Université Paris Dauphine **Licence MIE 1re année**
Algorithmique et programmation 1 **2025-2026**

2. Boucle répétitive while (TP)

Exercice 2.9 *(Debugage d'une solution de l'exercice 1.7)*

L'équipe Python a écrit une solution pour l'exercice 1.7 sur l'équation du second degré. Malheureusement le programme ne fonctionne pas ! Vous allez devoir le corriger et le faire fonctionner.

Pour récupérer le fichier, aller sur Moodle et dans la semaine 3, copier, dans votre répertoire personnel, le fichier `Programme à corriger : Debug_Equation2ndDegre.py`.

```
# Ce programme calcule les solutions
#  d'une équation du second degré
a = input("Donner le parametre a : ")
b = input("Donner le parametre b : ")
c = input("Donner le parametre c : ")
d = b*b - 4ac
print("la valeur de delta est", d)
# le cas ou il y a deux solutions reelles
if (d > 0):
    print("Il y a deux racines reelles distinctes : x1 =",
    -b + sqrt(d)/2*a," et x2 = ",  -b - sqrt(d)/2*a)
# le cas ou une seule solution reelle
elif (d == 0):
    print("Il y a une unique racine reelle : x=", -b/2*a)
# le cas ou aucune solution reelle
else:
    print("Il y a deux racines complexes distinctes : z1 =",
    -b/2*a,"+", sqrt(d),"i", " et z2 = ",
    -b/2*a, "-", sqrt(d),"i")
```

1. À quoi servent les lignes commençant par # ? Vous aident-elles à comprendre ce que fait ce programme ?
2. Exécuter le programme ci-dessus tel quel. Quel est le message d'erreur ? Que faut-il corriger dans le programme pour résoudre le problème ?
3. L'instruction `print("la valeur de delta est",d)` est-elle nécessaire dans le programme ? Si non, pourquoi l'ajouter ?
4. Exécuter le programme en donnant 2 pour a, -8 pour b et 6 pour c. Quel est le message d'erreur ? Que faut-il corriger dans le programme pour lever ce message ?
5. Exécuter le programme en donnant 2 pour a, -8 pour b et 6 pour c. Le programme s'exécute-t-il ? Comment s'assurer que la solution proposée est la bonne ? Si ce n'est pas la bonne que faut-il corriger ? Où dans le programme ? Faire les corrections correspondantes.

6. Exécuter le programme en donnant 2 pour a, 4 pour b et 4 pour c. Quel est le message d'erreur ? Que faut-il corriger dans le programme pour résoudre le problème ?
7. Le cas où l'équation admet une unique solution réelle a-t-il été testé ? Proposer un jeu de valeur permettant de l'exécuter.
8. Exécuter le programme en donnant 0 pour a, 2 pour b et 6 pour c. Quel est le message d'erreur ? Que faut-il corriger dans le programme pour résoudre le problème ?

Exercice 2.10 *(Mise en œuvre de l'exercice 2.1)*

Écrire un programme Python qui lit un réel $x \geq 0$ et un entier $n \geq 0$, et affiche la valeur de x^n (sans utiliser l'opérateur puissance).

Exercice 2.11 *(Mise en œuvre de l'exercice 2.2)*

Écrire un programme Python qui affiche la somme de tous les entiers impairs positifs, inférieurs à 100.

Exercice 2.12 *(Mise en œuvre de l'exercice 2.3)*

Écrire un programme Python qui lit un entier (supposé strictement positif), affiche la somme de tous ses diviseurs stricts (c'est-à-dire différents du nombre) et précise si ce nombre est premier (c'est-à-dire qu'il n'a que 1 comme diviseur strict). Par exemple, si l'on saisit 8, il affiche 7 (car somme de ses diviseurs stricts qui sont 1, 2 et 4).

Exercice 2.13 *(Mise en œuvre de l'exercice 2.4)*

Un entier est dit parfait s'il est égal à la somme de ses diviseurs stricts (6 par exemple est parfait). Écrire un programme Python qui lit un entier n et affiche tous les nombres parfaits inférieurs ou égaux à n.

Exercice 2.14

Le Plus Petit Commun Multiple (PPCM) de deux entiers positifs a et b est le plus petit entier p tel que p soit à la fois multiple de a et de b.

Afin de calculer le PPCM de a et b, on applique l'algorithme suivant qui consiste à calculer les multiples successifs de a et b, notés m_a et m_b, jusqu'à arriver à l'obtention du PPCM.

(i) initialiser m_a et m_b ;

(ii) tant que le PPCM n'est pas trouvé, on augmente la plus petite des deux valeurs m_a et m_b ;

(iii) afficher le PPCM.

1. Comment initialiser m_a et m_b ?
2. Comment identifier que le PPCM est trouvé ?
3. De combien doit-on augmenter à chaque itération m_a ? et m_b ?
4. Écrire en Python le programme complet : il doit lire deux nombres saisis par l'utilisateur (on les supposera entiers et positifs), calculer leur PPCM et l'afficher.

Exercice 2.15

Étant donné un livret bancaire rémunéré au taux annuel de 1,5 %, sur lequel 6 000 euros auront été déposés le 1er janvier 2017, écrire un programme en Python permettant de savoir au bout de combien d'années la somme disponible sur ce livret aura dépassé 7 000 euros.

Nous vous rappelons que si l'on place une somme s le 1er janvier d'une année n sur un livret bancaire rémunéré au taux annuel de 1,5 %, la somme disponible le 1er janvier de l'année $(n+1)$ sera $s + s * 0,015$. *(La fonction logarithme n'est pas utilisable.)*

Exercice 2.16 *(Mise en œuvre de l'exercice 2.6)*

À l'aide de 2 boucles `while` emboîtées, écrire un programme Python qui affiche (la gestion précise de l'espacement n'est pas demandée) :

```
1   2   3   4   5
1   3   5   7   9
1   4   7   10  13
1   5   9   13  17
```

Exercice 2.17 *(Mise en œuvre de l'exercice 2.7)*

Écrire un programme Python qui lit un entier positif n, puis n entiers, et qui vérifie si la suite des n entiers saisis est triée de façon croissante.

Par exemple, si la valeur saisie pour n est 7 et les 7 valeurs saisies sont : -1, 3, 7, 8, 11, 234, 300, le programme affichera "les 7 valeurs sont triées de façon croissante", alors que le programme affichera "les 7 valeurs ne sont pas triées" si les entiers saisis sont : 1, 2, 7, -2, 4, 5, 200.

Exercice 2.18 *(Mise en œuvre de l'exercice 2.8)*

Écrire un programme Python qui lit un entier positif n, puis n entiers positifs, et qui vérifie si parmi les n entiers saisis la somme des entiers pairs est égale à la somme des entiers impairs. Le programme doit refuser les valeurs nulles ou négatives.

Par exemple, si la valeur saisie pour n est 5 et les 5 valeurs saisies sont 2, 3, 2, 3, 2, le programme affichera "la somme des nombres pairs est égale à la somme des nombre impairs", alors que le programme affichera "la somme des nombres pairs n'est pas égale à la somme des nombres impair" si les valeurs saisies sont 2, 3, 4, 3, 1. Si la valeur saisie pour n (ou une des n valeurs) est par exemple -2, le programme demandera à nouveau d'insérer une valeur, jusqu'à ce que celle-ci soit positive.

Exercice 2.19

On joue à lancer 3 dés : un dé rouge à 6 faces (numérotées de 1 à 6), un dé vert à 8 faces (numérotées de 1 à 8) et un dé bleu à 10 faces (numérotées de 1 à 10). On veut déterminer combien il y a de façons d'obtenir un nombre n donné en additionnant les trois faces des dés. Par exemple, il y a une seule façon d'obtenir 3 qui est de faire 1 avec le dé vert, 1 avec le dé rouge et 1 avec le dé bleu. Il y a 3 façons d'obtenir 4 :

— 1 avec le dé vert, 1 avec le dé rouge et 2 avec le dé bleu, ou bien,
— 1 avec le dé vert, 2 avec le dé rouge et 1 avec le dé bleu, ou bien,
— 2 avec le dé vert, 1 avec le dé rouge et 1 avec le dé bleu.

Écrire le programme **Python** qui implémente l'algorithme suivant :

1. demander à l'utilisateur de saisir une valeur entière ;

2. à l'aide de boucles imbriquées, compter le nombre de possibilités permettant d'obtenir ce nombre comme somme des dés rouge, vert et bleu ;
3. afficher le résultat.

Exemple d'exécution :

affichage Entrez un nombre entier :

saisie 5

affichage Il y a 6 façon(s) de faire 5 avec ces trois dés.