Activité – Vrai/Faux

Exercice 1

Calculer les valeurs des expressions suivantes pour : (i) $x = 0$ (ii) $x = 10$ (iii) $x = -20$

a. $x < 10$ and $x > -10$

b. $x < -10$ or $x > 10$

c. $x <= 10$ and $x*x >= 100$

d. $(x > -25$ and $x < -5)$ or $(x > 5$ and $x < 25)$

Exercice 2

On considère l'expression : $\text{not}(A \text{ or not } B) \text{ and } B$

Pour quelles valeurs des variables booléennes *A* et *B* cette expression a pour valeur `True` ? Utiliser une table de vérité.

Exercice 3

Vérifier les identités ci-dessous, appelées lois de De Morgan, en complétant leurs tables de vérité.

a. $\text{not}(A \text{ and } B) == (\text{not } A) \text{ or } (\text{not } B)$

b. $\text{not}(A \text{ or } B) == (\text{not } A) \text{ and } (\text{not } B)$

TP – Introduction au langage Python

Exercice 4

Insérer le minimum de parenthèses dans les expressions suivantes pour que les égalités soient correctes.

a. 4 + 5 * 2 + 3 == 29 **b.** 5 + 2 * 3 + 4 == 25 **c.** 2 + 3 * 5 + 4 == 45

Exercice 5

Ecrire un code qui demande deux nombres *a* et *b* à l'utilisateur, puis calcule et affiche leur moyenne.

Exercice 6

Marc doit écrire un code qui permet d'échanger les contenus de deux variables *a* et *b* (la valeur de *a* dans *b* et vice-versa) données par l'utilisateur et de les afficher. Pour cela, il propose le code suivant :

```
1 a = int(input("Valeur de a?")
2 b = int(input("Valeur de b?")
3 print("Valeurs d'origine: a=",a," b=",b)
4 a=b
5 b=a
6 print("Valeurs échangées: a=",a," b=",b)
```

Expliquer pourquoi ce code est incorrect et ne fournira pas le résultat demandé. Comment corriger ce problème ?

TP – Si, sinon si, sinon

Exercice 7

Sans écrire le code, déterminer ce qu'affiche chacun des codes ci-dessous.

a.
```
1 if 12 * 2 == 24 :
2     print("Logique !")
```

b.
```
1 if 12 * 2 == True :
2     print("Logique !")
```

c.
```
1 if (12 * 2 == 23) == False :
2     print("Logique !")
3 print("Ou pas...")
```

d.
```
1 if 12 * 2 == 23 :
2     print("Logique !")
3 else :
4     print("Pas logique")
```

Exercice 8

Ecrire un code qui demande une température *T* à l'utilisateur et affiche :

- "Il fait très froid" pour $T < -5$
- "Il fait froid" pour $-5 \leq T < 5$
- "Il fait frais" pour $5 \leq T < 15$
- "Il fait bon" pour $15 \leq T < 25$
- "Il fait un peu chaud" pour $25 \leq T < 30$
- "Il fait chaud" pour $60 \leq T < 35$
- "Il fait très chaud" pour $T \geq 35$

Attention : Le code ne peut contenir qu'une seule fois l'instruction `if`.

TP – Encore, et encore, et encore…

Exercice 9

Sans écrire le code, déterminer combien de fois s'exécute le corps de chacune des boucles ci-dessous. Quelle est la valeur de *s* à la fin de l'exécution ?

a.
```
1 | s = 0
2 | for i in range(10) :
3 |     s = s + i
```

b.
```
1 | for i in range(1, 6) :
2 |     s = 1
3 |     s = s * i
```

c.
```
1 | s = 0
2 | while s < 20 :
3 |     s = s + 5
```

d.
```
1 | s = 1
2 | while s != 100 :
3 |     s = s * 2
```

Exercice 10

On considère le code ci-dessous.

```
1 | r = 0
2 | while b != 0 :
3 |     r = a % b
4 |     a = b
5 |     b = r
```

Quelle est la valeur de *a* à la fin de l'exécution lorsque :

a. $a = 10$ et $b = 3$

b. $a = 10$ et $b = 15$

Exercice 11

Réécrire le code ci-dessous en remplaçant la boucle bornée (`for`) par une boucle non bornée (`while`).

```
1 | Q = float(input("Quelle est la quantité initiale ? "))
2 | for i in range(5) :
3 |     Q = Q * 1,33
4 | print(Q)
```

Exercice 12

a. Ecrire une boucle bornée (`for`) qui affiche les nombres pairs entre 2 et 20.

b. Ecrire une boucle non bornée (`while`) qui fasse la même chose.

c. Mêmes questions pour afficher les nombres pairs entre 20 et 2 par ordre décroissant.

Exercice 13

Ecrire un code qui affiche les chiffres d'un nombre *N* en base décimale donné par l'utilisateur à partir du dernier, un par ligne.

Exemple : pour le nombre N = 1 234, le programme affiche :
4
3
2
1

Exercice 14

Dans un vivarium, une colonie d'insectes voit son nombre d'individus multiplié par 3 chaque jour. Ils sont 10 le premier jour.

Ecrire un code qui demande à l'utilisateur le nombre maximal d'insectes que peut contenir le vivarium puis qui affiche le nombre de jours avant que ce maximum soit dépassé (ou atteint) ainsi que le nombre d'individus à cette date.

TP – Les fonctions

Exercice 15

On considère le code ci-dessous :

```
1  def double(x) :
2      print(x * 2)
3
4  a = 21
5  a = double(a)
```

a. Quelle est la valeur de *a* à la fin de l'exécution du code ?

b. Qu'affiche le programme ?

Exercice 16

Les codes suivants sont-ils corrects ? Si oui, qu'affichent-ils ? Si non, quelles sont les erreurs ?

a.

```
1  pi = 3,14
2  def aire-cercle(rayon)
3      return pi * r ** 2
4  print(aire_cercle(10))
```

b.

```
1  def f(a):
2      a = a * 2
3      return a
4  print(f(5), a)
```

c.

```
1  a = 100
2  def f(a):
3      a = a * 2
4      return a
5  print(f(5), a, f(a))
```

Exercice 17

On représente une date par 3 nombres : *j* le jour du mois, *m* le numéro du mois (de 1 à 12) et *a* l'année. On considère ici, pour simplifier, que tous les mois comportent 30 jours (soit 360 jours par année).

a. Ecrire une fonction `anterieur` qui prend deux dates (*j1*, *m1*, *a1* et *j2*, *m2*, *a2*) en arguments (dates après J.-C.) et renvoie `True` si la première date est antérieure (avant) la seconde et `False` sinon. . Les préconditions sur les arguments doivent être vérifiées dans la fonction.

b. Ecrire une fonction *age* qui prend en arguments la date de naissance d'une personne et une autre date (la date de calcul) et affiche l'âge de la personne à la date de calcul, en année. Les préconditions sur les arguments doivent être vérifiées dans la fonction.

Exemple : age(24,3,1981,10,2,1996) renvoie 14,8 ans

Exercice 18

Les règles du jeu de Pierre-Papier-Ciseaux (aussi appelé Chifoumi) sont les suivantes : chaque joueur choisi de jouer Pierre ou Papier ou Ciseaux. Un joueur gagne pour :

- Pierre bat Ciseaux (Pierre écrase Ciseaux)
- Ciseaux bat Papier (Ciseaux coupe Papier)
- Papier bat Pierre (Papier recouvre Pierre)

On cherche ici à écrire un programme qui joue à Pierre-Papier-Ciseaux.

a. Ecrire une fonction `Choix` qui prend en arguments les choix (1 = Pierre ; 2 = Papier ; 3 = Ciseaux) de deux joueurs *A* et *B*, et renvoie +1 si *A* gagne, -1 si *B* gagne et 0 s'il y a égalité.

b. Ecrire une fonction `Tour` qui tire les choix des deux joueurs au hasard (avec `randint`) et détermine celui qui gagne, sachant qu'en cas d'égalité, un nouveau tirage a lieu. La fonction renvoie +1 si *A* gagne et -1 si *B* gagne.

c. Ecrire un programme qui initialise les scores et joue 50 tours, en affichant à chaque fois le numéro de tour, les choix sous forme "Pierre", "Papier", "Ciseaux" et les scores des joueurs *A* et *B* (sur des lignes séparées). A la fin, on affiche le joueur qui a gagné la partie.

(Note : La modification des fonctions précédente est nécessaire pour les affichages)

Exercice 19

On veut calculer la valeur x pour laquelle une fonction $f(x)$ définie dans le code vaut une valeur donnée v.

a. Ecrire une fonction ValeurX prenant comprenant une boucle qui fait varier x d'une valeur donnée *min* à une valeur donnée *max* (les deux valeurs étant données directement dans le code) par pas de valeur *pas*. La boucle s'arrête pour une valeur x telle que $f(x) \leq v$ et $f(x+pas) > v$ (on suppose ici que la valeur v existe) et la fonction renvoie x. Les valeurs *min*, *max*, *pas* et v sont données en arguments.

Test : Appliquer l'algorithme à la fonction *sin(x)* avec *min* = 0, *max* = π, *pas* = 0,1 et v = 0,5 ; La fonction renvoie alors x = 0,6. Utiliser la bibliothèque math.

b. Pour affiner l'estimation de x, on applique le même procédé avec *min* = x et *max* = x+*pas*, en utilisant maintenant un pas 10 fois plus petit. On peut ainsi continuer à affiner l'estimation en réduisant l'intervalle de recherche et le pas.
Ecrire une fonction Estimation utilisant la fonction ValeurX, qui permet d'estimer la valeur de x en affinant le pas de *pasini* jusqu'à ce qu'elle soit d'un pas *ordre* donné, le pas étant 10 fois plus petit à chaque itération. La fonction prend en arguments les valeurs *min*, *max*, *pasini*, *ordre* et v.

Test : Appliquer l'algorithme à la fonction *sin(x)* avec *min* = 0, *max* = π, *pasini* = 0,1, ordre = 1E-6 v = 0,5 ; La fonction renvoie alors $x = \pi/6 \simeq 0{,}523599$.