

Chapitre 2 : Alternatives - TD

Exercice 1 : Trace de programme*

Donner la trace du programme, c'est-à-dire donner l'affichage obtenu par l'exécution du programme.

```
In [ ]: x = 12
        print(x > 12)
        print(x < 11 or (x > 40 and x < 100))
        print(x != 9)
        test = not(x > 10 and x <= 12) and x%2 == 0
        print(test)
        test = x>=10 and test
        print(test)
        test = not(test)
        print(test)
        test = not(test)
        print(test)
```

Exercice 2 : Affichage d'un test**

1. Écrire un programme demandant à l'utilisateur de saisir un nombre entier. Si ce nombre est pair et positif ou s'il est impair et compris entre 5 (inclus) et 25 (inclus), alors le programme affiche True. Dans le cas contraire, le programme affiche False.

Rappel : Il existe un opérateur appelé *modulo* dont le symbole est % qui permet de calculer le reste de la division euclidienne. Par exemple, $3\%2$ vaut 1.

2. Donner un jeu d'essai (4 tests significatifs) et prévoir les résultats.

Exercice 3 : Valeur absolue d'un nombre*

Écrire un programme demandant à l'utilisateur de saisir un nombre et affichant la valeur absolue de ce nombre.

Rappel : La valeur absolue d'un nombre est sa valeur numérique sans son signe. Par exemple, la valeur absolue de -4 et de +4 est 4. Pour représenter la valeur absolue, on utilise la notation $|\dots|$. Ainsi, on écrit : $|-4| = |+4| = 4$.

Exercice 4 : Le jeu des différences*

Quelle différence y a-t-il entre les deux programmes suivants ?

```
In [ ]: A=5<2
        if A:
            print("V")
        else:
            print("F")
```

```
In [ ]: A=5<2
        if A == True:
            print("V")
        if A==False:
            print("F")
```

Exercice 5 : Questions sur le if*

Question 1 : Indentation

Qu'est-ce qu'une *indentation* ? Quel est l'intérêt d'une indentation ?

Question 2 : Else

Le else est-il obligatoire après un if ? Après un if/elif ?

Question 3 : Imbrication de if

Dans le programme suivant quelles instructions dépendent du premier if ? du deuxième if ?

```
In [ ]: print(1)
        if A>2:
            print(2)
            if B>=A:
                print(3)
            print(4)
        print(5)
```

Qu'affiche le programme dans les différents cas suivants ?

- quand A = 1 et B = 6,
- quand A = 4 et B = 5,
- quand A = 3 et B = 0.

Exercice 6 : Alternatives élémentaires*

Pour quelle(s) valeur(s) de A l'instruction `print("B")` est-elle exécutée dans chacun des quatre codes ?

```
In [ ]: if A > 10:
        print("A")
        else:
        print("B")
```

```
In [ ]: if A > 10:
        print("A")
        elif A > 200:
        print("B")
```

```
In [ ]: if A > 10:
        print("A")
        if A > 200:
        print("B")
```

```
In [ ]: if A > 10 and A < 10:
        print("A")
        else:
        print("B")
```

Pour quelle(s) valeur(s) de A l'instruction `print("C")` est-elle exécutée ?

```
In [ ]: if A < 100:
        print("A")
        elif A >= 100:
        print("B")
        else:
        print("C")
```

Exercice 7 : Réduction du nombre de tests**

Peut-on réduire le nombre de tests dans le programme suivant ? Si oui, comment ?

```
In [ ]: A=float(input())
        if A <= 10:
        print("A")
        elif A > 10 and A <= 50:
        print("B")
        elif A > 50 and A < 100:
        print("C")
        elif A >= 100:
        print("D")
```

Exercice 8 : Différence de deux nombres**

Écrire un algorithme qui demande deux nombres entiers à l'utilisateur et calcule la différence du plus grand nombre avec le plus petit nombre, quel que soit l'ordre de saisie.

Exercice 9 : Calcul de gabarit**

On définit le gabarit d'un objet en fonction de sa taille. Le gabarit peut prendre les valeurs "Grand", "Moyen" ou "Petit" (qui sont des chaînes de caractères) selon que la taille, qui est un nombre entier, est respectivement supérieure ou égale à 10, comprise entre 4 (inclus) et 10 (non inclus) ou strictement inférieure à 4.

Écrire un programme qui demande à l'utilisateur la taille d'un objet et affiche après l'avoir déterminé le gabarit correspondant. On s'attachera à ne pas faire de test inutile.

Exercice 10 : Tri de trois nombres**

Écrire un algorithme qui déclare 3 variables a, b et c initialisées avec 3 nombres entiers saisis par l'utilisateur. L'algorithme devra ensuite modifier les valeurs de a, b et c pour que a contienne la plus petite valeur saisie, c la plus grande valeur saisie et b la troisième valeur. L'algorithme affichera ensuite les valeurs de la plus petite à la plus grande.

Exercice 11 : Racines d'un polynôme du second degré**

Écrire un algorithme qui calcule et affiche les racines réelles d'un polynôme du second degré.

NB : Ce sont les racines réelles de l'équation $ax^2 + bx + c = 0$

Prendre soin de bien analyser le problème de savoir ce qui est très précisément demandé pour répertorier les données du problème.

Remarque : Le calcul des racines nécessite de calculer la racine carrée d'un nombre. En python, on obtient la racine carrée d'un nombre x grâce à l'appel `math.sqrt(x)`. Il faut auparavant avoir importé le module `math` grâce à l'instruction `import math`.