

1. Algorithmes et premières instructions (exercices de TD)

Exercice 1.1

1. Indiquer précisément les actions engendrées par les instructions Python suivantes :

```
x = 3.5
print(x)
print("x")
print("x=", x)
v = float(input("Saisir au clavier une valeur réelle"))
print("x=", v)
```

2. Combien vaut la variable x à la fin de l'exécution du programme ?

Exercice 1.2

Écrire un algorithme en pseudo-code permettant de lire une valeur entière, d'incrémenter cette valeur de 1, puis de l'afficher.

Exercice 1.3

Écrire un algorithme en pseudo-code permettant de calculer l'image d'un nombre x (saisi au clavier par l'utilisateur) par la fonction

$$f(x) = 3x + 1$$

et d'afficher le résultat.

Exercice 1.4

Écrire un algorithme en pseudo-code permettant de lire deux nombres et de les stocker dans deux variables, puis d'échanger les valeurs de ces deux variables.

Exercice 1.5

Traduire en expressions booléennes les phrases suivantes :

- « x est compris au sens large entre 4,7 et 8. »
Donner la table de vérité du `and`.
- « y est divisible par 2 ou z n'est pas divisible par 3 (`or` inclusif). »
Donner la table de vérité du `or`.
- « Une et une seule des conditions suivantes est vraie : 1) u n'est pas nul, 2) v a la même valeur que u . »
A cet effet, l'opérateur `or` **eX**clusif, connu sous le nom `xor`, renvoie, étant donnés deux opérandes, le résultat `True` si et seulement si les deux opérandes ont des valeurs distinctes.

- (a) Donner la table de vérité de $A \text{ xor } B$.
- (b) L'opérateur `xor` n'est pas prédéfini en Python. Comment peut-on modéliser $A \text{ xor } B$ à l'aide uniquement des opérateurs `not`, `and` et `or`? Le vérifier en établissant les tables de vérité respectives.

Exercice 1.6

On donne le programme suivant :

```

1. a = float(input())
2. b = int(input())
3. if a > b: b = b+1
4. if b < 4:
5.     if a < 0:
6.         a = a-b
7.         else:
8.             b += int(a)
9.             print("moins de 4")
10. else:
11.     print("plus de 4")
12. b -= 4
13. print("a=", a)
14. print("b=", b)

```

Donner l'affichage produit par le programme précédent si l'utilisateur tape :

- 1. 2.3 puis -2
- 2. 4 puis 3
- 3. -1 puis 1

Exercice 1.7

Écrire en pseudo-code un algorithme qui lit trois nombres a , b et c , résout l'équation $ax^2 + bx + c = 0$ et affiche les solutions réelles (on distinguera 3 cas : $\Delta < 0$, $\Delta = 0$ et $\Delta > 0$).

Exercice 1.8

Dans le programme Python ci-dessous, pour quelles valeurs des différents paramètres verra-t-on s'afficher chacun des messages ?

```

if (niveau <= 2):
    if not(section == 1):
        if (discipline > 1):
            print("Lycéen en section générale non littéraire")
        elif (discipline == 1):
            print("Lycéen en section générale littéraire")
        else: print("Lycéen qui n'est pas en section générale")
elif (niveau <= 6):
    print("c'est un collégien")
else: print("il est trop petit")

```

Exercice 1.9

Écrire un algorithme en pseudo-code qui lit un entier n et affiche "Ajourné" si $n < 10$, "Passable" si $10 \leq n < 12$, "AB" si $12 \leq n < 14$, "B" si $14 \leq n < 16$ et "TB" si $n \geq 16$.

Exercice 1.10

Écrire en pseudo-code un algorithme qui lit 3 nombres et affiche "oui" si l'un des 3 est égal à la somme des 2 autres et "non" sinon.

Exercice 1.11

Écrire en pseudo-code, un algorithme qui lit 6 nombres a, b, c, d, e et f et affiche la résolution du système d'équations :

$$ax + by = c$$

$$dx + ey = f$$

Quelles sont les différentes issues de cette résolution ?