

# Module 7 : Les Tableaux

## Situations

### Anniversaire de Yassine

A l'occasion de son anniversaire le petit Yassine va distribuer des petites boîtes surprises contenant des bonbons à ses amis.



Figure 1, Boîtes surprises bonbons

Les boîtes ne contiennent pas un nombre équitable de bonbons car le petit Yassine a imaginé la méthode de distribution suivante de bonbons.

Les boîtes sont numérotées de 0 à  $na-1$ , avec  $na$  étant le nombre d'amis de Yassine qui sont invités à l'anniversaire.

Pour chaque bonbon l'enfant :

- tire un nombre au hasard,  $n_{bh}$ , compris entre 0 et  $na-1$
- met le bonbon dans la boîte numéro  $n_{bh}$

Pour quelques bonbons le jeu semble amuser l'enfant, mais très vite le petit se lasse et il ne veut plus continuer son jeu.

Observant son petit frère, Youssef, un lycéen en 2TI a décidé de l'aider. Pour cela, il a pensé à écrire un programme qui :

- Lit le nombre total de bonbons à distribuer,  $10 \leq nb \leq 200$
- Lit le nombre d'amis invités,  $3 \leq na \leq 10$
- Calcule, puis affiche le nombre de bonbons à mettre dans chacune des boîtes.

#### Exemple

```
Nombre de bonbons [10, 200] ? 30
Nombre d'amis [3, 10] ? 5
Boîte 0 contient 4 bonbons.
Boîte 1 contient 8 bonbons.
Boîte 2 contient 6 bonbons.
Boîte 3 contient 5 bonbons.
Boîte 4 contient 7 bonbons.
```

## Correction

```
from random import randint
nb = 0
while not 10 <= nb <= 200:
    nb = int(input("Nombre de bonbons [10, 200] ? "))
na = 0
while not 3 <= na <= 10:
    na = int(input("Nombre d'amis [3, 10] ? "))
boite = [0]*na
for i in range(nb):
    nbh = randint(0, na-1)
    boite[nbh] = boite[nbh] + 1
for i in range(na):
    print(f"Boîte {i} contient {boite[i]} bonbons.")
```

# Tombola

## Principe



Figure 2, Tombola

A l'occasion de son anniversaire, une entreprise organise un Tombola pour ces employés. Tous les employés peuvent s'inscrire au Tombola sans frais dans le but de gagner des prix de valeur.

## Deux gagnants / Deux Prix

On suppose qu'il y a deux participants et deux prix à gagner.

- iPad
- iPhone

On remarque que dans ce premier cas il existe deux possibilités.

Cas	1	2
P1	iPad	iPhone
P2	iPhone	iPad

Ecrire un programme pour traiter ce cas.

## Solution Algorithmique

Programme tombola\_v1

Début

Ecrire("Nom du 1er participant : "); Lire(p1)

Ecrire("Nom du 2nd participant : "); Lire(p2)

t ← alea(1, 2)

Si t = 1 Alors

g1 ← "iPad"

g2 ← "iPhone"

Sinon

g2 ← "iPad"

g1 ← "iPhone"

Fin Si

Ecrire(p1, "gagne", g1)

Ecrire(p2, "gagne", g2)

Fin

Objet	Type
p1, p2	chaîne
t	entier
g1, g2	chaîne

## Trois participants, trois gagnants

On suppose qu'il y a trois participants et trois prix à gagner. Les deux prix précédents et un Samsung S9.

On remarque que dans ce second cas il existe six possibilités.

Cas	1	2	3	4	5	6
P1	iPad	iPad	iPhone	iPhone	Samsung S9	Samsung S9
P2	iPhone	Samsung S9	iPad	Samsung S9	iPhone	iPad
P3	Samsung S9	iPhone	Samsung S9	iPad	iPad	iPhone

Ecrire un programme pour traiter ce cas.

## Solution Algorithmique

Programme tombola\_v2

Début

```
Ecrire("Nom du 1er participant : "); Lire(p1)
Ecrire("Nom du 2nd participant : "); Lire(p2)
Ecrire("Nom du 3ème participant : "); Lire(p3)
t ← alea(1, 6)
Si t = 1 Alors
    g1 ← "iPad"
    g2 ← "iPhone"
    g3 ← "Samsung S9"
Sinon Si t = 2 Alors
    g1 ← "iPad"
    g2 ← "Samsung S9"
    g3 ← "iPhone"
Sinon Si t = 3 Alors
    g1 ← "iPhone"
    g2 ← "iPad"
    g3 ← "Samsung S9"
Sinon Si t = 4 Alors
    g1 ← "iPhone"
    g2 ← "Samsung S9"
    g3 ← "iPad"
Sinon Si t = 5 Alors
    g1 ← "Samsung S9"
    g2 ← "iPhone"
    g3 ← "iPad"
Sinon
    g1 ← "Samsung S9"
    g2 ← "iPad"
    g3 ← "iPhone"
Fin Si
Ecrire(p1, "gagne", g1)
Ecrire(p2, "gagne", g2)
Ecrire(p3, "gagne", g3)
```

Fin

Objet	Type
p1, p2, p3	chaîne
t	entier
g1, g2, g3	chaîne

## Cas réel

Dans le cas réel le nombre de participants est toujours très grand par rapport au nombre de prix. Il existe 120 cas où cinq participants gagnent cinq prix. Dans ce cas le nombre de possibilités est très grand et il n'est pas pratique d'écrire un programme qui dénombre tous ces cas possibles.

On demande d'écrire un programme qui :

- Saisit le nombre de prix ( $1 \leq np \leq 10$ )
- Saisit les noms des prix
- Saisit le nombre de participants ( $np \leq n \leq 20$ )
- Saisit les noms des participants
- Effectue le tirage aux sorts
- Affiche le nom des gagnants ainsi que leurs prix

Pour simplifier la résolution de ce problème on peut, par exemple :

- Stocker les noms des  $np$  prix dans un tableau **prix**
- Stocker les noms des  $n$  participants dans un second tableau **noms**
- Mélanger l'ordre des participants d'une manière aléatoire
- Prendre les  $np$  premiers participants

## Solution

```

from random import randint
# Saisie du nombre de prix
while not 1 <= np <= 10:
    np = int(input("Donner le nombre de prix (1 ≤ np ≤ 10) ? "))
# Déclaration du tableau des prix
prix = [""] * np
# Saisie des prix
for i in range(np):
    prix[i] = input(f"Donner le nom du {i+1}ème prix ? ")
# Saisie du nombre de participants
while not np <= n <= 20:
    n = int(input(f"Donner le nombre de participants ({np} ≤ n ≤ 20) ? "))
# Déclaration du tableau de participants
noms = [""] * n
# Saisie des noms des participants
for i in range(n):
    noms[i] = input(f"Donner le nom du {i+1}ème participant ? ")
# Mélanger l'ordre des noms
for i in range(n):
    j = randint(0, n-1)
    noms[i], noms[j] = noms[j], noms[i]
# Affichage des gagnants
for i in range(np):
    print(noms[i], "gagne", prix[i])

```

## Renforcement

### Inversion Tableau

Ecrire un programme qui permet d'inverser les éléments d'un tableau t de n entiers (n≥2).

**Exemple :**

Exemple

```

Donner la taille du tableau (>=2) : 7
t [ 0 ]= 11
t [ 1 ]= 2
t [ 2 ]= 33
t [ 3 ]= 5
t [ 4 ]= 6
t [ 5 ]= 7
t [ 6 ]= 1
Tableau inversé : 1, 7, 6, 5, 33, 2, 11,

```

### Extrêmes d'un tableau

Ecrire un programme qui permet d'afficher le maximum et le minimum dans un tableau t de n entiers (n≥2).

**Exemple :**

Exemple

```

Donner la taille du tableau (>=2) : 5
t [ 0 ]= 11
t [ 1 ]= 2
t [ 2 ]= 33
t [ 3 ]= 4
t [ 4 ]= 67
Minimum : 2 - Maximum : 67

```

## Eclatement selon la parité

Ecrire un programme qui permet de saisir les éléments d'un tableau t de n entiers non nuls ( $n \geq 2$ ) et l'éclater en deux selon la parité de chaque nombre.

**Exemple :**

Exemple

```
Donner la taille du tableau (>=2) : 6
t [ 0 ]= 5
t [ 1 ]= 4
t [ 2 ]= 2
t [ 3 ]= 7
t [ 4 ]= 6
t [ 5 ]= 5
Nombres pairs : 4, 2, 6,
Nombres impairs : 5, 7, 5
```

## Eclatement selon le signe

Ecrire un programme qui permet de remplir un tableau t par n entiers non nuls ( $n \geq 2$ ) et d'éclater en deux tableaux : tn (contenant les éléments négatifs de t) et tp (contenant les éléments positifs de t).

**Exemple :**

Exemple

```
Donner la taille du tableau (>=2) : 10
t [ 0 ]= 3
t [ 1 ]= 2
t [ 2 ]= -5
t [ 3 ]= 4
t [ 4 ]= -7
t [ 5 ]= -8
Positifs : 3, 2, 4,
Négatifs : -5, -7, -8,
```

## Somme des carrés des chiffres

Ecrire un programme qui permet de saisir les éléments d'un tableau t de n entiers de trois chiffres ( $n \geq 2$ ) et de remplir un tableau t2 de la façon suivante : les éléments de t2 sont égaux à la somme des carrés des chiffres de t1.

**Exemple :**

Exemple

```
Donner la taille du tableau (>=2) : 6
t1 [ 0 ]= 254
t1 [ 1 ]= 121
t1 [ 2 ]= 133
t1 [ 3 ]= 325
t1 [ 4 ]= 150
t1 [ 5 ]= 222
t2 : 45, 6, 19, 38, 26, 12,
```

## Première occurrence

Ecrire un programme qui permet de remplir un tableau t par n entiers non nuls ( $n \geq 2$ ), de garder la première occurrence de chaque élément.

**Exemple :**Exemple

```
Donner la taille du tableau (>=2) : 10
t [ 0 ]= 10
t [ 1 ]= 7
t [ 2 ]= 9
t [ 3 ]= 7
t [ 4 ]= 10
t [ 5 ]= 7
t [ 6 ]= 7
t [ 7 ]= 8
t [ 8 ]= 8
t [ 9 ]= 8
Le résultat est : 10, 7, 9, 8,
```

## Vas et viens

Soient t1 et t2 deux tableaux d'entiers de taille n ( $n \geq 2$ ). Ecrire un programme qui permet de transférer les éléments de t1 et t2 selon le principe suivant : Les éléments de rang impair de t1 seront rangés dans t2 en ordre inverse (de droite à gauche) et les éléments de rang pair de t1 seront rangés dans t2 (de gauche à droite).

**Exemple :**Exemple

```
Donner la taille du tableau (>=2) : 9
t1 [ 0 ]= 12
t1 [ 1 ]= 13
t1 [ 2 ]= 41
t1 [ 3 ]= 5
t1 [ 4 ]= 19
t1 [ 5 ]= 13
t1 [ 6 ]= 4
t1 [ 7 ]= 8
t1 [ 8 ]= 15
t2 : 12, 41, 19, 4, 15, 8, 13, 5, 13,
```

## Intersection de deux tableaux

Ecrire un programme qui permet de ranger et d'afficher les éléments qui figurent dans les deux tableaux de n entiers a et b ( $n \geq 2$ ).

**Exemple :**Exemple

```
Donner la taille du tableau (>=2) : 4
A[ 0 ]= 1
A[ 1 ]= 2
A[ 2 ]= 4
A[ 3 ]= 7
B[ 0 ]= 7
B[ 1 ]= 3
B[ 2 ]= 2
B[ 3 ]= 1
Les éléments communs sont : 1, 2, 7,
```

## Le marchand de légumes

Le marchand de légumes vend les pommes dans des sacs de la façon suivante :

- Des sacs d'une seule pomme,
- Des sacs de deux pommes,

- Des sacs de quatre pommes,
- Des sacs de huit pommes,

Le marchand refuse de vendre deux sacs qui contiennent le même nombre de pommes à un seul client.

1. Calculer le nombre minimal de sacs achetés par les clients suivants :
  - Islem veut 14 pommes,
  - Ayoub veut 3 pommes,
  - Alaa veut 8 pommes,
  - Amine veut 10 pommes.
2. Comment faire si je veux acheter 16 pommes sachant qu'il est interdit de prendre plus d'un sac des quatre catégories disponibles ?
3. Je veux acheter  $n$  pommes,  $0 \leq n \leq 31$ . Comment calculer le nombre de sacs de chaque catégorie ?
4. Ecrire le programme qui calcule le nombre de sacs nécessaires pour satisfaire la commande d'un client donné.
5. Ecrire le programme qui calcule le nombre de pommes qui correspond à un nombre de sacs donné.

#### Exemple (premier programme) :

##### Exemple

```
Nombre de pommes : 25
un sac de 1 pommes
un sac de 8 pommes
un sac de 16 pommes
```

#### Exemple (second programme) :

##### Exemple

```
Nbre de sac de 1 pommes ? 1
Nbre de sac de 2 pommes ? 0
Nbre de sac de 4 pommes ? 1
Nbre de sac de 8 pommes ? 0
Nbre de sac de 16 pommes ? 0
Vous avez acheté : 5 pommes
```

## Conversion de la base 10 à la base 16

Ecrire un programme qui saisit un nombre  $n$  compris entre 0 et 255 et le convertit en hexadécimal.

#### Exemple :

##### Exemple

```
Nombre à convertir : 241
241(10) = F1(16)
```

## Fusion de deux tableaux de même longueur

Ecrire un programme qui fusionne deux tableaux  $t1$  et  $t2$ , de même longueur  $n$  dans un troisième tableau  $t3$  de longueur  $n3 = 2 * n$ .

La fusion se fera de la façon suivante :

- Placer les éléments de  $t1$  dans des cases impaires de  $t3$
- Placer les éléments de  $t2$  dans des cases paires de  $t3$

t1

9	2	4	7
---	---	---	---

0123

t2

7	3	2	1
---	---	---	---

0123

t3

7	9	3	2	2	4	1	7
---	---	---	---	---	---	---	---

01234567

**Exemple :**Exemple

```
Taille du tableau t1 (>=2) : 4
t1[ 0 ]= 9
t1[ 1 ]= 2
t1[ 2 ]= 4
t1[ 3 ]= 7
t2[ 0 ]= 7
t2[ 1 ]= 3
t2[ 2 ]= 2
t2[ 3 ]= 1
t3 : 7, 9, 3, 2, 2, 4, 1, 7
```

## Fusion de tableaux selon leurs parités

Ecrire un programme qui fusionne deux tableaux t1 et t2, de longueur n1 et n2 différentes, dans un troisième tableau t3 de longueur n3 = n1 + n2.

La fusion se fera de la façon suivante :

- Placer les éléments pairs de t1, puis les éléments pairs de t2
- Placer les éléments impairs de t1 et enfin les éléments impairs de t2

**Exemple :**Exemple

```
Taille du tableau t1 (>=2) : 4
t1[ 0 ]= 9
t1[ 1 ]= 2
t1[ 2 ]= 4
t1[ 3 ]= 7
Taille du tableau t2 (>=2) : 7
t2[ 0 ]= 7
t2[ 1 ]= 3
t2[ 2 ]= 2
t2[ 3 ]= 1
t2[ 4 ]= 5
t2[ 5 ]= 8
t2[ 6 ]= 4
t3 : 2, 4, 2, 8, 4, 9, 7, 7, 3, 1, 5
```

## Segmentation d'un tableau

Ecrire un programme qui segmente un tableau t de n entiers de la façon suivante :

- Les éléments inférieurs à t[0] sont placés au début du tableau
- Les éléments supérieurs à t[0] sont placés à la fin du tableau
- L'élément t[0] est placé au milieu du tableau

Les nombres du tableau sont des nombres aléatoires compris entre 1 et 999.

**Exemple :**Exemple

```
Taille du tableau t (>=2) : 10
Avant segmentation : [203, 32, 515, 19, 762, 762, 884, 157, 459, 337]
Après segmentation : [19, 32, 157, 203, 762, 884, 762, 459, 337, 515]
```

**Explication :**

203 32 515 19 762 762 884 157 459 337

Les éléments inférieurs à 203 sont en rouge



Les éléments supérieurs à 203 sont en vert

## Station de services la plus proche

Un automobiliste en manque de carburant veut aller à la station de services la plus proche pour faire le plein.

Ecrire un programme qui :

- Saisit le nombre de stations de services  $n$  qui se trouvent aux alentours
- Remplit un tableau  $dist$  par les distances des stations de services
- Affiche le numéro de station de la plus proche ainsi que sa distance.

## Doublure d'acteurs

Lors de tournage d'un film certaines scènes doivent être doublées par une autre personne qui possède une certaine ressemblance à l'acteur principal.

Disposant d'un tableau qui contient le degré de ressemblance de  $n$  candidats avec l'acteur principal sélectionner le candidat adéquat.

On suppose que la caractéristique de l'acteur est représentées par un entier entre 0 et 100.

On suppose que les caractéristiques des doublures sont aussi représentées par des entiers entre 0 et 100 stockés dans un tableau.

Ecrire un programme qui :

- saisit la caractéristique de l'acteur principal  $0 \leq car \leq 100$ ,
- saisit le nombre de doublures (nombre de personnes qui peuvent remplacer l'acteur)  $2 \leq n \leq 10$ ,
- remplit un tableau  $t$  par les caractéristiques des doublures,
- affiche le candidat qui ressemble le plus à l'acteur principal.

### Exemple

Caractéristique de l'acteur ? 78

Nombre de doublures ? 10

Caractéristiques des doublures : [58, 55, 79, 91, 44, 4, 44, 2, 4, 22]

Le candidat est n° 2 sa caractéristique est 79