙΑΣΕ

ΣΕΙΡΑ "BENZINEΣ", "-", "ΑΓΟΡΕΣ ΥΛΙΚΩΝ", "-",

"ΑΤΟΜΙΚΑ", "-", "ΜΙΣΘΟΙ", "-", "ΕΞΩΤΕΡΙΚΑ", "+", "ΚΑΤΑΘΕΣΗ", "-", "ΑΝΑΛΗΨΗ", "+", "*", "**"

```

ΔΙΑΒΑΣΕ A$,B$
K=1
ΕΝΩ A$<>"*" {
ΠΡΟΣΘΗΚΗ κ$, "ΛΟΓΑΡΙΑΣΜΟΙ", K,A$, B$
ΔΙΑΒΑΣΕ A$,B$
K=K+1
}

```

Αναγνωριστικό: ΑΦΑΙΡΕΣΗ [DELETE]
ΑΦΑΙΡΕΣΗ ονομα_βασης\$, ονομα_πινακα\$, ονομα_πεδιου\$, τιμη_πεδιου\$
ΑΦΑΙΡΕΣΗ ονομα_βασης\$ διαγράφει την βάση

Εδώ δεν δίνουμε αριθμό εγγραφής, ούτε SQL, πρέπει να δώσουμε όνομα και τιμή πεδίου
ΑΦΑΙΡΕΣΗ "πρωτη", "ενα", "ΟΝΟΜΑ", "ΑΛΦΑ ΒΕΤΑ FROM ΔΕΛΤΑ"

Για την δεύτερη περίπτωση:
Πρώτα κοιτάει το σύστημα αν η βάση δεδομένων είναι δική του, αν έχει δηλαδή το σωστό κωδικό και μετά τη σβήνει. Δεν μπορεί να σβήσει οποιαδήποτε βάση!

Μπορούμε με μια εντολή ΕΚΤΕΛΕΣΗ να σβήσουμε πολλαπλές εγγραφές
Εκτέλεση "HELP2000", "DELETE FROM COMMANDS WHERE GROUPNUM =22;"

Αναγνωριστικό: ΒΑΣΗ [BASE]
1) Χρήση με αριθμητική παράμετρο (βάση πινάκων)
ΒΑΣΗ 1 πίνακες με βάση 1
ΒΑΣΗ 0 πίνακες με βάση 0
2) Χρήση με αλφαριθμητική παράμετρο (δημιουργία ή διαγραφή και δημιουργία κενής βάσης)
ΒΑΣΗ ονομα_βασης\$
η βάση θα είναι στο ΚΑΤ\$+ονομα_βασης\$+".mdb"

Δημιουργεί μια βάση διαγράφοντας όμως εκείνη που θα βρει με το ίδιο όνομα
Η βάση θα είναι κενή από αρχεία (πίνακες αλλά εδώ για να ξεχωρίσουμε από τους πίνακες στη μνήμη θα λέμε αρχεία τους πίνακες της βάσης). Κάθε αρχείο μπορεί να έχει αρκετά πεδία και μπορούμε να ορίσουμε ποια πεδία θα αποτελούν την τάξη ταξινόμησης.

δες εντολές

ΑΡΧΕΙΟ για να φτιάξεις τη δομή του αρχείου στη βάση

ΤΑΞΗ για να φτιάξεις τη δομή της ταξινόμησης

Με την ΔΟΜΗ διαβάζουμε τη δομή της Βάσης σε Αρχεία, και τη δομή των Αρχείων σε πεδία και πεδία τάξης.

Με την ΠΡΟΣΘΗΚΗ προσθέτουμε στοιχεία σε σειρές (εγγραφές)

Με την ΕΚΤΕΛΕΣΗ εκτελούμε μια εντολή για τη βάση (δεν περιμένουμε να μας επιστρέψει πεδία)

Με την ΑΝΑΚΤΗΣΗ διαβάζουμε στοιχεία (από εγγραφή)

Με την ΔΙΑΓΡΑΦΗ διαγράφουμε εγγραφή

Με την ΑΝΑΖΗΤΗΣΗ διαβάζουμε με επιλογή φίλτρου

Με την ΔΕΙΞΕ μεταφέρουμε μέρος στήλης στο μενού επιλογής.

Με την ΣΥΜΠΙΕΣΗ επισκευάζουμε και τακτοποιούμε την βάση

Με την ΚΛΕΙΣΕ ΒΑΣΗ κλείνουμε την σύνδεση με τη βάση

' προσθήκη από την έκδοση 7

Από την έκδοση 7 έχουμε αλλαγή του τύπου της βάσης mdb από Access 7 σε ADO αντικείμενα, προσαρμοσμένο όμως στην ίδια λειτουργία με τις άλλες εκδόσεις. Παράλληλα αυτή η έκδοση υποστηρίζει πεδία Unicode.

Με την ΚΛΕΙΣΕ ΒΑΣΗ ονομαβασης αποδεσμεύουμε μη αυτόματα το ADO αντικείμενο.

Αναγνωριστικό: ΒΑΣΗ.ΠΑΡΟΧΟΣ [DB.PROVIDER]

ΒΑΣΗ.ΠΑΡΟΧΟΣ "Microsoft.Jet.OLEDB.4.0", "Jet OLEDB", "PASSWORD"

το password είναι για τη κρυπτογράφηση της βάσης

Αναγνωριστικό: ΒΑΣΗ.ΧΡΗΣΤΗΣ [DB.USER]

ΒΑΣΗ.ΧΡΗΣΤΗΣ "dbusername","dbpassword"

Αναγνωριστικό: ΔΕΙΞΕ [VIEW]

ΔΕΙΞΕ ονομα_βασης\$, ονομα_πινακα\$, απο_εγγραφη, εως_εγγραφη

ΔΕΙΞΕ ονομα_βασης\$, "select distinct ονομα from ενα", 1,5

ΔΕΙΞΕ ονομα_βασης\$, "ενα", 1,5

Εμφανίζει μια λίστα με στοιχεία από πίνακα ή ερώτημα SQL. Τα απο_εγγραφη και εως_εγγραφη, είναι αριθμοί που δηλώνουν τις εγγραφές από τον πίνακα ή το ερώτημα που θέτουμε.

Δες εντολή ΕΠΙΛΟΓΗ

ΠΡΟΣΟΧΗ στην LIKE σε ερωτήματα SQL δεν χρησιμοποιείται το * αλλά το % (λειτουργούσε παλιά λόγω του DAO, αλλά λόγω της χρήσης του αντικειμένου ADO θέλει αυτή την αλλαγή, διαφορετικά το * θα το βλέπει ως χαρακτήρα, οπότε δεν θα γυρνάει κάτι)

Αναγνωριστικό: ΔΟΜΗ [STRUCTURE]

1) ΔΟΜΗ ονομα_βασης\$

2) ΔΟΜΗ ονομα_βασης\$, ονομα_πινακα\$

3) Δες τη Διάρθρωση η εντολή Δομή δημιουργεί δομές διάρθρωσης μνήμης (ως σύνθετους τύπους)

ΦΟΡΜΑ 80,50

ΒΑΣΗ.ΠΑΡΟΧΟΣ "" ' ρυθμίζει τον πάροχο στον βασικό (ενδέχεται να αλλαχθεί αυτόματα αν δεν μπορεί να ανοίξει μια βάση η M2000)

α=ΣΤΗΛΗ ' κρατάμε στο α το πλάτος νοητής στήλης στην οθόνη

ΤΥΠΩΣΕ \$(0,8) ' ορίζουμε πλάτος 8

ΕΠΙΛΟΓΗ ' σβήνουμε την λίστα επιλογών

ΑΡΧΕΙΑ + "mdb" ' βάζουμε στην λίστα επιλογών όλα τα αρχεία τύπου "mdb"

' από τον τρέχον κατάλογο

ΑΝ ΕΠΙΛΟΓΕΣ>0 ΤΟΤΕ { ' αν υπάρχουν επιλογές στην λίστα

ΤΥΠΩΣΕ "Επέλεξε ένα αρχείο:";

ΕΠΙΛΟΓΗ ! ' εμφάνιση της λίστας για να επιλέξουμε

ΑΝ ΕΠΙΛΟΓΗ>0 ΤΟΤΕ {

ΒΑΣΗ\$=ΕΠΙΛΟΓΗ\$(ΕΠΙΛΟΓΗ)

ΤΥΠΩΣΕ ΠΕΖ\$(ΒΑΣΗ\$)+".mdb"

ΤΥΠΩΣΕ "Δομή"

ΤΥΠΩΣΕ "Αρχείο:";

ΔΟΜΗ ΒΑΣΗ\$

ΔΙΑΒΑΣΕ ΑΡΧΕΙΑ ' εδώ δεν έχουμε την εντολή ΑΡΧΕΙΑ

' αλλά φτιάχνουμε μια μεταβλητή με το ίδιο όνομα και δίνουμε τιμή

' από τον σωρό

ΤΥΠΩΣΕ ΓΡΑΦΗ\$(ΑΡΧΕΙΑ)

ΑΝ ΑΡΧΕΙΑ>0 ΤΟΤΕ {

ΓΙΑ Ι=1 ΕΩΣ ΑΡΧΕΙΑ {

ΔΙΑΒΑΣΕ ΟΝΟΜΑΤΑ\$, ΤΑΞΕΙΣ

ΤΥΠΩΣΕ Ι,") "+ΟΝΟΜΑΤΑ\$

ΔΟΜΗ ΒΑΣΗ\$, ΟΝΟΜΑΤΑ\$

ΔΙΑΒΑΣΕ ΠΕΔΙΑ, ΤΑΞΕΙΣ

' αν υπάρχει αριθμός στο σωρό και όχι λέξη τότε...

ΑΝ ΠΕΔΙΑ>0 ΤΟΤΕ {

ΓΙΑ Π=1 ΕΩΣ ΠΕΔΙΑ {

```

        ΔΙΑΒΑΣΕ ΠΕΡΙΓΡΑΦΗ$, ΤΥΠΟΣ$, ΜΗΚΟΣ
        ΤΥΠΩΣΕ "",Π,") "+ΠΕΔΙΟ$(ΠΕΡΙΓΡΑΦΗ$,17),"",ΤΥΠΟΣ$, ΜΗΚΟΣ
    }
}
ΑΝ ΤΑΞΕΙΣ>0 ΤΟΤΕ {
    ΤΥΠΩΣΕ "", "ΤΑΞΗ:"
    ΔΙΑΒΑΣΕ ΠΕΔΙΑ
    ΑΝ ΠΕΔΙΑ>0 ΤΟΤΕ {
        ΓΙΑ Π=1 ΕΩΣ ΠΕΔΙΑ {
            ΔΙΑΒΑΣΕ ΠΕΡΙΓΡΑΦΗ$, ΤΥΠΟΣ$
            ΤΥΠΩΣΕ "",Π,") "+ΠΕΔΙΟ$(ΠΕΡΙΓΡΑΦΗ$,17),"",ΤΥΠΟΣ$
        }
    }
}
}
}
}
}
}
}
ΤΥΠΩΣΕ $(0,α)

```

\ Μπορούμε να χρησιμοποιούμε μπλοκ μέσα στη δομή και κάθε φορά στο τέλος του μπλοκ επιστρέφει ο μετρητής στην αρχή του μπλοκ, ώστε να βάλουμε και άλλα με σκοπό να έχουμε για ίδια μνήμη διαφορετικό τρόπο προσπέλασης (λέγονται unions σε δομές). Είναι προχωρημένο θέμα. Εδώ ένα απλό παράδειγμα (χωρίς unions):

\ ψηφίο (1 byte), ακέραιος (2 byte), μακρύς (4 byte) είναι αριθμοί χωρίς πρόσημο (πχ το ψηφίο παίρνει τιμές από 0 έως 255)

\ ο διπλός είναι 8 byte. Χαρακτήρες γράφουμε ανά δυο byte, για Unicode. Υπάρχει τρόπος να γράψουμε και να διαβάσουμε σε Ansi, με ένα Byte ανά χαρακτήρα.

\ η άλφα είναι δείκτης σε διάρθρωση και μπορεί να υπάρχει σε ομάδες, σε πίνακες, σε σωρούς, σε καταστάσεις. Η διάρθρωση μνήμης μπορεί να αλλάξει μέγεθος χωρίς να αλλάξει τύπο και χωρίς να σβήσει ότι έχει. Ουσιαστικά είναι ένα αντικείμενο που κρατάει μια περιοχή μνήμης.

```

Δομή γράμματα_ονόματος {
    γρα ως ακέραιος*20
}
Δομή η_δομή_μου {
    άλφα ως ψηφίο
    για_στοίχιση ως ψηφίο*3
    Κάπα ως μακρύς*20
    όνομα ως γράμματα_ονόματος*10
    Κάπα2 ως διπλός
}

```

Διάρθρωση κενή άλφα ως η_δομή_μου*100

Αναγνωριστικό: ΕΚΤΕΛΕΣΗ [EXECUTE]

1. Για βάσεις δεδομένων:

ΕΚΤΕΛΕΣΗ βάση, αλφαριθμητικό εκτέλεσης

ΕΚΤΕΛΕΣΗ βάση, αλφαριθμητικό εκτέλεσης, χρόνος εξόδου

εξ ορισμού χρόνος εξόδου = 30 (ελάχιστο 10)

Σε μια βάση δεδομένων μπορούμε να εκτελέσουμε εντολές όπως CREATE TABLE, INSERT INTO, SELECT

Αυτό γίνεται με την ΕΚΤΕΛΕΣΗ. Στη Select επιστρέφεται αντικείμενο Recordset, με αντίγραφο των στοιχείων που ζητάμε.

Μπορούμε να φτιάξουμε ένα Αρχείο στη βάση με αυτή την εντολή (με χρήση της Create Table), ή περισσότερα με χρήση του ; μεταξύ των εντολών.

Δες το παράδειγμα στα αγγλικά εδώ EXECUTE

2. Για εκτέλεση κώδικα

Εκτέλεση Κώδικα ΔιάρθρωσηΑ, 0

Εκτελεί το κώδικα σε Διάρθρωση που φτιάξαμε ως Διάρθρωση Κώδικα

Πρέπει να υπάρχει το 0xC3 ως επιστροφή από τη κλήση αλλιώς υπάρχει πρόβλημα!

\\ Φτιάχνουμε μια διάρθρωση μνήμης για δεδομένα

Διάρθρωση Δεδομένα ως Μακρύς*10

Επιστροφή Δεδομένα, 1:=500

Διάρθρωση Κώδικα Εκτελέσιμη ως ψηφίο*1024

\\ δες εδώ <https://defuse.ca/online-x86-assembler.htm>

\\ μετατρέπει την Assembly σε γλώσσα μηχανής

Διεύθυνση=0

\\ x86 Γλώσσα Μηχανής

\\ Το Δεδομένα(0) γυρνάει την διεύθυνση του πρώτου μακρύ (Long)

\\ Το Δεδομένα(1) γυρνάει την διεύθυνση του δεύτερου μακρύ

ΚωδΜακρύς(0xb8, 5100) ' mov eax, 0xb8 ' 10 στο eax

ΚωδΨηφΨηφ(0x83, 0xC0, 5) ' add eax, 0x5 ' πρόσθεσε 5 στο eax

ΚωδΨηφΜακρύς(0x3, 0x5, Δεδομένα(1)) ' add eax, [Δεδομένα(1)] ' πρόσθεσε στο eax το 500 από το δεύτερο μακρύ στο Δεδομένα

ΚωδΜακρύς(0xa3, Δεδομένα(0)) ' mov [Δεδομένα(0)], eax

Σημ : ΚωδΨηφ(0x31, 0xC0) ' xor eax, eax ' σημαίνει eax=0 ' χωρίς αυτό θα πάρουμε το 5605 στο M

```

Επιστροφή() ' Επιστροφή
\\ Τέλος Κώδικα
Δες οκ {
    Εκτέλεση Κώδικα Εκτελέσιμη, 0
}
M=Δυαδικό.Ακέραιο(Λάθος)
Δεκαεξ M
Τύπωσε M
Τύπωσε Λάθος, οκ
Τύπωσε Εκφρ(Δεδομένα, 0) ' 5605

```

```

Ρουτίνα Επιστροφή()
    Επιστροφή Εκτελέσιμη, Διεύθυνση:=0xC3
    Διεύθυνση++
Τέλος Ρουτίνας
Ρουτίνα ΚωδΨηφΨηφ()
    Επιστροφή Εκτελέσιμη, Διεύθυνση:=Αριθμός, Διεύθυνση+1:=Αριθμός,
    Διεύθυνση+2:=Αριθμός
    Διεύθυνση+=3
Τέλος Ρουτίνας
Ρουτίνα ΚωδΨηφ()
    Επιστροφή Εκτελέσιμη, Διεύθυνση:=Αριθμός, Διεύθυνση+1:=Αριθμός
    Διεύθυνση+=2
Τέλος Ρουτίνας
Ρουτίνα ΚωδΜακρύς()
    Επιστροφή Εκτελέσιμη, Διεύθυνση:=Αριθμός, Διεύθυνση+1:=Αριθμός ως Μακρύς
    Διεύθυνση+=5
Τέλος Ρουτίνας
Ρουτίνα ΚωδΨηφΜακρύς()
    Επιστροφή Εκτελέσιμη, Διεύθυνση:=Αριθμός, Διεύθυνση+1:=Αριθμός,
    Διεύθυνση+2:=Αριθμός ως Μακρύς
    Διεύθυνση+=6
Τέλος Ρουτίνας

```

Αναγνωριστικό: ΕΠΙΣΤΡΟΦΗ [RETURN]

ΕΠΙΣΤΡΟΦΗ "πρωτη","SELECT * FROM ενα WHERE ONOMA LIKE 'MAGELAN%'" , , "EARTH THERE","DATA2","DATA3"

ΠΡΟΣΟΧΗ στην LIKE σε ερωτήματα SQL δεν χρησιμοποιείται το * αλλά το % (λειτουργούσε παλιά λόγω του DAO, αλλά λόγω της χρήσης του αντικειμένου ADO θέλει αυτή την αλλαγή,

διαφορετικά το * θα το βλέπει ως χαρακτήρα, οπότε δεν θα γυρνάει κάτι)

Ο πίνακας ενα έχει τέσσερα πεδία και εμείς επιστρέφουμε σε μια εγγραφή τα τρία από τα τέσσερα πεδία. Το πρώτο πεδίο το αφήνουμε ως έχει (είναι το όνομα). Προσέξτε τα δυο συνεχόμενα κόμματα.

ΕΠΙΣΤΡΟΦΗ "ΟΝΟΜΑΒΑΣΗΣ", "ΟΝΟΜΑΑΡΧΕΙΟΥ" ΣΤΟ ΑΡΙΘΜΟΣΚΑΤΑΓΡΑΦΗΣ, ΠΕΔΙΟ1, ΠΕΔΙΟ2,....

μπορούμε να παραλήψουμε ένα πεδίο με διπλό κόμμα.

Αυτή την εντολή θα την χρησιμοποιήσουμε όταν το αρχείο χρησιμοποιείται ως πίνακας αναφοράς, οπότε η θέση της καταγραφής είναι σημαντική αφού καταχωρείται σε άλλα αρχεία στην βάση.

Η εντολή Επιστροφή χρησιμοποιείται και για άλλη λειτουργία.

1) Για επιστροφή από απλές ρουτίνες

Διαμέσου αλφα

Τέλος

αλφα:

Τύπωσε "οκ"

Επιστροφή

2) Με αντικείμενα Κατάσταση, Πίνακας, Σωρός

Κατάσταση A=1:="Γεια", "αλλο":="Ναι"

\ \ οι καταστάσεις καταχωρούν βάσει κλειδιών

Επιστροφή A, "αλλο":="Όχι", 1:="Γειά σου"

Τύπωσε A\$("αλλο"), A\$(1)

\ \ ο AA είναι επαναλήπτης

AA=Κάθε(A)

Τύπωσε "Κλειδί", "Τιμή"

Ενώ AA { Τύπωσε εκφρ\$(AA!), εκφρ\$(AA)}

B=(1,2,3,4,5)

\ \ Η Πίνακας(B) φέρνει το πρώτο στοιχείο

Επιστροφή B,3:=400, 0:=Πίνακας(B)+1, 1:=Πίνακας(B,1)**2 ' με βάση 0 , το 3 είναι το 4ο στοιχείο

Τύπωσε B

Τύπωσε Πίνακας(B,3)

\ \ Ο BB είναι επαναλήπτης

BB=Κάθε(B)

Ενώ BB { Τύπωσε γραφή\$(Πίνακας(BB),"00000")}

\ \ Ο σωρός έχει κλειδιά θέσεις και μετακινεί εύκολα στοιχεία χωρίς αντιγραφές

M=Σωρός:=1,2,3,4,5

Επιστροφή M, 4:=400, 1:=ΤιμήΣωρού(M)+1, 2:=ΤιμήΣωρού(M,2)**2 ' Οι σωροί έχουν βάση 1
 Τύπωσε M
 Τύπωσε ΤιμήΣωρού(M,4)
 \ \ Ο MM είναι επαναλήπτης
 MM=Κάθε(M,-1,1) \ \ ή Κάθε(M Τέλος Έως Αρχή) ή Κάθε(M -1 Εως 1)
 Ενώ MM {Τύπωσε Γραφή\$(ΤιμήΣωρού(MM))}
 Σωρός M {Φέρε 4 : ΦέρεΠίσω 5} \ \ μετακίνηση του 4ου στοιχείου στην 5η θέση
 Τύπωσε M

Αναγνωριστικό: ΠΡΟΣΘΗΚΗ [APPEND]

1) Για πίνακες σε βάσεις δεδομένων

ΠΡΟΣΘΗΚΗ ονομα_βασης\$, ονομα_πινακα\$, πεδιο1, πεδιο2\$, πεδιο3\$, πεδιο4,....

Παραδείγματα

ΠΡΟΣΘΗΚΗ ΒΑΣ\$, "ΕΓΓΡΑΦΗ", ΠΡΑΞΕΙΣ, ΓΡΑΦΗ\$(ΣΗΜΕΡΑ, "SHORT DATE"), ΚΩΔΙΚΟΣ, ΠΟΣΟ

ΠΡΟΣΘΗΚΗ ΒΑΣ\$, "ΟΦΕΙΛΕΤΕΣ", ΚΩΔ.ΠΕΛ, ΠΡΑΞΕΙΣ, ΤΙΜΗ(ΑΠΟΚ\$(ΠΟΣΟ\$))

2) Για Καταστάσεις και δείκτες σε πίνακες

α=(1,2,3,4)

Προσθήκη α, (5,6,7,8), (9,10,11,12) \ \ ,

Κατάσταση β=1,2,3,4:=300,5

Προσθήκη β, 100:=500, "Γεια":=1000, "οκ":="εντάξει"

Τύπωσε β\$("οκ"), β(4)

Τύπωσε β

Αναγνωριστικό: ΣΥΜΠΙΕΣΗ [COMPRESS]

ΣΥΜΠΙΕΣΗ ονομα_βασης\$

το όνομα βάσης πρέπει να έχει το τύπο π.χ. .mdb

χρησιμοποίησε το ΚΛΕΙΣΕ ΒΑΣΗ κατ\$+ονομα_βασης\$+".mdb"

για να κλείσεις τυχόν ανοικτή σύνδεση με τη βάση

Αναγνωριστικό: ΤΑΞΗ [ORDER]

ΤΑΞΗ κ\$, "ΛΟΓΑΡΙΑΣΜΟΙ", "ΤΙΤΛΟΣ", ΑΥΞΟΥΣΑ

Μπορούμε να δηλώσουμε και ΦΘΙΝΟΥΣΑ και μέχρι δέκα πεδία συνολικά

Χρειάζεται μόνο μια φορά να δηλωθεί η τάξη

Αν δεν δώσουμε πεδία τότε διαγράφουμε την ταξινόμηση για αυτόν τον πίνακα (εδώ ΛΟΓΑΡΙΑΣΜΟΙ)

Ομάδα: ΗΧΟΙ ΚΑΙ ΤΑΙΝΙΕΣ - SOUNDS AND MOVIES

Αναγνωριστικό: ΕΝΤΑΣΗ [VOLUME]

Μεταβάλλουμε την ένταση του ήχου

ΕΝΤΑΣΗ 100 ' το μέγιστο

ΕΝΤΑΣΗ 0 ' το ελάχιστο

ΕΝΤΑΣΗ 100 , 50 ' ΚΑΝΟΝΙΚΟ (ΜΙΣΟ/ΜΙΣΟ) όπως ΕΝΤΑΣΗ 100

ΕΝΤΑΣΗ 100 , 0 ' ΔΕΞΙΟ ΚΑΝΑΛΙ ΜΟΝΟ

ΕΝΤΑΣΗ 100 , 100 ' ΑΡΙΣΤΕΡΟ ΚΑΝΑΛΙ ΜΟΝΟ

ΕΝΤΑΣΗ 100 , 25 ' ΑΡΙΣΤΕΡΟ ΚΑΝΑΛΙ 100%, ΔΕΞΙΟ 50%

ΕΝΤΑΣΗ 100 , 75 ' ΑΡΙΣΤΕΡΟ ΚΑΝΑΛΙ 50%, ΔΕΞΙΟ 100%

ΕΝΤΑΣΗ 50 , 25 ' ΑΡΙΣΤΕΡΟ ΚΑΝΑΛΙ 50%, ΔΕΞΙΟ 25%

ΕΝΤΑΣΗ 50 , 75 ' ΑΡΙΣΤΕΡΟ ΚΑΝΑΛΙ 25%, ΔΕΞΙΟ 50%

ΕΝΤΑΣΗ 25 , 50 ' ΑΡΙΣΤΕΡΟ ΚΑΝΑΛΙ 25%, ΔΕΞΙΟ 25%

η μεταβλητή ΕΝΤΑΣΗ επιστρέφει τη πρώτη τιμή (την μέγιστη ένταση). Αυτή η τιμή διαβάζεται απευθείας από τα Windows, για το τρέχον πρόγραμμα (περιβάλλον M2000)

Διαβάζουμε την ένταση από την μεταβλητή ΕΝΤΑΣΗ

Αναγνωριστικό: ΕΠΙΛΕΞΕ.ΟΡΓΑΝΟ [CHOOSE.ORGAN]

ΕΠΙΛΕΞΕ.ΟΡΓΑΝΟ

ΑΝ ΕΠΙΛΟΓΗ>0 ΤΟΤΕ Α=ΕΠΙΛΟΓΗ

Αναγνωριστικό: ΗΧΟΓΡΑΦΗΣΗ [SOUNDREC]

1) ΗΧΟΓΡΑΦΗΣΗ ' Ξεκινάει 8bit, 11025khz, 1 channel.

2) ΗΧΟΓΡΑΦΗΣΗ ΝΕΑ "filename.wav", 41000 stereo hifi

χωρίς να ξεκινάει

stereo για 2 κανάλια, hifi για 16bit

χωρίς stereo θα πάρουμε 1 κανάλι, χωρίς Hifi θα πάρουμε 8 bit

μπορούμε να επιλέξουμε 11025, 22050, 41000 ή άλλη τιμή για δειγματοληψία

3)ΗΧΟΓΡΑΦΗΣΗ ΕΙΣΑΓΩΓΗ

Από τη τρέχουσα θέση

4)ΗΧΟΓΡΑΦΗΣΗ OverWrite

ηχογράφηση από την αρχή πατώντας πάνω στην εγγραφή

5) ΗΧΟΓΡΑΦΗΣΗ ΑΠΟΚΟΠΗ 10 ΕΩΣ 20

κόβει από 10ms μέχρι και 20ms

6) ΗΧΟΓΡΑΦΗΣΗ ΔΙΑΚΟΠΗ

σταματάει την ηχογράφηση

7) ΗΧΟΓΡΑΦΗΣΗ ΔΟΚΙΜΗ

παίζει ότι έχουμε γράψει

8) ΗΧΟΓΡΑΦΗΣΗ ΘΕΣΗ 10

αν γίνεται ηχογράφηση τότε αλλάζουμε θέση και συνεχίζεται η ηχογράφηση

αν δεν γίνεται ηχογράφηση απλά μετακινούμε τον "δρομέα"

9) ΗΧΟΓΡΑΦΗΣΗ ΣΩΣΕ

με χρήση του ονόματος που είχαμε αρχικά δώσει

10) ΗΧΟΓΡΑΦΗΣΗ ΣΩΣΕ "filename.wav"

σώνουμε με χρήση του ονόματος που μόλις πήραμε!

21) ΗΧΟΓΡΑΦΗΣΗ ΤΕΛΟΣ

κλείνει το αντικείμενο και αποδίδει μνήμη στο σύστημα.

ΕΝΤΑΣΗ 100

ΗΧΟΓΡΑΦΗΣΗ new dir\$+"alfa.wav", 22050 STEREO HIFI

ΑΝΑΛΥΤΗΣ

ΤΥΠΩΣΕ "εκκίνηση εγγραφής"

ΗΧΟΓΡΑΦΗΣΗ ΕΙΣΑΓΩΓΗ

ΕΠΑΝΕΛΑΒΕ

ΤΥΠΩΣΕ over \$(1), ΕΠΑΝ\$(1), ΗΧΟΓΡΑΦΗΣΗΣ.ΕΠΙΠΕΔΟ)

ΤΥΠΩΣΕ Part \$(3, ΠΛΑΤΟΣ), ΦΟΡΤΟΣ 1000

ΑΝΑΝΕΩΣΗ 100

if ΠΑΤΗΜΕΝΟ(32) ΤΟΤΕ ΕΞΟΔΟΣ

ΑΝΑΜΟΝΗ 1

ΠΑΝΤΑ

M=ΦΟΡΤΟΣ

ΗΧΟΓΡΑΦΗΣΗ ΔΙΑΚΟΠΗ

ΤΥΠΩΣΕ ΠΑΝΩ

ΤΥΠΩΣΕ "ΤΕΛΟΣ ΗΧΟΓΡΑΦΗΣΗ - ΑΡΧΗ ΔΟΚΙΜΗΣ"

ΕΝΤΑΣΗ 100

ΑΝΑΛΥΤΗΣ

ΗΧΟΓΡΑΦΗΣΗ ΔΟΚΙΜΗ

ΕΠΑΝΕΛΑΒΕ

ANAMONH 1
MEΧΡΙ ΦΟΡΤΟΣ>M
ΤΥΠΩΣΕ "ΤΕΛΟΣ ΔΟΚΙΜΗΣ"
ΗΧΟΓΡΑΦΗΣΗ ΣΩΣΕ
ΤΥΠΩΣΕ "Η ΗΧΟΓΡΑΦΗΣΗ ΣΩΘΗΚΕ ΣΤΟ ΔΙΣΚΟ"
ΗΧΟΓΡΑΦΗΣΗ ΤΕΛΟΣ

Αναγνωριστικό: ΗΧΟΣ [SOUND]
ΗΧΟΣ αρχείο_ήχου_wav
ΗΧΟΣ "" σταματάει το προηγούμενο ήχο

Αναγνωριστικό: ΛΟΓΟΣ [SPEECH]
Λογος "Agglikla"
Λογος "Pediki phoni" !
Λογος "Hena Hena Grama" #

Λογος "Agglikla", φωνη
Λογος "Pediki phoni" !. φωνη
Λογος "Hena Hena Grama" #, φωνη

ΤΥΠΩΣΕ Λογος
τυπωνει τον αριθμό φωνών

Για φωνη= 1 εως Λογος {
 Τυπωσε Λογος\$(φωνη)
}

Τυπώνει τα ονόματα των φωνών

Αναγνωριστικό: ΜΕΛΩΔΙΑ [TUNE]
ΜΕΛΩΔΙΑ 100, "CAC"
ΜΕΛΩΔΙΑ "C3BC#"
διάστημα μέσα στο αλφαριθμητικό είναι παύση, το νούμερο μας λει τον αριθμό οκτάβας

παίζει νότες διαμορφώνοντας εσωτερικά μια σειρά από τόνους. Δες την ΤΟΝΟΣ
προσοχή σταματάει τα νήματα!
Ανώτερο σύστημα που δουλεύει σε ξεχωριστά νήματα δημιουργείται με τις εντολές ΦΩΝΗ και
ΠΑΙΞΕ όπου εκτός

```
Module A {  
  Module AA {  
    For i=1 To 10 {  
      Tone 50, 150  
      Tone 25, 300  
    }  
  }  
  For i=1 To 5 {  
    AA  
    Wait 200  
    AA  
    Wait 800  
  }  
}  
Module B {  
  For i = 1 To 10 {  
    TUNE "cac "  
    Wait 200  
  }  
}  
Module C {  
  For i=100 To 8000 Step 20 {  
    Tone 2,i  
  }  
}
```

Αναγνωριστικό: ΜΟΥΣΙΚΗ [MUSIC]
1) ΜΟΥΣΙΚΗ ΚΑΤ\$+"ΒΗΤΑ.MID", άλλο_κομμάτι\$, ...

Μπορούμε άμεσα να σταματήσουμε την μουσική με μια εντολή χωρίς όρισμα
ΜΟΥΣΙΚΗ
Η μεταβλητή ΜΟΥΣΙΚΗ δίνει ΑΛΗΘΕΣ αν η μουσική εκτελείτε ή ΨΕΥΔΕΣ αν έχει πάψει.

π.χ. ΕΝΩ ΜΟΥΣΙΚΗ { ΤΥΠΩΣΕ "ΜΠΡΑΒΟ " ; }

Η μεταβλητή ΔΙΑΡΚΕΙΑ: Δίνει την διάρκεια σε δευτερόλεπτα ενός πολυμέσου (δίνει μέχρι και τα χιλιοστά) κατά την εκτέλεση αυτού.

2) ΜΟΥΣΙΚΗ ΚΑΤ\$+"ΒΗΤΑ.MID";

Ο μεταφραστής περιμένει να τελειώσει η μουσική και μετά συνεχίζει την ροή προγράμματος.

Η εντολή ΜΟΥΣΙΚΗ παίζει και MP3, wav κλπ.

Δες το ΜΟΥΣΙΚΗ.ΜΕΤΡΗΤΗΣ

3) ΜΟΥΣΙΚΗ -1

σταματάει τη μουσική

4) ΜΟΥΣΙΚΗ 1.2

προχωράει τη μουσική στο δευτερόλεπτο 1.2

5) ΜΟΥΣΙΚΗ 0

ξεκινάει πάλι την μουσική

Αναγνωριστικό: ΜΠΙΠ [BEEP]

1) εκτελεί τον ήχο ΜΠΙΠ

ΜΠΙΠ

2) εκτελεί τον ήχο αριθμό 10 των windows

ΜΠΙΠ 32

Αναγνωριστικό: ΠΑΙΞΕ [PLAY]

Φωνη 1, 500, "c@2dc @2ef"

Παιξε 1, 19

Score 1, 500, "c@2dc @2ef"

Play 1, 19

παίζει την φωνή 1 με το όργανο 19

Παιξε 0 κλείνει όλες τις φωνές

Δες Φωνη

Δες Επιλεξε.Οργανο

Μπορούμε να βάλουμε και τις 16 φωνές να παίξουν μαζί π.χ. Παιξε 1,19,2,19,3,20,4,20,...
Κάθε φωνή έχει κάποιο χρόνο να τελειώσει. Αν ξαναδώσουμε την Play ενώ δεν έχουν τελειώσει
η ακόμα οι φωνές το παίξιμό τους θα δημιουργηθεί ένα μπάχαλο γιατί το σύστημα φτιάχνει νήμα
για κάθε κλήση φωνής-οργάνου και θα υπάρχουν πάνω από ένα νήμα για την ίδια φωνή. Το
σύστημα δεν κολλάει αλλά στέλνει εντολές πιο γρήγορα με συνέπεια να ακούγονται οι νότες με
μικρότερη διάρκεια.

Ηλεκτρική Κιθάρα:

score 1, 350, "c@2dc @2ef"

play 1, 29

Κουδούνι!!!

έχω βάλει το 2 μετά το C που σημαίνει 2η οκτάβα, πιο μπάσο δηλαδή από την 3η οκτάβα)

το @2 λέει να παίζει μισής χρονικής αξίας, και το διάστημα μετά τις dc λέει παύση μισής
χρονικής αξίας, επειδή έχει το @2 μετά

score 1, 350, "c2@2dc @2ef"

play 1, 9

αν δώσουμε τα παρακάτω θα παίξει η ταινία, η μουσική, και ο ήχος...μαζί (ο ήχος θα ξεκινήσει
ένα δευτερόλεπτο πιο μετά)

movie "second" \ \ αν υπάρχει το second.avi

wait 1000

sound "ringin" \ \ αν υπάρχει το ringin.wav

\ \ Πρόγραμμα Πιάνο (με συγχορδίες)

Πίνακας Νοτα1\$(10,3), Νοτα2\$(10,3)

ΓέμισεΠίνακα()

Φόρμα 60,30

Πένα 14

Οθονη 5

Διπλά

Αναφορά 2, "Πιανάκι 005"

Κανονικά

Αναφορά 2,{Οδηγίες

1-Έξοδος 3-Ξυλόφωνο 4-Πιάνο 5-Σαξόφωνο 8-Πλήκτρα/Νότες 9-Ρυθμός Ναι 0-Ρυθμός Όχι

Διάστημα - Δυναμώνει τις νότες

-- Γιώργος Καρράς --

```

}
Σχέδιο.Νημάτων Διαδοχικό
Παίξε 0 \ \ καθαρίζει το σύστημα
Γενική πλ$=" ", τέμπο=300, οργ=5, φ=0, εντ$="V90", φων(18), χρη(17) ' 0 ..17,
v=1
Για i=1 έως 16 { χρη(i)=Αληθές }
χρη(10)= Ψευδές \ \ για την drum machine
\ \ υπολογισμός θέσης του κλαβιέ!
μμ=2*(χ.σημεια Δια 14)
μμ2= μμ δια 2
κκ=γραμμή/ύψος*υ.σημεια
κκ1=υ.σημεια/ύψος*8
γραμ=υ.σημεια/ύψος*1.5
Οθόνη , Γραμμή+9
Τμήμα Πλήκτρο {
    Διάβασε α$, πατ, επιλ
    αν πατ τότε {
        Αν Θέση(πλ$,α$+"-")>0 τότε έξοδος
        Επομ()
        φων(επιλ)=φ
        Τύπωσε α$, φ
        Φωνή φ, τεμπο, α$+εντ$ : πλ$<=πλ$+α$+"-": Παίξε φ, οργ
    } αλλιώς {
        πλ$<=αλλαγή$(α$+"-", "", πλ$)
        Αν φων(επιλ)>0 τότε {
            Παίξε φων(επιλ), 0
            χρη(φων(επιλ))~
            φων(επιλ)=0
        }
    }
}
Ρουτίνα Επομ()
τοπικη i
για i=1 έως 16 {
    αν χρη(i) τότε έξοδος
}
αν i<17 τότε { φ<=i : χρη(i)~ } αλλιώς φ<=1: χρη(1)=αληθές : Τύπωσε "!!!!!!!"
Τέλος Ρουτίνας
}
\ \ κανάλι 10 είναι το drum machine
Νήμα { Φωνή 10,400,"CV90CC ab Cd eCC" : Παίξε 10,1 } ως Λ κάθε 60
Νήμα {
    πλήκτρο "A#2", πατημένο(κωδ("A")), 1

```

```

πλήκτρο "B2", πατημένο(κωδ("Z")), 2
πλήκτρο "C3", πατημένο(κωδ("X")), 3
πλήκτρο "C#3", πατημένο(κωδ("D")), 4
πλήκτρο "D3", πατημένο(κωδ("C")), 5
πλήκτρο "D#3", πατημένο(κωδ("F")), 6
πλήκτρο "E3", πατημένο(κωδ("V")), 7
πλήκτρο "F3", πατημένο(κωδ("B")), 8
πλήκτρο "F#3", πατημένο(κωδ("H")), 9
πλήκτρο "G3", πατημένο(κωδ("N")), 10
πλήκτρο "G#3", πατημένο(κωδ("J")), 11
πλήκτρο "A3", πατημένο(κωδ("M")), 12
πλήκτρο "A#3", πατημένο(κωδ("K")), 13
\\ https://msdn.microsoft.com/en-us/library/windows/desktop/dd375731(v=vs.85).aspx
πλήκτρο "B3", πατημένο(0xBC), 14 \\ VK_OEM_COMMA
πλήκτρο "C4", πατημένο(0xBE), 15 \\ VK_OEM_PERIOD
πλήκτρο "C#4", πατημένο(0xBA), 16 \\ VK_OEM_1
πλήκτρο "D4", πατημένο(0xBF), 17 \\ VK_OEM_2
} ως ΑΛ κάθε 50
Τύπωσε "οκ"
Νηματα
Νήμα Λ Κάθε 6000
Κύριο Έργο 100 {
    Εμφ_Κλαβιέ(μμ, κκ, μμ2, κκ1)
    Ανανέωση 1000
    αν πατημένο(κωδ("1")) τότε έξοδος
    αν πατημένο(κωδ("3")) τότε οργ<=14 : τέμπο<=100 ' ξυλόφωνο
    αν πατημένο(κωδ("4")) τότε οργ<=5 : τέμπο<=300 ' Πιάνο
    αν πατημένο(κωδ("5")) τότε οργ<=65 : τέμπο<=5000 ' Σαξόφωνο
    αν πατημένο(κωδ("8")) τότε ν=1-ν
    αν πατημένο(κωδ("9")) τότε Νήμα Λ Ξεκίνα
    αν πατημένο(κωδ("0")) τότε Νήμα Λ Κράτα
    αν πατημένο(32) τότε { εντ$ <= "V127" } Αλλιώς εντ$ <= "V90"
    Τύπωσε "-----"
}
Νήματα Σβήσε
Τύπωσε "Τέλος"
Ρουτίνα Εμφ_Κλαβιέ(π0,υ0, π1, υ1)
    Κλαβιέ(π0, υ0, π1, υ1, 0, &Νοτα1$())
    Κλαβιέ(π0-π1/2,υ0,π1,υ1*2/3, -1, &Νοτα2$())
    Ανανέωση 1000
Τέλος Ρουτίνας
Ρουτίνα Κλαβιέ(π0, υ0,π1, υ1, π3, &Ν$())
    Ένωσε Ν$() στο Ν()

```

π3-!

Τοπικές κ=-1, ι

Για ι=π0 εως 9*π1+π0 ανά π1

κ++

Αν Ν\$(κ,0)<>"" Τότε {

θέση ι+π3*π1/6, υ0

Αν π3 τότε {

βαψε π1-π3*π1/3-15,υ1-15, 7* (1-(φων(N(κ,2))>0))+1, 0,1

} Αλλιώς {

βαψε π1-15,υ1-15,15,7* (1-(φων(N(κ,2))=0))+1,1

}

θέση ι+π3*π1/6, υ0

βαψε @ π1-π3*π1/3,υ1,2,1

θέση ι+π3*π1/6, υ0+υ1-γραμ

πένα π3*15 { βαψε @ π1-π3*π1/3,γραμ,5,Ν\$(κ,ν)}

}

Επόμενο ι

Τέλος Ρουτίνας

Ρουτίνα ΓέμισεΠίνακα()

Τοπικές ν,π, ν\$, κ\$, νο

Σωρός Νέος {

Σειρά "A#2", "A", 1, 1, 2

Σειρά "B2", "Z", 2, 1, 1

Σειρά "C3", "X", 3, 2, 1

Σειρά "C#3", "D", 4, 3, 2

Σειρά "D3", "C", 5, 3, 1

Σειρά "D#3", "F", 6, 4, 2

Σειρά "E3", "V", 7, 4, 1

Σειρά "F3", "B", 8, 5, 1

Σειρά "F#3", "H", 9, 6, 2

Σειρά "G3", "N", 10, 6, 1

Σειρά "G#3", "J", 11, 7, 2

Σειρά "A3", "M", 12, 7, 1

Σειρά "A#3", "K", 13, 8, 2

Σειρά "B3", ",", 14, 8, 1

Σειρά "C4", ".", 15, 9, 1

Σειρά "C#4", ";", 16, 10, 2 \ 10 μικρά - λείπουν μερικά

Σειρά "D4", "/", 17, 10, 1 \ 10 μεγάλα πλήκτρα

Ενώ όχι κενό {

Διάβασε ν\$, κ\$, νο, ν, π

Αν π=1 τότε {

νοτα1\$(ν-1,0):= ν\$, κ\$, νο

} Αλλιώς {

```

        νοτα2$(v-1,0):= v$, κ$, νο
    }
}
}
Τέλος Ρουτίνας

```

Αναγνωριστικό: TAINIA [MOVIE, MEDIA]

Όπως και η ΜΟΥΣΙΚΗ μπορεί να εκτελεί σύγχρονα ή ασύγχρονα ταινίες (με αναμονή του προγράμματος ή όχι)

Επίσης υπάρχουν:

Η μεταβλητή TAINIA (ΑΛΗΘΕΣ αν παίζει ή ΨΕΥΔΕΣ αν έχει τελειώσει)

Η μεταβλητή ΔΙΑΡΚΕΙΑ

Αν κάνουμε απλό αριστερό κλικ του ποντικιού πάνω στην ταινία τότε αυτή κλείνει αμέσως.

1) Ασύγχρονη λειτουργία

TAINIA αρχαιο1\$, αρχαιο2\$

TAINIA (κλείνει αμέσως την ταινία)

2) Σύγχρονη λειτουργία

TAINIA αρχαιο1\$, αρχαιο2\$;

3) Προβολή με αλλαγή μεγέθους και θέσης

Μέχρι την έκδοση 2.1 οι ταινίες έβγαιναν στο κέντρο της οθόνης και στο μέγεθος που είχαν αρχικά.

TAINIA θεσηX, θεσηY, ταινια\$

TAINIA θεσηX, θεσηY, πλατος, ταινια\$

TAINIA θεσηX, θεσηY, πλατος, υψος, ταινια\$

Τα θεσηX, θεσηY έχουν σχέση με το Παραθυρο

Αν θέλουμε να τα κάνουμε σχετικά με το επίπεδο που τυπώνουμε τότε πρέπει να προσθέτουμε τα κινηση.χ και κινηση.ψ

Αν μετακινηθεί το επίπεδο...δεν μετακινείται η ταινία μαζί,πρέπει να δώσουμε εντολή για να μετακινηθεί. Δες νέες εντολές:

TAINIA θεσηX, θεσηY

TAINIA θεσηX, θεσηY, πλατος, υψος (από την έκδοση 6 μπορούμε να αλλάξουμε μέγεθος στην ταινία όποτε θέλουμε, καθορίζοντας διαφορετική αναλογία πλάτους με ύψος αλλάζουμε το πώς βλέπουμε την ταινία π.χ. 16/9 ή 4/3

Μπορούμε να δώσουμε και άλλες ταινίες να παίζουν σε σειρά αλλά όλες θα βγουν στο ίδιο μέγεθος και θέση.

Από την έκδοση 6 έχει επεκταθεί η λειτουργία της εντολής!

Εδώ είναι ένα πρόγραμμα που ανοίγει μια ταινία και τυπώνει δυο καρτέ από αυτήν.

ΤΑΙΝΙΑ ΦΟΡΤΩΣΕ "SECOND" ' φορτώνει την ταινία σε κατάσταση παύσης

ΤΑΙΝΙΑ ΣΤΟ 2 ' στο 2 δευτερόλεπτο (μπορούμε να δώσουμε δεκαδικά)

ΤΑΙΝΙΑ 1000+ΚΙΝΗΣΗ.Χ,1000+ΚΙΝΗΣΗ.Ψ ' βάζει το παράθυρό στο 1000,1000 του

επιπέδου που τυπώνουμε

ΤΑΙΝΙΑ ΔΕΙΞΕ ' Μας δείχνει την ταινία (είναι σε παύση)

ΑΝΑΜΟΝΗ 100 ' Στα Windows 10 θέλουν λίγο χρόνο για να δείξουν, αλλιώς δεν θα πάρουμε την εικόνα!

Α\$=ΠΑΡΕΚΑΡΕ\$ ' αντιγράφει το καρτέ (μονο αν βλέπουμε την ταινία)

ΑΝΤΙΓΡΑΨΕ 1000,1000 ΜΕ Α\$ ' αντιγράφουμε το καρτέ πίσω ακριβώς από την ταινία (η οποία εμφανίζεται σε δικό της επίπεδο)

ΤΑΙΝΙΑ ΚΡΥΨΕ ' τώρα κρύβουμε την ταινία

'(αλλά δεν το καταλαβαίνουμε γιατί βλέπουμε την ίδια εικόνα από

πίσω τυπωμένη)

ΤΑΙΝΙΑ ΣΤΟ 3 ' πάμε στο τρίτο δευτερόλεπτο

ΤΑΙΝΙΑ 1000+ΚΙΝΗΣΗ.Χ,1000+ΚΙΝΗΣΗ.Ψ ' καθορίζουμε άλλη θέση (μπορούμε να αλλάξουμε και το ύψος και πλάτος)

ΤΑΙΝΙΑ ΔΕΙΞΕ ' εμφανίζουμε το καρτέ

ΑΝΑΜΟΝΗ 100

Α\$=ΠΑΡΕΚΑΡΕ\$ ' το καταχωρούμε

ΑΝΤΙΓΡΑΨΕ 5000,1000 ΜΕ Α\$ ' και το τυπώνουμε

ΤΑΙΝΙΑ ' κλείνουμε την ταινία

Εντολές:

MOVIE LOAD "FILENAME OR PATH AND FILENAME"

MOVIE PLAY

MOVIE PAUSE

MOVIE HIDE

MOVIE SHOW

MOVIE RESTART

MOVIE TO 0.01

ΤΑΙΝΙΑ ΦΟΡΤΩΣΕ "SECOND" 'ΥΠΑΡΧΕΙ ΤΟ ΑΡΧΕΙΟ SECOND.AVI ΣΤΟΝ ΚΑΤΑΛΟΓΟ ΚΑΤ\$

ΤΑΙΝΙΑ ΠΑΙΞΕ

ΤΑΙΝΙΑ ΚΡΑΤΗΣΕ

ΤΑΙΝΙΑ ΚΡΥΨΕ

ΤΑΙΝΙΑ ΔΕΙΞΕ

ΤΑΙΝΙΑ ΞΕΚΙΝΑ

ΤΑΙΝΙΑ ΣΤΟ 0.01

Όταν είναι σε κατάσταση Play τότε η ταινία θα κλείσει στο πέρας. Σε κατάσταση Restart στο

πέρας δεν κλείνει, πρέπει να τη κλείσουμε με MOVIE. Επίσης στην κατάσταση Play με κλικ στην ταινία κλείνει. Σε κατάσταση Restart με κλικ σταματάει αν έπαιζε η ταινία ή ξεκινάει αν ήταν σταματημένη.

ΤΑΙΝΙΑ 1.2 είναι ίδιο με ΤΑΙΝΙΑ to 1.2

ΤΑΙΝΙΑ 0 είναι ίδιο με ΤΑΙΝΙΑ ΞΕΚΙΝΑ (και όχι με ΤΑΙΝΙΑ ΣΤΟ 0)

ΤΑΙΝΙΑ -1 είναι ίδιο με ΤΑΙΝΙΑ ΚΡΑΤΗΣΕ

Πως ελέγχουμε αν μια ταινία έχει φορτωθεί;

Αν έχει φορτωθεί ταινία τότε έχει Διάρκεια Duration

Τυπωσε Διάρκεια

Αν μια ταινία παίζει σε κατάσταση PLAY τότε δίνει στο MOVIE το -1

Αλλά μπορεί να τρέχει μουσική στο περιθώριο και τότε να έχουμε πάλι το ίδιο αποτέλεσμα

Τυπωσε ΤΑΙΝΙΑ, ΤΑΙΝΙΑ.ΜΕΤΡΗΤΗΣ

Η μεταβλητή ΤΑΙΝΙΑ.ΜΕΤΡΗΤΗΣ μας δίνει το δευτερόλεπτο (και τα δεκαδικά του) που εκείνη τη στιγμή δείχνει η ταινία.

Αν δεν έχουμε φορτώσει ταινία τότε ο μετρητής είναι στο -1

Μπορεί να εκμεταλλευθεί κανείς την μεταβλητή αυτή για να φτιάξει σημεία επιστροφής, ή αλλιώς κεφάλαια!

Αναγνωριστικό: ΤΟΝΟΣ [TONE]

Αυτή η εντολή δεν δουλεύει σε εικονικό περιβάλλον. Επειδή χρησιμοποιεί συσκευή που στο εικονικό περιβάλλον δεν μπορεί να αναπαρασταθεί.

Κάθε υπολογιστής από τα Windows Χρ και πάνω έχει γενήτρια τόνου

1) ΤΟΝΟΣ

παίζει 1000Hz για 0,1 του δευτερολέπτου

2) ΤΟΝΟΣ 200

παίζει 1000Hz για 0,200 του δευτερολέπτου

3) ΤΟΝΟΣ 200, 5000

παίζει 5000Hz για 0,200 του δευτερολέπτου

Στο τμήμα Β έχουμε την TUNE η οποία είναι μια σύνθετη TONE

Module A {Module AA {

For i=1 To 10 {

Tone 50, 150

Tone 25, 300

}

}

```

For i=1 To 5 {
AA
Wait 200
AA
Wait 800
}
}
Module B {For i = 1 To 10 {
TUNE "cac "
Wait 200
}
}
Module C {For i=100 To 8000 Step 20 {
Tone 2,i
}
}

```

Όταν δίνουμε εντολή ΤΟΝΟΣ ακούμε ένα ήχο και εκείνη την ώρα σταματάει κάθε νήμα να λειτουργεί! Δεν είναι παράξενο γιατί αυτή η λειτουργία γίνεται με το Kernel των windows το ίδιο που "σκανδαλίζει" τα νήματα του περιβάλλοντος. Μπορεί κανείς να παίξει έτοιμους ήχους με την Ηχος η οποία τρέχει παράλληλα! Αλλά το ίδιο κάνει και η Μουσική και η Παιξε.

```

I=0
THREAD {
PRINT I
I=I+1
} AS L
THREAD L INTERVAL 40
WAIT 1000
TONE 1000,3000
WAIT 1000

```

Αναγνωριστικό: ΦΩΝΗ [SCORE]
 SCORE 3, 1000, "C5F#@2B@2C5F#@2B"
 SCORE 1, 1000, "D@2E@2C#3 @2D5@2V90 @3F#4V127"
 ' / C # D D # E F F # G G # A A # B

'/
PLAY 1, 19, 3, 22 ' VOICE, INSTRUMENT

Υπάρχουν 16 φωνές

Φωνή 12, 1000,"C5"

βάζουμε στη φωνή 12 να παίξει μια νότα από την 5η οκτάβα για 1000 χιλιοστά

αν βάλουμε @2 θα παίξει σε χρόνο 1000/2, μέχρι @6 δηλαδή 1000/32

αν βάλουμε V και έναν αριθμό από 1 μέχρι 127 αλλάζουμε την ένταση

αν δεν βάλουμε γράμμα νότας και βάλουμε διάστημα τότε παίζει παύση με το διαιρέτη @ αν υπάρχει

Για να παίξουμε τη κάθε φωνή αρκεί να δώσουμε μια εντολή ΠΑΙΞΕ ή PLAY

όπου συνδιάζουμε φωνή και όργανο

η Παιξε 0 κλείνει την μουσική.

η Ένταση αλλάζει την ένταση του ήχου συνολικά.

η Επιλεξε.Όργανο δίνει το νούμερο στο όργανο που θα επιλέξουμε από μια λίστα οργάνων για να το χρησιμοποιήσουμε στην Παιξε (το διαβάζουμε με την ΕΠΙΛΟΓΗ)

Ομάδα: ΕΝΤΟΛΕΣ ΔΕΙΚΤΗ - MOUSE COMMANDS

Αναγνωριστικό: ΔΕΙΚΤΗ.ΜΟΡΦΗ [MOUSE.ICON]

ΔΕΙΚΤΗ.ΜΟΡΦΗ 2

ΔΕΙΚΤΗ.ΜΟΡΦΗ ΚΑΤ\$+"ταδε.ico"

ΔΕΙΚΤΗ.ΜΟΡΦΗ ΔΕΙΞΕ

ΔΕΙΚΤΗ.ΜΟΡΦΗ ΚΡΥΨΕ

Αναγνωριστικό: ΛΑΒΗ [JOYPAD]

ΛΑΒΗ 0,1

ενεργοποιεί τις λαβές (παιχνιδιών) 0 και 1

Χωρίς παράμετρο απενεργοποιεί τις Λαβές.

Υπάρχουν τέσσερις συναρτήσεις για να χρησιμοποιήσουμε:

ΛΑΒΗ(0)

ΛΑΒΗ.ΚΑΤΕΥΘΥΝΣΗ(0)

ΛΑΒΗ.ΑΝΑΛΟΓΙΚΟ.Χ(0)

ΛΑΒΗ.ΑΝΑΛΟΓΙΚΟ.Υ(0)

Κάθε φορά δίνουμε το νούμερο της λαβής. Στην πρώτη περίπτωση θα πάρουμε όλα τα πλήκτρα σαν ένα δυαδικό νούμερο.

Στη δεύτερη περίπτωση θα πάρουμε την κατεύθυνση

Καμία = 0

Αριστερά = 1
Δεξιά = 2
Πάνω = 3
Κάτω = 4
Πάνω_Αριστερά = 5
Κάτω_Αριστερά = 6
Πάνω_Δεξιά = 7
Κάτω_Δεξιά = 8

Στη τρίτη και τέταρτη περίπτωση θα έχουμε τις θέσεις του μοχλού (χωρίζεται ένας μοχλός πάνω στη λαβή σε διακριτά βήματα ανάλογα τη γωνία του κατά X και κατά Y)

Ομάδα: ΕΝΤΟΛΕΣ ΙΣΤΟΥ - BROWSER COMMANDS

Αναγνωριστικό: ΑΝΑΛΟΓΙΟ [BROWSER]

1) ΕΜΦΑΝΙΣΗ

ΑΝΑΛΟΓΙΟ "www.google.gr"

2) ΕΜΦΑΝΙΣΗ σε θέση που θέλουμε

ΑΝΑΛΟΓΙΟ "www.youtube.com",1000,1000,5000,3000

3) ΑΚΥΡΩΣΗ

ΑΝΑΛΟΓΙΟ ""

4) ΑΝΑΛΟΓΙΟ "αλφα"

φορτώνει αρχείο που έχει σωθεί στον προσωρινό φάκελο των WINDOWS

η μεταβλητή ΠΡΟΣΩΡΙΝΟΣ ή TEMPORARY\$ μας δίνει την διεύθυνση του προσωρινού φακέλου των windows. Π.χ. ΚΑΤΑΛΟΓΟΣ ΠΡΟΣΩΡΙΝΟΣ: ΑΡΧΕΙΑ "tmp"

Δες ΚΕΙΜΕΝΟ και ΓΕΝΙΚΑ για ΙΣΤΟΣΕΛΙΔΕΣ

Αναγνωριστικό: ΚΕΙΜΕΝΟ [TEXT, HTML]

Το κείμενο εδώ είναι με ένα Byte ανά χαρακτήρα (όχι Unicode)

Αυτή η εντολή είναι εδώ για δύο λόγους. Να δημιουργεί καταγραφές εύκολα (log file)

και να δημιουργεί φόρμες για το ΑΝΑΛΟΓΙΟ

Για unicode μπορούμε να ανοίξουμε αρχεία στο ΠΡΟΣΩΡΙΝΟΣ και για να τα σβήσουμε

μπορούμε να δώσουμε το όνομα με τον τύπο χωρίς το μονοπάτι στην εντολή ΚΕΙΜΕΝΟ χωρίς να προσθέσουμε μπλοκ με γραμμές. (δες 5)

Προσθήκη στην 7 έκδοση

```
ΚΕΙΜΕΝΟ utf-8 α$ {  
}
```

```
ΚΕΙΜΕΝΟ utf-16 α$ {  
}
```

1) Δημιουργία κειμένου σε προσωρινό φάκελο

```
α$="αλφα"
```

```
ΚΕΙΜΕΝΟ α$ {  
μπλα μπλα μπλα  
}
```

```
ΚΕΙΜΕΝΟ αλφα { ' εδώ το αλφα δεν είναι μεταβλητή αλλά όνομα  
μπλα μπλα μπλα  
}
```

2) Προσθήκη κειμένου

```
ΚΕΙΜΕΝΟ αλφα + {  
άλλη γραμμή, μπλα μπλα  
}
```

3) Δημιουργία Ιστοσελίδας και πέρασμα μεταβλητής

```
alfa$="GEORGE"
```

```
ΚΕΙΜΕΝΟ αλφα.html {<!doctype html public "-//w3c//dtd html 3.5//en">
```

```
<html>
```

```
<head>
```

```
<title>(Type a title for your page here)</title>
```

```
<meta name="GENERATOR" content="M2000">
```

```
<meta name="FORMATTER" content="M2000">
```

```
<meta content="this, other, that" name=keywords>
```

```
<meta content=All name=robots>
```

```
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1253">
```

```
<meta NAME="Author" CONTENT="GEORGE KARRAS">
```

```
</head>
```

```
<body bgcolor="#ffffff" text="#000000" link="#0000ff" vlink="#800080" alink="#ff0000">
```

```
Γεια Χαρά ##αλφα$##
```

```
</body>
```

```
</html>
```

```
}
```

4) Δημιουργία Ιστοσελίδας με φόρμα, πέρασμα τιμών και συλλογή τιμών σε μεταβλητές

```
Τμήμα α {
```

```
ΑΝΑΨΕ
```

```
username$ = "Γιωργος"
```

```
κειμενο αλφα.html {<!doctype html public "-//w3c//dtd html 3.2//en">
```

```
<html>
```

```

<head>
<title>(Type a title for your page here)</title>
<meta name="GENERATOR" content="M2000">
<meta name="FORMATTER" content="M2000">
<meta content="this, other, that" name=keywords>
<meta content=All name=robots>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1253">
<meta NAME="Author" CONTENT="GEORGE KARRAS">
</head>
<body bgcolor="CYAN" text="#000000" link="#0000ff" vlink="#800080" alink="#ff0000">
<CENTER>Χειριστής</CENTER></H1><HR>
<script> var m1;var p1;var total;function alfa() {
total="?onoma=2000";total=total+"&username="+escape(m1);total=total+"&password="+escape
(p1);document.title=total;setTimeout ("alfa()",100);return true}</script>
<FORM NAME="ValidForm" action="about:blank" >
<INPUT NAME="onoma" TYPE="HIDDEN" MAXLENGTH=4 SIZE="4" VALUE="1000">
<TABLE ALIGN=CENTER BORDER=0 RULES=NO COLS="2">
<TBODY>
<TR VALIGN=TOP >
<TD >Όνομα:</TD>
<TD ><INPUT NAME="username" TYPE="TEXT" id="uname" MAXLENGTH=16 SIZE="16"
VALUE="##username###" ></TD>
</TR>
<TR VALIGN=TOP >
<TD >Συνθηματικό:</TD>
<TD ><INPUT NAME="password" ID="pass" TYPE="PASSWORD" MAXLENGTH=8
SIZE="8"></TD>
</TR>
</TABLE>
<CENTER><BR><HR>
<input type=button value="Αποστολή" onclick="m1=uname.value;p1=pass.value;alfa() ">
</CENTER>
</FORM>
</body>
</html>
}
onoma$=""
αναλογιο αλφα.html
ενω onoma$="" {ανανεωση} ' προσοχή εδώ ...η ανανέωση χρειάζεται για να πιάσουμε την
αποστολή από την σελίδα.
λιστα
αναλογιο ""
κειμενο αλφα.html ' εδώ σβήνουμε το προσωρινό αρχείο

```

```
Τυπωσε"ΟΚ":α$=κομ$
}
α
```

5) ΚΕΙΜΕΝΟ ονομα.τυπος
ΚΕΙΜΕΝΟ "ονομα.τυπος" ή έκφραση αλφαριθμητική
σβήνει το αρχείο προσωρινο\$+"ονομα.τυπος"

ΔΕΣ ΓΕΝΙΚΑ ΓΙΑ ΙΣΤΟΣΕΛΙΔΕΣ

Ομάδα: ΚΟΙΝΕΣ ΦΟΡΜΕΣ - COMMON DIALOGUES

Αναγνωριστικό: ΑΝΟΙΓΜΑ.ΑΡΧΕΙΟΥ [OPEN.FILE]
ΑΝΟΙΓΜΑ.ΑΡΧΕΙΟΥ
ΑΝΟΙΓΜΑ.ΑΡΧΕΙΟΥ ένα.ονομα\$
ΑΝΟΙΓΜΑ.ΑΡΧΕΙΟΥ ένα.ονομα\$, κατάλ\$
ΑΝΟΙΓΜΑ.ΑΡΧΕΙΟΥ ένα.ονομα\$, κατάλ\$, τίτλος\$
ΑΝΟΙΓΜΑ.ΑΡΧΕΙΟΥ ένα.ονομα\$, κατάλ\$, τίτλος\$, τυποςαρχείων\$
Το αποτέλεσμα θα είναι στον σωρό

Υπάρχει και η ΑΡΧΕΙΟ\$() που παίρνει από κανένα μέχρι δυο παραμέτρους, με πρώτη το τίτλο του διαλόγου και δεύτερη τον ή τους τύπους των αρχείων (τους διαχωρίζουμε με τη κάθετο |)

```
ΤΜΗΜΑ ΑΡ {
ΠΑΡΑΘΥΡΟ 12,5000;
ΟΘΟΝΗ 0,0
ΑΝΑΨΕ
{
' Και στην νέα έκδοση λειτουργεί ο παλιός τρόπος που ορίζαμε τους τύπους αρχείων
' ΑΝΟΙΓΜΑ.ΑΡΧΕΙΟΥ "",ΚΑΤ$+"..\media\","Ανοιγμα Αρχείου Εικόνας","Εικόνα
(*.bmp)|*.bmp|Φωτογραφία (*.jpg)|*.jpg"
' Η νέα φόρμα επιλογής αρχείου έχει πιο απλοποιημένο αλφαριθμητικό για να δηλώσουμε
τύπους αρχείων
' Αυτό δεν θα είναι συμβατό με παλιές εκδόσεις
ΑΝΟΙΓΜΑ.ΑΡΧΕΙΟΥ "",ΚΑΤ$+"..\media\","Ανοιγμα Αρχείου Εικόνας","bmp|jpg"
ΔΙΑΒΑΣΕ ΕΙΚ$
ΑΝ ΕΙΚ$<>"" ΤΟΤΕ {
ΕΝΩ ΔΕΙΚΤΗΣ=1 {}
ΕΙΚΟΝΑ ΕΙΚ$,Χ.ΣΗΜΕΙΑ,Υ.ΣΗΜΕΙΑ
```

```
ΑΝΑΝΕΩΣΗ
ΕΝΩ (ΔΕΙΚΤΗΣ=0) ΚΑΙ (ΕΝΚΟΜ$="") {}
} ΑΛΛΙΩΣ ΕΞΟΔΟΣ
ΚΥΚΛΙΚΑ
}
ΜΠΙΠ
}
ΑΡ
```

Αναγνωριστικό: ΑΝΟΙΓΜΑ.ΕΙΚΟΝΑΣ [OPEN.IMAGE]
ΑΝΟΙΓΜΑ.ΕΙΚΟΝΑΣ
ΑΝΟΙΓΜΑ.ΕΙΚΟΝΑΣ ένα.ονομα\$
ΑΝΟΙΓΜΑ.ΕΙΚΟΝΑΣ ένα.ονομα\$, κατάλ\$
ΑΝΟΙΓΜΑ.ΕΙΚΟΝΑΣ ένα.ονομα\$, κατάλ\$, τίτλος\$
ΑΝΟΙΓΜΑ.ΕΙΚΟΝΑΣ ένα.ονομα\$, κατάλ\$, τίτλος\$, τυποσαρχείων\$
Το αποτέλεσμα θα είναι στον σωρό

ο τυποσαρχείων\$ εξ ορισμού είναι "BMP|JPG|GIF|WMF|EMF|DIB|ICO|CUR"

Ο συγκεκριμένος διάλογος δείχνει τις εικόνες και τα σχέδια σε προεπισκόπηση.

Αναγνωριστικό: ΑΠΟΘΗΚΕΥΣΗ.ΩΣ [SAVE.AS]
ΑΠΟΘΗΚΕΥΣΗ.ΩΣ
ΑΠΟΘΗΚΕΥΣΗ.ΩΣ ένα.ονομα\$
ΑΠΟΘΗΚΕΥΣΗ.ΩΣ ένα.ονομα\$, κατάλογος\$
ΑΠΟΘΗΚΕΥΣΗ.ΩΣ ένα.ονομα\$, κατάλογος\$, τίτλος\$
ΑΠΟΘΗΚΕΥΣΗ.ΩΣ ένα.ονομα\$, κατάλογος\$, τίτλος\$, τυποσαρχείων\$
Το αποτέλεσμα θα είναι στον σωρό
Όπως και το προηγούμενο έτσι και εδώ απλά ζητάμε το όνομα (με το μονοπάτι) ενός αρχείου.

Αναγνωριστικό: ΕΠΙΛΕΞΕ.ΓΡΑΜΜΑΤΟΣΕΙΡΑ [CHOOSE.FONT]
ΕΠΙΛΕΞΕ.ΓΡΑΜΜΑΤΟΣΕΙΡΑ

```
ΤΜΗΜΑ Α {
    ΤΜΗΜΑ ΛΕΓΕ {
```

```

        ΔΙΑΒΑΣΕ A
        ΑΝ A=0 ΤΟΤΕ { ? "NO"} ΑΛΛΙΩΣ ? "YES"
    }
    ΕΠΙΛΕΞΕ>ΓΡΑΜΜΑΤΟΣΕΙΡΑ
    ΑΝ ΑΡΙΘΜΟΣ>0 ΤΟΤΕ {
        ΔΙΑΒΑΣΕ ΓΡ$,ΜΕΓ,ΚΩΔ,ΠΛΑΓ,ΦΑΡΔ
        ΤΥΠΩΣΕ "ΓΡΑΜΜΑΤΟΣΕΙΡΑ:"; ΓΡ$
        ΤΥΠΩΣΕ "ΜΕΓΕΘΟΣ:"; ΜΕΓ
        ΤΥΠΩΣΕ "ΚΩΔΙΚΟΠΟΙΗΣΗ:"; ΚΩΔ
        ΤΥΠΩΣΕ "ΠΛΑΓΙΑ"; : ΛΕΓΕ ΠΛΑΓ
        ΤΥΠΩΣΕ "ΠΛΑΓΙΑ"; : ΛΕΓΕ ΦΑΡΔ
    } ΑΛΛΙΩΣ {
        ΤΥΠΩΣΕ "ΤΙΠΟΤΑ...."
    }
}

```

Αναγνωριστικό: ΕΠΙΛΕΞΕ.ΧΡΩΜΑ [CHOOSE.COLOR]

1) Βγάζει τον εξ ορισμού διάλογο για επιλογή χρώματος

ΕΠΙΛΕΞΕ.ΧΡΩΜΑ

2) Όπως η ένα αλλά μας δείχνει και ένα συγκεκριμένο χρώμα που θέλουμε

ΕΠΙΛΕΞΕ.ΧΡΩΜΑ χρωμα

με βελάκια αριστερά και δεξιά περιστρέφουμε την πληροφορία χρώματος ώστε με τα άνω και κάτω να αλλάζουμε το πιο ακραίο δεξιά χρώμα κατά ένα.

Μπορούμε ακόμα και να ολισθήσουμε την τιμή στο κάτω μέρος του διαλόγου, και αυτό ισοδυναμεί με διπλό κλικ.

Αναγνωριστικό: ΚΑΤΑΛΟΓΟΣ [DIR]

ΚΑΤΑΛΟΓΟΣ "C:"

ΚΑΤΑΛΟΓΟΣ "..\αλφα" (ανεβαίνει ένα επίπεδο και κατεβαίνει στο αλφα - αν υπάρχει!)

ΤΥΠΩΣΕ ΚΑΤ\$

DIR "C:\"

PRINT DIR\$

ΚΑΤΑΛΟΓΟΣ ΧΡΗΣΤΗ (κάνει τον κατάλογο του χρήστη username\M2000\ τωρινό κατάλογο)

ΚΑΤΑΛΟΓΟΣ ? (ανοίγει ένα διάλογο επιλογής φακέλου)

ΚΑΤΑΛΟΓΟΣ ? "C:\\" (ανοίγει ένα διάλογο επιλογής φακέλου με κορυφή τον C:\")

ΚΑΤΑΛΟΓΟΣ ? "C:\", "ΓΡΑΨΕ ΚΑΤΙ"

ΚΑΤΑΛΟΓΟΣ ? "?" (βάζει και το ΝΕΟΣ ΦΑΚΕΛΟΣ)

ΚΑΤΑΛΟΓΟΣ ? "C:\?" (αρχικός φάκελος και επιλογή ΝΕΟΣ ΦΑΚΕΛΟΣ)

ΚΑΤΑΛΟΓΟΣ ? "C:\?", "ΓΡΑΨΕ ΚΑΤΙ, η βάλε και δυο τρεις γραμμές"

ΚΑΤΑΛΟΓΟΣ {μια γραμμή

μια άλλη

και μια άλλη } ? "C:\?"

Αν ακυρώσουμε την εισαγωγή φακέλου τότε δεν αλλάζει ο τρέχον φάκελος.

Αν προσθέσουμε το ΣΤΟ Κ\$ τότε βάζουμε το μονοπάτι του φακέλου στο Κ\$ και δεν αλλάζουμε τον τρέχον φάκελο

' Δεν μπορούμε να ανοίξουμε δυο φορές τον επιλογέα φακέλου/αρχείου

' Μπορούμε όμως να το ανοίξουμε ενώ έχουμε νήματα να τρέχουν μαζί.

' η εντολή ΔΕΣ φτιάχνει την μεταβλητή ΕΝΤΑΞΕΙ και βάζει -1 αν δεν βγάλει λάθος το μπλοκ εντολών

' η ΟΧΙ όπως και η ΔΕΝ αντιστρέφουν το ΑΛΗΘΕΣ με ΨΕΥΔΕΣ και το ΨΕΥΔΕΣ με ΑΛΗΘΕΣ

' όπου ΑΛΗΘΕΣ = -1 και ΨΕΥΔΕΣ=0

' Υπάρχουν και τα ΑΛΗΘΗΣ και ΨΕΥΔΗΣ επίσης

ΚΑΤΑΛΟΓΟΣ ΧΡΗΣΤΗ

ΜΕΤΑ 200 {

 ΤΥΠΩΣΕ "ΕΛΕΓΧΟΣ"

 ΔΕΣ ΕΝΤΑΞΕΙ {

 ΚΑΤΑΛΟΓΟΣ ?

 }

 ΑΝ ΟΧΙ ΕΝΤΑΞΕΙ ΤΟΤΕ ΤΥΠΩΣΕ ΛΑΘΟΣ\$

}

ΚΑΤΑΛΟΓΟΣ ?

Αναγνωριστικό: ΡΥΘΜΙΣΕΙΣ [SETTINGS]

ΡΥΘΜΙΣΕΙΣ

Ανοίγει μια φόρμα ρυθμίσεων οι οποίες θα λειτουργήσουν άμεσα (αλλά θα σβήσει η οθόνη και θα τυπωθεί το αρχικό μήνυμα - τρεις γραμμές πάνω αριστερά) και θα φτιαχτεί και χωριστή οθόνη ολίσθησης

Αναγνωριστικό: ΤΙΤΛΟΣ [TITLE]

ΤΙΤΛΟΣ "παράδειγμα της M2000" είναι ίδιο με ΤΙΤΛΟΣ "παράδειγμα της M2000", 1
ΤΙΤΛΟΣ "Άλλος", 0

Εμφανίζει στην γραμμή εργασιών (δίπλα στο ΕΝΑΡΞΗ) ένα πλαίσιο που περιέχει τον τίτλο που δώσαμε και το εικονίδιο της M2000. Μπορούμε να αλλάξουμε το εικονίδιο, δες ΕΙΚΟΝΙΔΙΟ.

Το πλαίσιο που εμφανίζεται είναι χρήσιμο γιατί:

- 1) Μας επαναφέρει στην M2000 μετά από εναλλαγή προγραμμάτων με τα alt και tab
- 2) Μετά από επιλογή του μας εμφανίζει αμέσως την M2000
- 3) Μας δίνει την δυνατότητα με άνοιγμα του σχετικής λίστας επιλογών (δεξί κλικ πάνω στο πλαίσιο) να κλείσουμε αμέσως την γλώσσα
- 4) Επιπλέον από την έκδοση 6.2.3 μπήκε και μια δεύτερη παράμετρος για Ελαχιστοποίηση στο πλαίσιο. Με 0 έχουμε μόνο το πλαίσιο στη γραμμή εργασιών και εξαφανίζουμε το παράθυρο του Περιβάλλοντος! Με 1 το εμφανίζουμε. Οι διάλογοι άνοιγμα αρχείου και σώσιμο, καθώς και επιλογή χρώματος, γραμματοσειράς, ιδιότητες εκτυπωτή, εμφανίζονται και χωρίς το περιβάλλον να εμφανίζεται! Το ίδιο επίσης ισχύει και για το ΡΩΤΑ(ή ASK(
- 5) Μπορούμε από το πλαίσιο, στη σχετική λίστα να επιλέξουμε Ελαχιστοποίηση ή Επαναφορά. Μπορούμε κατά βούληση να αλλάζουμε τον τίτλο και το εικονίδιο. Επίσης με μια επανάληψη της εντολής χωρίς όρισμα αφαιρούμε το πλαίσιο αυτό. Αν αφαιρέσουμε το πλαίσιο τότε αν το περιβάλλον είχε αποκρυφτεί θα εμφανιστεί.

Αναγνωριστικό: ΥΠΟΚΑΤΑΛΟΓΟΣ [SUBDIR]

Δες {

ΥΠΟΚΑΤΑΛΟΓΟΣ ΑΛΦΑ

}

ΚΑΤΑΛΟΓΟΣ ΑΛΦΑ

Δημιουργούμε τον ΑΛΦΑ αν δεν υπάρχει και τον επιλέγουμε

Ομάδα: ΑΡΙΘΜΗΤΙΚΑ - ARITHMETIC FUNCTIONS

Αναγνωριστικό: #ΑΘΡ([#SUM(

Τύπωσε (1,2,3)#αθρ()

Πίνακα A(10)=1

Τύπωσε A()#αθρ()=10

μ=(1,2,3,4)

Τύπωσε μ#αθρ()=10

Στη $\alpha=1, \beta=2, \gamma=3$
Τύπωσε $(\alpha, \beta, \gamma)\#\alpha\theta\rho()$

\| δεν μπορούμε να βάλουμε πριν τον αυτόματο πίνακα πρόσημο (πχ το $-(1,2,3)\#\alpha\theta\rho()$ βγάζει λάθος)

\| όμως μπορούμε αν χρησιμοποιήσουμε δείκτη, όπως ο μ , ή πίνακας όπως το $A()$

Αναγνωριστικό: #ΑΝΑΠ([#REV()

Τύπωσε $(1,2,3)\#\alpha\alpha\pi()$
3 2 1

Αναγνωριστικό: #ΑΝΟΙΓΜΑ([#EXPANSE()
 $\alpha=(,)$
? $\alpha\#\alpha\alpha\alpha\alpha\alpha(100)\#\mu\alpha\zeta\acute{\iota}("=", 1)\#\alpha\theta\rho()=100$

Αναγνωριστικό: #ΑΝΤ([#MAP()

$\alpha=(1,2,3,4,5,6,7,8,9)$
πακ2=λάμδα ->{
 φερε 2
 βάλε γραμμα α +"-"+γραμμα α
}
μονός=λάμδα (χ)-> χ υπόλοιπο 2=1
 $\beta=\alpha\#\phi\acute{\iota}\lambda\tau\rho(\mu\omicron\nu\acute{o}\varsigma, (,))$
Τύπωσε β ' 1,3,5,7,9
αντ1=λάμδα ->{
 Βάλε ΧαρΚωδ α (αριθμός+64)
}
 $\zeta=\alpha\#\phi\acute{\iota}\lambda\tau\rho(\mu\omicron\nu\acute{o}\varsigma)\#\alpha\alpha\alpha\alpha\alpha(αντ1)$
Τύπωσε ζ ' A C E G I
Τύπωσε $"["+\alpha\#\phi\acute{\iota}\lambda\tau\rho(\mu\omicron\nu\acute{o}\varsigma)\#\alpha\alpha\alpha\alpha\alpha(αντ1)\#\pi\alpha\kappa\alpha(\pi\alpha\kappa2, "CODE")+"]"="[CODE-A-C-E-G-I]"$

Αναγνωριστικό: #ΑΡΧΗ([#START()

$\alpha=(1,2,3,4)$
 $\beta=(1,2,3)$
? $\alpha\#\alpha\rho\chi\eta(\beta)\#\tau\epsilon\lambda((100,))$

// 1 2 3 1 2 3 4 100

Αναγνωριστικό: #ΔΕΝΕΧΕΙ([#NOTHAVE(
ΤΥΠΩΣΕ (1,2,3,4)#ΔΕΝΕΧΕΙ(1,2,5)=ΑΛΗΘΗΣ
ΤΥΠΩΣΕ (1,2,3,4,1,2)#ΔΕΝΕΧΕΙ(3->2,3)=ΑΛΗΘΗΣ ' από το 4ο
ΤΥΠΩΣΕ (1,2,3,4,1,2)#ΔΕΝΕΧΕΙ(0->(2,2))=ΑΛΗΘΗΣ ' από το 1ο

Αναγνωριστικό: #ΕΚΦΡ([#EVAL(
Η #εκφρ() χρησιμοποιείται σε πίνακες πχ Α() ή δείκτες σε πίνακες. Μπορεί να προστεθεί σε υπάρχουσες εντολές τύπου #χχ(). Διαφέρει από την #Τιμή() σε σχέσει με πίνακες γιατί η #Τιμή() γυρίζει αντίγραφο πίνακα ενώ η #εκφρ() γυρνάει τον κανονικό δείκτη. Αν θέλουμε τιμές αλφαριθμητικών τότε χρησιμοποιούμε την #Εκφρ\$(). Αν στην θέση που ζητάμε υπάρχει λάμδα συνάρτηση ή αντικείμενο που γυρνάει τιμή και ίσως ζητούν και δεδομένα, τότε η #Εκφρ() αφενός οφείλει να παρέχει τα δεδομένα και αφετέρου εκτελεί (κάτι που δεν κάνει η #Τιμή()). Επιπλέον αν το αντικείμενο είναι Πίνακας ή Κατάσταση ή Σωρός τότε αν δώσουμε παράμετρο θα δώσει ότι βρει στη θέση που δείχνει η παράμετρος, και αν δώσουμε περισσότερες τότε εφόσον αυτό που βρίσκει είναι αντικείμενο προχωράει το ίδιο. Αν δώσουμε παράμετρο και δεν υπάρχει αντικείμενο θα βγει λάθος - το πιθανότερο να κάνει έξοδο το τμήμα στο οποίο εκτελείται, ή στο μπλοκ που εκτελείται.

μ=(1,2,(100,(200,201,202),300),4)
Τύπωσε μ#Εκφρ(2,1,2)=202
Τύπωσε μ#Τιμή(2)#Τιμή(1)#Τιμή(2)=202
Κλάση Άλφα {
 Αξία (x){
 =500*x
 }
}
μ=(1,2,(Λάμδα (x)->500*x), Άλφα())
Τύπωσε μ#Εκφρ(2,3)=500*3, μ#Εκφρ(3,3)=500*3
Κατάσταση Ουρά κ=1,2,3:=300,4
μ=(1,2,κ)
Τύπωσε κ(3)=300, κ(2!)=300, μ#Εκφρ(2,2)=300
\\ η απλή Κατάσταση δεν εγγυάται τη θέση των στοιχείων (αλλάζουν μετά από διαγραφή)
\\ στη Κατάσταση τύπου ουρά, διαγράφουμε μόνο από το τέλος, οπότε η θέση όσων υπάρχουν δεν αλλάζει.
\\ και στις δυο περιπτώσεις η ταξινόμηση αλλάζει θέση, αλλά στην τύπου ουρά, γίνεται σταθερή ταξινόμηση, και τα όμοια κλειδιά δεν αλλάζουν μεταξύ τους θέση. Στην κανονική Κατάσταση δεν

μπαίνουν όμοια κλειδιά.

Κατάσταση κ1=1,2,3:=1000,4,5

μ=(1,2,κ1)

Αφαίρεση κ1, 2, 4

Αν Υπάρχει(κ1, 3) Τότε κ1_θέση_στοιχείου=Εκφρ(κ1!)

Τύπωσε μ#Εκφρ(2,κ1_θέση_στοιχείου)=1000

\\ για να δούμε με κλειδί από το μ χρειάζεται μια ένωση με άλλου τύπου διεπαφή

Ένωσε μ στο μ()

Τύπωσε μ(2)(3)=1000

Αναγνωριστικό: #ΕΧΕΙ([#HAVE(]

ΤΥΠΩΣΕ (1,2,3,4)#ΕΧΕΙ(1,2)=ΑΛΗΘΗΣ

ΤΥΠΩΣΕ (1,2,3,4,1,2)#ΕΧΕΙ(3->1,2)=ΑΛΗΘΗΣ ' από το 4ο

ΤΥΠΩΣΕ (1,2,3,4,1,2)#ΕΧΕΙ(0->(1,2))=ΑΛΗΘΗΣ ' από το 1ο

Αναγνωριστικό: #ΘΕΣΗ([#POS(]

Δεν μπορούμε να βάλουμε και αριθμούς και αλφαριθμητικά στην λίστα αναζήτησης

α=(1,1,0,1,1,1,1,0,1,1,0)

Τύπωσε α#θέση(1,0,1) ' 1 σημαίνει 2η θέση

Τύπωσε α#θέση(5->1,0,1) ' 6 σημαίνει 7η θέση

α=("hello","1", "hello","2")

Τύπωσε α#θέση(0->"hello", "2")=2 ' δίνει τη 3η θέση

Αναγνωριστικό: #ΜΑΖΙ([#MAT(]

α=(1,2,3,4,5)

? α#μαζί("+=", 100)#γραφή\$()="101 102 103 104 105"

Αναγνωριστικό: #ΜΕΓ([#MAX(]

α=(1,2,3)

Τύπωσε α#μεγ()=3

$\zeta=-1$

Τύπωσε $(1,2,3,-3)\#μεγ(\zeta)=3, \zeta=2$

Αναγνωριστικό: #ΜΕΡΟΣ([#SLICE(]

$\alpha=(1,2,3,4,5)$

Τύπωσε $\alpha\#μερος(1,2) \text{ ' } 2 \ 3$

Τύπωσε $\alpha\#μερος(1,2)\#αναπ() \text{ ' } 3 \ 2$

Τύπωσε $\alpha\#μερος(2) \text{ ' } 3 \ 4 \ 5$

Τύπωσε $\alpha\#μερος(,2) \text{ ' } 1 \ 2 \ 3$

Αναγνωριστικό: #ΜΙΚ([#MIN(]

Τύπωσε $(1,2,3)\#μικ()=1$

$\alpha=(1,2,3)$

Τύπωσε $\alpha\#μικ()=1$

Πίνακας $\alpha(3)$

$\alpha(0)=1,2,3$

Τύπωσε $\alpha()\#μικ()$

$\zeta=-1$

Τύπωσε $\alpha()\#μικ(\zeta)=1, \zeta=0$

Αναγνωριστικό: #ΠΑΚ([#FOLD(]

$\alpha=(1,2,3,4,5,6,7,8,9)$

πακέτο1=Λάμδα ->{

 Βάλε αριθμός+αριθμός

}

Τύπωσε $\alpha\#πακ(πακέτο1)=\alpha\#αθρ()$

Τύπωσε $\alpha\#πακ(πακέτο1,10)=\alpha\#αθρ()+10$

Δες φίλτρο(), #αντ(), #πακ\$(), #θέση()

Αναγνωριστικό: #ΤΑΞΙΝΟΜΗΣΗ([#SORT(]

Τύπωσε $(10,3,4,1)\#ΤΑΞΙΝΟΜΗΣΗ(1)$

10 3 4 1

Τύπωσε $(10,3,4,1)\#ΤΑΞΙΝΟΜΗΣΗ()$

1 4 3 10

Τύπωσε $(10,3,4,1)\#ΤΑΞΙΝΟΜΗΣΗ(0,1,3)$

10 1 3 4

Αναγνωριστικό: #ΤΕΛ([#END(]

$\alpha=(1,2,3,4)$

$\beta=(1,2,3)$

? $\alpha\#αρχη(\beta)\#τελ((100,))$

// 1 2 3 1 2 3 4 100

Αναγνωριστικό: #ΤΙΜΗ([#VAL(]

$\alpha=(1,2,3)$

Τύπωσε $\alpha\#Τιμή(0)=1$

Αναγνωριστικό: #ΥΛΙΚΟ([#STUFF(]

Επιστρέφει τιμές ως έχουν (όχι όπως τις τροποποιεί η ΤΙΜΗ()) αν είναι αλφαριθμητικό, δες το παράδειγμα)

ΚΛΑΣΗ ΑΛΦΑ {

$\chi=10$

}

ΠΙΝΑΚΑΣ $\alpha(10)$

$\zeta=\DeltaΕΙΚΤΗΣ(ΑΛΦΑ())$

$\alpha(3):="OK", 100, (1,2,3,4), \zeta$

$\mu=\alpha()$

? $\mu\#ΥΛΙΚΟ(6).\chi$

$\zeta=>\chi++$

? $\mu\#ΥΛΙΚΟ(6).\chi$

$\zeta\zeta=\mu\#ΥΛΙΚΟ(6)$

? $\zeta ΕΙΝΑΙ \zeta\zeta$

? $\alpha(3)$

? $\mu\#ΥΛΙΚΟ(3)="OK", \mu\#ΤΙΜΗ(3)=0, \mu\#ΤΙΜΗ\$(3)="OK"$

? $\mu\#ΥΛΙΚΟ(4)=100, \mu\#ΤΙΜΗ(4)=100, \mu\#ΤΙΜΗ\$(4)="100"$

? $\mu\#ΥΛΙΚΟ(5)\#ΜΕΓ()$

Αναγνωριστικό: #ΦΙΛΤΡΟ([#FILTER(]

$\alpha=(1,2,3,4,5,6,7,8,9)$
πακ1=λάμδα ->{
 βάλε αριθμός+αριθμός
}
ζυγός=λάμδα (χ)->χ υπόλοιπο 2=0
 $\beta=\alpha\#\phi\acute{\iota}\lambda\tau\rho(\zeta\upsilon\gamma\acute{o}\varsigma, (,))$
Τύπωσε $\beta \cdot 2 \cdot 4 \cdot 6 \cdot 8$
Τύπωσε $\alpha\#\phi\acute{\iota}\lambda\tau\rho(\zeta\upsilon\gamma\acute{o}\varsigma)\#\pi\alpha\kappa(\pi\alpha\kappa1)=20$

Αναγνωριστικό: ΑΕΦΑΠ([RTAN()
Εφαπτομένη σε ακτίνα
ΤΥΠΩΣΕ ΑΕΦΑΠ(π/4)
Δες Εφαπ() για μοίρες

Αναγνωριστικό: ΑΗΜ([RSIN()
Επιστρέφει το ημίτονο γωνίας σε ακτίνα
Δες το ΗΜ()

Αναγνωριστικό: ΑΚ([INT()
>ΤΥΠΩΣΕ ΑΚ(12.5)
12
Δίνει πάντα το μικρότερο ή ίσο ακέραιο.

για 1.9 δίνει 1
για 1.1 δίνει 1
για -1.9 δίνει -2
για -1.1 δίνει -2

Αυτή η συνάρτηση δεν συνδέεται με την στρογγυλοποίηση
είτε την αυτόματη όπως εδώ
 $\alpha\%=-1.2$
Θα πάρει το -1
 $\alpha\%=-1.8$
Θα πάρει το -2
είτε την κατ' απαίτηση με την Στρογγ()
Τύπωσε Στρογγ(-1.8, 0) δίνει -2
Μπορούμε να κάνουμε στρογγυλοποιήσεις άμεσα στην εμφάνιση
Μπορούμε να ορίσουμε τη μορφή του σημείου των δεκαδικών
με την Τοπικό. Πχ Τοπικό 1032 θα βάλε το κόμμα

Πάντα όμως στο κώδικα το σημείο δεκαδικών είναι η τελεία.
Τύπωσε Μορφή\$("{0:0} {0:1} {0:2:-8}", -1.8)

Αναγνωριστικό: ΑΚΕΡΑΙΟ.ΔΥΑΔΙΚΟ([SINT(
α=0xFAAA
Τύπωσε ΑΚΕΡΑΙΟ.ΔΥΑΔΙΚΟ(α), ΑΚΕΡΑΙΟ.ΔΥΑΔΙΚΟ(α,2) ' το δεύτερο είναι αρνητικός
α=0xFFFFFAAA
Τύπωσε ΑΚΕΡΑΙΟ.ΔΥΑΔΙΚΟ(α) ' αρνητικός
Τύπωσε α ' ως ακέραιος αριθμός χωρίς πρόσημο
Δεκαεξ ΔΥΑΔΙΚΟ.ΑΚΕΡΑΙΟ(ΑΚΕΡΑΙΟ.ΔΥΑΔΙΚΟ(α))

\\ Χρησιμοποίησε το ΔΥΑΔΙΚΟ.ΑΚΕΡΑΙΟ() για το ανάποδο

Αναγνωριστικό: ΑΝ([IF(
1) Ternary operator
1.1 Με τιμές -1 και 0 (Αληθές και Ψευδές), θέλει μόνο δυο στοιχεία, μπορούμε να έχουμε κενά.
Εκτελεί μόνο μια έκφραση μετά το ->

K=5 : N=3

Τύπωσε ΑΝ(K>N -> 100, 200)

M=Αληθές \\ Ψευδές

Συναρτηση ΔυοΤιμές (X,Y) {
=X**Y

}

\\ Αν το M είναι αληθές τα X και Y θα γίνουν 10 και 20, αλλιώς θα γίνουν 30 και 40

\\ το !(1,2,3,4) σημαίνει να γραφεί ο πίνακας (1,2,3,4) στο σωρό τιμών κατά τη κλήση της συνάρτησης.

\\ εδώ το ΑΝ() γυρνάει έναν δείκτη σε πίνακα, το ! παίρνει το δείκτη και αντιγράφει όλα τα στοιχεία του πίνακα στο σωρό τιμών.

Τύπωσε ΔυοΤιμές(!ΑΝ(M -> (10,2), (30,4)))

1.2 Με τιμή από 1, εκτελεί την Νιοστή έκφραση (τις άλλες τις προσπερνάει χωρίς να τις εκτελέσει)

Τύπωσε ΑΝ(Τυχαίος(1,5) -> 10, 30, 45, 55, 60)

2) Elvis operator

2.1 για αντικείμενα

A(3)=ΑΝ(A(3)->,M) \\ όταν το A(3) είναι 0 ή άδειο, τότε θα πάρει τιμή το αντίγραφο του M (ή αν

είναι δείκτης σε κάτι θα πάρει το δείκτη)

2.2 για συναρτήσεις, λάμδα συναρτήσεις ή αντικείμενα τύπου ομάδα που γυρνούν τιμές, όπως στο παράδειγμα.

Κλάση μετρητής {

 x, μέγιστο

 Αξία {

 Αν .x>=.μέγιστο Τότε Έξοδος

 .x++

 =Αληθές

 }

Κλάση:

 Τμήμα μετρητής (.x, .μέγιστο) {}

}

A=μετρητής(1,2) ' 1

B=μετρητής(10,18) ' 8

Γ=μετρητής(5,6) ' 1

Δ=μετρητής(1,3) ' 2

μετρητής=1

\\ μέτρημα 1+8+1+2= 12 φορές

Ενώ Αν(A->,Αν(B->,Αν(Γ->,Δ))) {

 Τύπωσε B.x, Γ.x, Δ.x, μετρητής

 μετρητής++

}

Υπάρχει το Αν\$() που γυρνάει αλφαριθμητικά.

Αναγνωριστικό: ΑΠΟΛ([ABS(]

ΤΥΠΩΣΕ ΑΠΟΛ(-5)+5

10

Δίνει την απόλυτη τιμή.

Τύπωσε Απολ((1,-2i))=Υπολ((1,-2i))

Επιστρέφει το μέτρο του μιγαδικού αν δώσουμε μιγαδικό

Αναγνωριστικό: ΑΡΙΘΜΟΣ.ΠΑΡΑΓΡΑΦΟΥ([PARAGRAPH.INDEX(]

ΑΡΙΘΜΟΣ.ΠΑΡΑΓΡΑΦΟΥ(αντικείμενο_έγγραφο, μεταβλητή)

Επιστρέφει από το κωδικό παραγράφου τον αριθμό (δίνει το 1 αν ο κωδικός αντιστοιχεί στη πρώτη παράγραφο)

Δες Παράγραφος()

Αναγνωριστικό: ΑΡΧΕΙΟΥ.ΜΗΚΟΣ([FILELEN(]

\\ Παράδειγμα

Έγγραφο Α\$={πρώτη σειρά

δεύτερη

τρίτη

τέταρτη

}

UTF_16LE=0

Σώσε.Έγγραφο Α\$, "checkme.doc", UTF_16LE

Αν Υπάρχει("checkme.doc") τότε {

Τύπωσε Αρχείου.Μήκος("checkme.doc")

}

Αναγνωριστικό: ΑΡΧΕΙΟΥ.ΣΤΑΜΠΑ([FILE.STAMP(]

Άνοιξε "one.txt" Για Ευρεία Εξαγωγή Αποκλειστικά ως #κ

Τύπωσε #κ, "Γεια χαρά"

Τύπωσε #κ, "Γεια χαρά2"

Κλείσε #κ

Τύπωσε Γραφή\$(Αρχείου.Στάμπα("one.txt"),"hh:nn:ss dd/mm/yyyy") , "utc χρονοσήμανση δημιουργίας"

Τύπωσε Γραφή\$(Αρχείου.Στάμπα("one.txt" ,1),"hh:nn:ss dd/mm/yyyy") , "utc χρονοσήμανση δημιουργίας, 1"

Τύπωσε Γραφή\$(Αρχείου.Στάμπα("one.txt" ,-1),"hh:nn:ss dd/mm/yyyy"), "Τοπική χρονοσήμανση δημιουργίας, -1"

Τύπωσε Γραφή\$(Αρχείου.Στάμπα("one.txt" ,2),"hh:nn:ss dd/mm/yyyy"), "utc χρονοσήμανση εγγραφής, 2"

Τύπωσε Γραφή\$(Αρχείου.Στάμπα("one.txt" ,-2),"hh:nn:ss dd/mm/yyyy"), "Τοπική χρονοσήμανση εγγραφής, -2"

Αναγνωριστικό: ΑΣΥΝ([RCOS(]

Επιστρέφει το συνημίτονο γωνίας σε ακτίνια

Δες το ΣΥΝ()

Αναγνωριστικό: ΑΤΟΞ.ΕΦ([RATN(
ΤΥΠΩΣΕ αΤοξ.Εφ(1)=π/4
Δες Τοφ.Εφ() για μοίρες

Αναγνωριστικό: ΒΑΣΗ([MDB(
ΤΥΠΩΣΕ ΒΑΣΗ("HELP2000")
Δίνει -1

Σημαίνει ότι η βάση HELP2000.MDB είναι μια βάση της M2000 (έχει δηλαδή Password που ισοδυναμεί με ταυτότητα)

Αναγνωριστικό: ΔΑΠΕΔ([FLOOR(
Είναι ίδια με την Ακ()

Δίνει την ακέραια τιμή, την ίδια αν είναι ακέραιος ή την κοντινότερη μικρότερη
Δες Στρογγ(), Οροφ()

Αναγνωριστικό: ΔΕΙΚΤΗΣ([POINTER(
Δείκτες για ομάδες (αντικείμενα της M2000)
δ=δείκτης(όνομα_ομάδας) ' δείκτης σε επώνυμη ομάδα (η ομάδα ζει όσο το τμήμα ή η συνάρτηση που τη δημιουργήσε)
δ=δείκτης((όνομα_ομάδας)) ' αντίγραφο της ομάδας ως ανώνυμη ομάδα
δ=δείκτης(πίνακαςΑ(1,2)) ' δείκτης σε ομάδα σε πίνακα (ανώνυμη ομάδα)

\\ οι επώνυμες ομάδες έχουν ένα όνομα μεταβλητής και υπάρχουν μέσα στο μέρος που δημιουργήθηκαν, στο τμήμα ή τη συνάρτηση για όσο ο "δημιουργός" είναι ακόμη σε εκτέλεση. Αν πάρουμε δείκτη από επώνυμη ομάδα, και τον επιστρέψουμε από το τμήμα που τερμάτισε και διέγραψε την ομάδα, θα βγει λάθος αν προσπαθήσουμε να το χρησιμοποιήσουμε.

\\ οι ανώνυμες ομάδες ζουν ή με δείκτες ή σε αποθηκευτικά μέσα όπως σωροί, καταστάσεις και πίνακες. Οι ομάδες αυτές είναι μεταφερόμενος, ή μέσω δείκτη ή μέσω των αντικειμένων (αποθηκευτικών μέσω). Τα μέσα αυτά έχουν πάντα δείκτη. Δηλαδή το όνομα μεταβλητής τους είναι δείκτης.

Οι δείκτες ξεχωρίζουν από τις αναφορές επειδή μπορούν να αλλάξουν αυτό που δείχνουν. Αν περάσουμε ένα δείκτη με αναφορά τότε μπορούμε να τον αλλάξουμε και θα αλλάξει και πιο πίσω (από εκεί που τον περάσαμε), ενώ αν τον περάσουμε με τιμή, θα έχουμε έναν δεύτερο δείκτη με ίδια λειτουργία, αλλά αν αλλάξουμε αντικείμενο δεν θα αλλαχθεί πιο πίσω.

\ \ Παράδειγμα

Κλάση Άλφα {

$x=10$

}

\ \ το δ είναι δείκτης σε ομάδα (αντικείμενο)

\ \ τώρα δείχνει ένα ανώνυμο αντικείμενο.

\ \ Τα ανώνυμα αντικείμενα "πεθαίνουν" όταν ο τελευταίος δείκτης χαθεί

\ \ now point to a unnamed group. The unnamed group lives untill no pointers point to it.

δ=Δείκτης(Άλφα())

\ \ Αυτό είναι το ίδιο με αυτό δ->Άλφα()

Τύπωσε δ=>x

ζ=Άλφα()

\ \ now point to a named group (we can't use the pointer if ζ erased, επειδή has a reference only)

\ \ τώρα ο δείκτης δείχνει σε ένα επώνυμο αντικείμενο τύπου ομάδα. Το άλλο που έδειχνε έσβησε

\ \ Τα επώνυμα αντικείμενα "πεθαίνουν" όταν τερματίσει το τμήμα ή η συνάρτηση που τα δημιουργήσει.

\ \ ο δείκτης θα έχει ισχύ όσο το επώνυμο αντικείμενο υπάρχει, ή εφόσον του δώσουμε άλλο αντικείμενο

δ=Δείκτης(ζ)

\ \ Αυτό είναι το ίδιο με αυτό δ->ζ

Τύπωσε δ=>x

δ=>x+=10

Τύπωσε ζ.x=δ=>x ' επειδή ζ.x είναι το ίδιο με το δ=>x

δ=Δείκτης((ζ)) ' look the (), we get a copy of ζ as unnamed group.

\ \ Αυτό είναι το ίδιο με αυτό δ->(ζ)

Τύπωσε δ=>x

δ=>x+=10

Τύπωσε ζ.x<>δ=>x ' επειδή ζ.x δεν είναι το ίδιο με το δ=>x

Λίστα ' Δείχνει όλες τις μεταβλητές (εκτός από τις ιδιωτικές των ομάδων, και απ' ότι περιέχουν οι δείκτες)

Αναγνωριστικό: ΔΕΚ([FRAC(]

Τύπωσε Δεκ(1.234)

.234

επιστρέφει μόνο τα δεκαδικά, και πάντα θετική τιμή ή μηδέν

Αναγνωριστικό: ΔΕΝ [NOT_2]

Συνάρτηση αναστροφής

Είναι η ΟΧΙ με άλλο αναγνωριστικό

ΤΥΠΩΣΕ ΟΧΙ ΑΛΗΘΕΣ

ΤΥΠΩΣΕ ΟΧΙ (ΟΧΙ 2)

δίνει 2

Είναι ψευδοσυνάρτηση: Βάζει στη θέση της το $-1-1^*$

οπότε αν ακολουθεί το 0 τότε $-1-1^*0$ δίνει -1

αν ακολουθεί το -1 δίνει $-1-1^*(-1)$ δίνει 0

Αν ακολουθεί αρνητικό νούμερο θα δώσει θετικό μείον ένα, αν ακολουθεί θετικό νούμερο θα δώσει αρνητικό μείον ένα...

στο -5 θα δώσει 4

στο 4 θα δώσει -5

Άρα για οποιοδήποτε X υπάρχει μια $F(X)$ που δίνει ένα K όπου $X \neq K$ (το X είναι διάφορο του K) και ισχύει και το αντίστροφο ότι $F(K)=X$. Από αυτόν τον ορισμό είναι φανερό ότι ακόμα και για το μηδέν υπάρχει ένα μοναδικό νούμερο που δίνει το μηδέν στην ίδια συνάρτηση. Επίσης θα μπορούσε κανείς να χρησιμοποιήσει διαφορετικό τύπο και πάλι θα είχε μια συνάρτηση αναστροφής...π.χ. $-2-1^*(X)=K$ αν βάλω το -2 θα πάρω το 0 και αν βάλω το 0 θα πάρω το -2

Επειδή είναι ψευδοσυνάρτηση και γίνεται αφαίρεση δεν έχει την σειρά που έχουν οι εντολές ΚΑΙ και Η και ΑΠΟ

οπότε

η ΤΥΠΩΣΕ ΑΛΗΘΕΣ ΚΑΙ ΟΧΙ ΨΕΥΔΕΣ Η ΑΛΗΘΕΣ

η ΤΥΠΩΣΕ -1 ΚΑΙ $-1-1^*(-1)$ Η -1

δίνει -1 (ΑΛΗΘΕΣ)

Πρώτα θα γυρίσει το ...Η ΑΛΗΘΕΣ που θα βρει το ΟΧΙ ΨΕΥΔΕΣ (ως -1- -1), οπότε -1 η -1 δίνει -1 και μετά θα βρει το -1

Προσοχή πάντα από δεξιά αναλύει η γλώσσα τις εκφράσεις, αλλά τα ενδιάμεσα αποτελέσματα τα προωθεί από αριστερά προς τα δεξιά (με αναδρομική διαδικασία).

η ΤΥΠΩΣΕ ΑΛΗΘΕΣ ΚΑΙ ΟΧΙ (ΨΕΥΔΕΣ Η ΑΛΗΘΕΣ)

η ΤΥΠΩΣΕ -1 ΚΑΙ $-1-1^*(0 \text{ Η } -1)$

δίνει 0 (ΨΕΥΔΕΣ)

Αναγνωριστικό: ΔΙΑ [DIV]

\\ ακέραια διαίρεση

Τύπωσε 10 δια 2
5

Αναγνωριστικό: ΔΙΑ# [DIV#]
\\ Ευκλείδεια διαίρεση
 $\alpha = -20$
 $\beta = 6$
\\ Ευκλείδεια διαίρεση και υπόλοιπο
 $\gamma = \alpha \text{ Διά } \beta$
 $\delta = \alpha \text{ Υπόλ } \beta$
Τύπωσε γ , δ ' -4 4
Τύπωσε $\alpha = \beta * \gamma + \delta$
\\ Κανονική διαίρεση και υπόλοιπο
 $\gamma = \alpha \text{ Διά } \beta$
 $\delta = \alpha \text{ Υπόλ } \beta$
Τύπωσε γ , δ ' -3 -2
Τύπωσε $\alpha = \beta * \gamma + \delta$

Αναγνωριστικό: ΔΙΑΡΘΡΩΣΗ([BUFFER(
Διαρθ1=Διάρθρωση("όνομα_αρχείου")
φορτώνει σε μια νέα διάρθρωση μνήμης το αρχείο που θέλουμε και γυρίζει το δείκτη

Μπορούμε να φορτώνουμε εικόνες png και άλλες και να τις εμφανίζουμε με το Εικόνα και το Διάφανο

Αναγνωριστικό: ΔΙΑΣΤΑΣΗ([DIMENSION(
Πίνακας A(10,20)
Τύπωσε Διάσταση(α ())
2
Τύπωσε Διάσταση(α ()), Διάσταση(α ()),2)
10 20

Μπορούμε να διαβάζουμε πόσες διαστάσεις έχει ένας πίνακας και κάθε διάσταση τι μέγεθος έχει.

Μπορούμε να δώσουμε το πίνακα με σταθερό όνομα ή να δώσουμε αλφαριθμητικό.

Τύπωσε Διάσταση("α",1), Διάσταση("α",2)

Πίνακας Βάση 1, $\alpha(10)$ ' από 1 έως 10 η βάση μπήκε εδώ μόνο για το $\alpha()$

Βάση 1 ' Εδώ άλλαξε γενικά

Πίνακας B(10)

Τύπωσε Διάσταση(B(),0) ' δίνει την βάση του πίνακα 1 ή 0, εδώ 1

Τύπωσε Μήκος(B()) ' δίνει τον συνολικό αριθμό στοιχείων

Υπάρχει δυνατότητα για δείκτη σε πίνακα

A=(1,2,3,4,5) ' αυτοί οι πίνακες έχουν βάση 0, και είναι μονοδιάστατοι

Τύπωσε Διάσταση(A), Μήκος(A)

B=((1,2),(3,4)) ' είναι μονοδιάστατος πίνακας με στοιχεία πίνακες.

Οι δείκτες πάντα θα έχουν περιεχόμενο! Δεν έχουμε Null δείκτες στη M2000

Μπορούμε να βάλουμε δείκτη και σε κανονικό πίνακα

A=B()

' εδώ θα γίνει αντιγραφή πίνακα, γιατί το B() στην εκχώρηση παίρνει αντίγραφο

B()=B

Από την 9.4 έκδοση, αναθ. 4 μπορούμε να βάζουμε για κάθε διάσταση από που θα ξεκινάει και που θα τερματίζει. Αν τα βάλουμε ανάποδα τα γυρνάει αυτόματα ο διερμηνευτής

Πίνακας α(3 έως 10, -4 έως 8)

αρχή1=Διάσταση(α(),1, 0)

τέλος1=Διάσταση(α(),1, 1)

αρχή2=Διάσταση(α(),2, 0)

τέλος2=Διάσταση(α(),2, 1)

Τύπωσε "Διαστάσεις:"; Διάσταση(α())

Τύπωσε "Συνολικά στοιχεία:"; Μήκος(α())

Τύπωσε "Πίνακας των ";Διάσταση(α(),1);" γραμμών X ";Διάσταση(α(),2);" στηλών"

Τύπωσε "Πρώτη διάσταση από ";αρχή1; " έως "; τέλος1

Τύπωσε "Δεύτερη διάσταση από ";αρχή2; " έως "; τέλος2

Αναγνωριστικό: ΔΙΕΥΘΥΝΣΗΑΠΟ [ADDRESSOF]

Δες το Τμήμα()

Τα τμήματα που καλούνται από τη γλώσσα μηχανής, είναι σαν να εκτελούνται στο τμήμα που εκκινήθηκε η γλώσσα μηχανής.

(δεν διαγράφουν ότι δημιουργούν, βλέπουν τις τοπικές του τμήματος)

Διάβασε το παράδειγμα:

οθόνη

μακρυά=αν(τυχαίος>.5->ψευδές, αληθές)

τύπωσε αν\$(μακρυά=αληθές->"Χρήση dec ecx && jnz rel32","Χρήση loop rel8")

κλιπ=ψευδές

```
τύπωσε "Χρησιμοποίησε αυτό:"
τύπωσε "https://defuse.ca/online-x86-assembler.htm#disassembly2"
// τύπωσε ΕΛΕΓΧΟΣ.ΣΩΡΟΥ // (δίνει το μέγεθος του σωρού επιστροφής του διερμηνευτή σε
χρήση)
// τύπωσε ΕΛΕΓΧΟΣ.ΜΕΓΕΘΟΣ.ΣΩΡΟΥ //(το μέγεθος του σωρού επιστροφής, >30MB )
```

```
διάρθρωση κενή ΔυαδικάΣτοιχεία ως μακρύς*10
επιστροφή ΔυαδικάΣτοιχεία, 1:=500
διάρθρωση κώδικα Άλφα ως ψηφίο*1024
```

```
τμήμα Βήτα {
    τύπωσε :τύπωσε "Ολοκληρώθηκε"
}
τμήμα Βήτα2 {
    τύπωσε εκφρ(ΔυαδικάΣτοιχεία,0),
}
μετρ_προγρ=0
Πρόλογος_Κώδικα()
Κωδικοί_8μπιτ(0x51, 0x31, 0xC0) ' push ecx : xor eax, eax
Κωδικός_Μακρύς(0xb9, 100) ' mov ecx, 100
σημείο_κλήσης=μετρ_προγρ
Κωδικοί_8μπιτ(0x01, 0xc8) ' add eax, ecx (eax=eax+ecx)
Κωδικός_Μακρύς(0xa3, ΔυαδικάΣτοιχεία(0)) ' mov ds:ΔυαδικάΣτοιχεία(0), eax
    Κωδικοί_8μπιτ(0x51, 0x50) ' push ecx: push eax
    ΚλήσηΤμήματος(ΔιεύθυνσηΑπό Βήτα2)
    Κωδικοί_8μπιτ(0x58, 0x59) ' pop eax : pop ecx
αν μακριά τότε
    Κωδικοί_8μπιτ(0x49) ' dec ecx
    Επανάληψη2(σημείο_κλήσης)
αλλιώς
    Επανάληψη(σημείο_κλήσης) // 2 bytes (dec ecx)
τέλος αν
ΚλήσηΤμήματος(ΔιεύθυνσηΑπό Βήτα)
Κωδικοί_8μπιτ(0x59, 0x31, 0xC0) ' pop ecx : xor eax,eax
```

```
Επιστροφή()
έγγραφο κειμ$
για i=0 έως μετρ_προγρ-1
    κειμ$=δεκαεξ$(εκφρ(Άλφα, i),1)
επόμενο
τύπωσε "Κώδικας:";κειμ$
αν κλιπ τότε πρόχειρο κειμ$: έξοδος
Δες Καλά {
```

```

αναλυτής
εκτέλεση κώδικα Άλφα, 0
τύπωσε φόρτος
}
M=δυναδικό.ακέραιο(Λάθος) // επιστρέφει τον eax
τύπωσε εκφρ(ΔυναδικάΣτοιχεία,0)
τύπωσε ΕΛΕΓΧΟΣ.ΣΩΡΟΥ
τέλος
ρουτίνα Κωδικοί_8μπιτ()
    ενώ όχι κενό
        επιστροφή Άλφα, μετρ_προγρ:=αριθμός
        μετρ_προγρ++
    τέλος ενώ
τέλος ρουτίνας
ρουτίνα Κωδικός_Μακρύς()
    επιστροφή Άλφα, μετρ_προγρ:=αριθμός, μετρ_προγρ+1:=αριθμός ως μακρύς
    μετρ_προγρ+=5
τέλος ρουτίνας
ρουτίνα Πρόλογος_Κώδικα()
    Κωδικοί_8μπιτ(0x58, 0x59, 0x59, 0x59, 0x50)
    τέλος ρουτίνας
ρουτίνα Επιστροφή()
    επιστροφή Άλφα, μετρ_προγρ:=0xC3
    μετρ_προγρ++
τέλος ρουτίνας
ρουτίνα ΚλήσηΤμήματος()
    επιστροφή Άλφα, μετρ_προγρ:=0x68, μετρ_προγρ+1:=αριθμός ως μακρύς
    μετρ_προγρ+=5
    επιστροφή Άλφα, μετρ_προγρ:=0xE8, μετρ_προγρ+1:=τμήμα()-μετρ_προγρ-5-Άλφα(0) ως
μακρύς
    μετρ_προγρ+=5
τέλος ρουτίνας
// 0xE2 cb    LOOP rel8    D    Valid    Valid    Decrement ECX ; jump short av count ≠ 0.
ρουτίνα Επανάληψη()
    επιστροφή Άλφα, μετρ_προγρ:=0xE2, μετρ_προγρ+1:=(αριθμός-μετρ_προγρ-2+256)
    μετρ_προγρ+=2
τέλος ρουτίνας
// 0x49          dec    ecx // OF, SF, ZF, AF, and PF flags are set according έως the result.
// 0x85 jnz rel32
// by default this (αριθμός-μετρ_προγρ-5+0xFFFFFFFF) is type of modulo 2^32 (same as
binary.add())
ρουτίνα Επανάληψη2()
    επιστροφή Άλφα, μετρ_προγρ:=0x0F, μετρ_προγρ+1:=0x85, μετρ_προγρ+2:=

```

(αριθμός-μετρ_προγρ-5+0xFFFFFFFF) ως μακρύς
μετρ_προγρ+=6
τέλος ρουτίνας

Αναγνωριστικό: ΔΟΚΙΜΗ([TEST(
Μπορούμε με την εντολή Δοκιμή "Όνομα διακοπής" να αλλάζουμε το τίτλο της φόρμας Έλεγχος.
Η Συνάρτηση Δοκιμή() μπορεί να αφήνει διακοπές ροής από εντολές Δοκιμή μέχρι να εμφανιστεί
ο τίτλος που θέλουμε. Ή μέχρι να ικανοποιείται μια συνθήκη.

```
Για i=1 Έως 100 {  
    Αν i=5 Τότε Δοκιμή "Start 1", Δοκιμή("Point 1"), i  
    Αν i=55 Τότε Δοκιμή "Point 1"  
    Τύπωσε i  
}
```

Αναγνωριστικό: ΔΥΑΔΙΚΗ.ΠΕΡΙΣΤΡΟΦΗ([BINARY.ROTATE(
\\ -31 έως 31
α=0xFAAA000A
Δεκαεξ ΔΥΑΔΙΚΗ.ΠΕΡΙΣΤΡΟΦΗ(α, 1)
Δεκαεξ ΔΥΑΔΙΚΗ.ΠΕΡΙΣΤΡΟΦΗ(α, 2)
Δεκαεξ ΔΥΑΔΙΚΗ.ΠΕΡΙΣΤΡΟΦΗ(α, -1)
Δεκαεξ ΔΥΑΔΙΚΗ.ΠΕΡΙΣΤΡΟΦΗ(α, -2)

Αναγνωριστικό: ΔΥΑΔΙΚΟ([USGN(
α=8000000000
Τύπωσε Δυαδικό(α)=0xFFFFFFFF ' κόβει το α στο μέγιστο
α=-34324
Τύπωσε Δυαδικό(α)=0 ' κόβει το α στο ελάχιστο
α=34324
Τύπωσε Δυαδικό(α)=34324

Αναγνωριστικό: ΔΥΑΔΙΚΟ.ΑΚΕΡΑΙΟ([UINT(
' Επιστρέφει έναν χωρίς πρόσημο ακέραιο με ίδια δυαδικά ψηφία

$\alpha = -345$

Τύπωσε Δεκαεξ\$(ΔΥΑΔΙΚΟ.ΑΚΕΡΑΙΟ(α))

Δεκαεξ ΔΥΑΔΙΚΟ.ΑΚΕΡΑΙΟ(α), α ' Το δεύτερο τυπώνει ???- γιατί είναι αρνητικός.

Αναγνωριστικό: ΔΥΑΔΙΚΟ.ΑΝΤΙ([BINARY.NEG(]

Υπάρχει και ως ΔΥΑΔΙΚΟ.ΑΝΤΙΣΤΡΟΦΟ()

Δέχεται έναν αριθμό με πρόσημο 32 bit

και δίνει το αντίστροφο με ίδια bits (ψηφία) σε αριθμό χωρίς πρόσημο 32 bits

ο β παρακάτω θα πάρει έναν αρνητικό, τον -21555

Το αντίστροφό του είναι το 21554, διότι ως αναπαράσταση οι δυο αριθμοί ως δυαδικοί δίνουν το FFFFFFFF, το οποίο χωρίς πρόσημο είναι αυτό: 0xFFFFFFFF και με πρόσημο είναι αυτό: 0xFFFFFFFF& αλλά και τα δυο έχουν τα ίδια ψηφία ως δυαδική αναπαράσταση

$\beta = 0xFFFFABCD$

Τύπωσε $\alpha \geq 0 = \Psiευδ\acute{\epsilon}ς$

Τύπωσε $\beta = -21555$

Τύπωσε Δυαδικό.Αντίστροφο(β)=21554

Τύπωσε Ακέραιο.Δυαδικό(Δυαδικό.Αντίστροφο(β))=21554

Τύπωσε Δεκαεξ\$(Δυαδικό.Ακέραιο(β))="FFFFABCD", Δυαδικό.Ακέραιο(β)

Τύπωσε Δεκαεξ\$(Δυαδικό.Αντίστροφο(β)+Δυαδικό.Ακέραιο(β))="FFFFFFF"

Τύπωσε Δυαδικό.Όχι(Δυαδικό.Ακέραιο(β))=21554

Τύπωσε Δεκαεξ\$(Δυαδικό.Όχι(Δυαδικό.Ακέραιο(β))+Δυαδικό.Ακέραιο(β))="FFFFFFF"
 $\beta = 21555$

Τύπωσε Δεκαεξ\$(β)="00005433"

Τύπωσε Δυαδικό.Αντίστροφο(β)=4294945740

Τύπωσε Ακέραιο.Δυαδικό(Δυαδικό.Αντίστροφο(β))=-21556

Τύπωσε Δεκαεξ\$(Δυαδικό.Ακέραιο(β))="00005433", Δυαδικό.Ακέραιο(β)

Τύπωσε Δεκαεξ\$(Δυαδικό.Αντίστροφο(β)+Δυαδικό.Ακέραιο(β))="FFFFFFF"

Τύπωσε Δυαδικό.Όχι(Δυαδικό.Ακέραιο(β))=4294945740

Τύπωσε Δεκαεξ\$(Δυαδικό.Όχι(Δυαδικό.Ακέραιο(β))+Δυαδικό.Ακέραιο(β))="FFFFFFF"

Αναγνωριστικό: ΔΥΑΔΙΚΟ.ΑΠΟ([BINARY.XOR(]

$\alpha = 0xFAAA$

$\beta = 0x5550$

Δεκαεξ ΔΥΑΔΙΚΟ.ΑΠΟ(α, β)

Αναγνωριστικό: ΔΥΑΔΙΚΟ.Η([BINARY.OR(]

$\alpha = 0xFAAA$

$\beta = 0x5550$
Δεκαεξ ΔΥΑΔΙΚΟ.Η(α, β)

Αναγνωριστικό: ΔΥΑΔΙΚΟ.ΚΑΙ([BINARY.AND(
 $\alpha = 0xFAAA$
 $\beta = 0x5550$
Δεκαεξ ΔΥΑΔΙΚΟ.ΚΑΙ(α, β)

Αναγνωριστικό: ΔΥΑΔΙΚΟ.ΟΛΙΣΘΗΣΗ([BINARY.SHIFT(
\\ -31 έως 31
 $\alpha = 0xFAAA$
Δεκαεξ ΔΥΑΔΙΚΟ.ΟΛΙΣΘΗΣΗ($\alpha, 1$)
Δεκαεξ ΔΥΑΔΙΚΟ.ΟΛΙΣΘΗΣΗ($\alpha, 2$)
Δεκαεξ ΔΥΑΔΙΚΟ.ΟΛΙΣΘΗΣΗ($\alpha, -1$)
Δεκαεξ ΔΥΑΔΙΚΟ.ΟΛΙΣΘΗΣΗ($\alpha, -2$)

Αναγνωριστικό: ΔΥΑΔΙΚΟ.ΟΧΙ([BINARY.NOT(
Παίρνει αριθμό χωρίς πρόσημο από 0x0 έως 0xFFFFFFFF
Τύπωσε Δεκαεξ\$(Δυαδικό.Όχι(0xAAAAAAAA)) '55555555

Δες Δυαδικό.Αντί()

Αναγνωριστικό: ΔΥΑΔΙΚΟ.ΠΡΟΣΘΕΣΗ([BINARY.ADD(
ΔΥΑΔΙΚΟ.ΠΡΟ() ή ΔΥΑΔΙΚΟ.ΠΡΟΣΘΕΣΗ(
Κάνει πρόσθεση δυο δυαδικών αριθμών 32bit, χωρίς πρόσημο, και δίνει το υπόλοιπο από 2^{32} ,
η με άλλα λόγια αφαιρεί την τυχόν υπερχείλιση.

ΔΕΚΑΕΞ ΔΥΑΔΙΚΟ.ΠΡΟ(0xF0000000,0xF0000001)
0xE0000001

Αναγνωριστικό: ΔΥΟΜΙΣΑ([HILOWWORD(]

α=0xFAAA

β=0x5550

Δεκαεξ ΔΥΟΜΙΣΑ(α,β)

Τύπωσε ΚάτωΜισό(ΔΥΟΜΙΣΑ(α,β))=β

Τύπωσε ΠάνωΜισό(ΔΥΟΜΙΣΑ(α,β))=α

Αναγνωριστικό: ΕΓΓΡΑΦΕΣ([RECORDS(]

Δέχεται ως όρισμα τον αριθμό ενός ανοιχτού αρχείου και δίνει των αριθμό εγγραφών αν είναι τύπου ΠΕΔΙΩΝ ή το μέγεθος σε bytes αν είναι οτιδήποτε άλλο.

Αναγνωριστικό: ΕΓΓΡΑΦΟΥ.ΛΕΞΕΙΣ([DOC.WORDS(]

Έγγραφο α\$="Γεια χαρά και ξανά γεια σου Κόσμε"

ν=Εγγράφου.Λέξεις(α\$)

Λέξεις α\$

Για ι=1 Έως ν

Τύπωσε Κεφ\$(Γράμμα\$)

Επόμενο ι

Το παράδειγμα δίνει τις λέξεις σε ταξινομημένη σειρά, χωρίς να ομαδοποιήσει τις όμοιες.

Το ν επιστρέφει τον αριθμό των λέξεων

η εντολή λέξεις δίνει ανάλογα την επιλογή Εγγράφου.Λέξεις() ή Εγγράφου.Μοναδικές.Λέξεις()

Αναγνωριστικό: ΕΓΓΡΑΦΟΥ.ΜΗΚΟΣ([DOC.LEN(]

Α\$={1η γραμμή

δεύτερη γραμμή

}

ΕΓΓΡΑΦΟ Α\$

ΤΥΠΩΣΕ ΕΓΓΡΑΦΟΥ.ΜΗΚΟΣ(Α\$) ' υπολογίζει το μήκος χωρίς να φτιάξει αντίγραφο

ΤΥΠΩΣΕ ΜΗΚΟΣ(Α\$) ' εδώ πρώτα φτιάχνει ένα αντίγραφο και μετά υπολογίζει το μήκος

Δες εντολή ΕΓΓΡΑΦΟ

Αναγνωριστικό: ΕΓΓΡΑΦΟΥ.ΜΟΝΑΔΙΚΕΣ.ΛΕΞΕΙΣ([DOC.UNIQUE.WORDS(]

Έγγραφο α\$="Γεια χαρά και ξανά γεια σου Κόσμε"

$v = \text{Εγγραφου.Μοναδικές.Λέξεις}(α\$)$

Λέξεις $α\$$

Για $i=1$ Έως v

 Τύπωσε $\text{Κεφ}\$(\text{Γράμμα}\$)$

Επόμενο i

Το παράδειγμα επιστρέφει ταξινομημένη τη λίστα των λέξεων και σε κάθε λέξη βάζει σε παρένθεση τον αριθμό εμφανίσεων.

το v είναι ο αριθμός, αλλά για να πάρουμε τις λέξεις τις βάζει η εντολή Λέξεις στο σωρό τιμών

Αναγνωριστικό: $\text{ΕΓΓΡΑΦΟΥ.ΠΑΡ}(\quad [\text{DOC.PAR}()])$

Έγγραφο $A\$, B\%$

$A\$ = \{$

$\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha$

$\beta\beta\beta\beta\beta\beta\beta\beta$

$\gamma\gamma\gamma\gamma\gamma\gamma\gamma\gamma$

$\}$

$B\$ = \{$

$\}$

Τύπωσε $\text{Εγγραφου.Παρ}(A\$), \text{Εγγραφου.Παρ}(B\$)$

Επιστρέφει τον αριθμό παραγράφων

Αναγνωριστικό: $\text{ΕΓΓΡΑΨΙΜΟ}(\quad [\text{WRITABLE}()])$

$\text{ΤΥΠΩΣΕ ΕΓΓΡΑΨΙΜΟ}("a:")$

Αναγνωριστικό: $\text{ΕΓΚΥΡΟ}(\quad [\text{VALID}()])$

1) Έγκυρο(αριθμητική παράσταση)

Τυπωσε $\text{Εγκυρο}(\Pi(12))$

 Τυπώνει -1 αν υπάρχει στοιχείο πίνακα Π με δείκτη 12

Τυπωσε $\text{Εγκυρο}(\pi/v)$

 αν το v είναι μηδέν θα δώσει 0, αν δεν υπάρχουν οι μεταβλητές θα δώσει 0, αν υπάρχουν και βγαίνει αποτέλεσμα θα δώσει -1

μεταξύ των παρενθέσεων της ΕΓΚΥΡΟ μπορούμε να βάλουμε ότι παράσταση θέλουμε και θα μας δώσει αντί για την τιμή αν είναι η παράσταση σωστή ή λάθος

Εκτός από αυτή την συνάρτηση για να βρούμε λάθη και να τα ξεπεράσουμε, μπορούμε να χρησιμοποιήσουμε την $\Delta\text{Ε}\Sigma$ όπου ελέγχει ολόκληρη ομάδα εντολών και μπορούμε να

διαβάσουμε το λάθος σε νούμερο ΛΑΘΟΣ ή σε περιγραφή ΛΑΘΟΣ\$ χωρίς να σταματήσει η ροή του προγράμματος!

2) Έγκυρο(@αναγνωριστικό)

σταθερή M=1000

X=10

Z\$="αλφα"

Πίνακας A()

N=(1,2,3,4,5)

Λ=Σωρός:=1,2,3,4,5

Ομάδα Αλφα { X=10, Y=50, A\$="οκ"}

Τύπωσε Έγκυρο(@X), Έγκυρο(@Z\$), Έγκυρο(@A()), Έγκυρο(@M)

Τύπωσε Έγκυρο(@N), Έγκυρο(@Λ), Έγκυρο(@Αλφα.X), Έγκυρο(@Αλφα.A\$)

δίνει -1 γιατί υπάρχει σαν αναγνωριστικό. δεν εκτελεί παράσταση

Για αναγνωριστικά τμημάτων και συναρτήσεων χρησιμοποιούμε την Τμήμα()

3) Έγκυρο(@αναγνωριστικό1 ως αναγνωριστικό2)

Κοιτάει αν το αναγνωριστικό1 είναι του τύπου του αναγνωριστικό2

Αν τα αναγνωριστικά είναι ομάδες τότε κοιτάει αν το πρώτο αναγνωριστικό έχει ίδια μέλη με αυτό του δεύτερου. Μπορεί το πρώτο αναγνωριστικό να έχει περισσότερα μέλη, αλλά αρκεί να έχει τα ίδια με το δεύτερο αναγνωριστικό. Η σύγκριση γίνεται σε ονόματα μόνο. Επίσης και τα ονόματα τμημάτων και συναρτήσεων, και των ομάδων στις ομάδες συμμετέχουν.

Αν το αναγνωριστικό δύο έχει διαφορετική υπερίκλαση από το ένα τότε δεν κοιτάει καθόλου τι μέλη υπάρχουν όμοια, και γυρίζει 0, δηλαδή ψευδές.

Αναγνωριστικό: ΕΙΚΟΝΑ([IMAGE(]

1) Παράγει μια διάρθρωση μνήμης από μια εικόνα σε αλφαριθμητικό.

Εικ1=Εικόνα(α\$)

2) Παράγει μια διάρθρωση μνήμης από μια εικόνα (όχι emf/wmf) σε αντικείμενο διάρθρωση εικόνα

Έχει φτιαχτεί για PNG

Εικ1=Εικόνα(α, X1[, [Π1, Y1,] T1])

X1 χρώμα που εφαρμόζεται σε εικονοστοιχεία που έχουν δηλωθεί ως διάφανα (στο PNG)

Π1,Y1 πλάτος και ύψος σε twips

T1=4 κάνει αντιστροφή οριζόντια (παίζουν 16 τρόποι)

3) Qr Code

Εικ1=Εικόνα("http://www.vbforums.com")

// χρώμα 0 - 15 ή Χρώμα(K, Π, M) K, Π, M: 0 - 255 ή #FF0077 (html color)

Εικ1=Εικόνα("http://www.vbforums.com", χρώμα(255, 0, 128))

// Αντοχή στα λάθη: 0 - 3 (0 εξ ορισμού)

Εικ1=Εικόνα("http://www.vbforums.com", χρώμα(255, 0, 128), 3)
Εικ1=Εικόνα("http://www.vbforums.com", , 3)

Αναγνωριστικό: ΕΙΚΟΝΑ.Υ([IMAGE.Υ(
ΤΥΠΩΣΕ ΕΙΚΟΝΑ.Υ(A\$)

Το A\$ πρέπει να περιέχει εικόνα. Η τιμή είναι σε twips και μας δίνει το ύψος της εικόνας.

Το A πρέπει να είναι διάρθρωση μνήμης με εικόνα (φορτωμένο αρχείο)
ΤΥΠΩΣΕ ΕΙΚΟΝΑ.Υ(A)

Αναγνωριστικό: ΕΙΚΟΝΑ.Υ.ΣΗΜΕΙΑ([IMAGE.Υ.PIXELS(
ΤΥΠΩΣΕ ΕΙΚΟΝΑ.Υ.ΣΗΜΕΙΑ(A\$)

Το A\$ πρέπει να περιέχει εικόνα. Η τιμή είναι σε σημεία (pixels) και μας δίνει το ύψος της εικόνας.

ΤΥΠΩΣΕ ΕΙΚΟΝΑ.Υ.ΣΗΜΕΙΑ(A)

Το A πρέπει να είναι διάρθρωση μνήμης με εικόνα (φορτωμένο αρχείο με Εικόνα())

Αναγνωριστικό: ΕΙΚΟΝΑ.Χ([IMAGE.Χ(
ΤΥΠΩΣΕ ΕΙΚΟΝΑ.Χ(A\$)

Το A\$ πρέπει να περιέχει εικόνα. Η τιμή είναι σε twips και μας δίνει το πλάτος της εικόνας.

ΤΥΠΩΣΕ ΕΙΚΟΝΑ.Χ(A)

Το A πρέπει να είναι διάρθρωση μνήμης με εικόνα (φορτωμένο αρχείο με Εικόνα())

Αναγνωριστικό: ΕΙΚΟΝΑ.Χ.ΣΗΜΕΙΑ([IMAGE.Χ.PIXELS(
ΤΥΠΩΣΕ ΕΙΚΟΝΑ.Χ.ΣΗΜΕΙΑ(A\$)

Το A\$ πρέπει να περιέχει εικόνα. Η τιμή είναι σε σημεία (pixels) και μας δίνει το πλάτος της εικόνας.

ΤΥΠΩΣΕ ΕΙΚΟΝΑ.Χ.ΣΗΜΕΙΑ(A)

Το A πρέπει να είναι διάρθρωση μνήμης με εικόνα (φορτωμένο αρχείο με Εικόνα())

Αναγνωριστικό: ΕΙΝΑΙ [IS]

Τελεστής Είναι

Παραλλαγή Είναι Τύπος για έλεγχο τύπων ομάδων

\\ μόνο για αντικείμενα α και β

Τύπωσε α είναι β

Για ομάδες μόνο:

```
Κλάση Αλφα {X=10}
M=Αλφα()
Τύπωσε M είναι τύπος Αλφα
Ομάδα Κάππα {
    Τύπος: Ένα, Δύο
    X=10
}
Τύπωσε Κάππα είναι τύπος Δύο
```

```
Αναγνωριστικό: ΕΚΘ( [EXP()]
? ΕΚΘ(ΛΦ(3))==3
? ΕΚΘ((1,0i))=ΕΚΘ(1)
```

```
Αναγνωριστικό: ΕΚΦΡ( [EVAL()]
X=2
ΤΥΠΩΣΕ ΕΚΦΡ("X**2")
δίνει το 4
```

Προσοχή στην ΕΓΚΥΡΟ(
η ΕΓΚΥΡΟ("X**2") δίνει -1 όχι γιατί είναι ελεγμένη η έκφραση X^{**2} αλλά γιατί ανάμεσα στις παρενθέσεις υπάρχει αλφαριθμητικό και αυτό είναι έγκυρο!

Για να πιάσουμε το λάθος πριν γίνει θα πρέπει να βάλουμε την ΕΚΦΡ("κατι") σε μια δομή ΔΕΣ

```
ΔΕΣ ΚΚ {
ΤΥΠΩΣΕ ΕΚΦΡ("X**2")
}
Αν Όχι ΚΚ Τότε Τυπωσε"Πρόβλημα με την έκφραση"
' δεν σταματάει η ροή του προγράμματος
```

Αν δεν βάλουμε την ΔΕΣ τότε θα παραχθεί λάθος και αυτό θα βγει παραέξω, μέχρι να το πιάσει κάποια άλλη ΔΕΣ παραπάνω η αν δεν υπάρχει να βγούμε στην γραμμή εντολών και να πάρουμε το μήνυμα ότι υπάρχει λάθος έκφραση το αλφαριθμητικό.

```
Αναγνωριστικό: ΕΝΚΟΜ( [INKEY()]
έλεγχος πληκτρολογίου με καθυστέρηση για πάτημα πλήκτρου σε αυτό το διάστημα.
Τύπωσε Ενκομ(1000)
```

A=1
Nήμα {
 A++
} ως M κάθε 10
Τύπωσε Ενκομ(2000)
Τύπωσε A

Αναγνωριστικό: ΕΝΩΣΗ([CONS()
α=(1,2,3,4,5)
β=Ένωση(α, (6,7,8), α)
Τύπωσε β

Αναγνωριστικό: ΕΠΟΜΕΝΑ([CDR()
α=((1,2), (3,4), (5, 6))
b=Πρώτο(α)
Τύπωσε Πρώτο(b) ' 1
Τύπωσε Πρώτο(Πρώτο(α)) ' 1
Τύπωσε Επόμενα(Πρώτο(Επόμενα(Επόμενα(α)))) ' 6

Αναγνωριστικό: ΕΤΟΙΜΟ([READY()
Επειδή σε ένα γεγονός σε φόρμα μπορεί να έχει ξεκινήσει μια φορά και να ξεκινήσει και δεύτερη (υπάρχει περίπτωση ότι αφήσουμε μια φόρμα για άλλη και πάλι την φέρουμε στο προσκήνιο), όταν φορτώνουμε ένα έγγραφο με την Φόρτωση.Έγγραφο μπορούμε να ξέρουμε αν η μεταβλητή που έχουμε δηλώσει ως έγγραφο είναι απασχολημένη ή όχι και ανάλογα να πράξουμε
Αν Όχι Έτοιμο(εγγραφ\$) Τότε Έξοδος
Φόρτωση.Έγγραφο εγγραφ\$, όνομα_αρχείου\$

Αν η φόρμα έχει όνομα Φόρμα1, τότε η εντολή επίπεδο μπορεί να χρησιμοποιηθεί για να δώσει στη φόρτωση.έγγραφο "ανάδραση", ώστε αν τερματίσει η φόρμα να στείλει μήνυμα για να τερματίσει και η φόρτωση.

Αν Όχι Έτοιμο(εγγραφ\$) Τότε Έξοδος
Επίπεδο Φόρμα1 {
 Φόρτωση.Έγγραφο εγγραφ\$, όνομα_αρχείου\$

}

Αναγνωριστικό: ΕΦΑΠ([TAN(
Επιστρέφει την εφαπτομένη μιας γωνίας σε μοίρες
Επιστρέφει μιγαδικό ως εφαπτομένη μιγαδικού
Τύπωσε αΕφαπ(πi/4)
Δες Τοξ.εφ()

Αναγνωριστικό: Η [OR]
Τελεστής Η

Τύπωσε Αληθές Ή Ψευδές

Αναγνωριστικό: ΗΜ([SIN(
Τύπωσε ΗΜ(45)
Επιστρέφει το ημίτονο μιας γωνίας σε μοίρες
Επιστρέφει το ημίτονο μιγαδικού αν δώσουμε μιγαδικό.

Δες ΑΗΜ() για ακτίνια

Αναγνωριστικό: ΗΜΕΡΑ([DATE(
Μετατροπή μέρας από αλφαριθμητικό σε αριθμητικό
Α=ημερα("2015-10-11")
Τυπωσε Γραφη\$(Α ,"LONG DATE")
Α=ημέρα("March 7 2009", 1033)
Τυπωσε Γραφη\$(Α ,"LONG DATE")
Τύπωσε Ημερα\$(Α, 1033, "LONG DATE")

Αναγνωριστικό: ΘΕΣΗ([INSTR(
Δέχεται ως όρισμα δυο αλφαριθμητικά ή και μια αριθμητική τιμή. Δίνει τη θέση του δεύτερου αλφαριθμητικού στο πρώτο ή μηδέν αν δεν το βρει πουθενά. Αν δώσουμε και μια τιμή ψάχνει από την θέση που ορίζει αυτή η τιμή. Αν η θέση που δίνουμε είναι μεγαλύτερη από το μήκος του πρώτου αλφαριθμητικού δίνει μηδέν.

```
ΤΥΠΩΣΕ ΘΕΣΗ("12345","2")
ΤΥΠΩΣΕ ΘΕΣΗ("ΑΒΓΔΕΑΒΓΔ","Α",2)
```

δίνει

2

6

\\ δουλεύει και με ANSI αλφαριθμητικά (ένα ψηφίο (byte) ανά χαρακτήρα)

Τοπικό 1033

Αναζήτησε\$ = Γραφή\$("aetabetAbet")

τι\$ = Γραφή\$("bet")

Τύπωσε Θέση(Αναζήτησε\$, τι\$ ως ψηφίο)=5

Τύπωσε Θέση(Αναζήτησε\$, τι\$, 6 ως ψηφίο)=9

Τύπωσε Θέση(Αναζήτησε\$, τι\$, 1 ως ψηφίο)=5

Αναζήτησε\$ = "aetabetAbet"

τι\$ ="bet"

Τύπωσε Θέση(Αναζήτησε\$, τι\$)=5

Τύπωσε Θέση(Αναζήτησε\$, τι\$, 6)=9

Τύπωσε Θέση(Αναζήτησε\$, τι\$, 1)=5

Αναγνωριστικό: ΘΕΣΗΔΕΞΙΑ([RINSTR(

A\$="12345612345612345"

N=ΘέσηΔεξιά(A\$,"34")

M=μηκος(α\$)+1

Ενώ N>0 {

 Τύπωσε N ' χαρακτήρες από αριστερά που βρίσκεται αυτό που βρέθηκε ή το 0

 Τύπωσε Μεσ\$(A\$, N) ' το Μεσ\$ δουλεύει από αριστερά

 N=ΘέσηΔεξιά(A\$,"34", M-N) 'χαρακτήρες από δεξιά

}

N=ΘέσηΔεξιά(A\$,"34")

Ενώ N>0 {

 Τύπωσε N ' χαρακτήρες από αριστερά που βρίσκεται αυτό που βρέθηκε ή το 0

 Τύπωσε Μεσ\$(A\$, N) ' το Μεσ\$ δουλεύει από αριστερά

 N=ΘέσηΔεξιά(A\$,"34", -N) 'χαρακτήρες από αριστερά (με το -)

}

\\ Χρήση με αλφαριθμητικά του ενός ψηφίου (byte) ανά χαρακτήρα (τα φτιάχνει η Γραφή\$())

Τοπικό 1033

Αναζήτησε\$ = Γραφή\$("aetabetAbet")

τι\$ = Γραφή\$("bet")

Τύπωσε ΘέσηΔεξιά(Αναζήτησε\$, τι\$ ως ψηφίο)=9

Τύπωσε ΘέσηΔεξιά(Αναζήτησε\$, τι\$, 1 ως ψηφίο)=9
Τύπωσε ΘέσηΔεξιά(Αναζήτησε\$, τι\$, 2 ως ψηφίο)=5

Αναζήτησε\$ = "aetabetAbet"

τι\$ ="bet"

Τύπωσε ΘέσηΔεξιά(Αναζήτησε\$, τι\$, 1)=9

Τύπωσε ΘέσηΔεξιά(Αναζήτησε\$, τι\$, 2)=5

Αναγνωριστικό: ΙΔΙΟΤΗΤΑ([PROPERTY()
δες Ιδιότητα\$()

Αναγνωριστικό: ΚΑΘΕ([EACH()

Επαναλήπτες

1) δίνουμε όλα τα στοιχεία του A στον επαναλήπτη N

N=ΚΑΘΕ(A)

2) δίνουμε τα στοιχεία 1,2,3 του A στον επαναλήπτη N

N=ΚΑΘΕ(A,1,3)

3) δίνουμε αόριστα, από Αρχή έως Τέλος, το ίδιο όπως αυτό ΚΑΘΕ(A,1,-1)

N=ΚΑΘΕ(A ΑΡΧΗ ΕΩΣ ΤΕΛΟΣ)

\ για τα 1,2,3

Ενω N {

Τύπωσε N^ \ δίνει το νούμερο όπως αλλάζει

}

Για βήμα μπορούμε μέσα στην επανάληψη να αλλάξουμε προγραμματισμό του επαναλήπτη, με μια νέα Κάθε.

Το A μπορεί να είναι Πίνακας, Κατάσταση, Σωρός

Κάθε επαναλήπτης έχει δικό του δρομέα και μπορούμε να βάλουμε φωλιασμένους επαναλήπτες για το ίδιο αντικείμενο

Αναγνωριστικό: ΚΑΙ [AND]

Τελεστής Και

Τύπωσε Αληθές Και Ψευδές

Αναγνωριστικό: ΚΑΤΩΜΙΣΟ([LOWWORD()

A=0xFFEE12C3

Τύπωσε Δεκαεξ\$(ΚΑΤΩΜΙΣΟ(Α),2)
Τύπωσε Δεκαεξ\$(ΠΑΝΩΜΙΣΟ(Α),2)

Δεκαεξ ΚΑΤΩΜΙΣΟ(Α), ΠΑΝΩΜΙΣΟ(Α)

Αναγνωριστικό: ΚΩΔ([ASC()
Επιστρέφει τον κωδικό ASCII του πρώτου χαρακτήρα ενός αλφαριθμητικού που δίνουμε για όρισμα.
Δες το ΧΑΡΚΩΔ()

Αναγνωριστικό: ΛΑΒΗ([JOYPAD()
Τύπωσε ΛΑΒΗ(0)
δίνει τη κατάσταση των κουμπιών πάνω στη λαβή 0 (καθε bit είναι και ένα πλήκτρο)

Αναγνωριστικό: ΛΑΒΗ.ΑΝΑΛΟΓΙΚΟ.Υ([JOYPAD.ANALOG.Υ()
ΤΥΠΩΣΕ ΛΑΒΗ.ΑΝΑΛΟΓΙΚΟ.Υ(0)
δίνει την διακριτή θέση της λαβής
Καλό είναι εδώ να υπάρχει ένα καλιμπράρισμα. Δηλαδή να ρωτάει η εφαρμογή "είναι τώρα ο μοχλός τέρμα πάνω"....και με το ok να καταγράφει την τιμή.
Χρειάζεται να ενεργοποιήσουμε τις λαβές. Π.χ. αν έχουμε 0 και 1 λαβή θα δώσουμε:
ΛΑΒΗ 0,1
και θα απενεργοποιήσουμε στο πέρας της εφαρμογής
ΛΑΒΗ

Αναγνωριστικό: ΛΑΒΗ.ΑΝΑΛΟΓΙΚΟ.Χ([JOYPAD.ANALOG.Χ()
ΤΥΠΩΣΕ ΛΑΒΗ.ΑΝΑΛΟΓΙΚΟ.Χ(0)
δίνει την διακριτή θέση της λαβής
Καλό είναι εδώ να υπάρχει ένα καλιμπράρισμα. Δηλαδή να ρωτάει η εφαρμογή "είναι τώρα ο μοχλός τέρμα δεξιά"....και με το ok να καταγράφει την τιμή.
Χρειάζεται να ενεργοποιήσουμε τις λαβές. Π.χ. αν έχουμε 0 και 1 λαβή θα δώσουμε:
ΛΑΒΗ 0,1
και θα απενεργοποιήσουμε στο πέρας της εφαρμογής
ΛΑΒΗ

Αναγνωριστικό: ΛΑΒΗ.ΚΑΤΕΥΘΥΝΣΗ([JOYPAD.DIRECTION(
ΤΥΠΩΣΕ ΛΑΒΗ.ΚΑΤΕΥΘΥΝΣΗ(0)

Σταθερή DirectionNone = 0 ' ΚΑΜΙΑ ΚΑΤΕΥΘΥΝΣΗ
Σταθερή DirectionLeft = 1 ' ΑΡΙΣΤΕΡΑ
Σταθερή DirectionRight = 2 ' ΔΕΞΙΑ
Σταθερή DirectionUp = 3 ' ΠΑΝΩ
Σταθερή DirectionDown = 4 ' ΚΑΤΩ
Σταθερή DirectionLeftUp = 5 ΑΡΙΣΤΕΡΑ ΚΑΙ ΠΑΝΩ
Σταθερή DirectionLeftDown = 6 ' ΑΡΙΣΤΕΡΑ ΚΑΙ ΚΑΤΩ
Σταθερή DirectionRightUp = 7 ' ΔΕΞΙΑ ΚΑΙ ΠΑΝΩ
Σταθερή DirectionRightDown = 8 ' ΔΕΞΙΑ ΚΑΙ ΚΑΤΩ

Πρέπει να έχουμε ενεργοποίηση την συγκεκριμένη λαβή
ΛΑΒΗ 0
και όταν δεν την χρειαζόμαστε την απενεργοποιούμε (όλες μαζί, όχι ξεχωριστά)
ΛΑΒΗ

Αναγνωριστικό: ΛΟΓ([LOG(
Επιστρέφει τον δεκαδικό λογάριθμο ενός αριθμού

Αναγνωριστικό: ΛΦ([LN(
Επιστρέφει τον φυσικό (νεπέριο) λογάριθμο ενός αριθμού

Αναγνωριστικό: ΜΕΓΑΛΟ([MAX(
Δουλεύει μόνο με μεταβλητές και στοιχεία πίνακα
Αν θέλουμε να εξάγουμε το μικρότερο από δυο ή περισσότερα και να περιλαμβάνονται και
αριθμητικές/αλφαριθμητικές παραστάσεις πρέπει να χρησιμοποιήσουμε την συνάρτηση
ΜΕΓΑΛΟ.ΣΕΙΡΑΣ(εκφρ1, εκφρ2) η οποία δεν έχει όριο στον αριθμό των στοιχείων (αλλά
πρέπει να είναι όλα ή αριθμητικές ή αλφαριθμητικές εκφράσεις)

A=10
B=3
Τύπωσε ΜΕΓΑΛΟ(A, B)

Πίνακας A(10)=10
A(3)+=10
Τύπωσε ΜΕΓΑΛΟ(A(3), A(5))

γ\$="Εντάξει"
δ\$="Γειά"
? Μεγάλο(γ\$, δ\$)="Εντάξει"
γ="Εντάξει"
δ="Γειά"
? Μεγάλο(γ, δ)="Εντάξει"

Δες ΜΙΚΡΟ(), ΜΙΚΡΟ.ΣΕΙΡΑΣ(), ΜΕΓΑΛΟ.ΣΕΙΡΑΣ()

Αναγνωριστικό: ΜΕΓΑΛΟ.ΣΕΙΡΑΣ([MAX.DATA[
Η πρώτη έκφραση δηλώνει αν θα γίνει σύγκριση αριθμών ή αλφαριθμητικών

Τύπωσε ΜΕΓΑΛΟ.ΣΕΙΡΑΣ(1, 50*3,2)=150
Τύπωσε ΜΕΓΑΛΟ.ΣΕΙΡΑΣ("a", "c","b")="c"

Δες ΜΙΚΡΟ.ΣΕΙΡΑΣ()

Αναγνωριστικό: ΜΕΓΑΛΟΣΑΚΕΡΑΙΟΣ([BIGINTEGER[
A=ΜεγάλοςΑκέραιος("6864797660130609714981900799081393217269435300143305409394
4634591855431833976560521225596406614545549772963113914808580371219879997166
43812574028291115057151")
B=ΜεγάλοςΑκέραιος("162259276829213363391578010288127987979798")
Γ=A*B
Δ=B*Γ
Γ=Γ+500u
Τύπωσε "A=";A
Τύπωσε "B=";B
Τύπωσε "Γ=";Γ
Δ=Γ/B
Τύπωσε "Δ=";Δ
Τύπωσε "Γ/B υπόλοιπο="; Υπολ(Δ)
Τύπωσε "σύγκριση με το 500:";Υπολ(Δ)=500u
Μέθοδος A, "AnyBaseInput", "FFFFFFFF", 16
Τύπωσε "A=";A
Μέθοδος A, "AnyBaseInput", "FFFFFFFFFFFFFFFF", 16

```

Τύπωσε "A=";A
Μέθοδος A, "AnyBaseInput", "FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF", 16
Τύπωσε "A=";A
Μέθοδος A, "AnyBaseInput", "16"
Τύπωσε "A=";A
Με A, "OutputBase", 2
Τύπωσε "A=";A
Μέθοδος A, "intpower", ΜεγάλοςΑκέραιος("100", 2) ως A
Τύπωσε "A=";A
A=A-ΜεγάλοςΑκέραιος("10000000000", 2)
Τύπωσε "A=";A

```

Αναγνωριστικό: ΜΕΓΕΘΟΣ.Υ([SIZE.Υ()
επιστρέφει το μέγεθος πλαισίου κατά τον Υ άξονα που μπαίνει η επιγραφή
Τύπωσε Μέγεθος.Υ(επιγραφή\$, γραμματοσειρά\$, μέγεθος_σε_pt, γωνία_σε_ακτίνια,
έξτρα_κενό_twips)
Παράδειγμα: δεξ μέγεθος.χ()

Αναγνωριστικό: ΜΕΓΕΘΟΣ.Χ([SIZE.Χ()
επιστρέφει το μέγεθος πλαισίου κατά τον χ άξονα που μπαίνει η επιγραφή
Τύπωσε Μέγεθος.Χ(επιγραφή\$, γραμματοσειρά\$, μέγεθος_σε_pt, γωνία_σε_ακτίνια,
έξτρα_κενό_twips)

```

\\ Πλάγια για γραμματοσειρά
Πλάγια 1
γραμ$="Arial"
\\ θέτει 80 χαρακτήρες επί 50 γραμμές την κονσόλα
Φόρμα 80,50
\\ έξτρα_κενό 5 σημεία (pixels)
έξτρα_κενό=πλάτος.σημείου*5
τιμή_στοίχισης=3
έγγραφο επ$={M2000 Διερμηνευτής
Δεύτερη γραμμή
Τρίτη γραμμή
Τελευταία γραμμή
}
γραμμές=εγγράφου.παρ(επ$)-1
Αν γραμμές=0 Τότε γραμμές=1
αριθμός_μπλοκ=9
τωρινή_γωνία=πi/3

```

```

μέγεθος_γραμ=17
απόσταση_από_κέντρο=300
για i=1 έως αριθμός_μπλοκ {
    τιμή_στοίχισης=τυχαίος(1, 3)
    τωρινή_γωνία+=π/αριθμός_μπλοκ*2
    εκ=(εξτρα_κενό, 0)#val(τυχαίος(0,1))
    ππ=Μέγεθος.X(επ$,γραμ$,μέγεθος_γραμ,0 ,εκ)
    υυ=Μέγεθος.Y(επ$,γραμ$,μέγεθος_γραμ, 0,εκ)
    π1=Μέγεθος.X(επ$,γραμ$,μέγεθος_γραμ, τωρινή_γωνία, εκ)
    υ1=Μέγεθος.Y(επ$,γραμ$,μέγεθος_γραμ, τωρινή_γωνία, εκ)
    θέση χ.σημεία/2, υ.σημεία/2
    βήμα γωνία τωρινή_γωνία,ππ+απόσταση_από_κέντρο
    επίλεξε με τιμή_στοίχισης
    με 2
        βήμα γωνία τωρινή_γωνία+π/2, υυ/2-υυ/γραμμές/2
    με 3
        {
            βήμα γωνία τωρινή_γωνία+π/2, υυ/2
            βήμα γωνία τωρινή_γωνία, -ππ/2
        }
    αλλιώς με
        {
            βήμα γωνία τωρινή_γωνία+π/2, υυ/2-υυ/γραμμές
            βήμα γωνία τωρινή_γωνία,ππ/2
        }
    Τέλος επιλογής
    πένα τυχαίος(11, 15) {
        Επιγραφή επ$,γραμ$,μέγεθος_γραμ,τωρινή_γωνία, τιμή_στοίχισης,0,εκ
        θέση χ.σημεία/2, υ.σημεία/2
        βήμα γωνία τωρινή_γωνία,ππ/2+απόσταση_από_κέντρο
        βήμα γωνία τωρινή_γωνία+π/2, υυ/2
        χαραξε γωνία τωρινή_γωνία, ππ
        χαραξε γωνία τωρινή_γωνία-π/2, υυ
        χαραξε γωνία τωρινή_γωνία, -ππ
        χαραξε γωνία τωρινή_γωνία-π/2, -υυ
        θέση χ.σημεία/2, υ.σημεία/2
        βήμα γωνία τωρινή_γωνία,ππ+απόσταση_από_κέντρο
        βήμα -π1/2, -υ1/2
        χρώμα {πολύγωνο 1, π1, 0, 0,υ1, -π1, 0, 0,- υ1 }
    }
}
Πλάγια 0
περίμενε$=κομ$

```

Αναγνωριστικό: ΜΕΤΑΘΕΣΗ([SEEK(

\\ η Μετάθεση() είναι πάντα σε bytes, και η πρώτη θέση στο αρχείο δίνει το 1

Άδειασε

M=Σωρός

Σειρά "alfa", "beta", "gama", "epsilon", "delta"

Άνοιξε "a.dat" Για Ευρεία Εξαγωγή Ως #κ

Ενώ Όχι Κενό {

 Τύπωσε #κ, Γράμμα\$;

 Σωρός M { Σειρά Μετάθεση(#κ)} ' Σειρά comand push to end

}

Κλείσε #κ

α=1

β=Κάθε(M)

Άνοιξε "a.dat" Για Ευρεία Εισαγωγή Ως #κ

Ενώ β {

 Μετάθεση #κ, α

 Τύπωσε Εισαγωγή\$(#κ, (ΤιμήΣωρού(β)-α)/2) "χρήση του /2 για Ευρεία (UTF16LE)

 α=ΤιμήΣωρού(β)

}

Κλείσε #κ

Αναγνωριστικό: ΜΗΚΟΣ([LEN(

Επιστρέφει το μήκος ενός αλφαριθμητικού σε χαρακτήρες.

Σε πίνακες επιστρέφει τον αριθμό στοιχείων ανεξάρτητα από διαστάσεις

Σε καταστάσεις και σωρούς (αντικείμενα) επιστρέφει τον αριθμό στοιχείων

Αναγνωριστικό: ΜΗΚΟΣ.ΕΜΦ([LEN.DISPLAY(

\\Δίνει το μήκος εμφάνισης

Τύπωσε ΜΗΚΟΣ.ΕΜΦ("ăž") ' 2

Τύπωσε Μήκος("ăž") ' 4

Αναγνωριστικό: ΜΙΚΡΟ([MIN(

? ΜΙΚΡΟ(A,B), ΜΙΚΡΟ(A(3),B)

? ΜΙΚΡΟ(A%,B%(2))

Δέχεται μόνο μεταβλητές ή στοιχεία πίνακα και ακριβώς δύο (δέχεται πια και αλφαριθμητικά).

Αν θέλουμε να εξάγουμε το μικρότερο από δυο ή περισσότερα και να περιλαμβάνονται και αριθμητικές/αλφαριθμητικές παραστάσεις πρέπει να χρησιμοποιήσουμε την συνάρτηση

ΜΙΚΡΟ.ΣΕΙΡΑΣ(εκφρ1, εκφρ2) η οποία δεν έχει όριο στον αριθμό των στοιχείων (αλλά πρέπει να είναι όλα ή αριθμητικές ή αλφαριθμητικές εκφράσεις)

Παρακάτω είναι ένα πρόγραμμα σύγκρισης ταχύτητας.

Η μικρο() τρέχει στα δυο τρίτα τριχ χρόνου περίπου.

* δημιουργούμε τις μεταβλητές πριν εκκινήσουμε τον αναλυτή.

I=1 : A=10 : B=4 : Δ=0

ΑΝΑΛΥΤΗΣ

ΓΙΑ I=1 ΕΩΣ 1000 {

 Δ=ΜΙΚΡΟ.ΣΕΙΡΑΣ(A,B)

}

ΤΥΠΩΣΕ ΦΟΡΤΟΣ

ΑΝΑΛΥΤΗΣ

ΓΙΑ I=1 ΕΩΣ 1000 {

 Δ=ΜΙΚΡΟ(A,B)

}

ΤΥΠΩΣΕ ΦΟΡΤΟΣ

ΚΑΘΑΡΟ ' σβήνει τις μεταβλητές!

ΓΡΑΜΜΑ α="οκ", β="οχι", δ

ΑΝΑΛΥΤΗΣ

ΓΙΑ I=1 ΕΩΣ 1000 {

 δ=ΜΙΚΡΟ.ΣΕΙΡΑΣ(α, β)

}

ΤΥΠΩΣΕ ΦΟΡΤΟΣ

? δ

ΑΝΑΛΥΤΗΣ

ΓΙΑ I=1 ΕΩΣ 1000 {

 δ=ΜΙΚΡΟ(α, β)

}

ΤΥΠΩΣΕ ΦΟΡΤΟΣ

? δ

Αναγνωριστικό: ΜΙΚΡΟ.ΣΕΙΡΑΣ([MIN.DATA()

Το πρώτο στοιχείο δηλώνει αν θα γίνει σύγκριση με αριθμούς ή με αλφαριθμητικά

Τύπωσε ΜΙΚΡΟ.ΣΕΙΡΑΣ(1, 50*3,2)=1

Τύπωσε ΜΙΚΡΟ.ΣΕΙΡΑΣ("a", "c","b")="a"

Δες ΜΕΓΑΛΟ.ΣΕΙΡΑΣ()

Αναγνωριστικό: ΜΠΡΟΣΤΑ([FORWARD()

Εγγραφο α\$={αααααααα

βββββββββ

γγγγγγγγγγ}

μ=Παράγραφος(α\$, 0)

Αν Μπροστά(α\$,μ) Τότε {

Ενώ μ {

Τύπωσε Παράγραφος\$(α\$, (μ))

}

}

Αναγνωριστικό: ΟΜΑΔΑ([GROUP()

Ομάδα Άλφα {

Ιδιωτικό:

χ=0

Δημόσιο:

Αξία {

=.χ

.χ++

}

}

Τύπωσε Άλφα, Άλφα, Άλφα

Βήτα=Άλφα

Τύπωσε Βήτα , Βήτα, Βήτα ' 3 3 3 είναι τύπου double

Τύπωσε Τύπος\$(Βήτα)

Δέλτα=Ομάδα(Άλφα)

Τύπωσε Δέλτα, Δέλτα, Δέλτα ' 4, 5, 6

Τύπωσε Άλφα, Άλφα, Άλφα' 4, 5, 6

Αναγνωριστικό: ΟΜΑΔΑ.ΣΥΝΟΛΟ([GROUP.COUNT()
Δίνει τον αριθμό μελών μιας ομάδας
Δείτε το παρακάτω παράδειγμα (προχωρημένο κάπως)

Ομάδα Αλφα {

Ιδιωτικό:

χ=100

Δημόσιο:

α=10

β=20

Πίνακας Κ(20)=50

Τμήμα βήτα {

Τύπωσε .α, .β, .α*.β, .χ, .Κ()

}

}

Αλφα.Βήτα

Τύπωσε Ομάδα.Σύνολο(Αλφα) \\ 3 only variables and arrays

Άδειασε 'αδειάζουμε το Σωρό, για να τον γεμίσουμε με την Βάλε

Για ι=1 Έως Ομάδα.Σύνολο(Αλφα)

Τύπωσε "Ισχνή Αναφορά: "; Μέλος\$(Αλφα, ι), Μέλους.Τύπος\$(Αλφα, ι)

Βάλε Φίλτρο\$(Μέλος\$(Αλφα, ι),"(") ' αφαιρούμε το (- θα το δώσει ο πίνακας

Επόμενο ι

Σωρός ' Μας δείχνει τι έχει τώρα ο σωρός

' τώρα διαβάζουμε από το σωρό

Διάβασε &κ1(), &β1, &α1

Δες οκ {

Διάβασε &χ1 ' δεν θα πετύχει η αναφορά σε ιδιωτική μεταβλητή!

}

Αν όχι οκ Τότε Τύπωσε Λάθος\$

Τύπωσε κ1(3), β1, α1

β1+=100

Τύπωσε Αλφα.β

\\ ειδική εντολή διάβασε με το Από, διαβάζει μεταβλητές και πίνακες από ομάδα

\\ εδώ μπορούμε να συνδέσουμε την ιδιωτική με μεταβλητή.

Διάβασε Από Αλφα, χ2, α2, β2, μ()

Τύπωσε χ2, α2, β2, μ()

α2+=1000

χ2*=500

Αλφα.Βήτα

Αναγνωριστικό: ΟΡΙΣΜΑ([ARG()
Επιστρέφει το όρισμα ενός μιγαδικού
(ίδια τιμή από την συνάρτηση ΦΑΣΗ())

Αναγνωριστικό: ΟΡΟΦ([CEIL()

στρογγυλοποιεί στην επόμενη ακέραια τιμή

Αναγνωριστικό: ΟΧΙ [NOT]
Συνάρτηση αναστροφής
ΤΥΠΩΣΕ ΟΧΙ ΑΛΗΘΕΣ

ΤΥΠΩΣΕ ΟΧΙ (ΟΧΙ 2)
δίνει 2
Υπάρχει και η ΔΕΝ (ακριβώς ότι κάνει και η ΟΧΙ)
Αυτήν την έφτιαξα για να την βάζουμε έτσι:
ΑΝ ΔΕΝ ΥΠΑΡΧΕΙ(αρχείο_ταδε\$) ΤΟΤΕ {
}
το οποίο είναι πιο ωραίο από το
ΑΝ ΟΧΙ ΥΠΑΡΧΕΙ(....

Αναγνωριστικό: ΠΑΙΚΤΗΣ([PLAYER()
ΤΥΠΩΣΕ ΠΑΙΚΤΗΣ(12)
δίνει αληθής αν ο παίκτης με προτεραιότητα 12 είναι ορατός (μπορεί να μην είναι αν είναι εκτός της οθόνης (του μέρους που λέμε περιθώριο), αλλά μια κίνηση στην οθόνη θα τον δείξει).
Η Συνάρτηση Σύγκρουση() δεν κοιτάει αν ο παίκτης προς έλεγχο είναι ή όχι ορατός. Μπορούμε να φιλτράρουμε την έξοδο αυτής της συνάρτησης με την χρήση της Παίκτης()

Αναγνωριστικό: ΠΑΝΩΜΙΣΟ([HIGHWORD()
' η α δεν είναι ακέραιος, αλλά μπορεί να έχει ακέραιες τιμές.
α=0xFFFF0000
Τύπωσε ΠάνωΜισό(α) ' 65535

ΚάτωΜισό(), ΔυοΜισά(), Δεκαεξ\$()
Δεκαεξ

```
Αναγνωριστικό: ΠΑΡΑΓΡΑΦΟΣ( [PARAGRAPH()]  
Καθαρό  
Έγγραφο α$  
Για ι=1 έως 9 {  
    α$="aaaaaaaaaaaaaaaaaaaaaa"+γραφη$(ι)+{  
}  
}  
α$="aaaaaaaaaaaaaaaaaaaaaa"+γραφη$(ι)  
Για ι=10 έως 1 ανα 2 {  
    \\ διαγράφη  
    δ$=Παράγραφος$(α$, ι, -1)  
}
```

```

M=Παράγραφος(α$, 1-1)
Δ=Μπροστά(α$, Μ)
Ενώ Μ {
    Τύπωσε Αριθμός.Παραγράφου(α$,Μ), Μ
    Αναφορά Παράγραφος$(α$, (Μ))
}

```

Αναγνωριστικό: ΠΑΡΑΜ([ΠΑΡΑΜ(]

1) Ένθεση ορισμάτων στο κώδικα

\\ υπολογίζουμε τα ορίσματα που θα δώσουμε στην Τύπωσε και στην ΔεξΕδω

A=10

B=40

\\ και τις καταχωρούμε σε ένα αλφαριθμητικό

Σ\$=Παράθεση\$(A*B, A/B)

\\ η Παραμ() με αλφαριθμητικό βάζει τα αποτελέσματα στο κώδικα

Τύπωσε Παραμ(Σ\$)

Τμήμα ΔεξΕδω (X,Y) { Τύπωσε X, Y }

ΔεξΕδω Παραμ(Σ\$)

ΔεξΕδω A*B, A/B

2) Υπάρχει τρόπος να δώσουμε διαβάσουμε τις ιδιότητες/μεθόδους ενός αντικειμένου που

αναγνωρίζουν τα Windows.

\\ Για να δουλέψει το παρακάτω πρέπει να έχουμε το Excel στον υπολογιστή μας!

\\ Για να πάρουμε το GUID εκτελούμε στη γραμμή εντολών: ΛΙΣΤΑ COM ΣΤΟ B

\\ Θα μας βγάλει μια λίστα με αντικείμενα που αναγνωρίζει ο υπολογιστής

\\ Θα επιλέξουμε ένα και σε ένα τμήμα (εδώ το B) θα δώσει όλες τις γραμμές

\\ όπως η ακόλουθη με τα αντικείμενα και τους αριθμούς GUID

\\ οπότε με Σ B ανοίγουμε το τμήμα B και παίρνουμε ένα αντικείμενο

\\ Όλα τα αντικείμενα δεν μπορούν να οριστούν παρόλο που βλέπουμε γεαμμές με Όρισε.

Όρισε Worksheet "{00020820-0000-0000-C000-000000000046}"

\\ Η εντολή Παραμ(αντικείμενο) εξάγει μια κατάσταση (λίστα)

\\ Τα κλειδιά είναι τα ονόματα των ιδιοτήτων/συναρτήσεων

\\ και τα στοιχεία είναι τα ονόματα με τα ορίσματά τους

\\ Τα κλειδιά είναι σε κεφαλαία γράμματα

Άλφα=Παραμ(Worksheet)

Αναφορά Τύπος\$(Worksheet) ' Workbook

Αν μήκος(Άλφα)>1 Τότε {

\\ χρησιμοποιούμε την εντολή Αναφορά γιατί αφενός εμφανίζει κείμενο με παραγράφους, και αναδίπλωση λέξεων και αφετέρου σταματάει κάθε φορά που κάνει ολίσθηση στα 3/4 του ύψους της κονσόλας και περιμένει διάστημα ή κλικ με το ποντίκι.

\\ τα 8 πρώτα είναι του διπλού Interface των com αντικειμένων, τα αφήνουμε!

\\ M=Κάθε(Άλφα)

M=Κάθε(Άλφα, 9)

\\ το M^ είναι με βάση το 1 ο δρομέας του επαναλήπτη M

\\ εμείς θέλουμε με βάση το 0 για να δούμε τα κλειδιά στην πρώτη Εκφρ\$()

\\ η δεύτερη έχει ήδη ρυθμιστεί διάβασε παρακάτω

Ενώ M { Αναφορά 3, Έκφρ\$(Άλφα, M^1)+":"+'Έκφρ\$(Άλφα)

\\ Δεν χρειάζεται αυτό: Αναφορά 3, Άλφα\$(M^1!)

\\ γιατί η Άλφα έχει μια ιδιότητα index και αυτή ρυθμίζεται είτε από το M, είτε από τη προηγούμενη χρήση

\\ οπότε η Έκφρ\$(Άλφα) χρησιμοποιεί την ιδιότητα index

}

}

\\ και το Άλφα είναι αντικείμενο, αλλά δεν μπορούμε να το διαβάσουμε με την Παραμ()

\\ Με την Παραμ() διαβάζουμε ότι είναι "δημόσιο" στο Registry των Windows, και μπορούμε να το δημιουργήσουμε

\\ με την Όρισε.

Με Άλφα, "index" ως index

Αν Υπάρχει(Άλφα, "APPLICATION") Τότε {

\\ επειδή το Index έχει βάση το 0, βάζουμε +1 για να πάρουμε το νούμερο του Application στο WorkSheet

Τύπωσε Index+1, Εκφρ\$(Άλφα, Index), Εκφρ\$(Άλφα)

}

Αναγνωριστικό: ΠΑΤΗΜΕΝΟ([KEYPRESS(

Η συνάρτηση Πατημένο(αριθμός) παίρνει τον αριθμό πλήκτρου (σε πληκτρολόγιο και ποντίκι) και ελέγχει αν είναι πατημένο, μόνο όταν το παράθυρο του περιβάλλοντος της M2000 είναι στο προσκήνιο (δηλαδή όταν δουλεύουμε σε αυτό).

A=1

Επανάλαβε {

 Τύπωσε A

 A++

} Μέχρι Πατημένο(1) \ 1 είναι το αριστερό πλήκτρο του ποντικιού

Δεν υπάρχει εδώ ενδιαμέση μνήμη να κρατάει τα πατήματα στη σειρά που πατήθηκαν, αλλά άμεσος έλεγχος.

Εδώ υπάρχουν οι αριθμοί των πλήκτρων (λέγονται virtual keys)

[https://msdn.microsoft.com/en-us/library/windows/desktop/dd375731\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd375731(v=vs.85).aspx)

Αναγνωριστικό: ΠΙΝΑΚΑΣ([ARRAY(

Τυπωσε Πινακας("α",1,3)

είναι ισοδύναμο με

Τυπωσε α(1,3)

Υπάρχει αντίστοιχη ΠΙΝΑΚΑΣ\$("α\$",1,2)

για αλφαριθμητικά

πινακας α(10)

συναρτηση α {

 διαβασε α

 =α*2

}

α(5)=3

τυπωσε α(5), πινακας("α",5)

\Θα τυπώσει 3 3 γιατί οι πίνακες προηγούνται.

Τύπωσε α(* 5) \ \ θα τυπώσει 10 γιατί το * λέει ότι είναι συνάρτηση

Χρησιμοποιείται και με δείκτη σε πίνακα

A=(1,2,3,"Κείμενο",5)

Τύπωσε Πίνακας(A, 0) ' θα δώσει το 1
 Επιστροφή A, 0:=1000, 4:="Κείμενο B"
 Τύπωσε A, Πίνακας(A,0), Πίνακας\$(A,4)
 Προσθήκη A, (6,7,8,9)
 Τύπωσε Πίνακας(A, 8)
 Τύπωσε Μήκος(A)
 Ένωσε A στο A\$()
 Τύπωσε A\$(4)
 Τύπωσε Πίνακας(A\$(), 1)

Αναγνωριστικό: ΠΙΣΩ([BACKWARD(]
 Εγγραφο α\$={αααααααα
 ββββββββββ
 γγγγγγγγγγ
 }
 μ=Παράγραφος(α\$, Εγγράφου.παρ(α\$))
 Αν Πίσω(α\$,μ) Τότε {
 Ενώ μ {
 Τύπωσε Παράγραφος\$(α\$, (μ))
 }
 }

Αναγνωριστικό: ΠΟΛΙΚΟΣ([POLAR(]
 Δημιουργεί μιγαδικό
 α=(8, -3i)
 ?Πολικός(Υπολ(α), Όρισμα(α))=α
 ?Πολικός(Απολ(α), Φάση(α))=α

Αναγνωριστικό: ΠΡΩΤΟ([CAR(]
 α=((1,2), (3,4), (5, 6))
 b=Πρώτο(α)
 Τύπωσε Πρώτο(b) ' 1
 Τύπωσε Πρώτο(Πρώτο(α)) ' 1
 Τύπωσε Επόμενα(Πρώτο(Επόμενα(Επόμενα(α)))) ' 6

Αναγνωριστικό: ΡΙΖΑ([SQRT(
Επιστρέφει την τετραγωνική ρίζα ενός θετικού αριθμού.

Αναγνωριστικό: ΡΩΤΑ([ASK(
ΤΥΠΩΣΕ ΡΩΤΑ("ΟΛΑ ΚΑΛΑ")

Ανοίγει ένα παράθυρο μηνυμάτων και περιμένει ένα OK ή ένα Άκυρο.

Στο OK δίνει 1

Υπάρχει και ως ΡΩΤΑ\$()

Επιπλέον δέχεται προαιρετικούς παραμέτρους με τη σειρά:

Τίτλος

Λεζάντα για OK (με * στην αρχή είναι το επιλεγμένο, με σκέτο * είναι η προεπιλεγμένη λεζάντα και επιλεγμένο)

Λεζάντα για ΑΚΥΡΟ (με * στην αρχή είναι το επιλεγμένο, με σκέτο * είναι η προεπιλεγμένη λεζάντα και επιλεγμένο, με κενή λεζάντα , δηλαδή με "" δεν εμφανίζεται)

Αλφαριθμητικό για εικόνα, ή μονοπάτι όπου βρίσκεται στο δίσκο μια εικόνα

Αλφαριθμητικό που μετατρέπει το παράθυρο μνημάτων σε παραθυρο εισαγωγής (δέχεται μέχρι 100 χαρακτήρες)

Σε περίπτωση που έχουμε εισαγωγή και δώσουμε OK τότε στον σωρό θα υπάρχει το αλφαριθμητικό

Αναγνωριστικό: ΣΕΙΡΙΑΚΟΣ.ΔΙΣΚΟΥ([DRIVE.SERIAL(
Γυρίζει το αριθμό του δίσκου (ως ακέραια τιμή, μπορεί να είναι και αρνητικός)

' Μπορούμε να τον μετατρέψουμε σε ισοδύναμο ακέραιο χωρίς πρόσημο (με ίδια δυαδικά ψηφία)

' Δείτε το παράδειγμα (το πρώτο είναι σε μια γραμμή αλλά φαίνεται σε δυο)

Τύπωσε Δεκαεξ\$(ΠάνωΜισό(Δυαδικό.Ακέραιο(Σειριακός.Δίσκου("c:\"))),2) + "-" +
Δεκαεξ\$(ΚάτωΜισό(Δυαδικό.Ακέραιο(Σειριακός.Δίσκου("c:\"))),2)

Δεκαεξ Δυαδικό.Ακέραιο(Σειριακός.Δίσκου("c:\"))

Αναγνωριστικό: ΣΗΜ([SGN(
Τύπωσε Σημ(10),Σημ(0),Σημ(-5)

Δίνει: 1 0 -1

A=-5

Τύπωσε Σημ(A)

Αναγνωριστικό: ΣΗΜΕΙΟ([POINT(]

ΤΥΠΩΣΕ ΣΗΜΕΙΟ(A\$,10,10)

Τυπώνει το χρώμα της εικόνας που υπάρχει στο A\$ (αν δεν υπάρχει τότε βγαίνει λάθος) στο σημείο 10,10 (σε σημεία - pixels)

Παλιό=ΣΗΜΕΙΟ(A\$,10,10, Χρώμα(255, 0, 128))

Μας δίνει το παλιό χρώμα και βάζουμε νέο στο σημείο που θέλουμε (σε pixels, από το 0 πάνω αριστερά)

Για περισσότερους χειρισμούς πρέπει να το βάλουμε σε ένα επίπεδο, να εφαρμόσουμε εντολές σχεδίων και εικόνων, και μετά να το ξανασώσουμε στο αλφαριθμητικό.

Αναγνωριστικό: ΣΤΗΛΗ([TAB(]

Τύπωσε Στήλη(3)

Τυπώνει την θέση της τέταρτης στήλης

Τύπωσε @(Στήλη(2)),100, @(Στήλη(1)),400
400 100

Τύπωσε Στήλη*3=Στήλη(3)

-1

Τύπωσε \$(,10),Στήλη=10

Δες Στήλη, \$(

Αναγνωριστικό: ΣΤΡΟΓΓ([ROUND(]

1. Τύπωσε Στρογγ(12.123231, 3)

κάνει στρογγυλοποίηση στο 3 δεκαδικό.

2. Τύπωσε Στρογγ(12.123231764234432)

κάνει στρογγυλοποίηση στο 13 δεκαδικό (εξ ορισμού).

Αναγνωριστικό: ΣΥΓΚΡΙΝΕ([COMPARE(]

// ο τελεστής <=> δίνει -1, 0, 1 όπως και η ΣΥΓΚΡΙΝΕ() αλλά δουλεύει και με παραστάσεις (εκφράσεις)

```

// Η ΣΥΓΚΡΙΝΕ δουλεύει με πίνακες (με παρενθέσεις) και μεταβλητές
α%=5
β%=3
? ΣΥΓΚΡΙΝΕ(α%,β%)
    1
α=5
β=13
? ΣΥΓΚΡΙΝΕ(α,β)
    -1
// Ο B$( ) πίνακας παίρνει και αριθμητικές τιμές και αντικείμενα επειδή έχει Variant τύπο
ΠΙΝΑΚΑΣ β$(10)="αα" ' στοιχεία από 0 μέχρι 9 και εδώ βάζουμε σε όλα το "αα"
Ένωσε β$( ) στο γ( )
γ(3)=12345
? β$(3)="12345" ' το B$( ) γυρνάει σε αλφαριθμητικό τον αριθμό!
α$="αα"
? ΣΥΓΚΡΙΝΕ(α$,β$(3)), α$<=>β$(3)
? ΣΥΓΚΡΙΝΕ(α$,β$(4))
    0
α="ααα"
? ΣΥΓΚΡΙΝΕ(α$,α)
ΠΙΝΑΚΑΣ δ(100) ΩΣ ΓΡΑΜΜΑ = "BB" ' ΠΑΙΡΝΕΙ ΜΟΝΟ ΑΛΦΑΡΙΘΜΗΤΙΚΑ
? ΣΥΓΚΡΙΝΕ(α$,δ(3))
? ΣΥΓΚΡΙΝΕ(δ(3),α)

```

Συγκρίνει μόνο δυο μεταβλητές αριθμητικές ή αλφαριθμητικές. Μπορούμε να δώσουμε και στοιχεία πίνακα για σύγκριση.

Είναι πιο γρήγορη αυτή η σύγκριση γιατί οι τιμές περνάνε με αναφορά και όχι με αντιγραφή στην εσωτερική ρουτίνα.

Επιστρέφει 0 για ίσο, 1 για το πρώτο (αριστερά) μεγαλύτερο και -1 για το δεύτερο (δεξιά) μεγαλύτερο.

Μπορούμε να την χρησιμοποιήσουμε σε μια ΕΠΙΛΕΞΕ ΜΕ αν θέλουμε διαφορετική απόκριση σε κάθε μια από τις τρεις δυνατές τιμές 1 0 και -1.

Επίσης μόνο για αριθμητικές μεταβλητές υπάρχουν και οι γρήγορες ΜΙΚΡΟ() και ΜΕΓΑΛΟ()

Αν θέλουμε να συγκρίνουμε παραστάσεις υπάρχει ο spaceship operator <=>

Τύπωσε 10+2<=>10+2, "ααα"<=>"ββ"

Αναγνωριστικό: ΣΥΓΚΡΟΥΣΗ([COLLIDE()

Έλεγχος σύγκρουσης παικτών.

Τυπωσε Συγκρουση(2,100)

Ο έλεγχος γίνεται από το 1, αν είχαμε Συγκρουση(32,100) ο έλεγχος γίνεται από το 31 μέχρι το 1 σε σχέση με το 32. Συνεπώς για το 1 δεν έχουμε Σύγκρουση(1, 100). Αν ένας παίκτης είναι κρυμμένος (δεν φαίνεται) θα υπολογιστή για την σύγκρουση. Οπότε αν δεν θέλουμε να υπολογιστεί τον μετακινούμε έξω από το χώρο που πιθανόν κινείται. Μπορούμε να ξέρουμε αν ένας παίκτης δεν φαίνεται, γιατί εμείς τον κάνουμε να μην φαίνεται, άρα κρατάμε λογαριασμό

Τυπωσε Συγκρουση(2,100,10000,1800,12000,3800)

μας λέει αν ο παίκτης 2 συγκρούστηκε με το 1, το 100 σημαίνει ότι θα χρησιμοποιηθεί όλο (100%) το εμβαδόν του παίκτη (στο νοητό τετράπλευρο που σχηματίζει η εικόνα του). Μειώνοντάς του κάνουμε τις πλευρές του παίκτη αναίσθητες.

Τυπωσε Συγκρουση(2,100,10000,1800,12000,3800)

εδώ βλέπουμε αν ο παίκτης 2 καλύπτει το τετράπλευρο με κορυφή πάνω αριστερά 10000,1800 και κάτω δεξιά 12000,3800 (δηλαδή εδώ είναι απόλυτες οι συντεταγμένες, και όχι σχετικές με την πρώτη κορυφή)

Υπάρχουν 32 παίκτες όπου ο 1 δεν συγκρούεται με κανέναν ενώ ο 32 με όλους τους άλλους. Ο 32 είναι ο πιο κοντινός μας...δεν θα τον καλύψει κανείς. Κάθε παίκτης έχει μέγεθος και εικόνα. Μπορούμε να αλλάζουμε την εικόνα του και ταυτόχρονα και το μέγεθός του, και αυτό είναι το επίπεδό του. Από κάθε παίκτη μπορούμε να πάρουμε το επίπεδό του και να το χειριστούμε ως οθόνη.

Γιατί υπάρχει παίκτης και επίπεδο που αντιστοιχούν στο ίδιο πράγμα; Η απάντηση είναι απλή. Ο παίκτης χειρίζεται το επίπεδο με διαφορετικό τρόπο, πιο άμεσο, μετακίνηση, εμφάνιση και εξαφάνιση (αλλά παραμένει στην θέση του), αλλαγή εικόνας και αν έχει ή όχι κάποια διάφανα σημεία καθώς και αλλαγή προτεραιότητας. Αντίθετα το επίπεδο καλεί μια ομάδα εντολών και προετοιμάζει το επίπεδο για γραφικά και τύπωση γραμμάτων. Η ομάδα εντολών έχει εντολές που θα έπαιζαν και στην οθόνη. Μπορούμε να μετακινήσουμε το επίπεδο με την ίδια εντολή που μετακινούμε την οθόνη. Επιπλέον με την Τυπος μπορούμε να αλλάξουμε τύπο γραμμάτων (μέγεθος) καθώς και το μέγεθος του επιπέδου για να χωρέσουμε ας πούμε μια εικόνα! Επιπλέον μπορούμε να ορίσουμε στόχους πάνω στα επίπεδα όπως ορίζουμε και στην οθόνη.

Υπάρχει τρόπος να χειριστούμε εικόνες αντικείμενα βάζοντάς τα το ένα δίπλα στο άλλο ώστε να φτιάξουμε ένα τοπίο που θα βρίσκεται στο παρασκήνιο (δες την ΔΙΑΦΑΝΟ) και να χρησιμοποιήσουμε τους παίκτες για κινούμενες εικόνες και μερικά επίπεδα για να γράφουμε σκορ!

Αναγνωριστικό: ΣΥΖΥΓΗΣ([CONJUGATE(]

επιστρέφει τον συζυγή μιγαδικού

Αναγνωριστικό: ΣΥΝ([COS()
Επιστρέφει το συνημίτονο μιας γωνίας σε μοίρες
Αν περάσουμε Μιγαδικό δίνει το συνημίτονο μιγαδικού.
Για ακτίνια χρησιμοποίησε το αΣυν()

Αναγνωριστικό: ΣΥΝΑΡΤΗΣΗ([FUNCTION()
ΤΥΠΩΣΕ ΣΥΝΑΡΤΗΣΗ("ΟΝΟΜΑΣΥΝΑΡΤΗΣΗΣ", 1,4,5)
ΤΥΠΩΣΕ ΟΝΟΜΑΣΥΝΑΡΤΗΣΗΣ(1,4,5)
Συνάρτηση Άλφα (χ){
 Αν $x > 100$ Τότε {
 $=x * 500$
 } Αλλιώς $=x$
}
α\$=&Άλφα()
Τύπωσε Συνάρτηση(α\$,200), Συνάρτηση(α\$,50)

Αναγνωριστικό: ΣΥΧΝΟΤΗΤΑ([FREQUENCY()
Επιστρέφει την συχνότητα για την οκτάβα και τον αριθμό νότας

Αναγνωριστικό: ΣΩΡΟΣ([STACK()
α=Σωρός:=1,2,"αλφα",4,5
β=Σωρός:=6,7,8,9
β=Σωρός(α,β,α)
Τύπωσε α
Τύπωσε β

Αναγνωριστικό: ΤΑΞΗ([ORDER()
Τύπωσε Τάξη("ΑΛΦΑ1232","ΑΛΦΑ800")

κάνει σύγκριση με υπολογισμό των αριθμών μέσα στο αλφαριθμητικό. Εδώ θα δώσει 1, δηλαδή

το πρώτο είναι μεγαλύτερο.

δίνει -1 όταν είναι το δεξιό μεγαλύτερο, και μηδέν αν είναι ίσα. Μπορούμε να έχουμε αριθμούς ανάμεσα από γράμματα, και όσο βρίσκει ίδια γράμματα κάνει ταξινόμηση βάση αριθμού. Η συγκριμένη συνάρτηση χρησιμοποιείται στην Κατάσταση για ταξινόμηση ως αριθμός.

Αναγνωριστικό: ΤΑΥΤΙΣΗ([MATCH(

> Βάλε "αλφα",1,2

> Τύπωσε Ταύτιση("αα")

-1

> Τύπωσε Ταύτιση("ααγ")

-1

> Τύπωσε Φάκελος()

NNS εδώ εξάγει πάντα κεφαλαία και στα αγγλικά, για όλα όμως τα στοιχεία του σωρού

Η συνάρτηση Ταυτιση() κοιτάει από το παρεχόμενο αλφαριθμητικό αν αντιστοιχεί ο τύπος στο σωρό στην αντίστοιχη θέση ξεκινώντας από την κορυφή.

Το αλφαριθμητικό παίρνει τα παρακάτω(μικρά ή μεγάλα):

[Α]ριθμός, [Ο]μάδα, [Ε]γινός, [Δ]ιάρθρωση, [Κ]ατάσταση, [Π]ίνακας,[Λ]άμδα, [Γ]ράμματα, [Σ]ωρός

το Γ είναι για τα αλφαριθμητικά.

Άδειασε \ αδειάζει το σωρό

Πίνακας Α(10)

Κατάσταση Αλφα=1,2,3

Διάρθρωση Κενή Βήτα ως Μακρύς *10

Ομάδα Δέλτα {Χ=10}

Λ=Λάμδα->100

Συνάρτηση Ζήτα {=100}

Γεγονός Κ {Διάβασε Χ}

Α=Σωρός:=1,2,"οκ",3

Βάλε "οκ", Λ ,Α(), Αλφα, Βήτα,Κ, Δελτα, Ζήτα(), Σωρός(Α)

Τύπωσε Μέγεθος.Σωρού

Σωρός

Τύπωσε Φάκελος\$(), Ταύτιση("CNGEBIAFS"), Ταύτιση("ΣΑΟΕΔΚΠΛΓ")

δες Match(για τα αγγλικά γράμματα

Αναγνωριστικό: ΤΕΛΕΣΤΕΣ [OPERATORS]

Για αντικείμενα μόνο

ΕΙΝΑΙ

δείκτης αντικειμένου ΕΙΝΑΙ δείκτης αντικειμένου, δείκτης αντικειμένου ΕΙΝΑΙ ΤΙΠΟΤΑ

Για ομάδες

δείκτης_ομάδας είναι δείκτης_ομάδας, δείκτης_ομάδας είναι ομάδα
(δεν ισχύει το ομάδα είναι ομάδα)

Αριθμητικοί Τελεστές

το -3^2 δίνει το 9, το $0-3^2$ δίνει το -9

+ - * /

^ ** ύψωση σε δύναμη

Δια

Δια# (Ευκλείδεια διαίρεση)

Υπόλοιπο Υπόλ

Υπόλοιπο# Υπόλ# (Εκλείδειο υπόλοιπο)

< > <= >= = ==

<=> (τελεστής διαστημόπλοιο, δίνει μια από τρεις τιμές 1, 0, -1)

Η ΚΑΙ ΑΠΟ ΟΧΙ (OR AND XOR NOT)

(,) κενός πίνακας

(1,) πίνακας ενός στοιχείου

(1,2) πίνακας δυο στοιχείων

Τελεστές για μεταβλητές (πχ A++ ως μια εντολή)

Δεν ισχύουν για ιδιότητες ομάδων (όχι μεταβλητές ομάδων) πρέπει να ορίσουμε τους τελεστές

+=

-=

*=

/=

~! αλλαγή προσήμου

~ \ \ ψευδές <-> αληθές

++

--

Τελεστές για Αλφαριθμητικά

+

< > <= >= =

~ (όπως η Like στη VB6)

<=> (τελεστής διαστημόπλοιο, δίνει μια από τρεις τιμές 1, 0, -1)

Αναγνωριστικό: ΤΕΛΟΣ([EOF(]

Δέχεται ως όρισμα τον αριθμό ενός ανοιχτού αρχείου και δίνει ΑΛΗΘΕΣ αν η προηγούμενη ενέργεια διαβάσματος του αρχείου έφθασε στο τέλος αυτού.

Αναγνωριστικό: TIMH([VAL(
συνώνυμο ΑΞΙΑ(

1) Με αλφαριθμητική παράσταση

ΤΥΠΩΣΕ TIMH("123")

ΤΥΠΩΣΕ TIMH("1 2 3")=123

ΤΥΠΩΣΕ TIMH("123.233")

ΤΥΠΩΣΕ TIMH("123,233", ",")

ΤΥΠΩΣΕ TIMH("123-233", "-")

ΤΥΠΩΣΕ TIMH("123,233", 0)

ΤΥΠΩΣΕ TIMH("123,233", 1032)

ΤΥΠΩΣΕ TIMH("123.233", 1033)

2) Με αριθμητική παράσταση

\\ Χρησιμοποίησέ το για αλλαγές τύπου

Κάνε ΤύποςΠαράστασης\$(N)=Τύπος\$(N)

Τύπωσε ΤύποςΠαράστασης\$(10) ' Double

Τύπωσε ΤύποςΠαράστασης\$(Τιμή(10->0@)) ' Decimal

Τύπωσε ΤύποςΠαράστασης\$(Τιμή(10->0#)) ' Currency

Τύπωσε ΤύποςΠαράστασης\$(Τιμή(10->0&)) ' Long

Τύπωσε ΤύποςΠαράστασης\$(Τιμή(10@)) ' Double

Τύπωσε ΤύποςΠαράστασης\$(Τιμή(10#)) ' Double

Τύπωσε ΤύποςΠαράστασης\$(Τιμή(10&)) ' Double

Τύπωσε ΤύποςΠαράστασης\$(Τιμή(10#->0@)) ' Decimal

Τύπωσε ΤύποςΠαράστασης\$(Τιμή(10&->0@)) ' Decimal

Τύπωσε ΤύποςΠαράστασης\$(Τιμή(10@->0#)) ' Currency

Τύπωσε ΤύποςΠαράστασης\$(Τιμή(10&->0#)) ' Currency

Τύπωσε ΤύποςΠαράστασης\$(Τιμή(10@->0&)) ' Long

Τύπωσε ΤύποςΠαράστασης\$(Τιμή(10#->0&)) ' Long

αντί για μηδέν και σύμβολο μπορούμε να χρησιμοποιήσουμε ονόματα όπως:

Αριθμός (Decimal)

Λογιστικό (Currency)

Διπλός (Double)

Απλός (Single)

Μακρύς (Long)

Λογικός (Boolean) ' μόνο έτσι υπάρχει για μετατροπή, αλλιώς κάνουμε το (παράσταση)<>0 το οποίο δίνει αληθές για μη μηδενική τιμή.

A=Τιμή(B->λογιστικό)

3) Επιστροφή μήκους όπου υπάρχει ο αριθμός στην αρχή του αλφαριθμητικού, ή -1 αν δεν

υπάρχει

Συνάρτηση Αριθμός(α\$) {

 Κάνε β

 =Τιμή(Ψευδές->Λογικός)

 Δες {

 Αν ΕινΓρ Τότε {

 z=Τιμή(α\$,γράμμα\$, β)

 } Αλλιώς.Αν ΕινΑρ Τότε {

 z=Τιμή(α\$,αριθμός, β)

 } Αλλιώς z=Τιμή(α\$, "", β)

 =β>Μήκος(α\$)

 }

}

Συνάρτηση ΑκέραιοςΑριθμός(α\$) {

 Κάνε β

 =Τιμή(Ψευδές->Λογικός)

 Δες {

 z=Τιμή(α\$,"Ακ", β)

 =β>Μήκος(α\$)

 }

}

Τύπωσε ΑκέραιοςΑριθμός("1221213123213")=Αληθές

Τύπωσε ΑκέραιοςΑριθμός("1221213.123213")=Ψευδές

Τύπωσε Αριθμός("123131232131231231.23123123")=Αληθές

Τύπωσε Αριθμός("-123131232131231231.23123123e112")=Αληθές

Τύπωσε Αριθμός("-123131232131231231.23123123e112", ",")=Ψευδές

Τύπωσε Αριθμός("-123131232131231231.23123123e112", 1036)=Ψευδές

Τύπωσε Αριθμός("-123131232131231231.23123123e112", 1033)=Αληθές

Τύπωσε Τιμή("233.44sec", 1033)=233.44

α\$="233.44sec"

β=0

Τύπωσε Τιμή(α\$, 1033, β)=233.44

Αν β>0 Τότε Τύπωσε Mid\$(α\$, β)="sec"

\\ οποιοδήποτε αλφαριθμητικό με μήκος >1 για χαρακτήρα δεκαδικών εξαιρεί τα δεκαδικά.

Τύπωσε Τιμή(α\$, "??", β)=233

Αν β>0 Τότε Τύπωσε Mid\$(α\$, β)=".44sec"

Αναγνωριστικό: TIMΗΣΩΡΟΥ([STACKITEM(]

ΒΑΛΕ 1,2,3

ΤΥΠΩΣΕ ΤΙΜΗΣΩΡΟΥ(2), ΤΙΜΗΣΩΡΟΥ()

2 3

τιμησωρου(1) είναι η κορυφή του σωρού και είναι ίδιο όπως τιμησωρου()

Αν δεν υπάρχει τιμή στη θέση που ζητάμε τότε έχουμε λάθος

Αν δεν υπάρχει η θέση που ζητάμε τότε έχουμε λάθος

χρησιμοποιεί με την ΠΕΤΑ, το ΜΕΓΕΘΟΣ.ΣΩΡΟΥ και την ΤΙΜΗΣΩΡΟΥ\$() για να φτιάχνουμε πλαίσια παραμέτρων στο σωρό.

Αναγνωριστικό: ΤΜΗΜΑ([MODULE()

Τύπωσε Τμήμα(A)

δίνει -1 αν υπάρχει το τμήμα A

Τύπωσε Τμήμα()

επιστρέφει την διεύθυνση για κλήση από γλώσσα μηχανής δεξ το ΔιεύθυνσηΑπό

Τύπωσε Τμήμα(δέλτα!)

δίνει ακριβώς την τιμή που δίνει η έκφραση ΔιεύθυνσηΑπό δέλτα

Τμήμα δέλτα {

}

Τύπωσε Τμήμα(δέλτα!)=ΔιεύθυνσηΑπό δέλτα

Αναγνωριστικό: ΤΟΞ.ΕΦ([ATN()

Τύπωσε Τοξ.εφ(1) 'σε μοίρες

Τύπωσε αΤοξ.εφ(1) 'σε ακτίνια

επιστρέφει μιγαδικό αν δώσουμε μιγαδικό

Αναγνωριστικό: ΤΟΠΙΚΟ([LOCALE()

1) βρες το τοπικό για το αλφαριθμητικό

Τύπωσε Τοπικό("ΓΑΒ")

1032

2) Βρες αν τα γράμματα ανήκουν σε κάποιο τοπικό

Τύπωσε Τοπικό("ΓΑΒ",1032)

Τύπωσε Τοπικό("ΓΑΒ",1033)

0

Αναγνωριστικό: ΤΡΑΠ([BANK(

Είναι ίδια με την Round() της VB6

στρογγυλοποιεί στο μισό προς τον ζυγό αριθμό.

παίρνει και δεύτερη τιμή για να αφήσει δεκαδικά

Αναγνωριστικό: ΤΥΧΑΙΟΣ([RANDOM(

Τύπωσε Τυχαίος(1,10) \ \ δίνει έναν αριθμό μεταξύ 1 και 10

Τύπωσε Τυχαίος \ \ δίνει έναν αριθμό >=0 και μικρότερο του 1

X=Τυχαίος(!12345) \ \ βάζει την γεννήτρια τυχαίων από συγκεκριμένο βήμα και το προηγούμενο βήμα το θέτει σε μια στοίβα ενός επιπέδου

X=Τυχαίος(!) \ \ παίρνει από τη στοίβα το αποθηκευμένο βήμα

Δες Random()

Αναγνωριστικό: ΥΠΑΡΧΕΙ([EXIST(

1) Χρησιμοποιώντας μια αλφαριθμητική έκφραση

ΑΝ ΥΠΑΡΧΕΙ("C:\AUTOEXEC.BAT") ΤΟΤΕ ΤΥΠΩΣΕ "ΟΚ"

2) Χρησιμοποιώντας ένα αντικείμενο Κατάσταση ως πρώτο όρισμα (κάνει χρήση πίνακα κατακερματισμού)

Κάνει αναζήτηση και υψώνει μια σημαία στο αντικείμενο αν βρεθεί το ζητούμενο

Κατάσταση αλφα=1,2:=200,"βήτα":=1000,"100":="οκ"

Τύπωσε αλφα

Αν Υπάρχει(αλφα, 2) τότε Τύπωσε Εκφρ\$(αλφα), Εκφρ(αλφα)

Αν Υπάρχει(αλφα, 1) τότε Τύπωσε Εκφρ\$(αλφα)

Αν Υπάρχει(αλφα, 100) τότε Τύπωσε Εκφρ\$(αλφα)

Αν Υπάρχει(αλφα, "100") τότε Τύπωσε Εκφρ\$(αλφα)

Χωρίς αναζήτηση μπορεί να βγει λάθος!

Τύπωσε αλφα\$("100") 'οκ

Τύπωσε αλφα(500) ' θα βγει λάθος

3) Χρήση τρίτης παραμέτρου για ουρές (τύπος κατάστασης με όμοια κλειδιά)

α=ουρά:=1,2:="2α",2:="2β",3,4:="4α",4:="4β",5, 2:="3β"

μ=Κάθε(α)

Τύπωσε \$(4),"τιμή ως αλφαριθμητικό",@(στήλη(2)), "θέση - από το μηδέν",@(στήλη(4)),"κλειδί σε εισαγωγικά"

Τύπωσε \$(0),

Ενώ μ {

Τύπωσε εκφρ\$(μ),@(στήλη(2)), εκφρ(μ!),@(στήλη(4)), παράθεση\$(εκφρ\$(μ, μ^))

}

\\ νέα παράμετρος για την συνάρτηση Υπάρχει() όταν έχουμε πρώτη παράμετρο ουρά (είδος κατάστασης)

\\ με τιή 0 μας επιστρέφει τον αριθμό στοιχείων (ή 0 αν δεν υπάρχουν) με το ίδιο κλειδί

\\ το κλειδί μπορεί να είναι αλφαριθμητικό ή αριθμός (εσωτερικά είναι αλφαριθμητικό)

Τύπωσε Υπάρχει(α,2, 0)=3 ' αληθές

\\ μια θετική τιμή δηλώνει θέση (αν δεν υπάρχει δίνει 0 αν υπάρχει δίνει -1)

Αν Υπάρχει(α,2,1) Τότε Τύπωσε εκφρ\$(α)="2α" ' αληθές

\\ μια αρνητική τιμή δίνει θέση από το τέλος, με -1 το τελευταίο με το ζητούμενο κλειδί

\\ και εδώ επιστρέφει 0 αν δεν υπάρχει ή -1

Αν Υπάρχει(α,2,-3) Τότε Τύπωσε εκφρ\$(α)="2α" ' αληθές

Τύπωσε Υπάρχει(α,4,0)=2

\\ σε συνδυασμό με την Για μπορούμε να δούμε όλα τα στοιχεία με το ίδιο κλειδί

Για ι=-Υπάρχει(α,4,0) έως -1

Αν Υπάρχει(α,4, ι) Τότε Τύπωσε εκφρ\$(α), ' we get 4α 4β

Επόμενο ι

Τύπωσε

Αναγνωριστικό: ΥΠΑΡΧΕΙ.ΚΑΤΑΛΟΓΟΣ([EXIST.DIR(]

Τυπωσε υπαρχει.καταλογος("d:")

τυπώνει 0 αν δεν έχουμε CD

τυπώνει -1 αν έχουμε CD (υποθέτουμε ότι το D είναι το CD ROM)

μπορούμε να βρούμε ποιο είναι το CD-ROM

με την συνάρτηση ΟΔΗΓΟΣ\$()

Αναγνωριστικό: ΥΠΜΕΡ([CDATE(]

Υπολογισμός μέρας. Χρησιμοποιεί μια αριθμητική τιμή ως αριθμητικό μέρας και τρεις τιμές για έτη μήνες μέρες τα οποία προσθέτει ή αφαιρεί (ανάλογα το πρόσημο της κάθε τιμής) από την πρώτη τιμή

α=Σήμερα

Τύπωσε Ημέρα\$(ΥπΜερ(α, 0,4,0)) \\ 4 μήνες μετά

Τύπωσε Ημέρα\$(ΥπΜερ(α, 0,0,100)) \ 100 μέρες αργότερα
Τύπωσε Ημέρα\$(ΥπΜερ(α, 0,0,-1)) \ χθες

Αναγνωριστικό: ΥΠΟΔΥΝ([MODPOW()
Τύπωσε υποδυν(123456789324234324u, 32042304828u, 234234224442u)=130773979602u
Δίνει το (123456789324234324u^32042304828u) MOD 234234224442u

Αναγνωριστικό: ΥΠΟΛ [MOD]
Τύπωσε 13 υπολ 3
1
Τύπωσε 13.5 υπολ 3
1.5

Αναγνωριστικό: ΥΠΟΛ# [MOD#]
Εκλείδειο υπόλοιπο.
 $\alpha = -20$
 $\beta = 6$
\\ Ευκλείδεια διαίρεση και υπόλοιπο
 $\gamma = \alpha \text{ Διά } \beta$
 $\delta = \alpha \text{ Υπόλ } \beta$
Τύπωσε γ, δ ' -4 4
Τύπωσε $\alpha = \beta * \gamma + \delta$
\\ Κανονική διαίρεση και υπόλοιπο
 $\gamma = \alpha \text{ Διά } \beta$
 $\delta = \alpha \text{ Υπόλ } \beta$
Τύπωσε γ, δ ' -3 -2
Τύπωσε $\alpha = \beta * \gamma + \delta$

Αναγνωριστικό: ΥΠΟΛ([MOD()
Μόνο για μεγάλους ακέραιους και μιγαδικούς
Για τους μιγαδικούς γυρίζει το μέτρο ή απόλυτη τιμή. Ίδιο με την συνάρτηση ΑΠΟΛ() για μιγαδικούς.

Εδώ ο Β γίνεται μεγάλος ακέραιος.
Επιστρέφει το υπόλοιπο διαίρεσης ως μεγάλος ακέραιος
από τη τελευταία διαίρεση όπου διαβάζουμε το αποτέλεσμα της.

Τύπωσε $B^{*8484884} + \Upsilon\text{ΠΟΛ}(B) = A$ ' Αληθές

Τύπωσε Γραφή\$(β2,"dd-mm-yyyy hh:nn:ss")

$$\alpha\pi_3 = \alpha\pi_1 - \alpha\pi_2$$

```

κυκλ=π*2
κυκλ60=κυκλ/60
Ξυπνητήρι=0
ΞυπνητήριΟκ=Ψευδής
ΠαίξεΉχο=Ψευδής
ΟλόκληρεςΩρες=Ψευδής
Αναβοσβήνει=Ψευδής
ΑρχικόςΧρόνος=20*10 \ 20 δευτερόλεπτα
ΧρόνοςΜπιπ=ΑρχικόςΧρόνος
Ωρα=0
Ωρα12=0
λεπτό=υπωρα(0,0,1,0)
Θέση κχ, κυ
\ η χρώμα { } ; φτιάχνει περιοχή μόνο για εμφάνιση
Χρώμα { Κύκλος Γέμισμα 2, απ2+απ1 } ;
Θέση κχ-απ2-απ1, κυ-απ2-απ1
Βάψε 2*(απ2+απ1),2*(απ2+απ1), 7,1,1
Θέση κχ, κυ-απ1*3/4
Τίτλος "M2000 Ρολόι"
Επιγραφή "M2000 Ρολόι","Arial Black",18,0,2
Θέση κχ, κυ
\ με χρώμα 5 αντί για #2277ff γίνεται διάφανο
\ έχουμε δηλώσει το 5 (ματζέντα) στην εντολή Επιφάνεια
Πένα #2277ff {Κύκλος Γέμισμα #2277ff, απ3/2}
π=2
ι=0
Για ν=0 εως 59
    Αν ν υπολ 5 = 0 τότε {π=4} αλλιώς π=1
    Πάχος π {
        Βήμα γωνία ι, απ1
        Χάραξε γωνία ι, απ2
        Βήμα γωνία ι, -απ2-απ1
    }
    ι+=κυκλ60
Επόμενο ν
Κράτησε
Ανανέωση 1000
Κάθε 100 {
    Αναβοσβήνει~ \ αντιστροφή λογικής (Αναβοσβήνει = Δεν Αναβοσβήνει)
    Άφησε
    Ωρα=Τώρα
    Ωρα12=(τιμη(γραφη$(Ωρα,"h")) υπολ 12) *60+τιμη(γραφη$(Ωρα,"n"))
    Θέση κχ, κυ+απ1/2

```

```

    Αν Ωρα12<60 τότε {
        Επιγραφή Γραφη$(υπωρα(0,12,Ωρα12,τιμη(γραφη$(Ωρα,"s"))), "hh:nn:ss"), "Verdana",
18,0,2
    } Αλλιώς {
        Επιγραφή Γραφη$(υπωρα(0,0,Ωρα12,τιμη(γραφη$(Ωρα,"s"))), "hh:nn:ss"), "Verdana",
18,0,2
    }
Πένα -ΞυπνητήριΟκ*8+4 {
    Βήμα , +απ1/8
    Επιγραφή "+"Γραφη$(υπωρα(0,0,(ξυπνητήρι-Ωρα12+720) υπολ 720,0), "hh:nn "),
"Verdana", 18,0,2
    Βήμα , +απ1/8

    Αν ξυπνητήρι<60 τότε {
        Επιγραφή Γραφη$(υπωρα(0,12,ξυπνητήρι,0), "hh:nn"), "Verdana", 18,0,2
    } Αλλιώς {
        Επιγραφή Γραφη$(υπωρα(0,0,ξυπνητήρι,0), "hh:nn"), "Verdana", 18,0,2
    }
}
Θέση κχ, κυ
γωνία_ξυπν=-ξυπνητήρι*π/360+π/2
Αν Αναβοσβήνει ή όχι ΞυπνητήριΟκ τότε {
    Πάχος 2 {
        Βήμα Γωνία γωνία_ξυπν, απ3
        Χάραξε Γωνία γωνία_ξυπν, απ2, 13
        Βήμα Γωνία γωνία_ξυπν, -απ2
        Βήμα Γωνία γωνία_ξυπν, -απ3
    }
}
Αν ΟλόκληρεςΩρες Τότε {
    γωνία_ώρα=-(τιμη(γραφη$(Ωρα,"h")) υπολ 12)*π/6+π/2
} Αλλιώς γωνία_ώρα=-(τιμη(γραφη$(Ωρα,"h")) υπολ 12+τιμη(γραφη$(Ωρα,"n"))/60)*π/6+π/2
Πάχος 2 {
    Χάραξε Γωνία γωνία_ώρα+π/20, απ3/2
    Χάραξε Γωνία γωνία_ώρα-π/20, απ3/2
    Χάραξε Γωνία γωνία_ώρα+π/20, -απ3/2
    Χάραξε Γωνία γωνία_ώρα-π/20, -απ3/2
}
Αν ΟλόκληρεςΩρες Τότε {
    γωνία_λεπτά=τιμη(γραφη$(Ωρα,"n"))*π/30+π/2
} Αλλιώς {
    γωνία_λεπτά=-(τιμη(γραφη$(Ωρα,"n"))+τιμη(γραφη$(Ωρα,"s"))/60)*π/30+π/2
}

```

```

Πάχος 2 {
    Χάραξε Γωνία γωνία_λεπτά+πi/20, απ1/2
    Χάραξε Γωνία γωνία_λεπτά-πi/20, απ1/2
    Χάραξε Γωνία γωνία_λεπτά+πi/20, -απ1/2
    Χάραξε Γωνία γωνία_λεπτά-πi/20, -απ1/2
}
γωνία_δευτ=-τιμη(γραφη$(Ωρα,"s"))*πi/30+πi/2
Πάχος 2 {
    Χάραξε Γωνία γωνία_δευτ, απ3, 15
    Βήμα Γωνία γωνία_δευτ, -απ3
}
Αν Πατημένο(1) ή Πατημένο(0x25) τότε {
Αν Πατημένο(4) ή Πατημένο(0x26) τότε {ξυπνητήρι-=10: ΞυπνητήριΟκ=Αληθής} αλλιώς
ξυπνητήρι--
    ξυπνητήρι=(ξυπνητήρι+720) υπολ 720
} Αλλιώς.Αν Πατημένο(2) ή Πατημένο(0x27) τότε {
Αν Πατημένο(4) ή Πατημένο(0x26) τότε {ξυπνητήρι+=10: ΞυπνητήριΟκ=Αληθής} αλλιώς
ξυπνητήρι++
    ξυπνητήρι=(ξυπνητήρι+720) υπολ 720
}
Αν Πατημένο(32) τότε Έξοδος
Ανανέωση 1000
Αν ΞυπνητήριΟκ και ΧρόνοςΜπιπ>0 Τότε {
    Αν (τιμη(γραφη$(Ωρα,"h")) υπολ 12 )*60+τιμη(γραφη$(Ωρα,"n"))=ξυπνητήρι τότε {
        ΠαίξεΉχο=Αληθής
    }
}
Αν ΞυπνητήριΟκ και όχι (Πατημένο(2) ή Πατημένο(1)) Τότε {
    Αν Πατημένο(4) ή Πατημένο(0x28) Τότε ΠαίξεΉχο=Ψευδής: ΞυπνητήριΟκ=Ψευδής :
ΧρόνοςΜπιπ=ΑρχικόςΧρόνος
} Αλλιώς.Αν Πατημένο(4) ή Πατημένο(0x28) Τότε {
    ΞυπνητήριΟκ=Αληθής
    ΧρόνοςΜπιπ=ΑρχικόςΧρόνος
}
Αν ΠαίξεΉχο Τότε {
    Αν ΧρόνοςΜπιπ υπολ 3 =0 τότε Μπιπ
    ΧρόνοςΜπιπ--
    Αν ΧρόνοςΜπιπ=0 Τότε ΠαίξεΉχο=Ψευδής: ΞυπνητήριΟκ=Ψευδής
}

Αν Πατημένο(13) τότε ΟλόκληρεςΩρες~ \ \ αντιστροφή αληθές<>ψευδές
}
Οθόνη \ \ επαναφορά από το περιορισμό περιοχής της Χρώμα

```

Επιφάνεια 255

}

A:Τέλος

Αναγνωριστικό: ΦΑΣΗ([PHASE(]

Επιστρέφει τη φάση ενός μιγαδικού
(ίδια τιμή από την ΟΡΙΣΜΑ())

Αναγνωριστικό: ΧΑΡΚΩΔ([CHRCODE(]

Τυπωσε Χαρκωδ(Χαρκωδ\$(0x0645))
επιστρέφει την 16bit τιμή του πρώτου χαρακτήρα ενός αλφαριθμητικού.

Αναγνωριστικό: ΧΚΦ([HSL(]

Πένα ΧΚΦ(60,100,50) ' χροιά 60 μοίρες, κορεσμός 100%, φωτεινότητα 50%

Αναγνωριστικό: ΧΡΟΝΟΣ([TIME(]

Μετατροπή ώρας από αλφαριθμητικό σε αριθμητικό
Τύπωσε Χρόνος(Χρόνος\$("10:20"))

Αναγνωριστικό: ΧΡΩΜΑ([COLOR(]

ΠΕΝΑ ΧΡΩΜΑ(100,50,255)

Δίνει το χρώμα με κόκκινο 100, πράσινο 50 και μπλε 255

το μαύρο είναι το ΧΡΩΜΑ(0,0,0)

το άσπρο είναι το ΧΡΩΜΑ(255,255,255)

Πένα #FF0015

Πένα Χρώμα("FF0015")

Πένα Χρώμα (#FF0015)

Πένα Χρώμα (0X1500FF) \ \ στο δεκαεξαδικό είναι BGR, δηλαδή το νούμερο 15 σημαίνει (0,0,15) στο RGB.

Ομάδα: ΑΛΦΑΡΙΘΜΗΤΙΚΑ - STRING FUNCTIONS

Αναγνωριστικό: #ΓΡΑΦΗ\$([#STR\$(

Τύπωσε (1,2,3,4)#ΓΡΑΦΗ\$(

1 2 3 4

Τύπωσε (1,2,3,4)#ΓΡΑΦΗ\$(,)

1,2,3,4

Τύπωσε (1,2,3,4)#ΓΡΑΦΗ\$("-")

1-2-3-4

Τύπωσε (1,2,3,4)#ΓΡΑΦΗ\$("-", "0000")

0001-0002-0003-0004

Αναγνωριστικό: #ΕΚΦΡ\$([#EVAL\$(

Δες την #Εκφρ()

Κλάση άλφα\$ {

Αξία (χ) {

=Γραφή\$(500*χ)

}

}

ζ\$=άλφα\$(

Τύπωσε ζ\$(5)

μ=(άλφα\$(, άλφα\$(

κ\$=μ#Τιμή\$(0)

Τύπωσε κ\$(5)

Τύπωσε μ#Εκφρ\$(0, 5)

Τύπωσε "έγινε"

Αναγνωριστικό: #ΜΕΓ\$([#MAX\$(

α=("α", "β", "γ")

Τύπωσε α#μεγ\$()="γ"

ζ=-1

Τύπωσε α#μεγ\$(ζ)="γ", ζ=2

Αναγνωριστικό: #ΜΙΚ\$([#ΜΙΝ\$(

α=("α","β","γ")

Τύπωσε α#μικ\$()="α"

ζ=-1

Τύπωσε α#μικ\$(ζ)="α", ζ=0

Αναγνωριστικό: #ΠΑΚ\$([#FOLD\$(

α=("α","β","γ")

πακ1=λάμδα μ=1 ->{

 φέρε 2 :

 αν μ=1 τότε

 μ=0

 πέτα

 Βάλε ""+κεφ\$(γράμμα\$)+"""

 αλλιώς

 βάλε γραμμα\$+"", ""+""+κεφ\$(γράμμα\$)+"""

 τέλος αν

}

Τύπωσε "["+α#πακ\$(πακ1)+"]" ={"A","B","Γ"}

Αναγνωριστικό: #ΤΙΜΗ\$([#VAL\$(

α=(1,"A", 2, "B")

Τύπωσε α#Τιμή\$(1)="A"

Αναγνωριστικό: ΑΛΛΑΓΗ\$([REPLACE\$(

ΤΥΠΩΣΕ ΑΛΛΑΓΗ\$("A","B","AABAAABBBBAAA")

BBBBBBBBBBBBBB

Αναγνωριστικό: ΑΝ\$([IF\$(

ternary operator

Δες Αν()

\\ πέρνει -1 και 0

Τύπωσε Αν\$(αληθές -> "αληθές", "ψευδές")

\\ πέρνει τιμές 1,2,3,4... (εδώ μέχρι 4)

Τύπωσε Αν\$(3 -> "ένα", "δύο", "τρία", "τέσσερα")

Αναγνωριστικό: ΑΝΑΠ\$([STRREV\$(

Τύπωσε Αναπ\$("αβγδ")="δγβα"

Αναγνωριστικό: ΑΠΟΚ\$([TRIM\$(

ΤΥΠΩΣΕ "<" + ΑΠΟΚ\$(" AAA ") + ">"

<AAA>

αποκόπτει τα διαστήματα σε ένα αλφαριθμητικό και από τις δύο πλευρές

Τοπικό 1032

A\$=Γραφή\$(" Γιώργος ") ' μετατροπή σε ANSI κατά τοπικό1032 (ελληνικά)

Τύπωσε Μήκος(A\$)=4.5

B\$=Αποκ\$(A\$ ως ψηφίο)

Τύπωσε Χαρ\$(B\$)="Γιώργος" ' μετατροπή σε UTF-16LE κατά 1032

Αναγνωριστικό: ΑΠΟΚ.ΑΡ\$([LTRIM\$(

Τύπωσε ΑΠΟΚ.ΑΡ\$(" 123")="123"

κ\$=Γραφή\$(" 123")

Τύπωσε Χαρ\$(ΑΠΟΚ.ΑΡ\$(κ\$ ως ψηφίο))="123"

Τύπωσε ΑΠΟΚ.ΑΡ\$(κ\$ ως ψηφίο)=Γραφή\$("123")

Αναγνωριστικό: ΑΠΟΚ.ΔΕ\$([RTRIM\$(

Τύπωσε ΑΠΟΚ.ΔΕ\$("123 ")="123"

κ\$=Γραφή\$("123 ")

Τύπωσε Χαρ\$(ΑΠΟΚ.ΔΕ\$(κ\$ ως ψηφίο))="123"

Τύπωσε ΑΠΟΚ.ΔΕ\$(κ\$ ως ψηφίο)=Γραφή\$("123")

Αναγνωριστικό: ΑΡΙΣ\$([LEFT\$(
ΤΥΠΩΣΕ ΑΡΙΣ\$("ΑΒΓΔΕΖ",2)
ΑΒ

Δίνει ένα αριστερό τμήμα ενός αλφαριθμητικού, έτσι δίνουμε ένα αλφαριθμητικό και το μήκος του τμήματος αυτού. Αν δώσουμε μεγαλύτερο μήκος από αυτό του αλφαριθμητικού τότε δίνει όλο το αλφαριθμητικό

Τοπικό 1032

Α\$=Γραφη\$("Γιώργος") ' μετατροπή σε ANSI κατά τοπικό1032 (ελληνικά)
Τύπωσε Χαρ\$(Αρισ\$(Α\$,3 ως Ψηφίο))="Γιώ" ' το Χαρ\$() μετατροπή σε UTF-16LE κατά 1032

Αναγνωριστικό: ΑΡΙΣΤΕΡΟΜΕΡΟΣ\$([LEFTPART\$(
Τύπωσε ΔεξίΜέρος\$("123456789","5") ' μετά το 5
Τύπωσε ΑριστερόΜέρος\$("123456789","5") ' πριν το 5

Αναγνωριστικό: ΑΡΧΕΙΟ\$([FILE\$(
ΤΥΠΩΣΕ ΑΡΧΕΙΟ\$("ΚΕΙΜΕΝΟ","TXT")
ΦΟΡΤΩΣΕ ΑΡΧΕΙΟ\$() 'Επιλέγουμε αρχείο GSB
Η συνάρτηση ΑΡΧΕΙΟ\$() παίρνει από κανένα μέχρι δυο παραμέτρους, με πρώτη το τίτλο του διαλόγου και δεύτερη τον ή τους τύπους των αρχείων (τους διαχωρίζουμε με τη κάθετο |)

Αναγνωριστικό: ΑΥΛΟΣ\$([PIPENAME\$(
? ΑΥΛΟΣ\$(αλφα)

Αν δώσουμε νούμερο (το νήμα για τον αυλό) τότε θα μας δώσει τον "τόπο" του (όλο το μονοπάτι του, αφού ο αυλός λειτουργεί σαν αρχείο)
Αν δώσουμε μόνο το όνομα σε αλφαριθμητικό τότε δίνει το υπόλοιπο για να είναι σωστό μαζί με τον τόπο του.

Οι αυλοί λειτουργικά γίνονται γνωστοί σε ένα δίκτυο, επειδή συστατικό τους στοιχείο είναι το όνομα του υπολογιστή που εμφανίζεται στο δίκτυο. Εφόσον θέλει κάποιος να επικοινωνήσει από έναν άλλο υπολογιστή στο δίκτυο, τότε στο όνομα του αυλού πρέπει να βάλει το όνομα του

υπολογιστή που κατέχει τον αυλό. Ο κάθε άλλος (η κάθε άλλη εφαρμογή εκτός του υπολογιστή αυτού) θα μπορεί να χρησιμοποιεί την εντολή ΑΥΛΟΣ και να στέλνει δεδομένα στον αυλό που έχει γνωστοποιηθεί.

Αναγνωριστικό: ΒΑΛΕ.ΑΔΕΙΑ\$([ADD.LICENSE\$(
προσθέτει άδεια για χρήση αντικειμένου (με την όρισε)
Παίρνει ένα ή δυο ορίσματα. Το πρώτο είναι το αντικείμενο όπως ονομάζεται ή το νούμερό του, και το δεύτερο η άδεια
(είναι η συνάρτηση licence.add της VB6)
Info: [https://msdn.microsoft.com/en-us/library/aa277583\(v=vs.60\).aspx](https://msdn.microsoft.com/en-us/library/aa277583(v=vs.60).aspx)

Αναγνωριστικό: ΓΡΑΦΗ\$([STR\$(
ΤΥΠΩΣΕ ΓΡΑΦΗ\$(ΣΗΜΕΡΑ,"dddd")
Η συνάρτηση ΓΡΑΦΗ\$() έχει τρεις μορφές:

ΓΡΑΦΗ\$(αριθμός) ' το σημείο των δεκαδικών είναι πάντα το ".", στους θετικούς βάζει διάστημα στην αρχή ή - αν είναι αρνητικοί

Με Γραφή\$(αριθμός, "") ' παίρνουμε τον αριθμό με σημείο δεκαδικών (αν χρειάζεται βγάζει το σημείο)

Τύπωσε Γραφή\$(.34,"")

ΓΡΑΦΗ\$(αριθμός, φόρμα\$)

ΓΡΑΦΗ\$(γράμματα\$, φόρμα\$)

Παράδειγμα:

ΤΥΠΩΣΕ ΓΡΑΦΗ\$("ΑΒΓΔΕΖ","<")

Μας τυπώνει σε πεζά:

αβγδεζ

Η φόρμα που εμφανίζεται στη δεύτερη παράμετρο είναι ίδια με την πρώτη παράμετρο της \$(
ΔΕΣ ΤΥΠΩΣΕ

Μπορούμε να κρατάμε σε ένα αλφαριθμητικό την ANSI απεικόνιση μιας Unicode σειράς. Τα αλφαριθμητικά στην M2000 είναι του τύπου UTF16LE, δηλαδή δυο bytes κάθε χαρακτήρας. Αν κατασκευάσουμε ένα ANSI αλφαριθμητικό θα καταχωρηθεί ένας χαρακτήρας σε μια θέση των δυο bytes. Σε ένα αρχείο ANSI κάθε χαρακτήρας ANSI έχει ένα byte θέση. Όταν διαβάσουμε από ένα ANSI αρχείο τότε το σύστημα θα μας το μετατρέψει σε UTF16LE με βάση το Τοπικό Αναγνωριστικό (1032 για Ελληνικά). Αν γνωρίζουμε ότι το αρχείο είναι από άλλη γλώσσα τότε το

μετατρέπουμε σε ANSI με την ΓΡΑΦΗ\$(a\$,0) και το ξαναμετατρέπουμε σε UNICODE με την ΧΑΡ\$(a\$, 1049), ή πιο εύκολα με αλλαγή του τοπικού στο 1049.

α\$=Γραφή\$("Hello")

\\ το α\$ γράφεται σαν Ansi αλφαριθμητικό, ένα byte ανά χαρακτήρα.

Τύπωσε Μήκος(α\$)=2.5

Τύπωσε "Bytes:"=Μήκος(α\$)*2 ' 5 bytes

Τύπωσε Μήκος(χαρ\$(α\$))=5

Τύπωσε α\$

ΔΕΣ [ΦΟΡΜΑΡΙΣΜΑ_ΑΛΦΑΡΙΘΜΗΤΙΚΩΝ[

ΔΕΣ [ΦΟΡΜΑΡΙΣΜΑ_ΑΡΙΘΜΩΝ]

ΔΕΣ [ΦΟΡΜΑΡΙΣΜΑ_ΗΜΕΡΟΜΗΝΙΩΝ ΚΑΙ ΩΡΑΣ]

ΔΕΣ [ΦΟΡΜΑΡΙΣΜΑ_ΓΙΑ ΚΑΘΕ ΤΥΠΟ]

Αναγνωριστικό: ΔΕΚΑΕΞ\$([HEX\$(

a=0xFF12FFFF

ΤΥΠΩΣΕ ΔΕΚΑΕΞ\$(a),a

Νέα από την έκδοση 6.4 και μετά: Δεύτερη παράμετρος 1 έως 4 δηλώνει τα bytes που θα δείξει.

Με ένα τους δυο αριστερούς χαρακτήρες.

Τυπώνει το δεκαεξαδικό νούμερο

FF12FFFF 4279435263

Στην έκτη έκδοση μπορούμε να δίνουμε δεκαεξδικά νούμερα με πρόθεμα 0x

Η γλώσσα μπορεί να χειριστεί αριθμούς χωρίς πρόσημο 32bit

Το 0xFFFFFFFF& είναι το -1 (32 bit long)

Το 0xFFFF% είναι το -1 (16bit integer)

Δες το ΑΚΕΡΑΙΟ.ΔΥΑΔΙΚΟ().

Αναγνωριστικό: ΔΕΞΙ\$([RIGHT\$(

Δέχεται δυο ορίσματα. Μας δίνει ένα νέο αλφαριθμητικό από ένα που δίνουμε με μήκος το πολύ όσο μια τιμή που επίσης δίνουμε, λαμβάνοντας χαρακτήρες από την δεξιά πλευρά του αλφαριθμητικού.

```
A$="ΤΙΤΛΟΣ"  
A$=ΚΕΦ$(ΑΡΙΣ$(A$,1)) + ΜΕΣ$(ΠΕΖ$(A$),2)  
ΑΝ (ΔΕΞΙ$(A$,1)="σ" ΚΑΙ ΜΗΚΟΣ(A$)>1) ΤΟΤΕ A$=ΑΡΙΣ$(A$,ΜΗΚΟΣ(A$)-1)+"ς"  
ΤΥΠΩΣΕ A$  
Μας τυπώνει  
Τίτλος
```

```
Τοπικό 1032  
A$=Γραφη$("Γιώργος") ' μετατροπή σε ANSI κατά τοπικό1032 (ελληνικά)  
Τύπωσε Χαρ$(Δεξι$(A$,3 ως Ψηφίο))="γος" ' το Χαρ$() μετατροπή σε UTF-16LE κατά 1032
```

```
Αναγνωριστικό: ΔΕΞΙΜΕΡΟΣ$( [RIGHTPART$(  
Τύπωσε ΔεξιΜέρος$("123456789","5") ' μετά το 5  
Τύπωσε ΑριστερόΜέρος$("123456789","5") ' πριν το 5
```

```
Αναγνωριστικό: ΕΙΚ$( [BMP$(  
Δίνουμε ένα όνομα αρχείου BMP χωρίς κατάληξη και εξάγει αν υπάρχει το αρχείο το πλήρες  
όνομά του (με τη διεύθυνσή του και την κατάληξη BMP)
```

```
Αναγνωριστικό: ΕΙΣΑΓΩΓΗ$( [INPUT$(  
Σε απλό αρχείο ANSI  
N=1  
//Γράφουμε 20 χαρακτήρες, μετάθεση 20 bytes, μας επιστρέφει 40 Bytes, επειδή γίνεται  
μετατροπή σε UTF16LE  
A$=ΕΙΣΑΓΩΓΗ$(#ΑΡΧΕΙΟ, 20*2/N)  
A$=ΕΙΣΑΓΩΓΗ$(#ΑΡΧΕΙΟ, 20*2/N, 1032)  
A$=ΕΙΣΑΓΩΓΗ$(#ΑΡΧΕΙΟ, 20*2/N, -1) ' UTF8 δεν ξέρουμε ακριβώς πόσο μήκος θα έχει, το  
περισσότερο όμως 20 χαρακτήρες.  
Σε αρχείο με άνοιγμα ΓΙΑ ΕΥΡΕΙΑ  
N=2  
//Γράφουμε 20 χαρακτήρες, μετάθεση 40 bytes  
A$=ΕΙΣΑΓΩΓΗ$(#ΑΡΧΕΙΟ, 20*2/N)  
A$=ΕΙΣΑΓΩΓΗ$(#ΑΡΧΕΙΟ, 20) ' αφού N=2, το 2/N=1  
// Αν στην θέση υπάρχουν 5 Bytes σε ANSI ή άλλη μορφή τότε  
// με 2.5 θα πάρουμε 5 bytes. Εδώ δεν γίνεται μετατροπή σε UTF16LE  
A$=ΕΙΣΑΓΩΓΗ$(#ΑΡΧΕΙΟ, 2.5)  
// με μετατροπή κατά ANSI 1032
```

```
A$=ΕΙΣΑΓΩΓΗ$(#ΑΡΧΕΙΟ, 2.5, 1032)
// με μετατροπή κατά UTF8
A$=ΕΙΣΑΓΩΓΗ$(#ΑΡΧΕΙΟ, 2.5, -1)
```

Μπορούμε να μετακινήσουμε το δρομέα πριν διαβάσουμε, με την εντολή ΜΕΤΑΘΕΣΗ #ΑΡΧΕΙΟ, 100. Η εντολή ΜΕΤΑΘΕΣΗ λογαριάζει θέσεις σε Bytes πάντα. Μπορούμε να χρησιμοποιούμε την ΤΕΛΟΣ() για να γνωρίζουμε αν σημάνθηκε το τέλος στο διάβασμα του αρχείου (ο δρομέας πέρασε το τελευταίο χαρακτήρα, στους χαρακτήρες λογαριάζονται και οι αλλαγές γραμμών) Μπορούμε να διαβάσουμε UNICODE αν ορίσουμε ΓΙΑ ΕΥΡΕΙΑ ΕΙΣΑΓΩΓΗ στο ΑΝΟΙΞΕ

Η ΓΡΑΜΜΗ ΕΙΣΑΓΩΓΗΣ δεν φέρνει τις αλλαγές γραμμών αλλά σταματάει σε αυτές το διάβασμα και τοποθετεί τον δρομέα στην επόμενη γραμμή, αφήνοντας την αλλαγή γραμμή (που μπορεί να είναι δυο χαρακτήρες)

Δείτε τι συμβαίνει με το νούμερο που δίνουμε αν διαβάζουμε ένα ANSI αρχείο:

Αν διαβάσουμε με άνοιγμα για ANSI, η Εισαγωγή\$() θα μας δώσει τους χαρακτήρες που θέλουμε, αλλά πρέπει να ζητήσουμε διπλάσιους! Γιατί ο δρομέας πάει κατά ένα byte ενώ εμείς γράφουμε δυο στο αλφαριθμητικό. Η εισαγωγή\$() γυρνάει ότι της ζητήσουμε, αλλά στο άνοιγμα με ANSI εφαρμόζει μετατροπή βάσει του Τοπικού ή αν θέλουμε δίνουμε επιπλέον παράμετρο για μετατροπή.

Αν διαβάσουμε με άνοιγμα για UTF8, για να πάρουμε τους ANSI θα πρέπει να διαβάσουμε τα μισά, δηλαδή αν θέλουμε 40 χαρακτήρες θα ζητήσουμε 20, και ο δρομέας θα πάει 10*2 (επειδή έτσι πηγαίνει, ανά 2 Bytes για κάθε χαρακτήρα)

Δείτε τι συμβαίνει με το νούμερο που δίνουμε αν διαβάζουμε ένα ANSI αρχείο:

Αναγνωριστικό: ΕΚΦΡ\$([ΕVAL\$(

ΕΚΦΡΑΣΗ\$() ή ΕΚΦΡ\$(

είναι η αντίστοιχη ΕΚΦΡ\$() για αλφαριθμητικά

1) Χρήση της Εκφρ\$() για άμεση χρήση δείκτη (οι δείκτες σε μεταβλητές στη M2000 είναι αλφαριθμητικά)

α\$="οκ"

β\$=Ισχύη\$(α\$)

\\ δείτε παρακάτω ότι η β\$ έχει μια τελεία μετά το \$

β\$.="ναι" \\ η β\$ περιέχει δείκτη στην α\$, οπότε η τιμή "ναι" πάει στην α\$

Τύπωσε α\$

Τύπωσε Εκφρ\$(β\$.) \\ "ναι" \\ δες την τελεία μετά το \$

Τύπωσε Εκφρ\$(β\$)=β\$ \\ δίνει το περιεχόμενο του β\$, όχι αυτό που δείχνει αυτό!

2) Χρήση της Εκφρ\$() με αντικείμενο Κατάσταση

\\ οι καταστάσεις έχουν κλειδιά και τιμές, ή αν δεν έχουν τιμή σε κλειδί, τότε γυρνάει ως τιμή το κλειδί

Κατάσταση άλφα=1,"200",3:=500,4

β=Κάθε(άλφα)

Ενώ β {

\\ δείτε το ! στο πρώτο Εκφρ(), λέει στο διερμηνευτή ότι θέλουμε το κλειδί

Τύπωσε Εκφρ\$(β!), Εκφρ\$(β) ' Κλειδί, στοιχείο ως αλφαριθμητικό

}

3) Χρήση Εκφρ\$() με αντικείμενο Διάρθρωση. Δείτε πως διαβάζουμε αλφαριθμητικό από θέση στη μνήμη:

Δομή άλφα {

A as Μακρύς \\ 32 δυαδικών ψηφίων

β ως Ακέραιος*40 \\ ο Ακέραιος εδώ είναι 16 δυαδικών ψηφίων

}

Τύπωσε Μήκος(άλφα) \\ 4 byte + 2 byte * 40 = 84

Διάρθρωση Κενή β1 ως άλφα*20 \\ 84*20=1680 Bytes

Τύπωσε Μήκος(β1)

Τύπωσε Μήκος("Γεια χαρά!.....1234567") ' 42

Επιστροφή β1, 3!β:=Πεδίο\$("Γεια χαρά!.....1234567",40)

Τύπωσε Εκφρ\$(β1, 3!β, 40*2)

\\ διαβάζουμε από τη θέση μνήμης β1+3*84+4, 40*2 = 80 bytes

\\ ο αριθμός 3 δείχνει στην Εκφρ\$() ότι μας ενδιαφέρει το τέταρτο άλφα, που έχει διεύθυνση από την αρχή, το β1, όσο 3 άλφα, άρα 3*84, και επιπλέον θέλουμε να διαβάσουμε από το β πεδίο της δομής άλφα, δηλαδή αμέσως μετά το πρώτο, που είναι 4 bytes. Ζητάμε 2*40 bytes.

\\ εδώ γράφουμε χαρακτήρες ανά Byte

Επιστροφή β1, 10!β:=Γραφή\$("Γεια")

\\ διαβάζουμε σαν να είχαμε δυο byte ανά χαρακτήρα, και μετά γίνεται μετατροπή με την Χαρ\$()

\\ εδώ θέλουμε μόνο 4 Bytes

Τύπωσε Χαρ\$(Εκφρ\$(β1, 10!β, 2*2))

\\ ενώ εδώ βάζουμε σε Unicode 16LE, και θέλουμε 8 bytes για να διαβάσουμε

Επιστροφή β1, 10!β:="Γεια"

Τύπωσε Εκφρ\$(β1, 10!β, 8)

Αναγνωριστικό: ΕΝΩΣΗ.ΣΕΙΠΑΣ\$([UNION.DATA\$[

Τύπωσε Ένωση.Σειράς\$(65, 66,"C", "alfa", 0x2102)="ABCalfa©"

Αναγνωριστικό: ΕΠΑΝ\$([STRING\$(

Δέχεται δυο ορίσματα. Το πρώτο είναι ένα αλφαριθμητικό, το δεύτερο ένας αριθμός που δηλώνει πόσες φορές θα επαναλάβει το πρώτο σε ένα νέο που θα μας επιστρέψει

1) Τύπωσε Επαν\$("A", 3)

AAA

2) Τύπωσε Επαν\$(α\$)

επανεξάγει το αλφαριθμητικό με ορισμένες ακολουθίες escape (σε επιλεγμένους χαρακτήρες)

3) Τύπωσε Επαν\$(α\$ ως json)

επανεξάγει το αλφαριθμητικό με ακολουθίες escape κατά json

Παράδειγμα:

α\$={alfa

 "beta"

 gamma

}

Τύπωσε επαν\$(α\$)

Τύπωσε επαν\$(α\$ ως json)

Αναφορά Μορφή\$(επαν\$(α\$))

Αναφορά Μορφή\$(επαν\$(α\$ ως json))

4) Τύπωσε Επαν\$(α\$ ως Encode64)

εξ ορισμού βάζει αλλαγές γραμμών κάθε 60 χαρακτήρες

Τύπωσε Επαν\$(α\$ ως Encode64, 1) ' δεν βάζει αλλαγές γραμμών

Τύπωσε Επαν\$(α\$ ως Encode64, 0, 6) ' βάζει αλλαγές και βάζει και βάζει και 6 χαρακτήρες στην αρχή κάθε γραμμής

5) Τύπωσε Επαν\$(α\$ ως Decode64)

αποκωδικοποίηση BASE64

6) Τύπωσε Επαν\$(α\$ ως UTF8enc)

Κωδικοποίηση σε UTF8

7) Τύπωσε Επαν\$(α\$ ως UTF8dec)

Αποκωδικοποίηση από UTF8

8) Τύπωσε Επαν\$(α\$ ως URLdec)

Τύπωσε Επαν\$(α\$ ως URLdec +) ' αλλάζει τα + με διαστήματα

9) Τύπωσε Επαν\$(α\$ ως URLenc)

Υπάρχουν τρεις τρόποι κωδικοποίησης, ο απλός, ο HTML5 και ο RFC3986

α\$="http://foo.bar.com/~user-name/_subdir/*~.html"

Τύπωσε Επαν\$(α\$ ως URLencHTML5)

Τύπωσε Επαν\$(Επαν\$(α\$ ως URLencHTML5) ως URLdec)

Τύπωσε Επαν\$(α\$ ως URLencHTML5 +) ' διαστήματα με + και όχι %20

Τύπωσε Επαν\$(α\$ ως URLencRFC3986)

Τύπωσε Επαν\$(α\$ ως URLencRFC3986 +) ' διαστήματα με + και όχι %20

Αναγνωριστικό: ΕΠΙΛΟΓΗ\$([MENU\$(]

Μπορούμε να διαβάσουμε αλλά όχι να μεταβάλουμε οποιαδήποτε καταχώρηση στον επιλογέα.

Ο μέγιστος αριθμός στοιχείων μας επιστρέφεται από την αριθμητική συνάρτηση ΕΠΙΛΟΓΕΣ

Δες ΕΠΙΛΟΓΗ

Παράδειγμα

Σωρός Νέος {

Επιλογή \ \ καθαρίζει τον πίνακα

Σειρά "alpha","beta","gama","delta"

Για i =1 Έως 4 { Διάβασε α\$: Επιλογή + α\$ }

Επιλογή !

Αν Επιλογή>0 Τότε Τύπωσε Επιλογή\$(Επιλογή)

}

Αναγνωριστικό: ΕΦΑΡΜΟΓΗ.ΑΡΧΕΙΟΥ\$([FILE.APP\$(

Τύπωσε Εφαρμογή.Αρχείου\$("txt")

Δίνει το πρόγραμμα που ανοίγει τα αρχεία txt

Αναγνωριστικό: ΗΜΕΡΑ\$([DATE\$(

Δέχεται όρισμα μια αριθμητική παράσταση ημερομηνίας και δίνει την ημερομηνία ως αλφαριθμητικό.

A=ΣΗΜΕΡΑ

ΤΥΠΩΣΕ ΗΜΕΡΑ\$(A)

Τύπωσε ημέρα\$(Σήμερα,1033,"Long Date")

Τύπωσε ημέρα\$(Σήμερα,1033,"MMMMM d yyyy")

Τύπωσε ημέρα\$(Σήμερα,) ' χρήση σύντομης ελληνικής

Τύπωσε ημέρα\$(Σήμερα,) 'χρήση σύντομης όπως το ΤΟΠΙΚΟ

Τύπωσε ημέρα\$(Σήμερα,1033) ' χρήση σύντομης για το 1033

Αναγνωριστικό: ΗΧΟ\$([SND\$(

Δίνουμε ένα όνομα αρχείου WAV χωρίς κατάληξη και εξάγει αν υπάρχει το αρχείο το πλήρες όνομά του (με τη διεύθυνσή του και την κατάληξη WAV)

Αναγνωριστικό: ΙΔΙΟΤΗΤΑ\$([PROPERTY\$(

Η συνάρτηση Ιδιότητα\$() θέλει μια ιδιότητα που ορίσαμε με μεταβλητή με την Με αντικείμενο και επιστρέφει ένα αντικείμενο που μπορούμε να καταχωρήσουμε σε πίνακα ή σε κατάσταση
Ισχύει το ίδιο για την Ιδιότητα(), με τη διαφορά ότι δέχεται αριθμητική μεταβλητή για ιδιότητα που γυρίζει ή και παίρνει αριθμό

Ιδιότητες που παίρνουν παράμετρο δεν χρησιμοποιούνται από τις Ιδιότητα() και Ιδιότητα\$(). Εκεί χρησιμοποιούμε αντί για μεταβλητή, ένα όνομα πίνακα (χωρίς να ορίσουμε πίνακα). Ο πίνακας αυτός όταν χρησιμοποιείται χωρίς παράμετρο γυρίζει ότι και τα Ιδιότητα() και Ιδιότητα\$()

Τίτλος "?", 0 ' ελαχιστοποίηση κονσόλας

Όρισε Φόρμα1 Φόρμα

Με Φόρμα1, "title" ως τίτλος\$

Πίνακας α\$(10)

α\$(1)=Ιδιότητα\$(τίτλος\$)

Ένωσε τίτλος\$ στο μ\$

Συνάρτηση Φόρμα1.about {

\\ πάτα ctrl+F1

Περί "Βοήθεια", "Μπλα Μπλα, ctrl+tab αλλάζει τα παράθυρα"

μ\$="Ανοιγμα Βοήθειας"

Τίτλος μ\$

}

Συνάρτηση Φόρμα1.click {

\\ κάνε click στην φόρμα

α\$(1)="Νέος Τίτλος"

Τίτλος α\$(1)

}

Μέθοδος Φόρμα1, "show", 1

\\ με το κλείσιμο της φόρμας, η φόρμα παραμένει ως αντικείμενο

Τύπωσε τίτλος\$, μ\$, α\$(1) ' τυπώνει τρεις φορές τον τίτλο

\\ η φόρμα "Χαλάει" όταν την αποδεσμεύσουμε με εντολή

\\ ή στο πέρας εκτέλεσης αυτού του τμήματος

Όρισε Φόρμα1 Τίποτα

Τίτλος "Πίσω ξανά", 1

Αναγνωριστικό: ΙΣΧΝΗ\$([WEAK\$(

Η ισχνή\$() επιστρέφει ένα αλφαριθμητικό με ισχνή αναφορά σε αυτό που δίνουμε (πρέπει να υπάρχει το όνομα μιας μεταβλητής, ενός πίνακα (με δείκτες) ή μιας συνάρτησης)

\\ παράδειγμα με χρήση της ισχνής αναφοράς και της ένωσης

α\$="{=1000*αριθμός+αριθμός}"

\\ η α\$ έχει ισχνή αναφορά συνάρτησης (έχει το μπλοκ εντολών {})

Ένωσε ισχνη α\$ στο αα()

Τύπωσε αα(10,-100)

M=300

β\$=ισχνή\$(M)

\ \ η β\$ είναι ισχνή αναφορά στην M

Ένωσε ισχνη β\$ στο Δ

Τύπωσε Δ

Δ+=5000

Τύπωσε M, M=Δ, Εκφρ(β\$)

\ \ μπορούμε με μια τελεία μετά το β\$ να την χρησιμοποιούμε άμεσα

β\$.+=500

Τύπωσε M, M=Δ, Εκφρ(β\$)

\ \ Μπορούμε να έχουμε ισχνές αναφορές σε στοιχεία πίνακα

Πίνακας α(20)=10

Τύπωσε α()

α(2)+=10

Τύπωσε α(2)

\ \ η έκφραση χ-1 εκτελείτε πριν παραχθεί η ισχνή αναφορά

χ=3

α\$=ισχνή\$(α(χ-1))

Τύπωσε Έκφρ(α\$)

α\$.+=20

Τύπωσε Έκφρ(α\$)

Τύπωσε α(2), α(2)=Έκφρ(α\$)

Τύπωσε α\$

μ=100

α\$=ισχνή\$(μ)

α\$.+=20

Τύπωσε Έκφρ(α\$)

Τύπωσε α\$

Αναγνωριστικό: ΚΕΦ\$([UCASE\$(

1. Δέχεται σαν όρισμα ένα αλφαριθμητικό και δίνει το ίδιο αλλά με κεφαλαία γράμματα

2. Δεύτερη παράμετρος προαιρετική το τοπικό.

Δείτε το παράδειγμα. Το 1032 είναι για την Ελλάδα, ενώ το 1033 τα αγγλικά.

Κανονικά η λέξη "Γιώργος" έχει 7 χαρακτήρες των 14 συνολικά bytes

τοπικο 1032

α\$=γραφη\$("Γιώργος")

Τύπωσε χαρ\$(α\$) ' μετατροπή βάσει του 1032

Τύπωσε Μήκος(α\$)=3.5 ' μονάδες word=2 bytes, άρα 3.5*2=7bytes

τοπικό 1032

Τύπωσε Κεφ\$(χαρ\$(α\$), 0) ' δίνει λάθος

τοπικό 1033

Τύπωσε Κεφ\$(χαρ\$(α\$), 1032) ' ΟΚ
τοπικό 1033
Τύπωσε Κεφ\$(χαρ\$(α\$)) ' δίνει λάθος
τοπικό 1032
Τύπωσε Κεφ\$(χαρ\$(α\$)) ' ΟΚ

Αναγνωριστικό: ΚΡΥΦΟΣ\$([HIDE\$(]
Μετατρέπει ένα αλφαριθμητικό σε ένα κρυπτογραφημένο αλφαριθμητικό (διπλάσιου μήκους).
Δέχεται για όρισμα το προς κρυπτογράφηση αλφαριθμητικό, ένα αλφαριθμητικό για κωδικό
καθώς και έναν αριθμό που επίσης διαφοροποιεί τον κωδικό.
Το σύστημα κρυπτογράφησης παράγει κάθε φορά διαφορετικό αποτέλεσμα αλλά όλα τα
αποτελέσματα μπορούν να επαναφέρουν το αρχικό, αν γνωρίζουμε κωδικό και αριθμό. Ο
αριθμός μπορεί να χρησιμοποιηθεί έτσι ώστε πολλαπλά αλφαριθμητικά να κρυπτογραφούνται
με εξάρτηση του παρόντος από το προηγούμενο.

Αναγνωριστικό: ΛΟΓΟΣ\$([SPEECH\$(]
? λογος\$(1)
τυπώνει το όνομα της φωνής 1 από το σύστημα ομιλίας

Αναγνωριστικό: ΜΕΓΑΛΟ.ΣΕΙΡΑΣ\$([MAX.DATAS\$(]
? μεγαλο.σειρας\$("αα", "ββ")
αα
? μεγαλο.σειρας\$("κλμν", "αα", "ββ", "βα")
κλμν
Δες και το ΜΙΚΡΟ.ΣΕΙΡΑΣ\$() και ΜΙΚΡΟ.ΣΕΙΡΑΣ\$() για αριθμούς
υπάρχει και για αριθμούς ΜΕΓΑΛΟ.ΣΕΙΡΑΣ\$()

Αναγνωριστικό: ΜΕΛΟΣ\$([MEMBER\$(]
? μελος\$(αλφα, 3)
τυπώνει το όνομα του 3ου μέλους της ομάδας αλφα

δες ΟΜΑΔΑ

Αναγνωριστικό: ΜΕΛΟΥΣ.ΤΥΠΟΣ\$([MEMBER.TYPE\$(

Τύπωσε ΜΕΛΟΥΣ.ΤΥΠΟΣ\$(αλφα, 2)

τυπώνει το τύπο του 2ου μέλους από την ομάδα αλφα

δες ΟΜΑΔΑ

Αναγνωριστικό: ΜΕΡΟΣ\$([PIECE\$(

Τύπωσε Μέρος\$("αλφα.βήτα",".", 2) ' δίνει το βήτα

Τύπωσε Μέρος\$("αλφα.βήτα",".")(1) ' δίνει το βήτα

' το Α παίρνει αντικείμενο από την Μέρος\$() αν δεν δώσουμε αριθμό μέρους

' ο πίνακας ξεκινάει από το 0

A=Μέρος\$("αλφα.βήτα",".") ' είναι ίδιο με αυτό: Μέρος\$("αλφα.βήτα",".")()

Τύπωσε Πίνακας\$(A, 1) 'δίνει το βήτα

Τύπωσε A

Πίνακας B\$()

\\ διαφέρει αυτή η εκχώρηση από την A=...

\\ επειδή εδώ γίνεται αντιγραφή του πίνακα από το αριστερό μέρος στο δεξιό μέρος

B\$()=Μέρος\$("αλφα.βήτα",".")

Τύπωσε B\$()

Αναγνωριστικό: ΜΕΣ\$([MID\$(

Δέχεται δυο ή τρία ορίσματα.

Το πρώτο είναι ένα αλφαριθμητικό, το δεύτερο μια τιμή που δηλώνει μια θέση μέσα στο πρώτο από όπου θα μας επιστρέψει ένα νέο αλφαριθμητικό. Το μέγεθος αυτού του νέου θα είναι ανάλογο με το πόσοι χαρακτήρες ακολουθούν από την θέση που ορίσαμε έως το τέλος ή αν δώσουμε ένα τρίτο όρισμα όσο αυτό.

Σε περίπτωση που δεν υπάρχουν χαρακτήρες επειδή δώσαμε θέση μεγαλύτερη από το μέγεθος του αλφαριθμητικού τότε μας επιστρέφει το κενό αλφαριθμητικό.

Σε περίπτωση που ζητήσουμε περισσότερους χαρακτήρες από όσους βρει στο αλφαριθμητικό μας επιστρέφει όσους βρει.

δουλεύει και για αλφαριθμητικά με ένα ψηφίο ανά χαρακτήρα αρκεί να το δηλώσουμε:

Τοπικο 1032

A\$=Γραφη\$("Γιώργος") ' μετατροπή σε ANSI κατά τοπικό1032 (ελληνικά)

Τύπωσε Χαρ\$(Μεσ\$(Α\$,2,2 ως Ψηφίο))="ιώ" ' το Χαρ\$() μετατροπή σε UTF-16LE κατά 1032

Αναγνωριστικό: ΜΙΚΡΟ.ΣΕΙΡΑΣ\$([MIN.DATAS\$(

? μικρο.σειρας\$("αα","ββ")

αα

? μικρο.σειρας\$("κλμν","αα","ββ","βα")

αα

Δες και το ΜΕΓΑΛΟ.ΣΕΙΡΑΣ\$() και ΜΕΓΑΛΟ.ΣΕΙΡΑΣ() για αριθμούς
υπάρχει και για αριθμούς ΜΙΚΡΟ.ΣΕΙΡΑΣ()

Αναγνωριστικό: ΜΙΚΡΟΣ.ΚΑΤΑΛΟΓΟΣ\$([SHORTDIR\$(

? μικρος.καταλογος\$("C:\Documents and Settings\All Users\Documents\")

C:\DOCUME~1\ALLUSE~1\DOCUME~1\

δίνει το όνομα ενός αρχείου ή καταλόγου συμβατό με το DOS.

αν δεν υπάρχει το αρχείο ή ο κατάλογος τότε επιστρέφει ένα κενό αλφαριθμητικό

Αναγνωριστικό: ΜΟΡΦΗ\$([FORMAT\$(

1)

α\$=μορφη\$(αλφαριθμητικό\$, λίστα παραμέτρων)

το αλφαριθμητικό μπορεί να περιέχει σημάνσεις όπως αυτή {0} που δηλώνει την πρώτη παράμετρο στη λίστα. Αν έχουμε παραπάνω σημάνσεις δεν θα αλλάξουν με τιμές. Αν λείπουν σημάνσεις τότε οι αντίστοιχες τιμές θα υπολογιστούν (αν είναι εκφράσεις) και δεν θα χρησιμοποιηθούν. Δεν μπορούμε στη λίστα παραμέτρων να αφήσουμε κενή παράμετρο (δηλαδή συνεχόμενα κόμματα)

τύπωσε μορφη\$("γράμματα {1}", 12323, 45)

μπορούμε να βάλουμε {1:4} για να δηλώσουμε πόσα δεκαδικά θέλουμε σε αριθμούς (αν βάλουμε τη 2η παράμετρο λέξη τότε αυτή θα αναζητεί το {1} και δεν θα το βρει οπότε θα δούμε το {1:4} ως έχει. Δουλεύει και για την εκθετική μορφή π.χ. 1ε10

Προσθήκες, πεδίο τρίτο για το μέγεθος πεδίου, με αρνητικό νούμερο έχουμε κενό χώρο αριστερά, ενώ με θετικό νούμερο έχουμε χώρο δεξιά

Τύπωσε μορφή\$("{0:-20}{1:2:-6},{1:2:6}οκ", "όνομα δεξιά", 10.2)

2) Μετατροπή αλφαριθμητικού με escape κώδικες τύπου json (και javascript)

Αναφορά μορφή\$("Πρώτο\r\nΔεύτερο\r\nΤρίτο")

Αναφορά μορφή\$(μορφή\$("Πρώτο\r\nΔεύτερο {0}\r\nΤρίτο", 100))

Αναγνωριστικό: ΟΔΗΓΟΣ\$([DRIVE\$(

Τύπωσε Οδηγος\$("A:")

Δίνει περιφραστικά το τύπο οδηγού

Αναγνωριστικό: ΟΚΝ\$([LAZY\$(

Η Οκν\$() φτιάχνει μια Ανώνυμη συνάρτηση σε αλφαριθμητικό την οποία συνδέουμε σε μια συνάρτηση και την εκτελούμε, ή την εκτελούμε ως όρισμα αλφαριθμητικό στην Συνάρτηση()

χ=10

ζ=2

Τύπωσε Συνάρτηση(Οκν\$(χ**2+ζ)), Έκφρ("χ**2+ζ")

\\Παράδειγμα χρήσης με πέρασμα μέρους τμήματος σε άλλο τμήμα

καθαρό

ζ=100

Τμήμα Τοπικό_Τμήμα {

 Τύπωσε "αυτό το τμήμα μπορεί να κληθεί μόνο από το πατρικό"

}

Τοπικό_Τμήμα

Τμήμα Γενικό Δοκιμή1 {

 Στατική μ=10

 μ++

 Τύπωσε μ

}

\\ Αυτή δεν είναι μια οποιαδήποτε συνάρτηση!

\\ αλλά αποτελεί μέρος του τμήματος.

\\ τρέχει με το όνομα του τμήματος, και βλέπει ότι και το τμήμα!

\\ εφόσον το καλέσουμε σωστά!

Συνάρτηση Ψεύτικη (&Ανάδραση) {

 ζ++

 Ανάδραση=ζ

 Δοκιμή1

```
Τοπικό_Τμήμα  
}
```

```
Τμήμα Δέλτα { ' θα μπορούσε να γραφεί Τμήμα Δέλτα (&συναρ()) {}  
    Διάβασε &συναρ()  
    v=0  
    Κάλεσε συναρ(&v)  
    Τύπωσε v  
    Κάλεσε συναρ(&v)  
    Τύπωσε v  
    Δοκιμή1  
    Δοκιμή1  
}  
Δοκιμή1  
Δοκιμή1  
\\ Περνάμε τη Ψεύτικη συνάρτηση με τη μετατροπή της συνάρτησης Οκν$()  
Δέλτα Οκν$(&Ψεύτικη())  
Δοκιμή1
```

```
Αναγνωριστικό: ΟΜΑΔΑ$([GROUP$]  
Κλάση Άλφα$ {  
    Ιδιωτικό:  
        Όνομα$  
    Δημόσιο:  
        Αξία {  
            =.Όνομα$  
        }  
        Ιδιότητα Μήκος {Αξία}=0  
Κλάση:  
    Τμήμα alfa {  
        Διάβασε .Όνομα$  
        .[Μήκος]<=Μήκος(.Όνομα$)  
    }  
}
```

```
α$=Άλφα$("Hello")  
Τύπωσε α.Μήκος, α$, Τύπος$(α$), Τύπος$(α)  
β$=Ομάδα$(α$)  
Τύπωσε β.Μήκος, β$, Τύπος$(β$), Τύπος$(β)  
δ$=α$
```

Τύπωσε Τύπος\$(δ\$), δ\$

Αναγνωριστικό: ΟΝΟΜΑ.ΑΡΧΕΙΟΥ\$([FILE.NAME\$(

Μας δίνει το όνομα ενός αρχείου διαγράφοντας το μονοπάτι (τον τόπο εγγραφής)

περιλαμβάνει και τον τύπο

π.χ. αλφα.gsb είναι όνομα με τύπο gsb

Αναγνωριστικό: ΟΝΟΜΑ.ΑΡΧΕΙΟΥ.ΜΟΝΟ\$([FILE.NAME.ONLY\$(

Επιλογή \ \ καθαρίζει τη λίστα επιλογών

Αρχεία + "*" \ \ γεμίζει τη λίστα με ονόματα και τύπο αρχείων από το τρέχον φάκελο

Επιλογή ! \ \ ανοίγει τη λίστα για επιλογή

Αν Επιλογή>0 Τότε Τύπωσε Επιλογή\$(Επιλογή), Όνομα.Αρχείου.Μόνο\$(Επιλογή\$(Επιλογή))

Τύπωσε Όνομα.Αρχείου.Μόνο\$("c:\alfa beta\delta one.klm") \ \ δεν ελέγχει αν υπάρχει το αρχείο, δουλεύει και με διαστήματα

\ \ αφήνει αυτόματα όποια παράμετρος ακολουθεί το όνομα

Τύπωσε Όνομα.Αρχείου.Μόνο\$("c:\alfa beta.bin\delta one.klm -help fileother.txt")

Αναγνωριστικό: ΠΑΡΑΓΡΑΦΟΣ\$([PARAGRAPH\$(

Η παράγραφος\$() δίνει μια παράγραφο και μπορεί ταυτόχρονα να την διαγράψει από το έγγραφο

```
Εγγραφο Άλφα$={Πρώτη Παράγραφος
                  Δεύτερη Παράγραφος
                  Τρίτη Παράγραφος
                  }
```

Για i=3 έως 1 {

Αναφορά Παράγραφος\$(Άλφα\$, i)

}

παλία_παράγραφος\$=Παράγραφος\$(Άλφα\$, 2, -1)

Παρεμβολή στο 2 Άλφα\$=παλία_παράγραφος\$+{

μια άλλη Παράγραφος

και ακόμη μια

}

Κέντρο=2

Αναφορά Κέντρο, Άλφα\$

Αναγνωριστικό: ΠΑΡΑΘΕΣΗ\$([QUOTE\$]
εδώ χρησιμοποιώ το ? για ΤΥΠΩΣΕ

? ΠΑΡΑΘΕΣΗ\$("ΑΛΦΑ")

Τυπώνει το "ΑΛΦΑ"

δηλαδή βάζει εισαγωγικά

το παραθεση\$("") δίνει το "" το οποίο δεν είναι το κενό αλφαριθμητικό αλλά δυο χαρακτήρες εισαγωγικών.

α\$=ΠΑΡΑΘΕΣΗ\$(1,2*100,3+X,4,"κάτι",5,"κάτι άλλο")

βάζει στο αλφαριθμητικό τις τιμές από μια λίστα εκφράσεων, χωρίζοντάς τα με κόμμα και στα αλφαριθμητικά βάζει εισαγωγικά.

ΕΝΘΕΣΗ "ΣΕΙΡΑ "+ΠΑΡΑΘΕΣΗ\$(1,2*100,3+X,4,"κάτι",5,"κάτι άλλο")

η οποία είναι ίδιο με την

ΣΕΙΡΑ 1,2*100,3+X,4,"κάτι",5,"κάτι άλλο"

την παράθεση μπορούμε να την υπολογίσουμε πριν φτάσουμε στο σημείο με την εντολή ΣΕΙΡΑ να φορτώσουμε τον σωρό.

α\$=ΠΑΡΑΘΕΣΗ\$(1,2*100,3+X,4,"κάτι",5,"κάτι άλλο")

...άλλες εντολές....

ΕΝΘΕΣΗ "ΣΕΙΡΑ "+α\$

Αναγνωριστικό: ΠΑΡΑΜ\$([PARAM\$]

Παίρνει μια λίστα σταθερών τιμών και την ενθέτει στο κώδικα. Πρέπει το πρώτο στοιχείο να είναι αλφαριθμητικό, αλλιώς χρησιμοποιούμε την Παραμ()

α\$={10,30,"alfa",40}

α=(Παραμ(α\$))

Τύπωσε α

' Το πρώτο στοιχείο πρέπει να είναι αλφαριθμητικό

α\$={"hello", 10,30,"alfa",40}

α=(Παραμ\$(α\$))

Τύπωσε α

Τύπωσε Παραμ\$(α\$)

Αναγνωριστικό: ΠΕΔΙΟ\$([FIELD\$(

α\$=Πεδίο\$("Περιεχόμενο κείμενο", 40)

Δέχεται δυο ορίσματα ένα αλφαριθμητικό και έναν αριθμό. Εξάγει πάντα ένα αλφαριθμητικό με μήκος όσο ο αριθμός που δώσαμε. Αν το αλφαριθμητικό έχει μήκος μικρότερο από αυτόν τον αριθμό προσθέτονται διαστήματα δεξιά. Αν έχει μεγαλύτερο τότε αποκόπτονται οι πλεονάζοντες χαρακτήρες.

Αναγνωριστικό: ΠΕΖ\$([LCASE\$(

1. Δέχεται σαν όρισμα ένα αλφαριθμητικό και δίνει το ίδιο αλλά με πεζά γράμματα

2. Δεύτερη παράμετρος προαιρετική το τοπικό.

Δείτε το παράδειγμα. Το 1032 είναι για την Ελλάδα, ενώ το 1033 τα αγγλικά.

Κανονικά η λέξη "Γιώργος" έχει 7 χαρακτήρες των 14 συνολικά bytes

τοπικό 1032

α\$=γραφη\$("Γιώργος")

Τύπωσε χαρ\$(α\$) ' μετατροπή βάσει του 1032

Τύπωσε Μήκος(α\$)=3.5 ' μονάδες word=2 bytes, άρα 3.5*2=7bytes

τοπικό 1032

Τύπωσε Πεζ\$(χαρ\$(α\$), 0) ' δίνει λάθος

τοπικό 1033

Τύπωσε Πεζ\$(χαρ\$(α\$), 1032) ' OK

τοπικό 1033

Τύπωσε Πεζ\$(χαρ\$(α\$)) ' δίνει λάθος

τοπικό 1032

Τύπωσε Πεζ\$(χαρ\$(α\$)) ' OK

Αναγνωριστικό: ΠΙΝΑΚΑΣ\$([ARRAY\$(

Πίνακας Α\$(2,5,2)="γεια"

Τύπωσε ΠΙΝΑΚΑΣ\$("Α\$",1,4,1)

Τύπωσε Α\$(1,4,1)

α=(1,2,3,"λέξη")

α+=20

Τύπωσε α

Τύπωσε Πίνακας\$(α, 3)

Αναγνωριστικό: ΡΩΤΑ\$([ASK\$(

ΤΥΠΩΣΕ ΡΩΤΑ\$("ΟΛΑ ΚΑΛΑ")

Ανοίγει ένα παράθυρο μηνυμάτων και περιμένει ένα OK ή ένα Άκυρο.

Στο OK δίνει τη λεζάντα που έχουμε ορίσει για OK δεξ παρακάτω, ή την εξορισμού "ΕΝΤΑΞΕΙ"

Υπάρχει και ως ΡΩΤΑ()

Επιπλέον δέχεται προαιρετικούς παραμέτρους με τη σειρά:

Τίτλος

Λεζάντα για OK (με * στην αρχή είναι το επιλεγμένο, με σκέτο * είναι η προεπιλεγμένη λεζάντα και επιλεγμένο)

Λεζάντα για ΑΚΥΡΟ (με * στην αρχή είναι το επιλεγμένο, με σκέτο * είναι η προεπιλεγμένη λεζάντα και επιλεγμένο, με κενή λεζάντα , δηλαδή με "" δεν εμφανίζεται)

Αλφαριθμητικό για εικόνα, ή μονοπάτι όπου βρίσκεται στο δίσκο μια εικόνα

Αλφαριθμητικό που μετατρέπει το παράθυρο μηνυμάτων σε παραθυρο εισαγωγής (δέχεται μέχρι 100 χαρακτήρες)

Σε περίπτωση που έχουμε εισαγωγή και δώσουμε OK τότε στον σωρό θα υπάρχει το αλφαριθμητικό

Αναγνωριστικό: ΣΥΝΑΡΤΗΣΗ\$([FUNCTION\$(

ΤΥΠΩΣΕ ΣΥΝΑΡΤΗΣΗ\$("ΟΝΟΜΑΣΥΝΑΡΤΗΣΗΣ", 1,4,5)

ΤΥΠΩΣΕ ΟΝΟΜΑΣΥΝΑΡΤΗΣΗΣ\$(1,4,5)

Τύπωσε Συνάρτηση\$("{=Μεσ\$({1234567890},αριθμός,αριθμός})", 3, 4)

Συνάρτηση Άλφα\$(χ){

Αν χ>100 Τότε {

= "Μεγάλο"+γραφή\$(χ)

} Αλλιώς = "Μικρό"+γραφή\$(χ)

}

α\$=&Άλφα\$()

Τύπωσε Συνάρτηση\$(α\$,200), Συνάρτηση\$(α\$,50)

Αναγνωριστικό: ΣΧΔ\$([DRW\$(

Δίνουμε ένα όνομα αρχείου WMF χωρίς κατάληξη και εξάγει αν υπάρχει το αρχείο το πλήρες

όνομά του (με τη διεύθυνσή του και την κατάληξη WMF) αλλιώς παίρνουμε κενό αλφαριθμητικό

```

Αναγνωριστικό: ΣΩΡΟΣ$( [STACK$(]
α$={Παράγραφος 1
    Παράγραφος2
}
\\ Η σωρός$() βάζει τα αλφαριθμητικά με δείκτη μήκους
\\ έτσι μπορεί να πάρει ότι θέλουμε στο αλφριθμητικό!
\\ δεν παίρνει όμως αντικείμενα (χρησιμοποιούμε αντικείμενο σωρό, όπως το α, για να
περάσουμε αντικείμενα)
β$=Σωρός$(10, α$, 300)
Τύπωσε Φάκελος$(β$) ' NSN
α=Σωρός
Σωρός α {
    \\ τροφοδοσία σωρού από αλφαριθμητικό φτιαγμένο από το Σωρός$()
    \\ μπορούμε να πάρουμε μόνο αυτά που θέλουμε
    \\ το β$ θα μειωθεί αυτόματα
    Σωρός β$, "NS"
    \\ ή τα παίρνουμε όλα με Σωρός β$
}

```

```

Αναφορά ΤιμήΣωρού$(α, 2)
Σωρός β$
Διάβασε Κ
Τύπωσε Κ

```

```

Αναγνωριστικό: ΣΩΡΟΥΤΥΠΟΣ$( [STACKTYPE$(]
β=(1,2,3,4) \\ δείκτης σε αντικείμενο mArray
Πίνακας β1(10)=1 \\ αντικείμενο mArray
\\ στο σωρό μπαίνουν δείκτες προς τους πίνακες
α=Σωρός:=1,"hello", β, β1()
Σωρός α {
    \\ η Σωρός αντικείμενο {}
    \\ βάζει το αντικείμενο ως τρέχον σωρό
    \\ και στην έξοδο βάζει τον παλιό στη θέση του!
    Τύπωσε Φάκελος$() ' NSAA πάντα στα αγγλικά
    Τύπωσε Ταύτιση("NSAA") \\ αγγλικά γράμματα
    Τύπωσε Ταύτιση("ΑΓΠΠ") \\ ελληνικά γράμματα
}

```

```

δ=Κάθε(α) \\ Επαναλήπτης
Ενώ δ {

```

```

    Τύπωσε ΣωρούΤύπος$(δ) \ \ Τυπώνει ονόματα Number, String, mArray, mArray
}
Τύπωσε ΣωρούΤύπος$(α, 2)
Τύπωσε ΤιμήΣωρού(α,1), ΤιμήΣωρού(α)
Τύπωσε ΤιμήΣωρού$(α, 2)
Τύπωσε ΤιμήΣωρού(α, 3) ' 1 2 3 4
Τύπωσε Πίνακας(ΤιμήΣωρού(α, 3), 3) ' 4
Τύπωσε ΤιμήΣωρού(α, 4) ' 1 1 1 1 1 1 1 1 1 1

```

```

Αναγνωριστικό: TIMΗΣΩΡΟΥ$( [STACKITEM$(
ΒΑΛΕ "ΑΛΦΑ","ΒΗΤΑ"
ΤΥΠΩΣΕ TIMΗΣΩΡΟΥ$( ), TIMΗΣΩΡΟΥ$(2)
ΒΗΤΑ      ΑΛΦΑ

```

αν δεν υπάρχει στην θέση τιμή τότε παίρνουμε λάθος
αν δεν υπάρχει αλφαριθμητικό - λέξη- στη θέση που ζητάμε τότε παίρνουμε λάθος

Δουλεύει και για αντικείμενο σωρού

```

α=Σωρός
Σωρός α {
    Σειρά "άλφα", "βήτα"
}
Τύπωσε τιμήσωρού$(α), τιμήσωρού$(α,2)
\ \ old Σωρός deleted an α show new one
α=Σωρός:="άλφα1", "βήτα1"
Τύπωσε τιμήσωρού$(α), τιμήσωρού$(α,2)
Άδειασε \ \ Αδειάζει τον τρέχον σωρό
Σειρά "αλφα2", "βήτα2"
α=[] \ \ το α δείχνει παίρνει τον δείκτη του τρέχοντα σωρού και ο τρέχον σωρός παίρνει νέο
αντικείμενο
Τύπωσε Κενό \ \ -1 σημαίνει αληθές
Τύπωσε τιμήσωρού$(α), τιμήσωρού$(α,2)
Σειρά "ένα ακόμα" \ \ στον τρέχον σωρό
Σωρός α \ \ τώρα το α είναι κενό και ο τρέχον σωρός πήρε τα στοιχεία του α στο τέλος
Τύπωσε Μήκος(α) ' 0
Τύπωσε Μέγεθος.Σωρού ' 3
Σωρός \ \ μας δείχνει τα στοιχεία του στη κονσόλα
Διάβασε α1$, α2$, α3$
Τύπωσε Κενό \ \ -1 σημαίνει αληθές
α=Σωρός:=α1$, α2$, α3$

```

Τύπωσε Μήκος(α) ' 3

Αναγνωριστικό: ΤΙΤΛΟΣ\$([TITLE\$(
Τύπωσε τίτλος\$("γιώργος καρράς")
Γιώργος Καρράς

Αναγνωριστικό: ΤΙΤΛΟΣ.ΑΡΧΕΙΟΥ\$([FILE.TITLE\$(
Δίνει το τίτλο αρχείου, όπως φαίνεται στον explorer (το παράθυρο που βλέπουμε τα αρχεία ενός φακέλου)
Τύπωσε Τίτλος.Αρχείου\$("TXT")

Αναγνωριστικό: ΤΟΠΙΚΟ\$([LOCALE\$(
Φόρμα 60, 30
Διπλά
Τύπωσε "Example 1.0"
Κανονικά
Κύλιση Χώρισμα 2 ' ή Οθόνη ,2 για να καθαρίσει και το κάτω μέρος
Τοπικό 1033 ' για ελληνικά 1032
Για i=0 έως 255 {
 Αν Τοπικό\$(i)<>"" τότε {
 \\ Η Αναφορά σταματάει αυτόματα σε κάθε 3/4 των γραμμών που εμφανίζει στο κάτω μέρος
 \\ όπου έχουμε κύλιση οθόνης
 Αναφορά Μορφή\$(" {0:-3}. Τοπικό\$(0x{1})={2}", i, Δεκαεξ\$(i,1),Παράθεση\$(Τοπικό\$(i)))
 }
}

Αναγνωριστικό: ΤΟΠΟΣ\$([PATH\$(
\\ δίνει τα "μονοπάτια" ή "τόπο" στο δίσκο βάσει αρίθμησης, ειδικών φακέλων των Windows
Γενική Σταθερή CSIDL_ADMINTOOLS = 0x30
Γενική Σταθερή CSIDL_ALTSTARTUP = 0x1D
Γενική Σταθερή CSIDL_APPDATA = 0x1A
Γενική Σταθερή CSIDL_BITBUCKET = 0xA
Γενική Σταθερή CSIDL_COMMON_ADMINTOOLS = 0x2F
Γενική Σταθερή CSIDL_COMMON_ALTSTARTUP = 0x1E

Γενική Σταθερή CSIDL_COMMON_APPDATA = 0x23
Γενική Σταθερή CSIDL_COMMON_DESKTOPDIRECTORY = 0x19
Γενική Σταθερή CSIDL_COMMON_DOCUMENTS = 0x2E
Γενική Σταθερή CSIDL_COMMON_FAVORITES = 0x1F
Γενική Σταθερή CSIDL_COMMON_PROGRAMS = 0x17
Γενική Σταθερή CSIDL_COMMON_STARTMENU = 0x16
Γενική Σταθερή CSIDL_COMMON_STARTUP = 0x18
Γενική Σταθερή CSIDL_COMMON_TEMPLATES = 0x2D
Γενική Σταθερή CSIDL_CONNECTIONS = 0x31
Γενική Σταθερή CSIDL_CONTROLS = 0x3
Γενική Σταθερή CSIDL_COOKIES = 0x21
Γενική Σταθερή CSIDL_DESKTOP = 0x0
Γενική Σταθερή CSIDL_DESKTOPDIRECTORY = 0x10
Γενική Σταθερή CSIDL_DRIVES = 0x11
Γενική Σταθερή CSIDL_FAVORITES = 0x6
Γενική Σταθερή CSIDL_FONTS = 0x14
Γενική Σταθερή CSIDL_HISTORY = 0x22
Γενική Σταθερή CSIDL_INTERNET = 0x1
Γενική Σταθερή CSIDL_INTERNET_CACHE = 0x20
Γενική Σταθερή CSIDL_LOCAL_APPDATA = 0x1C
Γενική Σταθερή CSIDL_MYPICTURES = 0x27
Γενική Σταθερή CSIDL_NETHOOD = 0x13
Γενική Σταθερή CSIDL_NETWORK = 0x12
Γενική Σταθερή CSIDL_PERSONAL = 0x5 ' My Documents
Γενική Σταθερή CSIDL_MY_DOCUMENTS = 0x5
Γενική Σταθερή CSIDL_PRINTERS = 0x4
Γενική Σταθερή CSIDL_PRINTHOOD = 0x1B
Γενική Σταθερή CSIDL_PROFILE = 0x28
Γενική Σταθερή CSIDL_PROGRAM_FILES = 0x26
Γενική Σταθερή CSIDL_PROGRAM_FILES_COMMON = 0x2B
Γενική Σταθερή CSIDL_PROGRAM_FILES_COMMONX86 = 0x2C
Γενική Σταθερή CSIDL_PROGRAM_FILESX86 = 0x2A
Γενική Σταθερή CSIDL_PROGRAMS = 0x2
Γενική Σταθερή CSIDL_RECENT = 0x8
Γενική Σταθερή CSIDL_SENDTO = 0x9
Γενική Σταθερή CSIDL_STARTMENU = 0xB
Γενική Σταθερή CSIDL_STARTUP = 0x7
Γενική Σταθερή CSIDL_SYSTEM = 0x25
Γενική Σταθερή CSIDL_SYSTEMX86 = 0x29
Γενική Σταθερή CSIDL_TEMPLATES = 0x15
Γενική Σταθερή CSIDL_WINDOWS = 0x24

Λίστα

Τύπος\$(CSIDL_PROGRAM_FILES)

Αναγνωριστικό: ΤΟΠΟΣ.ΑΡΧΕΙΟΥ\$([FILE.PATH\$(]
μας δίνει το μονοπάτι ενός αρχείου (τον τόπο που θα γράψουμε ή που θα ανοίξουμε ένα αρχείο)
Περιλαμβάνει τον οδηγό (π.χ. το τμήμα C: του σκληρού δίσκου) και όλη τη σειρά φακέλων και υποφακέλων. Δεν περιλαμβάνει το όνομα του αρχείου.

Αναγνωριστικό: ΤΥΠΟΣ\$([TYPE\$(]
α=10
β=(1,2,3,4)
Τύπωσε Τύπος\$(α), Τύπος\$(β) ' double mArray

Αναγνωριστικό: ΤΥΠΟΣ.ΑΡΧΕΙΟΥ\$([FILE.TYPE\$(]
Δίνει τον τύπο ενός αρχείου, αφαιρώντας τόπο (μονοπάτι) και όνομα

Αναγνωριστικό: ΦΑΚΕΛΟΣ\$([ENVELOPE\$(]
1) Τύπωσε Φακελος\$(
τυπώνει ένα αλφαριθμητικό με τον τύπο των στοιχείων του σωρού "εργασίας", του τρέχοντος, δηλαδή του σωρού που είναι κοινός στα τμήματα, ενώ στις συναρτήσεις χρήστη είναι ιδιωτικός.
2) Τύπωσε Φακελος\$(α\$)
τυπώνει το φάκελο του σωρού α\$ (που έχει δημιουργηθεί με την Σωρός\$())

Τμήμα A {
α\$=Σωρος\$(1,3,"ονομα",4,"περιοχη",6)
Τύπωσε "NNSNSN"
Τύπωσε Φακελος\$(α\$)
}

Διαβάζουμε με την ΦΑΚΕΛΟΣ\$() τι περιλαμβάνει η μεταβλητή α\$ όπου έχουμε δώσει μια σειρά δεδομένων ως σωρός.

Τα γράμματα που επιστρέφει είναι στα αγγλικά (ενώ δέχεται στα ελληνικά)

Το F είναι για την συνάρτηση Λάμδα [N]umber, [G]roup, [E]vent, [B]uffer, [I]nventory, [A]rray, [F]unction, [S]tring, Sta[C]k

(οι συναρτήσεις που μπαίνουν με αναφορά - εκτός λάμδα- έχουν τύπο αλφαριθμητικού και γενικά ότι μπαίνει με αναφορά είναι αλφαριθμητικοί και η μετατροπή γίνεται στην Διάβασε)

A=10

Βάλε &A
Τύπωσε Φάκελος\$()
Διάβασε &B
B++
Τύπωσε A

Για να δούμε τι έχει ένα αντικείμενο σωρός πρέπει να το κάνουμε πρόσκαιρα τρέχον σωρό
α=Σωρός:=1,2,3,4
Σωρός α {
 Τύπωσε Φάκελος\$()
}

Αναγνωριστικό: ΦΑΝΕΡΟΣ\$([SHOW\$(
Μετατρέπει ένα κρυπτογραφημένο αλφαριθμητικό σε ένα αλφαριθμητικό (μισού μήκους).
Δέχεται για όρισμα το προς αποκρυπτογράφηση αλφαριθμητικό, ένα αλφαριθμητικό για κωδικό
καθώς και έναν αριθμό που επίσης διαφοροποιεί τον κωδικό.
δες ΚΡΥΦΟΣ\$()

Αναγνωριστικό: ΦΙΛΤΡΟΣ\$([FILTER\$(
? ΦΙΛΤΡΟΣ\$("ΑΒΓΔΕΖ","ΒΓΖ")
ΑΔΕ

καθαρίζει το πρώτο αλφαριθμητικό από τη λίστα χαρακτήρων στο δεύτερο αλφαριθμητικό.
δες την ΑΛΛΑΓΗ\$(
το παράδειγμα παραπάνω ως σειρές της ΑΛΛΑΓΗ\$(
α\$="ΑΒΓΔΕΖ"
α\$=ΑΛΛΑΓΗ\$(α\$, "Β")
α\$=ΑΛΛΑΓΗ\$(α\$, "Γ")
α\$=ΑΛΛΑΓΗ\$(α\$, "Ζ")

Αναγνωριστικό: ΦΩΤΟΣ\$([JPG\$(
Δίνουμε ένα όνομα αρχείου JPG χωρίς κατάληξη και εξάγει αν υπάρχει το αρχείο το πλήρες
όνομά του (με τη διεύθυνσή του και την κατάληξη JPG)

Αναγνωριστικό: ΧΑΡ\$([CHR\$(

Επιστρέφει έναν χαρακτήρα (ως αλφαριθμητικό) βάσει της αριθμητικής τιμής που δίνουμε ως όρισμα. Αντίθετη εντολή της ΚΩΔ(
ΤΥΠΩΣΕ ΧΑΡ\$(ΚΩΔ("λ"))
δες και το ΧΑΡΚΩΔ() και τα ΧΑΡΚΩΔ\$()

α\$=Γραφή\$("Hello")
\\ το α\$ γράφεται σαν Ansi αλφαριθμητικό, ένα byte ανά χαρακτήρα.
Τύπωσε Μήκος(α\$)=2.5
Τύπωσε "Bytes:"=Μήκος(α\$)*2 ' 5 bytes
Τύπωσε Μήκος(χαρ\$(α\$))=5
Τύπωσε α\$

Με αριθμό θα δώσει το ANSI αλφαριθμητικό για τιμές από 0 έως 255 ως Utf16LE χαρακτήρας, στην 1033 κωδικοσελίδα
? χαρ\$(65)

Με αριθμό και αριθμό θα δώσει το ANSI αλφαριθμητικό για τιμές από 0 έως 255 ως Utf16LE χαρακτήρας, στην κωδικοσελίδα που δίνουμε.
? χαρ\$(195, 1032)

Με αλφαριθμητικό και νούμερο, κάνει μετατροπές, εφόσον όμως δίνουμε με κωδικούς χαρακτήρων από 0 έως 255:
Δείτε πως θα το κάνουμε εδώ:

Τοπικό 1032
' παράγουμε ANSI, 1 byte ανά χαρακτήρα κατά 1032 (τοπικό ελλάδος)
' θα πάρουμε το μισό μήκος αλφαριθμητικό, αφού εξ ορισμού έχουμε 2 byte στο UTF16LE, που δουλεύει η M2000.
α\$=γραφή\$("ΓΙΩΡΓΟΣ ΚΑΡΡΑΣ")
' Αλλάζουμε σε τοπικό Δυτικής Ευρώπης
Τοπικό 1033
' Τώρα θα πάρουμε σε UTF16LE αλλά η αντικατάσταση κωδικών έγινε με τοπικό 1033!
' όμως για κάθε χαρακτήρα ο κωδικός θα είναι ίδιος με το κωδικό σε ANSI.
' Δεν έχουμε αλλαγή κωδικών. Διπλασιάζεται το μήκος
β\$=χαρ\$(α\$)
' Τώρα θα μετατρέψουμε τους κωδικούς (από 1033 σε 1032)
' αυτό κάνει η χαρ\$() όταν δίνουμε τοπικό, αλλά πρέπει αυτό που δίνουμε να είναι:
' με κωδικοποίηση UTF16LE και όλοι οι κωδικοί από 0 έως 255 (όπως κατά ANSI)
Τύπωσε χαρ\$(β\$, 1032)
\\ θα πάρουμε ΓΙΩΡΓΟΣ ΚΑΡΡΑΣ

Αναγνωριστικό: ΧΑΡΚΩΔ\$ ([CHRCODE\$(
Τύπωσε Χαρκωδ\$(0x0645)
επιστρέφει τον Unicode χαρακτήρα των 16bit
ή διπλό χαρακτήρα 2*16bit για χαρακτήρες με αριθμό > 0x10000

Αναγνωριστικό: ΧΡΟΝΟΣ\$([TIME\$(
Δέχεται όρισμα μια αριθμητική παράσταση ώρας (δεκαδικός αριθμός όπου τα δεκαδικά φανερώνουν κλάσμα της ημέρας) και δίνει την ώρα ως αλφαριθμητικό με ώρα και λεπτά.
Ω=ΤΩΡΑ
ΤΥΠΩΣΕ ΧΡΟΝΟΣ\$(Ω), ΓΡΑΦΗ\$(Ω, "hh:nn")
Τύπωσε χρόνος\$(Τώρα,)
Τύπωσε χρόνος\$(Τώρα,1033)
Τύπωσε πεζ\$(χρόνος\$(Τώρα,1033, "hh:mm:ss tt"))
Τύπωσε πεζ\$(χρόνος\$(Τώρα,1032, "hh:mm:ss tt"))
Τύπωσε πεζ\$(χρόνος\$(Τώρα,1033, "Long Time"))
Τύπωσε πεζ\$(χρόνος\$(Τώρα,1033, "Short Time"))
Τύπωσε Χρόνος\$() 'εμφανίζει τη ρυθμισμένη από το σύστημα ζώνη ώρας
Τύπωσε Χρόνος\$(Χρόνος("UTC")) ' εμφανίζει την τρέχουσα ώρα σε UTC (περιλαμβάνει και την
διαφορά θερινής ανάλογα την περίοδο)

Δες Χρόνος()

Ομάδα: ΜΕΤΑΒΛΗΤΕΣ ΣΥΣΤΗΜΑΤΟΣ - VARS READ ONLY

Αναγνωριστικό: ΑΝΑΛΟΓΙΟ\$ [BROWSER\$]
ΤΥΠΩΣΕ ΑΝΑΛΟΓΙΟ\$
Δίνει τον τίτλο της σελίδας που εμφανίζουμε στο ΑΝΑΛΟΓΙΟ

Αναγνωριστικό: ΑΥΤΟ [THIS]

1. το ΑΥΤΟ δηλώνει την ομάδα

Ομάδα Αλφα {
 X=10
 Τμήμα Βήτα {
 Τύπωσε Αυτό.X, X
 }
}

Αλφα.Βήτα

2. Αυτό στο Για Αυτό {}

Η Για Αυτό {} χρησιμοποιείται προσωρινούς ορισμούς.

Αναγνωριστικό: GRAMMA\$ [LETTER\$]

α\$=γραμμα\$

Δίνει το αλφαριθμητικό που βρίσκεται στην κορυφή του σωρού και το αφαιρεί. Αν ο ΣΩΡΟΣ δεν έχει τιμές ή έχει στην κορυφή μια αριθμητική τιμή τότε βγαίνει λάθος. Ελέγχουμε με τις KENO, EINΓΡ και EINAP.

νέες εντολές είναι η ΦΑΚΕΛΟΣ\$() και η ΜΕΓΕΘΟΣ.ΣΩΡΟΥ

Αναγνωριστικό: GRAMMATOSEIPA\$ [FONTNAME\$]

? GRAMMATOSEIPA\$

ΤΥΠΩΝΕΙ ΤΟ ΟΝΟΜΑ ΤΗΣ GRAMMATOSEIPAS ΠΟΥ ΚΑΝΟΥΜΕ ΧΡΗΣΗ

μπορούμε να αλλάζουμε το όνομα της γραμματοσειράς με την εντολή GRAMMATOSEIPA

κάθε επίπεδο όπως και το φόντο μπορεί να έχει τη δική του ξεχωριστή γραμματοσειρά.

Αναγνωριστικό: GRAMMESANAFORAS [REPORTLINES]

Επιστρέφει τον αριθμό γραμμών που μπορεί να τυπώσει η ΑΝΑΦΟΡΑ στη δοκιμή που κάναμε σε προηγούμενη χρήση της με αρνητικό νούμερο στον αριθμό γραμμών (δεν τυπώνει)

Δες την αναφορά. Δες το παράδειγμα εδώ: REPORTLINES

Αναγνωριστικό: GRAMMH [ROW]

Δίνει την θέση Υ του δρομέα χαρακτήρων

Δρομέας 2,2 : Τύπωσε Γραμμή

θέση και Γραμμή δίνουν το σημείο που βρίσκεται ο δρομέας χαρακτήρων.

Ο δρομέας μπορεί να αλλάζει κατά ΣΤΗΛΗ (δίνει το πλάτος στήλης).

Αναγνωριστικό: ΔΕΙΚΤΗΣ [MOUSE]

ΤΥΠΩΣΕ ΔΕΙΚΤΗΣ, ΔΕΙΚΤΗΣ.X, ΔΕΙΚΤΗΣ.Y

Η ΔΕΙΚΤΗΣ μας δίνει 1 αν πατάμε το αριστερό πλήκτρο του ποντικιού και 2 το δεξί, επίσης 3 αν πατηθούν και τα δυο μαζί ταυτόχρονα!

Υπάρχει και το Πατημένο(1) και Πατημένο(2) που διαβάζουν άμεσα αν τα πλήκτρα είναι πατημένα και δίνουν -1

Από την έκτη έκδοση προστέθηκαν ακόμα δυο τιμές για τα X και Y του δείκτη (ποντικιού). Αυτές λέγονται απόλυτες τιμές. Είναι οι τιμές σε σχέση με το περιθώριο. Όταν διαβάζουμε την Δεικτης.X σε ένα επίπεδο έστω το 2 τότε αυτό που παίρνουμε είναι η τιμή X του δείκτη σε σχέση με την αριστερή του πλευρά (σε TWIPS , μονάδα μέτρησης διάστασης οθόνης). Αν διαβάσουμε την ΔεικτηςA.X τότε θα διαβάσουμε την τιμή X του Δείκτη σε σχέση με την αριστερή πλευρά του περιθωρίου.

Περί twips. Στα windows τα pixels διαφέρουν από τα twips κατά μια αναλογία. Δηλαδή συγκεκριμένα pixels, εικονοστοιχεία, αντιστοιχούν σε συγκεκριμένα twips. Ουσιαστικά αυτό που έκαναν οι προγραμματιστές των windows ήταν να δώσουν μια νέα κλίμακα που να έχει σχέση με την φυσική μονάδα, τις ίντσες. Έτσι αν για παράδειγμα έχουμε μια οθόνη με 96 σημεία στην ίντσα, σημαίνει ότι έχουμε οθονη 96 dpi, ή κάθε εικονοστοιχείο θα αντιστοιχεί σε 1/96 της ίντσας. Επειδή πάλι δεν μπορούσαν να φορτώσουν το σύστημα με αριθμούς με δεκαδικά (κινητής υποδιαστολής) σκέφθηκαν να βάλουν ένα μεγάλο ακέραιο που να αντιστοιχεί σε μια ίντσα ώστε κάθε εικονοστοιχείο να αντιστοιχεί σε ένα ακέραιο αριθμό. Ο μαγικός αριθμός εδώ είναι το 1440. Κάθε εικονοστοιχείο των windows στα 96dpi είναι 15 twips. Έτσι 1440 twips είναι μια ίντσα. Μια υποθετική ίντσα, γιατί σε ορισμένα μονιτορ μπορούμε να επιλέξουμε διάφορες αναλύσεις....και τα Windows δεν θα αλλάξουν τα dpi, ενώ ας πούμε το πλάτος της οθόνης θα παραμένει το ίδιο σε διαφορετικές αναλύσεις. Ωστόσο σε όλα τα συστήματα υπάρχει τρόπος να ρυθμίσουμε τα dpi εμφανίζοντας έναν χάρακα στην οθόνη και ταιριάζοντάς τον με έναν εξωτερικό χάρακα να έχουμε για 1440 twips ακριβώς μια ίντσα. Όμως αυτό δεν το κάνει κανείς γιατί τα περισσότερα προγράμματα θα έχουν πρόβλημα σε διαφορετικά dpi (επειδή δεν σχεδιάστηκαν για τέτοιες αλλαγές).

Μπορούμε να έχουμε σε έναν εκτυπωτή μια ίντσα επί μια ίντσα ή 1440 twips επί 1440 twips (και να βγαίνει επακριβώς αυτό το μέγεθος, γιατί για τους εκτυπωτές αυτό συμβαίνει), αλλά να έχουμε διαφορετικά dpi στα οριζόντια σημεία από τα κάθετα. Δηλαδή κάθε σημείο να μην είναι τετράγωνο αλλά παραλληλόγραμμο! Αν έπρεπε να υπολογίζουμε σε εικονοστοιχεία τότε θα έπρεπε να υπολογίζουμε, να κρατάμε λογαριασμό, την αναλογία ύψους προς πλάτος του σημείου εκτύπωσης, δηλαδή να πολλαπλασιάζουμε, μια χρονοβόρα διαδικασία. Αντί γι' αυτό λοιπόν τα twips λειτουργούν με ακέραιους (άρα γρήγορα) και αφήνουν τους οδηγούς εκτύπωσης (ή το περιβάλλον όπως εδώ της M2000) να κάνουν τους τελικούς μετασχηματισμούς.

Τα παραλληλόγραμμα εικονοστοιχεία δεν είναι νεοτερισμός αλλά μια πάγια τακτική στην τηλεόραση. Δείτε το ΠΑΡΑΔΕΙΓΜΑ1

Δίνει τον αριθμό του πλήκτρου του ποντικιού όταν πατηθεί, αλλά δεν θα το δώσει και άλλη φορά ενώ συνεχίζουμε να το κρατάμε πατημένο

Δες την Δείκτης και Πατημένο()

Αναγνωριστικό: ΔΕΙΚΤΗΣ.Y [MOUSE.Y]
Μας δίνει την θέση Y του δείκτη πάνω στην ΟΘΟΝΗ

Αναγνωριστικό: ΔΕΙΚΤΗΣ.X [MOUSE.X]
Μας δίνει την θέση X του δείκτη πάνω στην ΟΘΟΝΗ

Αναγνωριστικό: ΔΕΙΚΤΗΣΑ.Y [MOUSEA.Y]
Δίνει την απόλυτη τιμή Y του δείκτη (ποντικιού) σε Twips, με αρχή την πάνω πλευρά του περιθωρίου.
Αν βρισκόμαστε σε μια ομάδα εντολών στο περιθώριο με την ΠΕΡΙΘΩΡΙΟ τότε ο Δεικτης.α ταυτίζεται με τον Δεικτης.υ

Αναγνωριστικό: ΔΕΙΚΤΗΣΑ.X [MOUSEA.X]
Δίνει την απόλυτη τιμή X του δείκτη (ποντικιού) σε Twips, με αρχή την αριστερή πλευρά του περιθωρίου.
Αν βρισκόμαστε σε μια ομάδα εντολών στο περιθώριο με την ΠΕΡΙΘΩΡΙΟ τότε ο Δεικτης.α ταυτίζεται με τον Δεικτης.χ

Αναγνωριστικό: ΔΙΑΔΙΚΤΥΟ [INTERNET]
Τύπωσε Διαδίκτυο
Τυπώνει Αληθής αν ο υπολογιστής έχει σύνδεση με το διαδίκτυο

Αναγνωριστικό: ΔΙΑΔΙΚΤΥΟ\$ [INTERNET\$]
Τύπωσε Διαδίκτυο\$
127.0.0.1

ή την δημόσια ip αν έχουμε σύνδεση

Αναγνωριστικό: ΔΙΑΡΚΕΙΑ [DURATION]

Τυπωσε ΔΙΑΡΚΕΙΑ

Μας δίνει την διάρκεια της ΤΑΙΝΙΑΣ ή της ΜΟΥΣΙΚΗΣ που εκτελείται.

Αναγνωριστικό: ΔΙΑΦΑΝΕΙΑ\$ [SPRITE\$]

Επιστρέφει αλφαριθμητικό με περιεχόμενο εικόνα από μία προηγούμενη εντολή ΔΙΑΦΑΝΕΙΑ. Η εικόνα αυτή είναι το τμήμα της οθόνης πριν αλλαχθεί από την ΔΙΑΦΑΝΕΙΑ.

Παράδειγμα (έχει προηγηθεί μια εντολή ΔΙΑΦΑΝΕΙΑ)

A\$=ΔΙΑΦΑΝΕΙΑ\$

B\$=ΔΙΑΦΑΝΕΙΑ\$

η A\$ περιέχει εικόνα η B\$ όμως δεν περιέχει εικόνα διότι η ΔΙΑΦΑΝΕΙΑ\$ δίνει μία φορά μόνο την εικόνα και αποδεσμεύει την μνήμη που βάσταγε την εικόνα.

Αναγνωριστικό: ΔΙΚΤΥΟ\$ [LAN\$]

? ΔΙΚΤΥΟ\$

επιστρέφει την IP του δικτύου μας.

Αναγνωριστικό: ΕΙΝΑΡ [ISNUM]

Αν υπάρχει αριθμός στην κορυφή του σωρού τότε δίνει ΑΛΗΘΕΣ

Αναγνωριστικό: ΕΙΝΓΡ [ISLET]

Αν υπάρχει λέξη στην κορυφή του σωρού τότε δίνει ΑΛΗΘΕΣ

(δεν βγαίνει λάθος αν)

Αναγνωριστικό: ΕΚΤΥΠΩΤΗΣ\$ [PRINTERNAME\$]

? ΕΚΤΥΠΩΤΗΣ\$

όνομα εκτυπωτή (και πόρτα)

Αναγνωριστικό: ΕΛ OSBIT [OSBIT]
Τύπωσε OSBIT
τυπώνει 32 ή 64 ανάλογα τι είναι τι είδους λειτουργικό σύστημα έχουμε
Δες IsWine, ΛΣ\$

Αναγνωριστικό: ΕΛΕΓΧΟΣ.ΜΕΓΕΘΟΣ.ΣΩΡΟΥ [MONITOR.STACK.SIZE]
Τύπωσε ΕΛΕΓΧΟΣ.ΜΕΓΕΘΟΣ.ΣΩΡΟΥ
Τυπώνει το μέγεθος του σωρού επιστροφής, που μπορούμε να χρησιμοποιήσουμε.
Τύπωσε ΕΛΕΓΧΟΣ.ΜΕΓΕΘΟΣ.ΣΩΡΟΥ-ΕΛΕΓΧΟΣ.ΣΩΡΟΥ
Τυπώνει το μέγεθος του ελεύθερου υπόλοιπου του σωρού.

Αναγνωριστικό: ΕΛΕΓΧΟΣ.ΣΩΡΟΥ [MONITOR.STACK]
Τύπωσε ΕΛΕΓΧΟΣ.ΣΩΡΟΥ
δίνει το μέγεθος του σωρού επιστροφής του διερμηνευτή σε χρήση
Οι μεταβλητές/πίνακες της M2000 δεν αυξάνουν τον σωρό.
Οι κλήσεις των ρουτινών επίσης δεν αυξάνουν τον σωρό. Δες Όριο.Αναδρομής
Οι κλήσεις συναρτήσεων και τμημάτων αυξάνουν τον σωρό
Οι παραστάσεις με τελεστές αυξάνουν τον σωρό
Δες ΕΛΕΓΧΟΣ.ΜΕΓΕΘΟΣ.ΣΩΡΟΥ και ΕΛΕΓΧΟΣ

Αναγνωριστικό: ΕΛΛΗΝΙΚΑ_μεταβλητή [GREEK_variable]
Τύπωσε ΕΛΛΗΝΙΚΑ
γυρνάει αληθής αν έχουμε μηνύματα διαλόγων στα ελληνικά
με Θέσε Διακόπτες "+GREEK" κάνουμε το ΕΛΛΗΝΙΚΑ αληθής
με Θέσε Διακόπτες "+GREEK" κάνουμε το ΕΛΛΗΝΙΚΑ ψευδής

Και η εντολή ΕΛΛΗΝΙΚΑ θέτει το ΕΛΛΗΝΙΚΑ αληθής αλλά θέτει και το Τοπικό στο 1032
Το ΛΑΤΙΝΙΚΑ κάνει το ανάποδο από το ΕΛΛΗΝΙΚΑ, οπότε θέτει την μεταβλητή ΕΛΛΗΝΙΚΑ
ψευδής και το τοπικό στο 1033

Αναγνωριστικό: ΕΝΑΟΝΟΜΑ\$ [TEMPNAME\$]
α\$=εναονομα\$
δημιουργούμε ένα νέο όνομα αρχείου για προσωρινή χρήση με κατάληξη .tmp
Μετά θα δημιουργήσουμε το αρχείο (θα είναι στο φάκελο των προσωρινών αρχείων). Μάλιστα

αυτό το όνομα είναι μοναδικό, οπότε κανένα άλλο πρόγραμμα δεν θα τύχει να δημιουργήσει το ίδιο. (και αυτή είναι η σκοπιμότητα της μεταβλητής μόνο ανάγνωσης)

Αναγνωριστικό: ENKOM\$ [INKEY\$]

Επιστρέφει έναν χαρακτήρα που έχουμε μόλις πληκτρολογήσει ή το κενό αλφαριθμητικό. Δεν περιμένει να πατηθεί κάποιο πλήκτρο (κομβίο), αλλά αν πατηθεί την στιγμή που τρέχει η εντολή τότε δίνει τον χαρακτήρα ως αλφαριθμητικό

Αναγνωριστικό: ΕΝΤΑΣΗ_μεταβλητή [VOLUME_as variable]

Δίνει την τρέχουσα ένταση. ΔΕΣ ΕΝΤΑΣΗ

Αναγνωριστικό: ΕΝΤΟΛΗ\$ [COMMAND\$]

ΤΥΠΩΣΕ ΕΝΤΟΛΗ\$

δίνει όλο το μονοπάτι μαζί με το όνομα αρχείου του τελευταίου φορτώματος

ΦΟΡΤΩΣΕ "αλφα"

ΣΩΣΕ "αλφα"

ή

ΣΩΣΕ ΕΝΤΟΛΗ\$

απλά το ΕΝΤΟΛΗ\$ έχει το φάκελο (μονοπάτι), το όνομα και τον τύπο (GSB) του αρχείου. Επειδή φορτώσαμε έστω το αλφα και θέλουμε να το σώσουμε μας ρωτά το σύστημα αν θέλουμε να γράψουμε πάνω στο προηγούμενο.

η εντολή αυτή ΣΩΣΕ ΕΝΤΟΛΗ\$ δίνεται και με CTRL+A

Αναγνωριστικό: ΕΠΙΛΟΓΕΣ [MENUITEMS]

Μας δίνει τον αριθμό των επιλογών που έχουμε ορίσει. ΔΕΣ ΔΕΙΞΕ, ΕΠΙΛΟΓΗ

Αναγνωριστικό: ΕΠΙΛΟΓΕΣ.ΦΑΝΕΡΕΣ [MENU.VISIBLE]

ΤΥΠΩΣΕ ΕΠΙΛΟΓΕΣ.ΦΑΝΕΡΕΣ

' Παράδειγμα με χρήση νήματος

```
Φ=1
ΝΗΜΑ {
    Φ++
    ΑΝ ΕΠΙΛΟΓΕΣ.ΦΑΝΕΡΕΣ ΤΟΤΕ ΤΥΠΩΣΕ Φ, ΕΠΙΛΟΓΗ$(ΕΠΙΛΟΓΗ)
} ΩΣ Ω
ΕΠΙΛΟΓΗ ΤΙΤΛΟΣ "Ο ΤΙΤΛΟΣ ΜΟΥ"
ΜΕΤΑ 100 {
    ΝΗΜΑ Ω ΚΑΘΕ 100
}
ΕΠΙΛΟΓΗ "1η Γραμμή", "2η Γραμμή"
ΤΥΠΩΣΕ ΕΠΙΛΟΓΗ
```

Αναγνωριστικό: ΕΠΙΛΟΓΗ_μεταβλητή [MENU_as variable]
Μας δίνει τον αριθμό της επιλογής μετά από μια ΔΕΙΞΕ ή ΕΠΙΛΟΓΗ.
Με 0 σημαίνει ότι δεν έγινε καμιά επιλογή.

Αναγνωριστικό: ΕΦΑΡΜΟΓΗ.ΚΑΤ\$ [APPRDIR\$]
μας δίνει τον κατάλογο της εφαρμογής, αν είναι όμως σωστά εγκατεστημένη αλλιώς είναι κενό
ΤΥΠΩΣΕ ΕΦΑΡΜΟΓΗ.ΚΑΤ\$

Αναγνωριστικό: ΗΧΟΓΡΑΦΗΣΗΣ.ΕΠΙΠΕΔΟ [SOUNDREC.LEVEL]
ΗΧΟΓΡΑΦΗΣΗ ΝΕΑ
ΑΝΑΛΥΤΗΣ
ΕΠΑΝΕΛΑΒΕ
 ΤΥΠΩΣΕ ΠΑΝΩ \$(1), ΕΠΑΝ\$("[", ΗΧΟΓΡΑΦΗΣΗΣ.ΕΠΙΠΕΔΟ)
 ΤΥΠΩΣΕ ΜΕΡΟΣ \$(3, ΠΛΑΤΟΣ), ΦΟΡΤΟΣ ΔΙΑ 1000
 ΑΝΑΝΕΩΣΗ 100
 ΑΝ ΠΑΤΗΜΕΝΟ(32) ΤΟΤΕ ΕΞΟΔΟΣ
 ΑΝΑΜΟΝΗ 1
ΠΑΝΤΑ
ΗΧΟΓΡΑΦΗΣΗ ΤΕΛΟΣ

Αναγνωριστικό: ΘΕΣΗ.Υ [POS.Υ]
Θέση Υ του δρομέα γραφικών

Αναγνωριστικό: ΘΕΣΗ.Χ [POS.Χ]
Θέση Χ του δρομέα γραφικών

Αναγνωριστικό: ΘΕΣΗ_μεταβλητή [POS]
Δίνει την θέση Χ του δρομέα χαρακτήρων

ΤΥΠΩΣΕ "ΑΒΓΔΕ",ΘΕΣΗ
ΤΥΠΩΣΕ ΘΕΣΗ

Αναγνωριστικό: ΙΔΙΟΤΗΤΕΣ\$ [PROPERTIES\$]
Η ΙΔΙΟΤΗΤΕΣ\$ επιστρέφει ένα αλφαριθμητικό με τις τρέχουσες ιδιότητες του εκτυπωτή

Εδώ είναι ένα παράδειγμα όπου αλλάζουμε μια φορά τις ιδιότητες και μετά ξαναγυρνάμε στις παλιές που είχε ο εκτυπωτής όπως τις είδαμε την πρώτη φορά.

B\$=ΙΔΙΟΤΗΤΕΣ\$
ΕΚΤΥΠΩΤΗΣ!
A\$=ΙΔΙΟΤΗΤΕΣ\$
ΙΔΙΟΤΗΤΕΣ B\$
ΕΚΤΥΠΩΤΗΣ!

Αναγνωριστικό: ΚΑΤ\$ [DIR\$]
Τυπωσε Κατ\$
Επιστρέφει τον τωρινό κατάλογο. Μπορεί να αλλαχτεί αυτός ο κατάλογος με την εντολή
ΚΑΤΑΛΟΓΟΣ

στο φάκελο χρήστη δίνει "." το οποίο μπορούμε να χρησιμοποιήσουμε όσο είμαστε σε αυτόν τον

κατάλογο

μπορούμε να μεταβούμε σε αυτόν με την εντολή (μπορούμε να βάλουμε και τόνους)
κατάλογος χρήστη

Αναγνωριστικό: ΚΑΤΑΣΤΑΣΗ.ΤΑΙΝΙΑΣ\$ [MOVIE.STATUS\$]

ΚΑΤΑΣΤΑΣΗ.ΤΑΙΝΙΑΣ\$

μας δίνει την κατάσταση ταινίας!

Αναγνωριστικό: KENO [EMPTY]

Τυπωσε KENO

Δίνει ΑΛΗΘΕΣ αν δεν υπάρχει στοιχείο στο ΣΩΡΟ

Δες και την ΜΕΓΕΘΟΣ.ΣΩΡΟΥ

ΜΕΓΕΘΟΣ.ΣΩΡΟΥ=0 είναι ισοδύναμο με το KENO

Αναγνωριστικό: ΚΙΝΗΣΗ.ΠΥ [MOTION.WY]

δίνει την θέση Υ του ΠΕΡΙΘΩΡΙΟΥ σε σχέση με την ΕΠΙΦΑΝΕΙΑ ΕΡΓΑΣΙΑΣ

ΚΙΝΗΣΗ.ΠΥ

και ΚΙΝΗΣΗ.ΥΠ είναι το ίδιο

Είναι η ιδιότητα TOP της κεντρική φόρμας της M2000 (που αναλογεί στο περιθώριο).

Αναγνωριστικό: ΚΙΝΗΣΗ.ΠΧ [MOTION.WX]

δίνει την θέση Χ του ΠΕΡΙΘΩΡΙΟΥ σε σχέση με την ΕΠΙΦΑΝΕΙΑ ΕΡΓΑΣΙΑΣ

Αναγνωριστικό: ΚΙΝΗΣΗ.Υ [MOTION.Y]

Δίνει την θέση Υ του άνω αριστερού σημείου της ΟΘΟΝΗΣ (η επιπέδου) σε σχέση με το ΠΕΡΙΘΩΡΙΟ

Αναγνωριστικό: ΚΙΝΗΣΗ.ΥΠ [MOTION.YW]

δίνει την θέση Υ του ΠΕΡΙΘΩΡΙΟΥ σε σχέση με την ΕΠΙΦΑΝΕΙΑ ΕΡΓΑΣΙΑΣ

Αναγνωριστικό: ΚΙΝΗΣΗ.Χ [MOTION.X]

Δίνει την θέση Χ του άνω αριστερού σημείου της ΟΘΟΝΗΣ σε σχέση με το ΠΕΡΙΘΩΡΙΟ

Χρησιμοποιείται και σε Επίπεδα σε σχέση με την Οθόνη

Αναγνωριστικό: ΚΙΝΗΣΗ.ΧΠ [MOTION.XW]

δίνει την θέση Χ του ΠΕΡΙΘΩΡΙΟΥ σε σχέση με την ΕΠΙΦΑΝΕΙΑ ΕΡΓΑΣΙΑΣ

Αναγνωριστικό: ΚΛΙΜΑΞ.Υ [SCALE.Y]

Μέγιστα σημεία Υ στην ΟΘΟΝΗ ή στο ΠΕΡΙΘΩΡΙΟ (ΔΕΣ ΠΕΡΙΘΩΡΙΟ)
συνώνυμο με Υ.ΣΗΜΕΙΑ

Αναγνωριστικό: ΚΛΙΜΑΞ.Χ [SCALE.X]

Μέγιστα σημεία Χ στην ΟΘΟΝΗ ή στο ΠΕΡΙΘΩΡΙΟ (ΔΕΣ ΠΕΡΙΘΩΡΙΟ)
συνώνυμο με Χ.ΣΗΜΕΙΑ

Αναγνωριστικό: ΚΟΜ\$ [KEY\$]

α\$=κομ\$

Ο μεταφραστής περιμένει να πατηθεί ένα πλήκτρο (κομβίο) και δίνει τον χαρακτήρα ως
αλφαριθμητικό

αυτή η εντολή βάζει σε αναμονή τον μεταφραστή μέχρι να πατηθεί ένα πλήκτρο.

αν το περιβάλλον είναι σε ελαχιστοποίηση τότε αυτόματα ανοίγει

Αναγνωριστικό: ΚΟΝΣΟΛΑ_μετ [DOS_var]

α=Κονσόλα

επιστρέφει τιμή από την τελευταία εντολή Κονσόλα

Αναγνωριστικό: ΚΩΔΙΚΟΣΕΛΙΔΑ [CODEPAGE]

Τύπωσε Κωδικοσελίδα

γυρνάει την τρέχουσα κωδικοσελίδα

Αναγνωριστικό: ΛΑΘΟΣ\$ [ERROR\$]

Τυπωσε Λαθος\$

τυπώνει την περιγραφή του λάθους μετά από μια ΔΕΣ

Αναγνωριστικό: ΛΑΘΟΣ.ΤΑΙΝΙΑΣ\$ [MOVIE.ERROR\$]

? ΛΑΘΟΣ.ΤΑΙΝΙΑΣ\$

μας δίνει το λάθος ταινίας!

Αναγνωριστικό: ΛΟΓΟΣ ως μεταβλητή [SPEECH as variable]

ΤΥΠΩΣΕ Λογος

τυπώνει τον αριθμό φωνών για λόγο

Για φωνη= 1 έως Λογος {

 Τυπωσε Λογος\$(φωνη)

}

Τυπώνει τα ονόματα των φωνών

Αναγνωριστικό: ΛΣ\$ [OS\$]

? ΛΣ\$

Μας δίνει το όνομα του λειτουργικού συστήματος

Αναγνωριστικό: ΜΕΓΕΘΟΣ.ΣΩΡΟΥ [STACK.SIZE]

ΒΑΛΕ 100, "123123"

ΤΥΠΩΣΕ ΜΕΓΕΘΟΣ.ΣΩΡΟΥ

Αναγνωριστικό: ΜΝΗΜΗ [MEMORY]
ΤΥΠΩΣΕ ΜΝΗΜΗ

Ο αριθμός που μας δίνει είναι σε Mega Bytes (εκατομμύρια χαρακτήρες), και περιλαμβάνει όλη την φυσική μνήμη που είναι διαθέσιμη στο σύστημα.

Πριν την έκτη έκδοση:

Ο αριθμός που μας δίνει είναι σε Mega Bytes (εκατομμύρια χαρακτήρες), και περιλαμβάνει όλη την μνήμη που είναι διαθέσιμη (μαζί με την εικονική δηλαδή του χώρου του δίσκου μας που μπορούμε να χρησιμοποιήσουμε)

Αναγνωριστικό: ΜΟΥΣΙΚΗ.ΜΕΤΡΗΤΗΣ [MUSIC.COUNTER]
Δίνει -1 αν δεν υπάρχει μουσική!
Δίνει το μετρητή (ανά χιλιοστό δευτερολέπτου)
Για όλη τη διάρκεια δεξ το ΔΙΑΡΚΕΙΑ
Η Μουσική.Μετρητής είναι ακριβώς η ίδια εντολή με την Ταινία.Μετρητής

Αναγνωριστικό: ΝΗΜΑΤΑ\$ [THREADS\$]
? ΝΗΜΑΤΑ\$
μας δίνει την λίστα νημάτων στο τρέχων τμήμα και σε όλα τα πατρικά προς τα πάνω.

Αναγνωριστικό: ΟΘΟΝΕΣ [MONITORS]
ΤΥΠΩΣΕ ΟΘΟΝΕΣ
Δίνει τον αριθμό οθονών (monitors)
Για να επιλέξουμε οθόνη:
Παράθυρο 12, 1
θα μας αλλάξει το παράθυρο της κονσόλας, φτιάχνοντας ένα νέο στη δεύτερη οθόνη
Αν έχουμε 2 οθόνες τότε το 0 και το 1 είναι οι οθόνες που επιλέγουμε. Αν δώσουμε >1 θα μας πάει στο 0
Η μεταβλητή Συσκευή μας δείχνει σε ποια οθόνη δουλεύουμε.

Με την εντολή ΕΛΕΓΧΟΣ από τη κονσόλα μπορούμε να δούμε σε ποια οθόνη (νούμερο) βγαίνει η κονσόλα και ποια είναι η βασική οθόνη.

Αναγνωριστικό: ΟΝΟΜΑ.ΤΜΗΜΑΤΟΣ\$ [MODULE.NAME\$]

Τμήμα Βήτα {

Τύπωσε Τμήμα\$

\\ το παραπάνω έχει κωδικούς

\\ από αυτούς το παρακάτω βγάζει το κανονικό όνομα

Τύπωσε όνομα.τμήματος\$

}

Κάλεσε Βήτα

Αναγνωριστικό: ΟΝΟΜΑ.ΧΡΗΣΤΗΣ [USER.NAME\$]

? ΟΝΟΜΑ.ΧΡΗΣΤΗ

επιστρέφει το όνομα χρήστη

επιλέγουμε ή και δημιουργούμε χρήστη με την παρακάτω εντολή

ΧΡΗΣΤΗΣ ονομα

Ο επόπτης έχει το όνομα του χρήστη των Windows. Αν φτιάξουμε χρήστη με αυτό το όνομα δεν θα είναι ο επόπτης. Για να επιστρέψουμε στον επόπτη πρέπει να δώσουμε τη κατάλληλη εντολή (υπάρχει γραμμένη στην βοήθεια και θα τη βρει κανείς όταν την διαβάσει προσεκτικά) μπορούμε να επιλέξουμε τον κατάλογο χρήστη με την εντολή ΚΑΤΑΛΟΓΟΣ ΧΡΗΣΤΗ με ενεργό ένα χρήστη εκτός των επόπτη δεν δουλεύουν οι εντολές ΚΟΝΣΟΛΑ και ΣΥΣΤΗΜΑ

Αναγνωριστικό: ΠΑΙΖΕΙΦΩΝΗ [PLAYSCORE]

Τύπωσε ΠΑΙΖΕΙΦΩΝΗ

αν ένα οποιοδήποτε νήμα μουσικής εκτελείται τότε επιστρέφει αληθής

Αναγνωριστικό: ΠΑΡΑΘΥΡΟ HWND [HWND]

Τύπωσε Παραθυρο

γυρνάει τον αριθμό του παραθύρου (χρησιμοποιείται σε κλήσεις εξωτερικών συναρτήσεων που ανήκουν στο λειτουργικό)

Όρισε μήνυμα Από "user32.MessageBoxW" {Μακρυσ Α, κειμενο\$, κειμενο2\$, Μακρυσ τυπος}

Τύπωσε μήνυμα(Παραθυρο, "Γεια χαρά", "Γιώργος", 2)

Αναγνωριστικό: ΠΑΡΑΜΕΤΡΟΙ\$ [PARAMETERS\$]

Διαβάζει παράμετρους που μπήκαν κατά τη κλήση ενός προγράμματος με κατάλληξη gsb. Μπορούμε να βάζουμε εντολές ή παραμέτρους με - ή + (αυτές γυρίζει η μεταβλητή) αν έχουμε γνωστές παραμέτρους πχ +DEC (δες Διακόπτες) στην αρχή του αλφαριθμητικού τότε διαβάζονται και αφαιρούνται από το αλφαριθμητικό. (αν βρει ο διερμηνευτής κάτι που δεν γνωρίζει ως διακόπτη σταματάει την ανάγνωση και επιστρέφει το υπόλοιπο - το πρώτο μη γνωστό αναγνωριστικό το έχει μετατρέψει σε κεφαλαία όταν το βάλει πάλι πίσω) Δες στην αγγλική έκδοση πρόγραμμα χρήσης.

Αναγνωριστικό: ΠΑΡΕΚΑΡΕ\$ [GRABFRAME\$]

A\$=ΠΑΡΕΚΑΡΕ\$

Μπορούμε να ανοίξουμε ένα βίντεο και να το μετακινήσουμε στο καρέ που θέλουμε και να πάρουμε ένα καρέ, στην ΠΑΡΕΚΑΡΕ\$ το καρέ πρέπει να το γράψουμε σε μια άλλη μεταβλητή...γιατί την επόμενη φορά που θα διαβάσουμε μπορεί το βίντεο να έχει μετακινηθεί σε άλλο καρέ ή να έχει κλείσει.

Αναγνωριστικό: ΠΕΔΙΟ_μεταβλητή [FIELD_as variable]

Δες εντολή ΠΕΔΙΟ

Η μεταβλητή μόνο ανάγνωσης ΠΕΔΙΟ γυρνάει την κατάσταση μιας εισαγωγής πεδίου. Έτσι μπορούμε να ξέρουμε αν ο χρήστης έχει πατήσει ESC ή βελάκια για να πάει στο επόμενο πεδίο

1 προηγούμενο (άνω βέλος)

-1 επόμενο (κάτω βέλος, πλήκτρο enter, πλήκτρο TAB)

99 ακύρωση (ESC)

-20 Home

20 End

-10 pageup

10 pagedown

1 Tab

-1 Shift Tab

Αναγνωριστικό: ΠΕΝΑ_μεταβλητή [PEN_variable]

Τυπωσε ΠΕΝΑ

δίνει το χρώμα σε αριθμό. Ο αριθμός είναι αρνητικός αν είναι RGB (δηλαδή γινόμενο των τριών χρωμάτων) ή αν είναι ένα από τα δεσμευμένα από το 1 έως το 15 (το 0 είναι το μαύρο και στις δυο περιπτώσεις)

Πενα 1
Τυπωσε Πενα
1
Πενα χρωμα(255,250,255)
Τυπωσε Πενα
-16775935

Αναγνωριστικό: ΠΕΡΙ\$ [ABOUT\$]

A\$=ΠΕΡΙ\$

διαβάζουμε την επιλογή από την φόρμα βοήθειας που ανήγει η εφαρμογή μας με την εντολή ΠΕΡΙ

Στην βοήθεια για την εφαρμογή μας που φτιάχνουμε με την ΠΕΡΙ μπορούμε να βάλουμε λέξεις σε τετράγωνε αγκύλες []

Αυτές οι λέξεις οδηγούνται σε μια ρουτίνα που έχουμε φορτώσει στο σύστημα με την ΠΕΡΙ ΚΑΛΕΣΕ

ΠΕΡΙ ! "ΤΙΤΛΟΣ: ΕΚΕΙΝΟ", 8000,6000, {Βοήθεια προγράμματος....

....[ΑΥΤΟ], [ΕΚΕΙΝΟ], [γράψε εδώ]

}

ΠΕΡΙ ΚΑΛΕΣΕ {

ΕΠΙΛΕΞΕ ΜΕ ΠΕΡΙ\$

ΜΕ "ΑΥΤΟ"

ΠΕΡΙ "ΤΙΤΛΟΣ: ΑΥΤΟ", 8000,6000, "ΔΕΣ [ΑΥΤΟ] ΚΑΙ [ΕΚΕΙΝΟ]"

ΜΕ "ΕΚΕΙΝΟ"

{

ΠΕΡΙ "ΤΙΤΛΟΣ: ΕΚΕΙΝΟ", 8000,6000, {Κείμενο....

.....άλλη γραμμή.....

....και άλλη.....

γύρνα σε [ΑΥΤΟ] η τράβα στο [ΑΛΛΟ]

}

}

ΑΛΛΙΩΣ

ΠΕΡΙ "ΤΙΤΛΟΣ: ΑΛΛΟ", 8000,6000, "ΔΕΣ [ΑΥΤΟ] ΚΑΙ [ΕΚΕΙΝΟ]"

ΤΕΛΟΣ ΕΠΙΛΟΓΗΣ

}

* ενώ η εισαγωγή περιμένει να γράψεις πάτα CTRL+F1

* θα ανοίξει η βοήθεια και μπορείς να πλογηθείς σε αυτήν

ΕΙΣΑΓΩΓΗ "Όνομα:", ν\$

* μπορούμε με μια σκέτη ΠΕΡΙ να απενεργοποιήσουμε τη βοήθεια

Αναγνωριστικό: ΠΛΑΤΟΣ [WIDTH_as variable]

Μας δίνει το πλάτος σε χαρακτήρες (ανάλογα της ΟΘΟΝΗΣ, του ΠΕΡΙΘΩΡΙΟΥ, της σελίδας του ΕΚΤΥΠΩΤΗ)

Αναγνωριστικό: ΠΛΑΤΟΣ.ΣΗΜΕΙΟΥ [TWIPSY]

Αν δώσει 15 (συνήθως αυτό είναι) σημαίνει ότι 15 twips είναι ένα εικονοστοιχείο, άρα στα 1440 twips (μια ιντσα) έχουμε 1440/15 εικονοστοιχεία, που ισούται με 96 dpi.

Που μας βοηθάει αυτή η γνώση; Όταν τυπώνουμε γραφικά, για να διευκολύνουμε το πρόγραμμα, να μην χρονοτριβεί, τυπώνουμε σε θέσεις σημείων, δίνοντας πολλαπλάσια του πλάτους σημείου σε κάθε περίπτωση που αναφερόμαστε σε συντεταγμένες πάνω στην οθόνη.

Προσοχή: Πάντα γυρίζει το πλάτος σημείου της οθόνης και όχι του εκτυπωτή, ακόμα και αν συμπεριλάβουμε την εντολή σε δομή ΕΚΤΥΠΩΣΕ { }

Στους εκτυπωτές τα dpi κατά Χ και κατά Υ μπορεί να διαφέρουν, που σημαίνει με απλά λόγια ότι οι εκτυπωτές κατά κανόνα δεν έχουν τετράγωνα εικονοστοιχεία αλλά παραλληλόγραμμα. Εκεί επειδή χρησιμοποιούμε twips..και επειδή το πλάτος εικονοστοιχείου είναι μικρό, πλησιάζει το 1...δεν μας ενοχλεί η απόκριση ως προς τον χρόνο ενός προγράμματος (στο φινάλε και ο εκτυπωτής θα αργήσει να δώσει το ζητούμενο, την εκτύπωση, πράγμα που δεν συμβαίνει αυτό στην οθόνη).

Δείτε και ΥΨΟΣ.ΣΗΜΕΙΟΥ ή TWIPSY

Αναγνωριστικό: ΠΛΑΤΟΣ.ΤΑΙΝΙΑΣ [MOVIE.WIDTH]

ΤΑΙΝΙΑ ΦΟΡΤΩΣΕ "SECOND"

ΤΥΠΩΣΕ ΠΛΑΤΟΣ.ΤΑΙΝΙΑΣ

Αναγνωριστικό: ΠΛΑΤΦΟΡΜΑ\$ [PLATFORM\$]

? πλατφορμα\$

επιστρέφει την περιγραφή της πλατφόρμας (π.χ. WINDOWS NT)

Αναγνωριστικό: ΠΡΟΣΩΡΙΝΟΣ\$ [TEMPORARY\$]

ΤΥΠΩΣΕ ΠΡΟΣΩΡΙΝΟΣ\$

C:\WINDOWS\TEMP\

Μας δίνει τον κατάλογο για προσωρινή αποθήκευση (δεν είναι απαραίτητα αυτός που

εμφανίζεται στο παράδειγμα εδώ)
Δες το ΕΝΑΝΟΜΑ\$

Αναγνωριστικό: ΠΡΟΧΕΙΡΟΣ\$ [CLIPBOARD\$]
τυπωσε Προχειρο\$
διαβάζουμε από το πρόχειρο ένα αλφαριθμητικό
Μπορούμε να βάλουμε στο πρόχειρο ένα αλφαριθμητικό με την εντολή Προχειρο
Χρησιμοποίησε το Πρόχειρο.Εικόνα\$ για να πάρεις εικόνα από το πρόχειρο

Δές ΠΡΟΧΕΙΡΟ.ΕΙΚΟΝΑ\$, ΠΡΟΧΕΙΡΟ.ΕΙΚΟΝΑ, ΠΡΟΧΕΙΡΟ.ΣΧΕΔΙΟ

Αναγνωριστικό: ΠΡΟΧΕΙΡΟ.ΕΙΚΟΝΑ [CLIPBOARD.IMAGE]
Φορτώνει εικόνα από το πρόχειρο σε διάρθρωση μνήμης
Α=ΠΡΟΧΕΙΡΟ.ΕΙΚΟΝΑ
ΕΙΚΟΝΑ Α, 10000

Δές ΠΡΟΧΕΙΡΟ.ΕΙΚΟΝΑ\$, ΠΡΟΧΕΙΡΟ.ΣΧΕΔΙΟ, ΠΡΟΧΕΙΡΟΣ\$

Αναγνωριστικό: ΠΡΟΧΕΙΡΟ.ΕΙΚΟΝΑ\$ [CLIPBOARD.IMAGE\$]
α\$=προχειρο.εικονα\$
αντιγράφουμε στο αλφαριθμητικό την εικόνα από το πρόχειρο, αν υπάρχει.

Δές ΠΡΟΧΕΙΡΟ.ΕΙΚΟΝΑ, ΠΡΟΧΕΙΡΟ.ΣΧΕΔΙΟ, ΠΡΟΧΕΙΡΟΣ\$

Αναγνωριστικό: ΠΡΟΧΕΙΡΟ.ΣΧΕΔΙΟ [CLIPBOARD.DRAWING]
Φορτώνουμε Emf (metafiles) από το πρόχειρο
Α=ΠΡΟΧΕΙΡΟ.ΕΙΚΟΝΑ
ΘΕΣΗ 1000, 1000
ΕΙΚΟΝΑ Α, 8000

Δείτε τα ΠΡΟΧΕΙΡΟ.ΕΙΚΟΝΑ, ΠΡΟΧΕΙΡΟ.ΕΙΚΟΝΑ\$, ΠΡΟΧΕΙΡΟΣ\$

Αναγνωριστικό: ΣΗΜΕΙΟ [POINT]
Δίνει το χρώμα του σημείου που δείχνει ο δρομέας γραφικών (της ΟΘΟΝΗΣ ή του ΠΕΡΙΘΩΡΙΟΥ - ΔΕΣ ΠΕΡΙΘΩΡΙΟ)

Μπορούμε με την Επίπεδο {} να αλλάξουμε τρέχον επίπεδο και να διαβάσουμε το χρώμα από Φόρμα ή Επίπεδο πάνω στην Οθόνη.

Αναγνωριστικό: ΣΗΜΕΡΑ [TODAY]

Δίνει έναν αριθμό που αντιπροσωπεύει την σημερινή μέρα.

ΤΥΠΩΣΕ ΗΜΕΡΑ\$(ΣΗΜΕΡΑ)

ΤΥΠΩΣΕ ΓΡΑΦΗ\$(ΣΗΜΕΡΑ, "LONG DATE")

Αναγνωριστικό: ΣΤΗΛΗ [TAB]

Δίνει το πλάτος των νοητών στηλών της οθόνης. ΔΕΣ ΤΥΠΩΣΕ

Τύπωσε @(στήλη*3), "οκ"

Τύπωσε @(στήλη(3)), "οκ"

Δες Στήλη(), \$()

Αναγνωριστικό: ΣΥΣΚΕΥΗ [WINDOW_variable]

Τύπωσε Συσκευή

Δίνει 0 αν έχουμε μια μόνο Οθόνη

ή μια τιμή από 0 μέχρι Οθόνες-1

Αναγνωριστικό: ΣΥΣΚΕΥΗ.ΠΡΟΒΟΛΗΣ\$ [MOVIE.DEVICE\$]

? ΣΥΣΚΕΥΗ.ΠΡΟΒΟΛΗΣ\$

μας δίνει το όνομα της συσκευής προβολής

Αναγνωριστικό: ΤΑΙΝΙΑ.ΜΕΤΡΗΤΗΣ [MOVIE.COUNTER]

Δίνει -1 αν δεν υπάρχει ταινία!

Δίνει το μετρητή (ανά χιλιοστό δευτερολέπτου)

Για όλη τη διάρκεια δες το ΔΙΑΡΚΕΙΑ

Αναγνωριστικό: ΤΑΙΝΙΑ_μεταβλητή [MOVIE_as variable]

Δίνει ΑΛΗΘΕΣ όταν έχουμε κάποια ταινία ή ένα μουσικό κομμάτι σε εκτέλεση.

συνώνυμα MEDIA, ΜΟΥΣΙΚΗ

Αναγνωριστικό: TIK [TICK]

Δίνει τον αριθμό τικ (παλμών) του συστήματος των νημάτων. Όσο είναι μηδέν σημαίνει ότι κανένα νήμα δεν έχει τρέξει.

Όρισε ρολοι Από "winmm.timeGetTime" {}

```
Για i=1 έως 100 {  
    Τύπωσε ρολοι()  
}
```

Αναγνωριστικό: TIMH [NUMBER]

Σηκώνει την αριθμητική τιμή της κορυφής του σωρού (και την διώχνει από εκεί). Αν είναι ο σωρός είναι κενός ή έχει λέξη και όχι αριθμό στην κορυφή τότε βγαίνει λάθος. Λέμε "σηκώνει" γιατί την αφαιρεί από το σωρό.

συνώνυμο ΑΡΙΘΜΟΣ

Μπορούμε να σηκώσουμε πίνακα ή ομάδα με τα Πίνακας και Ομάδα ως μεταβλητές μόνο για ανάγνωση

Μπορούμε να πάρουμε Κατάσταση ή Αντικείμενο σωρό από το σωρό με την ΤιμήΣωρού() ή ΤιμήΣωρού(1) (είναι το ίδιο)

Μετά με την Πέτα πετάμε την κορυφή.

Μπορούμε να γνωρίζουμε με την Ταύτιση() αν υπάρχουν συγκεκριμένοι τύποι τιμών στο σωρό

Μπορούμε με την ΣωρούΤύπος\$() να διαβάζουμε το τύπο (περιφραστικά στα αγγλικά)

Μπορούμε με την Φάκελος\$() να πάρουμε ένα αλφαριθμητικό με γράμμα σε κάθε θέση στο σωρό, που δείχνει τον τύπο του.

Αναγνωριστικό: ΤΜΗΜΑ\$ [MODULE\$]

? τμημα\$

μας δίνει το όνομα από το τρέχον τμήμα ή συνάρτηση

Αναγνωριστικό: ΤΥΠΟΣ_μεταβλητή [MODE_variable]

Δίνει το τύπο γραμμάτων σε σημεία pt (12 pt = 1440/72 twips - 1440 twips = 1 λογική ίντσα - σε εκτυπωτές είναι πραγματική ίντσα)

γράμματα 12 pt έχουν ύψος (πλασιού μιας γραμμής) (1440twips/72pt)*12pt=240twips

Μπορούμε να ορίζουμε ΤΥΠΟ με την εντολή ΤΥΠΟ.

Από την έκτη έκδοση μπορούμε να αλλάζουμε τύπο χωρίς να καθαρίζουμε την οθόνη, όπως και με την εντολή ΓΡΑΜΜΑΤΟΣΕΙΡΑ μπορούμε να αλλάζουμε τη μορφή των γραμμάτων. Στις προηγούμενες εκδόσεις έπρεπε να χρησιμοποιήσουμε την εντολή ΕΠΙΓΡΑΦΗ για να

τυπώσουμε με οποιαδήποτε γραμματοσειρά και οποιοδήποτε μέγεθος (υπάρχει ακόμα), με τη διαφορά ότι τυπώνει σε θέσεις γραφικών και όχι χαρακτήρων, και επίσης μπορούμε να δώσουμε και περιστροφή σε μοίρες.

Αναγνωριστικό: ΤΥΧΑΙΟΣ [RND]
Τύπωσε Τυχαίος \\ δίνει έναν αριθμό ≥ 0 και < 1

Αναγνωριστικό: ΤΩΡΑ [NOW]
Δίνει έναν αριθμό που αντιπροσωπεύει την τωρινή ώρα.
ΤΥΠΩΣΕ ΧΡΟΝΟΣ\$(ΤΩΡΑ)
ΤΥΠΩΣΕ ΓΡΑΦΗ\$(ΤΩΡΑ, "hh:mm AMPM")

$\alpha = \text{Σήμερα} + \text{Τώρα}$ '(όλα μαζί, το ακέραιο μέρος είναι η ημέρα, και τα δεκαδικά η ώρα)

Αναγνωριστικό: Υ.ΣΗΜΕΙΑ [Y.TWIPS]
Μέγιστα σημεία Υ στην ΟΘΟΝΗ ή στο ΠΕΡΙΘΩΡΙΟ (ΔΕΣ ΠΕΡΙΘΩΡΙΟ)
συνώνυμο με ΚΛΙΜΑΞ.Υ

Αναγνωριστικό: ΥΠΟΛΟΓΙΣΤΗΣ\$ [COMPUTER\$]
Δίνει το όνομα του υπολογιστή
Τυπωσε Υπολογιστης

Αναγνωριστικό: ΥΨΟΣ [HEIGHT]
Δίνει το ύψος σε χαρακτήρες της ΟΘΟΝΗΣ (ή του ΠΕΡΙΘΩΡΙΟΥ, ΔΕΣ ΠΕΡΙΘΩΡΙΟ)

Αναγνωριστικό: ΥΨΟΣ.ΣΗΜΕΙΟΥ [TWIPSY]
ΥΨΟΣ.ΣΗΜΕΙΟΥ
μας δίνει το ύψος σημείου της οθονης σε twips

Αναγνωριστικό: ΥΨΟΣ.ΤΑΙΝΙΑΣ [MOVIE.HEIGHT]
ΤΑΙΝΙΑ ΦΟΡΤΩΣΕ "SECOND"
ΤΥΠΩΣΕ ΥΨΟΣ.ΤΑΙΝΙΑΣ

Αναγνωριστικό: ΦΟΡΜΑ\$ [CONTROL\$]
ΤΥΠΩΣΕ ΦΟΡΜΑ\$

δίνει το όνομα της ενεργής φόρμα αν είναι μια από τα ΤΑΙΝΙΑ, ΚΥΡΙΟ, ΒΟΗΘΕΙΑ ή κάποιο παράθυρο χρήστη (φόρμα)

Αν έχουμε πίνακα παραθύρων τότε θα δώσει το όνομα της φόρμας με παρενθέσεις και το νούμερο της φόρμας.

Μπορούμε να ελέγχουμε αν στηνεφαρμογή μας είναι στο προσκήνιο ένα από αυτά τα τρία. Αν είναι κάποιο τότε παίρνουμε το όνομα. Αν έχουμε το περιβάλλον σε ελαχιστοποίηση τότε μας επιστρέφει ένα άδαιο αλφαριθμητικό.

Αναγνωριστικό: ΦΟΡΤΟΣ [TIMECOUNT]

Μας δίνει τον αριθμό χιλιοστών του δευτερολέπτου που έφθασε ένας εσωτερικός μετρητής του ΑΝΑΛΥΤΗ

Δες εντολή ΑΝΑΛΥΤΗΣ

Αναγνωριστικό: Χ.ΣΗΜΕΙΑ [X.TWIPS]

Μέγιστα σημεία Χ στην ΟΘΟΝΗ ή στο ΠΕΡΙΘΩΡΙΟ (ΔΕΣ ΠΕΡΙΘΩΡΙΟ)
συνώνυμο με ΚΛΙΜΑΞ.Χ

Αναγνωριστικό: ΧΡΩΜΑΤΑ [COLORS]

ΤΥΠΩΣΕ ΧΡΩΜΑΤΑ

τυπώνει τον αριθμό χρωμάτων που υποστηρίζει η επιφάνεια εργασίας (άρα και η οθόνη μας)

Ομάδα: ΣΤΑΘΕΡΕΣ - CONSTANTS

Αναγνωριστικό: ΑΚΕΡΑΙΟΣ [INTEGER]

Χρησιμοποιείται στην εντολή ΑΡΧΕΙΟ

Αναγνωριστικό: ΑΛΗΘΕΣ [TRUE _2]
ΔΙΝΕΙ -1
ΟΠΩΣ Η ΑΛΗΘΗΣ

Αναγνωριστικό: ΑΛΗΘΗΣ [TRUE]
Δίνει τιμή -1
(υπάρχει και η ΨΕΥΔΕΣ εκδ 4.)

Αναγνωριστικό: ΑΠΕΙΡΟ [INFINITY]
Τύπωσε ΑΠΕΙΡΟ, -ΑΠΕΙΡΟ
1.#INF -1.#INF

Αναγνωριστικό: ΑΠΛΟΣ [SINGLE]
χρησιμοποιείται στην εντολή ΑΡΧΕΙΟ

Αναγνωριστικό: ΑΥΞΟΥΣΑ [ASCENDING]
Δίνει τιμή 0
χρησιμοποιείται στην εντολή ΤΑΞΗ (ORDER)

Αναγνωριστικό: ΔΙΠΛΟΣ [DOUBLE_as constant]
χρησιμοποιείται στην εντολή ΑΡΧΕΙΟ

Αναγνωριστικό: ΔΥΑΔΙΚΟ_σταθ [BINARY_const]
ΔΥΑΔΙΚΟ
σταθερή που χρησιμοποιείται στις βάσεις δεδομένων

Αναγνωριστικό: ΕΚΔΟΣΗ_σταθερά [VERSION_as constant]

ΤΥΠΩΣΕ ΕΚΔΟΣΗ

τυπώνει τον αριθμό έκδοσης της γλώσσας, όπως η εντολή ΕΚΔΟΣΗ αλλά ως σταθερά

Αναγνωριστικό: ΕΛ ISWINE [ISWINE]

Τύπωσε ISWINE

επιστρέφει -1 αν ο διερμηνευτής τρέχει σε Linux με Wine

Αναγνωριστικό: ΗΜΕΡΟΜΗΝΙΑ [DATEFIELD]

χρησιμοποιείται στην εντολή ΑΡΧΕΙΟ

Αναγνωριστικό: ΚΕΙΜΕΝΟ_σταθερά [TEXT_as constant]

χρησιμοποιείται στην εντολή ΑΡΧΕΙΟ

Αναγνωριστικό: ΛΟΓΙΚΟΣ [BOOLEAN]

χρησιμοποιείται στην εντολή ΑΡΧΕΙΟ

Αναγνωριστικό: ΛΟΓΙΣΤΙΚΟ [CURRENCY]

χρησιμοποιείται στην εντολή ΑΡΧΕΙΟ

Αναγνωριστικό: ΜΑΚΡΥΣ_ΤΥΠΟΣ [LONG_TYPE]

ΜΑΚΡΥΣ

Τύπος για την εντολή Αρχείο

Υπάρχει και τύπος μεταβλητής

Αναγνωριστικό: ΠΙ [PI]

? πι

3.1415926535897932384626433832

Γενική πι=3.14159~ ' αναπρογραμματισμός

Αναγνωριστικό: ΥΠΟΜΝΗΜΑ [MEMO]

χρησιμοποιείται στην εντολή ΑΡΧΕΙΟ

Αναγνωριστικό: ΦΘΙΝΟΥΣΑ [DESCENDING]

Χρησιμοποιείται στην εντολή ΤΑΞΗ

και στην Ταξινόμηση

Αναγνωριστικό: ΦΟΡΜΑΡΙΣΜΑ_ΑΛΦΑΡΙΘΜΗΤΙΚΩΝ [FORMATTING_STRINGS]

Για την ΓΡΑΦΗ\$(και την εσωτερική συνάρτηση της ΤΥΠΩΣΕ την \$(

1)@

Θέση εκτύπωσης ενός χαρακτήρα αν υπάρχει στο προς φορμάρισμα αλφαριθμητικό, διαφορετικά τυπώνει το διάστημα

2)&

Θέση εκτύπωσης ενός χαρακτήρα αν υπάρχει στο προς φορμάρισμα αλφαριθμητικό, διαφορετικά δεν τυπώνει καθόλου χαρακτήρα

3)<

Μετατρέπει σε πεζά τους χαρακτήρες που ακολουθούν

4)>

Μετατρέπει σε κεφαλαία τους χαρακτήρες που ακολουθούν

5)!

Αλλάζει την καθορισμένη φορά εκτύπωσης

Αναγνωριστικό: ΦΟΡΜΑΡΙΣΜΑ_ΑΡΙΘΜΩΝ [FORMATTING_NUMBERS]

Για την ΓΡΑΦΗ\$(και την εσωτερική συνάρτηση της ΤΥΠΩΣΕ την \$(

1)0

Τυπώνει αριθμό ή το 0

2)#

Τυπώνει αριθμό ή τίποτα

3).

Θέση υποδιαστολής

4)%

Μετατροπή σε ποσοστό. Η έκφραση πολλαπλασιάζεται επί 100 και μπαίνει σε αυτή την θέση το σύμβολο %

5),

Διαχωριστής χιλιάδων

6) E- E+ e- e+

Επιστημονική γραφή αριθμών

Αναγνωριστικό: ΦΟΡΜΑΡΙΣΜΑ_ΓΙΑ ΚΑΘΕ ΤΥΠΟ [FORMATTING_ANY TYPE]

Για την ΓΡΑΦΗ\$(και την εσωτερική συνάρτηση της ΤΥΠΩΣΕ την \$(

\

τυπώνει τον επόμενο χαρακτήρα που θα βρει στο αλφαριθμητικό φορμαρίσματος

Εξαιρούνται τα : a, c, d, h, m, n, p, q, s, t, w, y, / και :

(" ... ")

Τυπώνει τους περιεχόμενους χαρακτήρες (εδώ ...)

Αναγνωριστικό: ΦΟΡΜΑΡΙΣΜΑ_ΗΜΕΡΟΜΗΝΙΩΝ ΚΑΙ ΩΡΑΣ [FORMATTING_DATE AND TIME]

1):

διαχωριστικό ώρας

2)/

διαχωριστικό ημερομηνίας

3)c

ολόκληρη ώρα/ημερομηνία

4)d

ημέρα ως αριθμός χωρίς 0 στην αρχή (1 - 31)

5)dd

όπως η d αλλά με αρχικό 0 αν χρειάζεται

6)ddd

ημέρα με τρία γράμματα

7)dddd

ημέρα με όνομα

8)dddddd

πλήρες ημερομηνία με ονόματα

9)w

αριθμός μέρα με νούμερο, 1 για Κυριακή και 7 για Σάββατο

10)ww

Αριθμός εβδομάδας έτους (1-54)

11)m

Αριθμός μήνα χωρίς αρχικό 0 (1-12))
12)mm
όπως το m αλλά με αρχικό 0
13)mmm
μήνας με τρία γράμματα
14)mmmm
ολόκληρο το όνομα του μήνα
15)q
Αριθμός τριμήνου του έτους(1-4)
16)y
Αριθμός μέρας του έτους (1-366)
17)yy
έτος με μόνο τα δυο τελευταία νούμερα
18)yyyy
αριθμός έτους (100-9999)
19)h
ώρα χωρίς αρχικό 0 (0 - 23)
20)hh
ώρα με αρχικό 0 αν χρειαστεί (0 - 23)
21)n
λεπτά χωρίς αρχικό 0 (0 - 59)
22)nn
λεπτά με αρχικό 0 αν χρειαστεί (00 - 59)
23)s
δευτερόλεπτα χωρίς αρχικό 0 (0 - 59)
24)ss
δευτερόλεπτα με αρχικό 0 αν χρειαστεί (00 - 59)
25)tttt
ώρα πλήρης μορφή
26)AM/PM
ενδεικτικό δωδεκάωρου με κεφαλαία
27)am/pm
ενδεικτικό δωδεκάωρου με πεζά
28)A/P
Ενδεικτικό δωδεκάωρου με ένα γράμμα
29)AMPM
Χρήση ενδεικτικού δωδεκάωρου του συστήματος

Αναγνωριστικό: ΨΕΥΔΕΣ [FALSE _2]
ΔΙΝΕΙ 0
ΟΠΩΣ Η ΨΕΥΔΗΣ

Αναγνωριστικό: ΨΕΥΔΗΣ [FALSE]

Δίνει τιμή 0

(υπάρχει και η ΨΕΥΔΕΣ - εκδ.4)

Αναγνωριστικό: ΨΗΦΙΟ [BYTE]

χρησιμοποιείται στην εντολή ΑΡΧΕΙΟ

Επίσης χρησιμοποιείται στην εντολή Δομή και στο αντικείμενο Διάρθρωση και στην Επιστροφή σε διάρθρωση, όπως και στην Εκφρ() όταν έχουμε ανάγνωση από θέση μνήμης.

Ομάδα: ΕΚΤΥΠΩΣΕΙΣ - PRINTINGS

Αναγνωριστικό: ΕΚΤΥΠΩΣΗ [PRINTING]

ΕΚΤΥΠΩΣΗ ΝΑΙ

ΕΚΤΥΠΩΣΗ ΔΙΕΚΟΨΕ

ΣΕΛΙΔΑ

ΕΚΤΥΠΩΣΗ ΟΧΙ

Μεταξύ των δυο αυτών εντολών μπορούμε να τυπώνουμε στον τρέχον εκτυπωτή.

Δες ΣΕΛΙΔΑ

Αυτή η εντολή υπάρχει λόγω συμβατότητας με προηγούμενες εκδόσεις.

Δες ΕΚΤΥΠΩΤΗΣ

Αναγνωριστικό: ΕΚΤΥΠΩΤΗΣ [PRINTER]

1) Στέλνουμε ένα τμήμα για εκτύπωση

ΕΚΤΥΠΩΤΗΣ {

...

}

2) Όπως η 1 αλλά αλλάζουμε και το μέγεθος των γραμμών

ΕΚΤΥΠΩΤΗΣ 10 {

...

}

3) Αλλαγή ιδιοτήτων εκτυπωτή

ΕΚΤΥΠΩΤΗΣ !

Η M2000 κρατάει για κάθε εκτυπωτή στοιχεία, τα οποία μπορεί να διαβάσει οποτεδήποτε. Αλλά μπορεί επίσης να τα αλλάξει ή ακόμα καλύτερα μπορεί να τα αποθηκεύσει χωριστά και να τα χρησιμοποιεί όποτε είναι αναγκαίο χωρίς να χρειάζεται ο χρήστης να ξαναρυθμίζει τον εκτυπωτή μέσω του διαλόγου των ιδιοτήτων. Δες το ΙΔΙΟΤΗΤΕΣ

4) Αλλαγή εκτυπωτή τρόπος με επιλογή από λίστα

ΕΚΤΥΠΩΤΗΣ ?

Ο αρχικός εκτυπωτής που δείχνει η λίστα είναι ο προεπιλεγμένος από τα Windows μπορούμε να διαβάσουμε το όνομα του εκτυπωτή από την εκτυπωτής\$

αφού ανοίξει μια φορά η λίστα θα γεμίσει ο πίνακας Επιλογή\$()

Η μεταβλητή επιλογή θα είναι 0 αν δεν έχουμε κάνει επιλογή (πχ πατήσαμε το Esc)

5) Αλλαγή εκτυπωτή τρόπος με απ' ευθείας επιλογή

ΕΚΤΥΠΩΤΗΣ 12, αλφαριθμητικό

το αλφαριθμητικό πρέπει να περιέχει το όνομα, διάστημα, παρένθεση, όνομα πόρτας, παρένθεση

παράδειγμα

ΕΚΤΥΠΩΤΗΣ?

ΤΥΠΩΣΕ ΕΠΙΛΟΓΗ\$(ΕΠΙΛΟΓΗ)

ΤΥΠΩΣΕ "-----"

ΕΚΤΥΠΩΤΗΣ 12, ΕΠΙΛΟΓΗ\$(ΕΠΙΛΟΓΗ)

ΕΠΙΛΟΓΗ!

6) Αναφορά επιλεγμένου εκτυπωτή

ΕΚΤΥΠΩΤΗΣ

Τυπώνει τον επιλεγμένο εκτυπωτή και αλλάζει γραμμή

? εκτυπωτής\$ \\ δίνει το όνομα και τη θύρα του εκτυπωτή.

Αναγνωριστικό: ΙΔΙΟΤΗΤΕΣ [PROPERTIES]

B\$=ΙΔΙΟΤΗΤΕΣ\$

ΕΚΤΥΠΩΤΗΣ !

A\$=ΙΔΙΟΤΗΤΕΣ\$

ΙΔΙΟΤΗΤΕΣ B\$

ΕΚΤΥΠΩΤΗΣ !

βλέπουμε τις ιδιότητες που είχε ο εκτυπωτής όπως τις είδαμε την πρώτη φορά.

Αναγνωριστικό: ΟΡΙΑ.ΕΚΤΥΠΩΤΗ [PRINTER.MARGINS]
Ορια.Εκτυπωτή (Όρια.Εκτύπωσης)
Ορια.Εκτυπωτή ' μηδενίζει τα όρια
Ορια.Εκτυπωτή ΔεξίΌριο
Ορια.Εκτυπωτή ΔεξίΌριο, ΠάνωΌριο
Ορια.Εκτυπωτή ΔεξίΌριο, ΠάνωΌριο, ΑριστερόΌριο
Ορια.Εκτυπωτή ΔεξίΌριο, ΠάνωΌριο, ΑριστερόΌριο, ΚάτωΌριο

Αναγνωριστικό: ΣΕΛΙΔΑ [PAGE]
ΣΕΛΙΔΑ

Με αυτήν την εντολή λέμε στον εκτυπωτή ό,τι έχουμε στείλει σ' αυτόν να το γράψει σε μια σελίδα. Μετά ότι θα γράψουμε θα εμφανισθεί στην επόμενη ΣΕΛΙΔΑ.

ΣΕΛΙΔΑ 1 ' Ζητάμε μια σελίδα portrait (το μήκος μεγαλύτερο από το πλάτος)
ΣΕΛΙΔΑ 0 ' Ζητάμε μια σελίδα landscape (το μήκος μικρότερο από το πλάτος)
ΣΕΛΙΔΑ ΚΑΘΕΤΗ
ΣΕΛΙΔΑ ΟΡΙΖΟΝΤΙΑ