

# Algorithmes et Pensée Computationnelle

## Programmation de base - Exercices avancés

Le but de cette séance est d'aborder des notions de base en programmation. Au terme de cette séance, l'étudiant sera capable de :

- Définir une variable, définir son type et sa valeur.
- Définir une fonction et comprendre son rôle.
- Utiliser des notions d'algèbre booléenne.
- Comprendre la notion d'entrée/sortie.

Les langages qui seront utilisés pour cette séance sont Java et Python. Assurez-vous d'avoir bien installé IntelliJ. Si vous rencontrez des difficultés, n'hésitez pas à vous référer au guide suivant : [tutoriel d'installation des outils et prise en main de l'environnement de travail](#).

Cette feuille d'exercices avancés vous permettra d'approfondir vos connaissances des notions vues en cours. Le code présent dans les énoncés se trouve sur Moodle, dans le dossier "Ressources".

## 1 Représentation de nombres entiers

### Question 1: (🕒 5 minutes) Conversion d'un nombre binaire (au format complément à 2) en base 10

Soit le nombre binaire suivant exprimé sur 8 bits au format complément à 2 :  $10010011_{(2)}$ . Convertissez ce nombre en base 10.

#### 💡 Conseil

- Utilisez le même tableau que dans la question 4 (complément à 2).

#### >\_ Solution

Il faut appliquer le processus de la question 4 (complément à 2), mais de façon inversée. Cela permet d'obtenir la valeur positive en binaire du nombre que l'on cherche. Ensuite, il faut convertir cette valeur en base 10 et puis multiplier par -1.

a	1	0	0	1	0	0	1	1
b	1	0	0	1	0	0	1	0
c	0	1	1	0	1	1	0	1

a : écrire le nombre en binaire

b : soustraire 1

c : inverser tous les bits

On obtient donc  $01101101_{(2)} = 109_{(10)}$

Pour finir, on obtient  $-109_{(10)}$  après multiplication par -1.

## 2 Bases en programmation

### Question 2: (🕒 5 minutes) Conversion des variables (Type casting) (Java ou Python)

Qu'afficheront les programmes suivants ?

### Python :

```
1 nombre_entier = 3
2 nombre_decimal = float(nombre_entier)
3 print(nombre_entier)
4 print(nombre_decimal)
```

### Java :

```
1 float nombre_decimal = 3.14f;
2 int nombre_entier = (int) nombre_decimal;
3 System.out.println(nombre_entier);
4 System.out.println(nombre_decimal);
```

#### Conseil

Attention, ces fonctions ne changent pas le type des variables, elles ne font que les convertir.

#### Solution

##### Python :

```
3
3.0
```

##### Java :

```
3
3.14
```

### Question 3: (🕒 10 minutes) Calculs (division) (Java ou Python)

Créez 2 variables *nb\_bonbons* avec pour valeur 11 et *nb\_personnes* avec pour valeur 3. Divisez la première variable par la deuxième et stockez le résultat dans une nouvelle variable *bonbons\_personnes*. Pour finir, calculez le nombre de bonbons restants via l'opérateur % (modulo) et stockez le résultat dans une nouvelle variable *reste*. Vous pouvez afficher les différentes variables pour voir leurs valeurs.

#### Conseil

Attention, en Python il existe 2 opérateurs de division, / effectue une division classique, tandis que // effectue une division entière. En Java, si vous travaillez uniquement avec des int, / effectuera une division entière tandis que si vous travaillez avec au moins un float, / effectuera une division classique. Vous pouvez aussi formater le type du résultat lorsque vous créez une variable.

## >\_ Solution

### Python :

```
1 #1
2 nb_bonbons = 11
3 nb_personnes = 3
4 bonbons_personnes = nb_bonbons // nb_personnes
5 reste = nb_bonbons % nb_personnes
6 print (nb_bonbons)
7 print(nb_personnes)
8 print(bonbons_personnes)
9 print(reste)
10
11 #2
12 nb_bonbons = 11
13 nb_personnes = 3
14 bonbons_personnes = nb_bonbons / nb_personnes
15 print (nb_bonbons)
16 print(nb_personnes)
17 print(bonbons_personnes)
```

### Java :

```
1 // 1
2 int nb_bonbons = 11;
3 int nb_personnes = 3;
4 int bonbons_personnes = nb_bonbons / nb_personnes;
5 int reste = nb_bonbons % nb_personnes;
6 System.out.println(nb_bonbons);
7 System.out.println(nb_personnes);
8 System.out.println(bonbons_personnes);
9 System.out.println(reste);
10
11 // 2
12 float nb_bonbons = 11;
13 int nb_personnes = 3;
14 float bonbons_personnes = nb_bonbons / nb_personnes;
15 System.out.println(nb_bonbons);
16 System.out.println(nb_personnes);
17 System.out.println(bonbons_personnes);
```

### Question 4: (🕒 10 minutes) Manipulation des chaînes de caractères (Java ou Python)

Il est possible d'obtenir la longueur d'une chaîne de caractère (ou d'une liste ou d'un dictionnaire) en utilisant la fonction `len()`. Gardez votre variable `mon_mot` et créez une nouvelle variable nommée `ln_mon_mot` contenant le nombre de caractère de la variable `mon_mot`, puis une nouvelle variable `moitie` contenant la première moitié de la variable `mon_mot` (utilisez la variable que vous venez de créer). Affichez le résultat.



### Conseil

La fonction présentée dans l'énoncé de la question n'est valable que pour python. L'équivalent pour Java est la fonction `length()`.

## >\_ Solution

### Python :

```
1 mon_mot = "Hard But Cool !!"
2 ln_mon_mot = len(mon_mot)
3 moitie_mon_mot = mon_mot[:ln_mon_mot//2]
4 print(ln_mon_mot)
5 print(moitie_mon_mot)
```

### Java :

```
1 String mon_mot = "Hard But Cool !!";
2 int ln_mon_mot = mon_mot.length();
3 String moitie = mon_mot.substring(0,ln_mon_mot/2);
4 System.out.println(ln_mon_mot);
5 System.out.println(moitie);
```

### 3 Opérateurs et conditions Booléennes (Python uniquement)

#### Question 5: (🕒 20 minutes) Le juste prix

Dans le programme suivant, nous vous donnons un nombre aléatoire entre 0 et 30 dans la variable *number*, écrivez un programme qui demande à l'utilisateur de deviner le nombre tiré au sort. L'utilisateur a 5 chances pour le trouver. S'il se trompe, donnez-lui un indice (le nombre qu'il a écrit est-il plus grand ou plus petit que celui qu'il cherche?). Vous pouvez vous amuser à modifier le nombre de chances ou le nombre de possibilités (par exemple 10 chances pour trouver un nombre entre 0 et 100).

```
1 # Programme écrit en Python
2 from random import randint
3 number = randint(0,30)
4
5 #Votre code
```

## >\_ Solution

```
1 # Programme écrit en Python
2 from random import randint
3 number = randint(0, 30)
4 x = int(input("Choisissez un nombre: "))
5 if x == number:
6     print("Yeeah!")
7 elif x < number:
8     print("Trop bas!")
9 else:
10    print("Trop haut!")
11
12 x = int(input("Choisissez un nombre: "))
13 if x == number:
14     print("Yeeah!")
15 elif x < number:
16     print("Trop bas!")
17 else:
18     print("Trop haut!")
19
20 x = int(input("Choisissez un nombre: "))
21 if x == number:
22     print("Yeeah!")
23 elif x < number:
24     print("Trop bas!")
25 else:
26     print("Trop haut!")
27
28 x = int(input("Choisissez un nombre: "))
29 if x == number:
30     print("Yeeah!")
31 elif x < number:
32     print("Trop bas!")
33 else:
34     print("Trop haut!")
35
36 x = int(input("Choisissez un nombre: "))
37 if x == number:
38     print("Yeeah!")
39 elif x < number:
40     print("Trop bas!")
41 else:
42     print("Trop haut!")
```

Le problème avec cette solution est le suivant : Si le joueur trouve la réponse, le jeu va continuer, une façon plus propre et correcte de coder ce jeu est d'utiliser une boucle (prochain chapitre).

```
1 from random import randint
2 number = randint(0, 30)
3 for i in range(5):
4     x = int(input("Choisissez un nombre: "))
5     if x==number:
6         print("Yeah!")
7         break
8     elif x<number:
9         print("Trop petit!")
10    else:
11        print("Trop grand!")
```

Ici le code est plus concis et permet de s'arrêter lorsque le joueur aura trouvé la bonne réponse.

### Question 6: (🕒 20 minutes) Pierre, Feuille, Ciseaux

Demandez à l'utilisateur d'entrer soit pierre, soit feuille, soit ciseaux. L'ordinateur choisira son coup au hasard (s'il choisi 1 ce sera pierre, si c'est 2 ce sera feuille et si c'est 3 ce sera ciseaux). Les règles sont les règles classiques, une manche gagnante.

```
1 # Programme écrit en Python
2 from random import randint
3 number = randint(1,3)
4
5 #Votre code
```

#### >\_ Solution

```
1 # Programme écrit en Python
2 from random import randint
3 number = randint(1,3)
4 if number == 1 :
5     ordi = "pierre"
6 elif number == 2 :
7     ordi = "feuille"
8 else :
9     ordi = "ciseaux"
10
11 player = input("Choisissez un signe (pierre, feuille, ciseaux) : ")
12
13 print("ordi a choisi " + ordi)
14
15 if player != "pierre" and player != "feuille" and player != "ciseaux" :
16     print("symbole invalide")
17 else :
18     if ordi == "pierre" :
19         if player == "pierre" :
20             print("égalité")
21         elif player == "feuille" :
22             print("gagné")
23         else :
24             print("perdu")
25     elif ordi == "feuille" :
26         if player == "pierre" :
27             print("perdu")
28         elif player == "feuille" :
29             print("égalité")
30         else :
31             print("gagné")
32     else :
33         if player == "pierre" :
34             print("gagné")
35         elif player == "feuille" :
36             print("perdu")
37         else :
38             print("égalité")
```

Vous pouvez également utiliser une boucle pour augmenter le nombre de manches.