

# DEVOIR SURVEILLÉ D'INFORMATIQUE — 8 JANVIER 2020

*Durée : 1 heure — Calculatrices interdites*

Les réponses aux questions 1 à 5 de l'exercice 4 sont à apporter sur le sujet, qui est donc à rendre avec votre copie.

---

## EXERCICE 1 — LISTE

1. Proposer un programme qui demande à l'utilisateur de saisir un entier strictement positif  $N$ , puis une liste de  $N$  nombres entiers, et qui affiche le maximum de ces entiers.
2. Modifier le programme précédent pour qu'il précise si au moins l'un des entiers saisis par l'utilisateur est pair.
3. Modifier le programme de la question 1 pour qu'il affiche la valeur moyenne des  $N$  nombres saisis par l'utilisateur.

## EXERCICE 2 — CHÂÎNES DE CARACTÈRES

Proposer un programme qui demande à l'utilisateur de saisir une phrase, et qui supprime les espaces de cette phrase avant d'afficher le résultat.

## EXERCICE 3 — CHÂÎNES DE CARACTÈRES

Dans cet exercice, on se propose de coder quelques fonctions portant sur les chaînes de caractères.

1. Ecrire un programme qui demande à l'utilisateur de saisir une chaîne de caractères (appelée **MOT**) et un caractère (appelé **CAR**) ; ce programme doit afficher **True** si le caractère **CAR** appartient à la chaîne **MOT**, et **False** sinon.
2. On souhaite écrire un script qui demande à l'utilisateur de saisir une chaîne de caractères **MOT** et un caractère **CAR**, et qui affiche la liste des positions où apparaît le caractère **CAR** dans la chaîne **MOT**.

Par exemple, si l'utilisateur saisit la chaîne "Quel beau texte", et le caractère "u", l'affichage produit doit être : [1,8].

Compléter le code ci-dessous (recopier et compléter sur votre copie seulement les lignes 6-7-8) :

```

1  print ('Saisissez un mot')
2  MOT =input()
3  print ('Saisissez un caractère')
4  CAR =input()
5  LPOS =[]
6  for # A Compléter
7  # A Compléter
8  # A Compléter
9  print (LPOS)
```

3. Comment compléter (rapidement) le programme précédent pour qu'il affiche également le nombre de fois où le caractère **CAR** apparaît dans la chaîne **MOT** ?
4. (\*) Proposer un programme demandant à l'utilisateur de saisir une phrase au clavier, et qui affiche la voyelle ( $a, e, i, o, u$  ou  $y$ ) qui a été la plus utilisée dans cette phrase, ainsi que le nombre de fois où elle apparaît.

**EXERCICE 4** — NOËL CANADIEN

Pour Noël, un groupe d'individus décide de s'offrir des cadeaux en suivant la procédure suivante : chaque membre du groupe choisit au hasard le nom d'un(e) autre membre (distinct de lui-même) à qui offrir un cadeau.

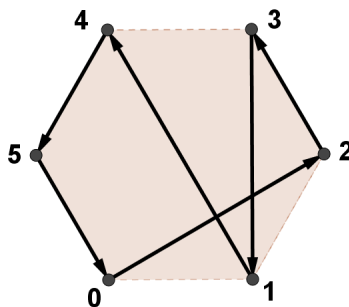
Le but de cet exercice est de représenter à l'aide de listes en Python le résultat d'une telle procédure, et d'étudier quelques programmes sur ce thème.

**Représentation et structure des données.** Nous supposons que les individus sont numérotés de 0 à  $(n-1)$ , où  $n$  est le nombre total d'individus. Tout au long du problème,  $n$  désignera un entier supérieur ou égal à 2.

Les résultats du tirage au sort pourront être représentés sous la forme d'une liste, comme expliqué dans les deux exemples ci-dessous.

**Exemple 1.** Pour fixer les idées, on suppose dans cet exemple que le groupe est constitué de 6 individus, représentés par les entiers de  $\llbracket 0, 5 \rrbracket$ . Supposons que le tirage au sort ait produit les résultats suivants :

- 0 offre un cadeau à 2;
- 1 offre un cadeau à 4;
- 2 offre un cadeau à 3;
- 3 offre un cadeau à 1;
- 4 offre un cadeau à 5;
- 5 offre un cadeau à 0.

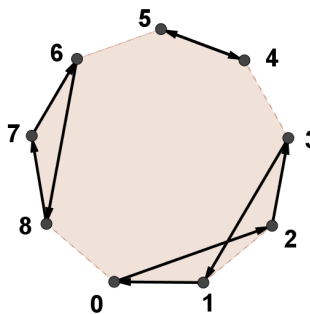


Ces résultats pourront être symbolisés par la liste :  $[0, 2, 3, 1, 4, 5]$ .

Nous appellerons **représentation** une telle liste.

**Exemple 2.** On suppose cette fois que le groupe est constitué de 9 individus. Il se peut que le tirage au sort produise des sous-ensembles disjoints d'individus, comme l'illustre le résultat du tirage au sort suivant :

- 0 offre un cadeau à 2;
- 1 offre un cadeau à 0;
- 2 offre un cadeau à 3;
- 3 offre un cadeau à 1;
- 4 offre un cadeau à 5;
- 5 offre un cadeau à 4;
- 6 offre un cadeau à 8;
- 7 offre un cadeau à 6;
- 8 offre un cadeau à 7.



Ces résultats seront alors symbolisés par la **représentation** :  $[0, 2, 3, 1, 'X', 4, 5, 'X', 6, 8, 7]$ .

Les caractères 'X' permettront donc de séparer les trois sous-listes  $[0, 2, 3, 1]$ ,  $[4, 5]$  et  $[6, 8, 7]$ . En outre, de telles sous-listes seront appelés des **cycles**, et plus précisément nous dirons que :

$[0, 2, 3, 1]$  est un **cycle de longueur 4**,  $[4, 5]$  un **cycle de longueur 2** et  $[6, 8, 7]$  un **cycle de longueur 3**.

**En résumé.**

- Dans un groupe de  $n$  individus, chaque individu est représenté par un unique entier compris entre 0 et  $(n - 1)$  (et réciproquement chaque entier de  $\llbracket 0, n - 1 \rrbracket$  correspond à un unique individu). Dans ce problème, on assimilera donc un individu du groupe à un entier compris entre 0 et  $(n - 1)$  ;
- Une **représentation** est une liste correspondant au résultat d'un tirage au sort ;
- La lecture de cette liste permet de savoir que l'entier (l'individu) situé à la  $i$ -ème place offre un cadeau à celui situé à la  $(i + 1)$ -ème place (sauf dans quelques cas particuliers qui ne vous ont pas échappé).
- Un **cycle de longueur**  $p$  est une sous-liste de  $p$  entiers (de  $p$  individus) qui s'offrent des cadeaux entre eux.
- Si un tirage au sort produit plusieurs cycles, ceux-ci seront séparés par un ' $X$ ' dans la **représentation**.

**Partie I - Généralités sur les représentations**

**QUESTION 1** — On procède à un tirage au sort dans un groupe de  $n$  individus. Quelle sont les longueurs minimale et maximale d'un cycle dans la représentation de ce tirage au sort ?

Longueur minimale =

Longueur maximale =

**QUESTION 2** — Il apparaît assez clairement qu'il existe plusieurs représentations pour un même tirage au sort. Pour illustrer cette affirmation, une autre représentation correspondant à la situation de l'exemple 2 est :

- |   |   |
|---|---|
| A. <input type="checkbox"/> [0, 1, 3, 2, 'X', 5, 4, 'X', 8, 6, 7] | C. <input type="checkbox"/> [8, 7, 6, 'X', 5, 4, 'X', 3, 1, 0, 2] |
| B. <input type="checkbox"/> [4, 5, 'X', 0, 2, 1, 3, 'X', 8, 7, 6] | D. <input type="checkbox"/> [6, 7, 8, 'X', 4, 5, 'X', 0, 1, 2, 3] |

**Partie II - Cas particulier d'une représentation à un seul cycle**

Dans cette partie, on suppose que le tirage au sort a donné lieu à un unique cycle (comme dans l'exemple 1) au sein d'un groupe de  $n$  individus. On note RES la **représentation** de ce tirage au sort : RES est donc une liste de longueur  $n$ . Dans cette partie, on considèrera que la liste RES et l'entier  $n$  sont déjà définis.

**QUESTION 3** — Parmi les instructions proposées ci-dessous, lequel (ou lesquels) permet(tent) de savoir qui est le premier individu  $K_0$  de la représentation, ainsi que l'individu à qui  $K_0$  doit faire un cadeau ?

- |  |  |
|--|--|
| A. <input type="checkbox"/> $\begin{array}{l} \text{print('Le premier individu est', RES[1])} \\ \text{print('Il offre un cadeau à', RES[2])} \end{array}$ | C. <input type="checkbox"/> $\begin{array}{l} \text{print('Le premier individu est', RES[0])} \\ \text{print('Il offre un cadeau à', RES[0] + 1)} \end{array}$ |
| B. <input type="checkbox"/> $\begin{array}{l} \text{print('Le premier individu est', RES[0])} \\ \text{print('Il offre un cadeau à', RES[1])} \end{array}$ | D. <input type="checkbox"/> $\begin{array}{l} \text{print('Le premier individu est', RES[0])} \\ \text{print('Il offre un cadeau à', 'X')} \end{array}$        |

**QUESTION 4** — L'individu  $K$  a oublié à qui il doit offrir un cadeau... Quel est le programme (ou quels sont les programmes) lui permettant de retrouver de qui il s'agit ?

A. ☐

```
1 n = len(RES)
2 for i in range(n):
3     if RES[i] == K:
4         who = RES[i+1]
5     print(K, 'offre un cadeau à', who)
```

B. ☐

```
1 n = len(RES)
2 print(K, 'K offre un cadeau à', RES[K+1])
```

C. ☐

```
1 n = len(RES)
2 who = n+2
3 for i in range(n-1):
4     if RES[i] == K:
5         who = RES[i+1]
6     if who > n:
7         who = RES[0]
8     print(K, 'offre un cadeau à', who)
```

### Partie III - Cas général

On revient au cas général d'un groupe de  $n$  individus. On **ne suppose plus** que le tirage au sort a donné lieu à un unique cycle. On note RES la **représentation** de ce tirage au sort : RES est donc une liste de longueur au moins égale à  $n$ . Dans cette partie, on considèrera encore une fois que la liste RES et l'entier  $n$  sont déjà définis.

**QUESTION 5** — Parmi les programmes proposés ci-dessous, quel est celui (ou quels sont ceux) permettant de déterminer le nombre de cycles contenus dans RES ?

A. ☐

```
1 print('Il existe', len(RES) - n + 1, 'cycles')
```

B. ☐

```
1 NbCycles = 0
2 for i in range(len(RES)):
3     if RES[i] == 'X':
4         NbCycles = NbCycles + 1
5     print('Il existe', NbCycles, 'cycles')
```

C. ☐

```
1 NbCycles = 1
2 n = len(RES)
3 for i in range(n):
4     if RES[i] == 'X':
5         NbCycles = NbCycles + 1
6     print('Il existe', NbCycles, 'cycles')
```

**QUESTION 6** — (Qui m'a offert ce magnifique cadeau?). Complétez-le programme ci-dessous qui demande à l'utilisateur de saisir un entier  $K$ , et qui retourne l'entier  $L$  qui a offert un cadeau à  $K$ .

```
1 print('Bonjour! Choisissez un entier:')
2 K = int(input())
3 L = -1
4
5
6
7
8
9
10
11
12
13
14 print(L, 'a offert un cadeau à', K)
```