

Correction Révision – contrôle de saisie

1. Remplir une variable par un entier composé de 4 chiffres.
2. Remplir une variable par un entier ne contenant pas le chiffre 0.
3. Remplir une variable par un entier ou chaque chiffre possède la même parité que sa position.
4. Remplir une variable par un entier symétrique (exemples : 1221, 131, 145541)
5. Remplir une variable par un entier composé de chiffres distincts.

Caractère

6. Remplir une variable par une lettre.
7. Remplir une variable par une consonne.
8. Remplir une variable par un symbole.

Chaîne

9. Remplir une variable par une chaîne de longueur 10 caractères.
10. Remplir une variable par une chaîne composée uniquement de chiffres.
11. Remplir une variable par une chaîne composée uniquement de lettres.
12. Remplir une variable par une chaîne où chaque deux mots sont séparés par un seul espace.
13. Remplir une variable par une chaîne où chaque mot finit par un point.
14. Remplir une variable par une chaîne dont le nombre de mots est impair.
15. Remplir une variable par une chaîne où chaque mot commence et finit par la même lettre (deux mots sont séparés par un seul espace).
16. Remplir une variable par une chaîne contenant toutes les voyelles.
17. Remplir une variable par une chaîne contenant 26 lettres distinctes minuscules prises au hasard.

Tableau

18. Remplir un tableau de n entiers positifs.
19. Remplir un tableau par n entiers d'une manière symétrique.
20. Remplir un tableau par n nombres binaires.
21. Remplir un tableau par n nombres hexadécimaux.
22. Remplir un tableau par n lettres.
23. Remplir un tableau par n entiers distincts.
24. Remplir un tableau par n entiers aléatoires entre 5 et 50.
25. Remplir un tableau par n réels aléatoires entre 1 et 20.
26. Remplir un tableau par n lettres majuscules d'une manière aléatoire.
27. Remplir un tableau contenant uniquement les chiffres 1, 2 et 3 et deux cases successives sont différentes.
28. Remplir un tableau par des entiers, la saisie s'arrête par l'introduction d'une valeur négative.
29. Remplir un tableau par n chaînes ou la longueur de chaque chaîne ne dépasse pas sa position.
30. Remplir un tableau par n chaînes de même longueur.
31. Remplir un tableau par n entiers strictement croissant.
32. Remplir un tableau par n entiers dont l'utilisateur choisira leur emplacement dans le tableau (afficher à chaque fois les emplacements libres).

Contrôle de saisie

N°	Traitement	
1	0-Def proc remplir (var n : entier) 1-répéter Lire(n) Jusqu'à (n>=1000) et (n<=9999) 2- Fin remplir	
2	0-Def proc remplir (var n : entier) 1-répéter Lire(n) Convch(n,ch) Jusqu'à pos("0",ch)=0 2- Fin remplir	
3	0-Def proc remplir (var n : entier) 1-répéter Lire(n) Convch(n,ch) Jusqu'à Fn verif (ch) 2- Fin remplir	0-Def Fnverif(ch :chaîne):Booléen 1-i=0 Répéter i=i+1 valeur(ch[i],x,e) v = x mod 2 = i mod 2 jusqu'à (v=faux) ou (i= long(ch)) 2- verif=v 3- Fin verif
4	0-Def proc remplir (var n : entier) 1-répéter Lire(n) Convch(n,ch) Jusqu'à Fn verif (ch) 2- Fin remplir	0-Def Fnverif(ch :chaîne):Booléen 1-i=0 Répéter i=i+1 v = ch[i] = ch[long(ch)-i+1] jusqu'à (v=faux) ou (i= long(ch)) 2- verif=v 3- Fin verif
5	0-Def proc remplir (var n : entier) 1-répéter Lire(n) Convch(n,ch) Jusqu'à Fn verif (ch) 2- Fin remplir	0-Def Fnverif(ch :chaîne):Booléen 1-i=0 Répéter i=i+1 v = pos(ch[i],ch) = i jusqu'à (v=faux) ou (i= long(ch)) 2- verif=v 3- Fin verif
6	0-Def proc remplir (var c : caractère) 1-répéter Lire(c) Jusqu'à majus(c) dans ["A".."Z"] 2- Fin remplir	

7	0-Def proc remplir (var c : caractère) 1-répéter Lire(c) Jusqu'à (majus(c) dans ["B".."Z"]) et Non (majus(c) dans ["E","I","O","U","Y"]) 2- Fin remplir	
8	0-Def proc remplir (var c : caractère) 1-répéter Lire(c) Jusqu'à non(majus (c) dans ["A".."Z","0".."9"]) 2- Fin remplir	
9	0-Def proc remplir (var ch : chaîne) 1-répéter Lire(ch) Jusqu'à long(ch)=10 2- Fin remplir	
10	0-Def proc remplir (var ch : chaîne) 1-répéter Lire(ch) Jusqu'à FN verif(ch) 2- Fin remplir	0-Def Fnverif(ch :chaîne):Booléen 1-i=0 Répéter i=i+1 v=ch[i] in ["0".."9"] jusqu'à (v=faux) ou(i= long(ch)) 2- verif=v 3- Fin verif
11	0-Def proc remplir (var ch : chaîne) 1-répéter Lire(ch) Jusqu'à FN verif(ch) 2- Fin remplir	0-Def Fnverif(ch :chaîne):Booléen 1-i=0 Répéter i=i+1 v=majus(ch[i]) in ["A".."Z"] jusqu'à (v=faux) ou(i= long(ch)) 2- verif=v 3- Fin verif
12	0-Def proc remplir (var ch : chaîne) 1-répéter Lire(ch) Jusqu'à pos (" ", ch)=0 2- Fin remplir	
13	0-Def proc remplir (var ch : chaîne) 1-répéter Lire(ch) Jusqu'à FN verif(ch) 2- Fin remplir	0-Def Fnverif(ch :chaîne):Booléen 1-ch=ch+" " Répéter P = pos (" ",ch) v=ch[p-1]="." Efface(ch,1,p) jusqu'à (v=faux) ou(long(ch)=0) 2- verif=v 3- Fin verif

14	0-Def proc remplir (var ch : chaîne) 1-répéter Lire(ch) Jusqu'à Fn verif (ch) mod 2 = 1 2- Fin remplir	0-Def Fnverif(ch :chaîne):entier 1-v $\square$ 1 Répéter P $\square$ pos (" ",ch) Si p<> 0 alors v $\square$ v+1 fin si Efface(ch,1,p) jusqu'à (p=0) 2- verif $\square$ v 3- Fin verif
15	0-Def proc remplir (var ch : chaîne) 1-répéter Lire(ch) Jusqu'à FN verif(ch) 2- Fin remplir	0-Def Fnverif(ch :chaîne):Booléen 1-ch $\square$ ch+" " Répéter P $\square$ pos (" ",ch) v $\square$ majus(ch[p-1]) =majus(ch[1]) Efface(ch,1,p) jusqu'à (v=faux) ou(long(ch)=0) 2- verif $\square$ v 3- Fin verif
16	0-Def proc remplir (var ch : chaîne) 1-répéter Lire(ch) Jusqu'à FN verif(ch) 2- Fin remplir	0-Def Fnverif(ch :chaîne):Booléen 1-voy $\square$ 'aeiouy' i $\square$ 0 Répéter i $\square$ i+1 v $\square$ (pos(voy[i],ch)<>0) or (pos(majus(voy[i]),ch)<>0) jusqu'à (v=faux) ou(i= 6) 2- verif $\square$ v 3- Fin verif
17	0-Def proc remplir (var ch : chaîne) 1-ch $\square$ "" pour i de 1 à 26 faire Répéter c $\square$ aléa(26)+97 Jusqu'à pos(c,ch) =0 Ch $\square$ ch+c Fin pour 2- Fin remplir	
18	0-Def proc remplir (var t : tab ; n : entier) 1-pour i de 1 à n faire répéter Lire(t[i]) Jusqu'à t[i]>=0 Finpour 2-fin remplir	

19	0-Def proc remplir (var t : tab ; n : entier) 1-pour i de 1 à n div 2 faire Lire(t[i]) t[n-i+1] $\square$ t[i] Finpour 2- si n mod 2 = 1 alors lire(t[n div 2+1]) Fin si 2-fin remplir	
20	0-Def proc remplir (var t : tab ; n : entier) 1-pour i de 1 à n faire répéter Lire(t[i]) Jusqu'à Fn verif(t[i]) Finpour 2-fin remplir	0-Def Fnverif(ch :chaîne):Booléen 1-i $\square$ 0 Répéter i $\square$ i+1 v $\square$ ch[i] dans ["0","1"] jusqu'à (v=faux) ou(i= long(ch)) 2- verif $\square$ v 3- Fin verif
21	0-Def proc remplir (var t : tab ; n : entier) 1-pour i de 1 à n faire répéter Lire(t[i]) Jusqu'à Fn verif(t[i]) Finpour 2-fin remplir	0-Def Fnverif(ch :chaîne):Booléen 1-i $\square$ 0 Répéter i $\square$ i+1 v $\square$ majus(ch[i]) dans["0".."9","A".."F"] jusqu'à (v=faux) ou(i= long(ch)) 2- verif $\square$ v 3- Fin verif
22	0-Def proc remplir (var t : tab ; n : entier) 1-pour i de 1 à n faire répéter Lire(t[i]) Jusqu'à majus(t[i]) in ["A".."Z"] Finpour 2-fin remplir	
23	0-Def proc remplir (var t : tab ; n : entier) 1-lire(t[1]) pour i de 2 à n faire répéter Lire(t[i]) Jusqu'à Fn verif (t[i] ,t,i-1)=faux Finpour 2-fin remplir	0-Def Fnverif(x:entier ;t:tab ; k:entier):Booléen 1-i $\square$ 0 Répéter i $\square$ i+1 v $\square$ t[i] =x jusqu'à (v=vrai)ou(i= k) 2- verif $\square$ v 3- Fin verif
24	0-Def proc remplir (var t : tab ; n : entier) 1-pour i de 1 à n faire t[i] $\square$ aléa (46)+5 Finpour 2-fin remplir	

SHS

25	0-Def proc remplir (var t : tab ; n : entier) 1-pour i de 1 à n faire t[i] ← aléa (46)+5 Finpour 2-fin remplir
26	0-Def proc remplir (var t : tab ; n : entier) 1-pour i de 1 à n faire t[i] ← CHR(aléa (26)+65) Finpour 2-fin remplir
27	0-Def proc remplir (var t : tab ; n : entier) 1-répéter Lire(t[1]) Jusqu'à t[1] dans [1..3] pour i de 2 à n faire répéter Lire(t[i]) Jusqu'à (t[i] dans [1..3]) et (t[i] <> t[i-1]) Finpour 2-fin remplir
28	0-Def proc remplir (var t : tab ; var n : entier) 1-n ← 0 Répéter Lire(x) Si x ≥ 0 alors n ← n+1 t[n] ← x fin si jusqu'à x < 0 Finpour 2-fin remplir
29	0-Def proc remplir (var t : tab ; n : entier) 1-pour i de 1 à n faire répéter Lire(t[i]) Jusqu'à long(t[i]) ≤ i Finpour 2-fin remplir
30	0-Def proc remplir (var t : tab ; n : entier) 1- Lire(t[1]) pour i de 2 à n faire répéter Lire(t[i]) Jusqu'à long(t[i])=long(t[1]) Finpour 2-fin remplir

31	0-Def proc remplir (var t : tab ; n : entier) 1- Lire(t[1]) pour i de 2 à n faire répéter Lire(t[i]) Jusqu'à t[i]>t[i-1] Finpour 2-fin remplir	
32	0-Def proc remplir (var t : tab ; n : entier) 1-pour i de 1 à n faire P[i] ← i Fin pour 2-pour i de 1 à n faire Répéter Ecrire("les emplacement libre sont :") Affiche(p,n) Ecrire ("saisir la position de l'entier") Lire(k) Jusqu'à (fn verif (k,p,n)) Ecrire("saisir l'entier") Lire(t[k]) P[k] ← 0 Finpour 2-fin remplir	0-Def Fnverif(k:entier ; p:tab ; n:entier):Booléen 1-i ← 0 Répéter i ← i+1 v ← p[i] = k jusqu'à (v=vrai) ou (i= n) 2- verif ← v 3- Fin verif 1- Def proc affiche (p :tab ; n : entier) 2- Pour i de 1 à n faire Si p[i] <> 0 alors ecrire(p[i],'-') Fin si Fin pour 2-Fin affiche

Exercice 7 (chaîne) :

```

program ex7chaîne;
uses wincrt;
var
  ch1,ch2:string;
functionverif( ch:string):boolean;
  var i:integer; v:boolean;
begin
  i:=0;
  repeat
    i:=i+1;
  v:= ch[i] in ['0'..'9'] ;
  until (v=false) or ( i=length(ch));
  verif:=v;
end;
procedureremplir(varch:string);
begin
  repeat
    write('ch :' );
    readln(ch);
  until (length(ch) <=250) and (verif(ch));
end;

functionajuste(ch1,ch2:string): string;
var i,l:integer;
  begin
    l:=length(ch2)-length(ch1);
    for i:=1 to l do
      ch1:='0'+ch1;
    ajuste:=ch1;
  end;
function somme (ch1,ch2:string):string;
var
  r,i,x,y,e,s :integer;
  res,chs:string;
begin
  if length(ch1) <length(ch2)
  thench1:=ajuste(ch1,ch2)
  else ch2:=ajuste(ch2,ch1);
  r:=0; res:="";

```

```

for i:=length(ch1) downto 1 do
begin
  val(ch1[i],x,e);
  val(ch2[i],y,e);
  s:=x+y+r;
  if s>9 then r:=1
  else r:=0;
  str (s mod10 ,chs);
  res:=chs+res;
end;
if r= 1 thenres:='1'+res;
somme:=res;

end;

begin
  remplir(ch1);
  remplir(ch2);
  write(somme(ch1,ch2));
end.

```

Exercice 8 (tableau) :

```

program ex8;
uses wincrt;
type tab=array[1..50] of real;
var n:integer;
t:tab ;
procedure saisir(var n:integer);
begin
repeat
write('saisir taille');
readln(n);
until n in[1..10];
end;
procedure remplir(var t:tab; n:integer);
var i:integer;
begin
for i:=1 to n do
begin
write('t[' ,i ,']');
readln(t[i]);
end;
end;

```

```

procedure trait(t:tab ; n:integer);
var
i,j,r:integer;
begin

for i:=1 to n do
begin
r:=1;
for j:=1 to n do
begin
if t[j] > t[i] then
begin
r:=r+1 ;
end;
end;
writeln('l'élève ',i,':moyenne:',t[i], 'rang',m);
end;
end;
begin
saisir(n);
remplir (t,n);
trait(t,n);
end.

```

Exercice 6 (chaîne)

```

program ex6;
uses wincrt;
var
g,ch:string;
function verif ( ch:string):boolean;
var i:integer; v:boolean;
begin
i:=0;
repeat
i:=i+1;
v:= ch[i] in ['0','1'] ;
until (v=false) or ( i=length(ch));
verif:=v;
end;
procedure remplir(var ch:string);
var c:char; i:integer;
begin
repeat
write('ch : '); readln(ch);
until (ch<>'') and (verif(ch)) and (length(ch)=8);
end;

```

```

function codec (ch:string):string;
var res,nbch:string; i,j,nb:integer;
begin
res:=""; i:=1;
repeat
j:=i+1; nb:=1;
while(ch[i]=ch[j]) and (j<=8) do
begin
nb:=nb+1; j:=j+1;
end;
if nb=1 then begin
res:=res+ch[i];
i:=i+1;
end
else begin
str(nb,nbch);
res:=res+nbch+ch[i];
i:=j;
end;
until i>8; codec:=res; end;
begin remplir(ch);
write(codec(ch)); end.

```

SHS