Fiche 1

Initiation à Python

1 Les variables

Exercice 1 :

1. Écrire un programme qui demande des coordonnées (abscisses puis ordonnées) de deux point A et B, puis retourne les coordonnées du milieu du segment $[AB]$.
2. Écrire un programme qui demande des coordonnées (abscisses puis ordonnées) de deux point A et B, puis retourne les coordonnées du symétrique de A par rapport à B.

Exercice 2 :
Écrire un programme qui demande à l'utilisateur d'entrer deux entiers, et affiche le résultat de la division euclidienne :
Exemple : Si l'utilisateur entre 25 puis 3, l'algorithme affiche :

```
25 = 3 x 8 + 1
```

Exercice 3 :

1. Affecter aux variables a et b les valeurs respectives 123 et 456.
   En utilisant une variable temporaire (appelée `temp`), inverser les affectations (a prendra la valeur de b et inversement).
2. En considérant que a et b sont des entiers, inverser les valeurs sans utiliser une variable temporaire.

Exercice 4 : Écrire une procédure qui demande à l'utilisateur d'entrer son nom puis son prénom, et affiche une phrase d'accueil :

```
bonjour, votre nom est ..., et votre prenom est ...
```

2 Les conditionnelles

Exercice 5 :
Sans utiliser l'ordinateur, donner la valeur de la variable `a` à la fin de la procédure dans chacun des cas suivants :

1.
```
>>> a = 8
>>> x = 3
>>> y = "bonjour"
>>> if (x>3) or (y <= "tout") :
                a = 14
        else :
                a = 2
```

2.
```
>>> a = 8
>>> x = 2.5
>>> y = 5
>>> if (x>3) :
        if ( y == 5 ) :
            a = 7
        else :
            a = 15
    else :
        if ( y % 5 == 1 ) :
            a = 48
```

3.
```
>>> a = 8
>>> x = 3
>>> y = 5
>>> z = 10
>>> if (x > -1) and (y <= 0) or (z == 10 ) :
        a = a+4
>>> if (x == -1) and ((y <= 0) or (z == 10 )) :
        a = a+2
>>> if (x == -1) or (y <= 0) and (z == 10 ) :
        a = a+7
>>> if (x == 3) or (y <= 0) and (z < 5 ) :
        a = a+1
```

Exercice 6 : Écrire une procédure demande à l'utilisateur son nom et son age, et affiche indiquant si le nom est inférieur au prénom (dans l'ordre alphabétique), et si l'utilisateur est adulte.

3 Les boucles

Exercice 7 :

1. Écrire une procédure qui affiche la table de multiplication d'un entier entré par l'utilisateur.
2. Écrire une procédure qui décompte, en partant de 10, et finie par "Partez !".

Exercice 8 :

1. Écrire une procédure qui calcule la somme des entiers consécutifs entre deux entiers entrés par l'utilisateur.
2. Écrire une procédure qui calcule le produit des entiers consécutifs entre deux entiers entrés par l'utilisateur.

Exercice 9 : On utilisera une fonction qui renvoie aléatoirement un entier entre deux entiers donnés :

```
>>> from random import randint
>>> randint(8,12)
8
```

1. Écrire une procédure qui simule le lancer de trois dés de 6, et renvoie le score obtenu.
2. Écrire une procédure qui compte le nombre d'essais pour obtenir trois 6.
3. Écrire une procédure qui compte le nombre d'essais pour obtenir la combinaison "421".

4 Les fonctions

Exercice 10 : Écrire une fonction `signe` qui renvoie 1 si la donnée est positif, 0 si elle est nulle et -1 si elle est négative.

Exercice 11 :

1. Écrire la fonction `carre` qui prend en argument un entier n et renvoie la valeur de n^2.
2. Écrire la fonction `est_rectangle` qui prend en argument un triplet (a, b, c) d'entiers naturel, et renvoie le booléen vrai si ce sont les longueurs d'un triangle rectangle, faux sinon.

Exercice 12 : Écrire la fonction `jeu` associé au jeu du nombre mystère : l'algorithme détermine aléatoirement une nombre entre 0 et 100, et l'utilisateur doit le retrouver. A chaque essai, un compteur est incrémenté, et un message indique si il est trop bas ou trop haut.
La fonction ne prend pas d'argument et renvoie le nombre d'essais.

Exercice 13 : On utilisera les fonctions mathématiques en faisant appel à la bibliothèque `math` :

```
from math import *
```

On définira un point du plan par un couple $(\mathtt{x}, \mathtt{y})$, exemple :

```
point_A = (5,9)
```

On définira un cercle du plan par un tuple $((\mathtt{x}, \mathtt{y}), \mathtt{r})$, où le point $(\mathtt{x}, \mathtt{y})$ est le centre, et `r` le rayon, exemple :

```
c_1 =( point_A , 4)
```

1. Écrire la fonction `centre` , qui prend en argument un cercle, et renvoie le centre du cercle.
2. Écrire la fonction `rayon`, qui prend en argument un cercle, et renvoie le rayon du cercle.
3. Écrire la fonction `position`, qui prend en argument un cercle et un point du plan, et renvoie -1 si le point est à l'intérieur du disque, 0 s'il est sur le cercle, et 1 s'il est à l'extérieur du disque.
4. Écrire la fonction `intersection`, qui prend en argument deux cercles et renvoie `True` si les deux disques ont des points communs, et `False` sinon.