
Problèmes de révision

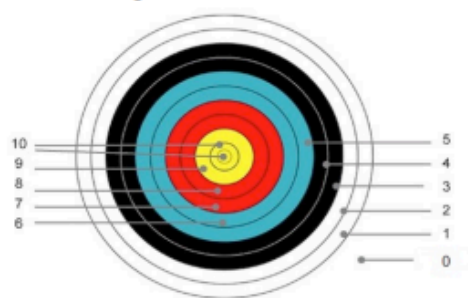
Problème1 :

Le jeu de tir à l'arc consiste à tirer des flèches en utilisant un arc vers un endroit cible.

Un club de tir à l'arc organise une compétition entre ses joueurs. Cette compétition consiste à donner à chaque joueur trois essais de tir.

A chaque tir, le joueur gagne de **0** à **10** points selon l'endroit touché par la flèche sur une cible composée de **10** cercles concentriques comme l'illustre la figure ci-contre.

Après le 3^{ème} essai, un score est calculé pour chaque joueur représentant la somme des résultats de ses trois essais, et ce pour classer les joueurs.



On se propose d'écrire un programme Pascal permettant :

- de remplir un tableau **A** par **N** noms de joueurs participant à cette compétition (avec $2 \leq N \leq 20$),
- de calculer le score de chaque joueur sachant que le résultat d'un essai est un entier positif inférieur ou égal à **10**,
- d'afficher la liste des joueurs ainsi que les scores correspondants en commençant par le meilleur.

Exemple :

Pour **N** = 4 et le tableau **A** suivant :

A	Mohamed ABIDI	Fatma AYADI	Anis OMRI	Imen LOUIZ
	1	2	3	4

En considérant que les résultats des trois essais ainsi que les scores calculés des quatre joueurs sont les suivants :

Joueur	Essai n°1	Essai n°2	Essai n°3	Score
Mohamed ABIDI	8	10	9	27
Fatma AYADI	10	7	10	27
Anis OMRI	7	9	7	23
Imen LOUIZ	9	9	10	28

La liste des joueurs affichée sera la suivante :

Imen LOUIZ avec un score de 28
Mohamed ABIDI avec un score de 27
Fatma AYADI avec un score de 27
Anis OMRI avec un score de 23

Problème2 :

Un spécialiste en psychologie analytique veut faire une étude sur les heures qui se disent heures miroirs et ce, à partir des heures de naissances des nouveau-nés. Les heures miroirs sont classées selon trois types à savoir les heures miroirs doublées, les heures miroirs triplées et les heures miroirs inversées. Pour commencer cette étude, il a demandé auprès de l'administration d'un hôpital de lui fournir, pour une journée donnée, la liste des noms et prénoms des nouveau-nés ainsi que les heures de leurs naissances.

Selon la représentation horaire HH:MM, l'heure miroir est dite :

- **doublée** si le nombre qui représente l'heure est le même que celui qui représente les minutes (*exemples* 13:13, 22:22, 15:15, ...)
- **triplée** s'il existe une séquence de trois chiffres identiques et consécutifs (*exemples* 22:22, 22:29, 10:00, ...)
- **inversée** si les chiffres qui représentent les minutes sont l'inverse de ceux qui représentent l'heure (*exemples* 13:31, 14:41, 22:22, ...)

NB : Une heure miroir peut être à la fois doublée, triplée et inversée.

Pour faire cette étude, on se propose d'écrire un programme Pascal qui permet de remplir deux tableaux **NP** et **HN** respectivement par :

- les noms et les prénoms des **N** nouveau-nés de la journée,
- l'heure de naissance de chacun,

puis d'afficher :

- la liste des nouveau-nés ayant une heure de naissance miroir, en précisant pour chacun le type de l'heure miroir correspondant (doublée, triplée ou inversée),
- le nombre des nouveau-nés ayant l'heure de naissance miroir doublée,
- le nombre des nouveau-nés ayant l'heure de naissance miroir triplée,
- le nombre des nouveau-nés ayant l'heure de naissance miroir inversée.

Exemple :

Pour **N** = 10 et les tableaux **NP** et **HN** suivants :

NP	Rania	Karim	Hatem	Sahar	Maha	Fedi	Amin	Dina	Samir	Abir
	Kefi	Salhi	Krid	Feki	Sessi	Krayem	Douaji	Selmi	Siala	Zekri
HN	06:05	11:14	12:23	13:13	14:44	15:51	20:33	21:11	22:22	23:07

Le programme affiche :

Les nouveau-nés ayant l'heure de naissance miroir sont :

Karim Salhi : heure miroir triplée
Sahar Feki : heure miroir doublée
Maha Sessi : heure miroir triplée
Fedi Krayem : heure miroir inversée
Dina Selmi : heure miroir triplée
Samir Siala : heure miroir doublée
Samir Siala : heure miroir triplée
Samir Siala : heure miroir inversée

Le nombre des nouveau-nés ayant l'heure de naissance miroir doublée est 2.

Le nombre des nouveau-nés ayant l'heure de naissance miroir triplée est 4.

Le nombre des nouveau-nés ayant l'heure de naissance miroir inversée est 2.

Problème3 :

On appelle "**WORD CHAIN**" une succession de chaînes ordonnées alphabétiquement où chaque chaîne diffère de celle qui la précède par un seul caractère.

Exemple :

La succession suivante de mots est un "**WORD CHAIN**" : HEAL HEAD DEAD DEAA
BEAA

En effet, cette succession est ordonnée selon un ordre alphabétique décroissant et chaque mot diffère d'un seul caractère par rapport au mot qui le précède.

On se propose d'écrire un programme permettant de vérifier si un ensemble de **N** chaînes de caractères forme un "**WORD CHAIN**". Sachant que :

- Les **N** chaînes de caractères sont de même longueur **M** donnée (avec $3 < M < 6$ et $4 < N < 10$).
- Les chaînes sont distinctes.
- Chaque chaîne est formée uniquement par des lettres majuscules.

Travail à faire :

- 1) Ecrire une analyse du programme principal en décomposant le problème en modules.
- 2) Ecrire un algorithme pour chaque module envisagé, tout en dressant les tableaux de déclaration des objets

Problème4 :

Un laboratoire de recherche scientifique réalise des expériences en biologie. Une expérience est caractérisée par son nom et son résultat (échec, neutre ou succès). L'informatisation de la gestion des résultats des expériences de ce laboratoire demande l'utilisation des deux tableaux suivants :

TE (EXPÉRIENCE) : est un tableau contenant les noms des expériences. Le nom est une chaîne de caractères non vide et ne dépassant pas 30 caractères.

TR (RESULTAT) : est un tableau qui contient les résultats des différentes expériences codés de la façon suivante :

- -1 : si le résultat est un échec
- 0 : si le résultat est neutre
- 1 : si le résultat est succès

NB : TE [i] et TR [i] respectent respectivement le nom et le résultat de l'expérience N ° i

Ecrire un programme qui permet de :

- Lire un entier N (longueur des deux tableaux TE et TR) avec $1 \leq N \leq 100$.
- Remplir les tableaux TE et TR respectivement par le nom de chaque expérience et le résultat correspondant
- Réordonner le tableau TE selon le résultat obtenu (succès, neutre puis échec). Si plusieurs expériences ont abouti au même résultat, on les place selon l'ordre alphabétique de leur nom
- Afficher le tableau TE

Correction problème n°1

Nom : Problème1

Résultat = proc affiche(A,S,N)

A=Proc Tri(A,S,N)

S=Proc Score(A,S,N)

A,N=Proc Remplir(A,N)

Fin Problème1

TDNT

Type
Tabc=tableau de 50 chaînes
Tabe=tableau de 50 entiers

TDOG

O	T/N	R
A	Tabc	Contient les noms des joueurs
S	Tabe	Contient les scores
N	Entier	Contient le nombre des joueurs
Remplir	Procédure	Permet de remplir N et le tableau A
Tri	//	Classer les joueurs selon leurs scores
Affiche	//	Afficher la liste des joueurs
Score	//	Remplir les scores

0)def proc remplir(var A :tabc ; var n :entier)

1)répéter

Lire(n)

Jusqu'à n dans [2..20]

2) pour i de 1 à n faire

Répéter

Lire (A[i])

Jusqu'à **FN verif (A[i])**

Fin pour

3) fin remplir

```
1) def proc affiche (a :tabc ; s :tabe ; n :entier)
2) ecrire("la liste des joueurs :")
3) pour i de 1 à n faire
    ecrire (A[i], "avec un score de ", S[i])
    fin pour
4) fin affiche
```

```
1) def proc score( A :tabc ; var S :tabe ; n :entier)
2) pour i de 1 à n faire
    ecrire(A[i])
    s[i] ← 0  y ← 0 {2eme solution}
    pour j de 1 à 3 faire
        répéter
            ecrire("essai n° ",j," :")
            lire(x)
            jusqu'à x dans [0..10]
            s[i] ← s[i]+x  y ← y+x
    fin pour
    s[i] ← y
fin pour
3) fin score
```

```
1) def proc tri (var s :tabe ; var a :tabc ; n :entier)
2) répéter
    ok ← vrai
    pour i de 1 à n-1 faire
        si s[i] < s[i+1] alors ok ← faux
            X ← s[i]
            s[i] ← s[i+1]
            s[i+1] ← X
    ch ← a[i]
```

$a[i] \leq a[i+1]$

$a[i+1] \leq ch$

Finsi

TDOL

Fin pour

Jusqu'à ok=**Vrai**

2) fin tri

~~Vrai~~—faux 28/27/27/23

Vrai 28/27/27/23

Correction problème n°2

Nom : Problème2

Résultat = proc affiche(NP,HN,n)

NP=proc remplir1(NP,n)

HN=proc remplir2(HN,n)

N=proc taille(n)

Fin Problème2

TDNT

Type
Tab=tableau de 50 chaînes

TD OG

O	T/N	R
HN	Tab	Contient les heures de naissance
NP	Tab	Contient les noms des nouveau-nés
N	Entier	Contient la taille des tableaux
Affiche	Procédure	Afficher les résultats
Remplir1	//	Remplir le tableau NP
Remplir2	//	Remplir le tableau HN
Taille	//	Remplir n

1) Def proc taille(**var** n :entier)

2) Répéter

Lire(n)

Jusqu'à n dans [3..20]

3) Fin taille

1) Def proc remplir1(**var** t :tab ; n :entier)

2) Pour i de 1 à n faire

Répéter

Lire (t[i])

Jusqu'à(**FN** verif (t[i])) et (long(t[i])<=30)

Fin pour

3) Fin remplir1

TDOL

0)def proc remplir2 (var t :tab ; n :entier) TDOL

1) Pour i de1 à n faire

 Répéter

 Lire (t[i])

 Valeur(souschaîne(t[i],1,2),a,e)

 valeur(souschaîne(t[i],4,2),b,e)

 Jusqu'à (t[i][3]=" :") et (a dans [0..23])

 Et (b dans [0..59]) et (long(t[i])=5)

 Fin pour

2)Fin remplir

0) def preoc affiche (a :tab ; h :tab ; n :entier)

1) écrire("les nouveau-nés ayant l'heure de naissance miroir sont :")

2)d₀ t₀ y₀

pour i de 1à n faire

Si souschaîne(h[i],1,2)= souschaîne(h[i],4,2)

 alors d₀d+1

 ecrire(a[i]," : heure miroir doublée")

Fin si

Ch₀h[i]

Si (ch[1]=ch[2])et(ch[2]=ch[4])) ou (ch[2]=ch[4])et(ch[4]=ch[5]))

 Alors t₀t+1

 ecrire(a[i]," : heure miroir triplée")

Fin si

Si (ch[1]=ch[5]) et (ch[2]=ch[4])

 Alors y₀y+1

 ecrire(a[i]," : heure miroir inversée")

Fin si

Fin pour

3) écrire("le nombre des nouveau-nés ayant l'heure de naissance miroir doublée est ",d)

4) écrire("le nombre des nouveau-nés ayant l'heure de naissance miroir triplée est ",t)

5) écrire("le nombre des nouveau-nés ayant l'heure de naissance miroir inversée est ",y)

6) fin affiche

TDOL