

.....

Session d'exercices – Les structures de contrôle

## Equation quadratique [Difficulté: \*\*]

L'objectif de cet exercice est d'écrire le programme `quadratic_equation.py` pour calculer les solutions des équations de la forme

$$ax^2 + bx + c = 0 \quad (1)$$

où  $a, b, c$ , et  $x$  sont des nombres réels. Le programme doit demander à l'utilisateur de saisir les valeurs des coefficients ( $a$ ,  $b$ , et  $c$ ), résoudre l'équation, et afficher les résultats à l'écran.

**Note: Il faut lire toutes les pages AVANT de commencer à écrire le programme !**

Pour obtenir une saisie utilisateur, nous avons vu qu'il faut utiliser la fonction `input()` de Python. Par défaut, cette fonction renvoie la saisie de l'utilisateur sous forme de chaîne de caractères (string). Notez que, comme la saisie est renvoyée sous forme de chaîne, vous devez la convertir en entier (integer) ou en nombre à virgule flottante (float) avant de réaliser des opérations arithmétiques. Rappelez-vous des fonctions `int()` et `float()` qui prennent un argument (une valeur ou une variable) et le convertissent en un entier ou un nombre à virgule flottante avec cette valeur.

- a != 0

Pour  $a \neq 0$ , l'équation quadratique peut être résolue en deux étapes. Calculer dans un premier temps le discriminant delta:

$$\Delta = b^2 - 4ac \quad (2)$$

Si  $\Delta > 0$ , l'équation a deux solutions:

$$x_1 = \frac{-b - \sqrt{\Delta}}{2a} \quad (3)$$

$$x_2 = \frac{-b + \sqrt{\Delta}}{2a} \quad (4)$$

Si  $\Delta = 0$ , l'équation a une solution unique:

$$x_1 = \frac{-b}{2a} \quad (5)$$

Si  $\Delta < 0$ , l'équation n'a pas de solution réelle. Dans ce cas, le programme doit afficher le message `This equation has no solution.` à l'écran.

- a == 0

Dans le cas générique, il peut arriver que certains des coefficients soient égaux à zéro. Si  $a = 0$ , aucune des relations (3), (4), (5) ne peut être utilisée. Les solutions doivent être trouvées grâce à une approche différente. L'équation devient alors:

$$bx + c = 0 \quad (6)$$

Si  $b \neq 0$ , une seule solution existe:

$$x = \frac{-c}{b} \quad (7)$$

Si  $b = 0$  et  $c = 0$  alors tout  $x \in \mathbb{R}$  satisfait l'équation. Dans ce cas, le programme doit afficher le message `Every real number is a solution of this equation.` à l'écran.

Si  $b = 0$  et  $c \neq 0$  alors aucune solution n'existe. Dans ce cas, le programme doit afficher le message `This equation has no solution.` à l'écran.

## Exemples

```
Enter the coefficients of the quadratic function ax^2+bx+c
a = 3
b = -18
c = 24
delta : 36.0
x1 : 2.0
x2 : 4.0
```

```
Enter the coefficients of the quadratic function ax^2+bx+c
a = 2
b = 3
c = 5
delta : -31.0
This equation has no solution.
```

```
Enter the coefficients of the quadratic function ax^2+bx+c
a = 0
b = 9
c = 4
x = -0.4444444444444444
```

## Comment calculer la racine carrée?

Afin de calculer la racine carrée d'un nombre flottant et positif, on peut utiliser la fonction `sqrt` (*square root*), qui se trouve dans la bibliothèque des fonctions mathématiques appelée `math`.

Afin d'utiliser les fonctions de cette bibliothèque, il faut insérer dans les premières lignes du programme l'expression: `import math`.

La fonction `sqrt` prend comme argument un nombre de type `float` et renvoie comme résultat un nombre de type `float` aussi. Par exemple, pour calculer la racine carrée de la variable `m` (initialisée à 49 dans cet exemple) et afficher le résultat, il faut écrire les instructions suivantes:

---

```
import math
m = 49 # déclaration de la variable m avec la valeur 49
racine2 = math.sqrt(m); # appel a la fonction sqrt pour calculer la racine caree de 49
```

```
print(f"square root of {m:.2f} = {racine2:.2f}")  
# afficher un texte formate ou la racine caree est calculee a 2 chiffres apres la virgule
```

L'exemple ci-dessus utilise *f-strings*: pour créer des chaînes formatées.

`:.2f` est un spécificateur de format qui permet de contrôler l'affichage des valeurs numériques. Voici sa signification détaillée:

- `:` introduit la spécification de format;
- `.2` signifie que on souhaite afficher le nombre avec **2 chiffres après la virgule**;
- `f` signifie que le nombre est au format flottant (nombré décimal).

Cela garantit que les nombres sont toujours affichés avec deux chiffres après la virgule.

```
>>> print(f"square root of {m:.2f} = {racine2:.2f}")  
square root of 49.00 = 7.00
```

## Comment calculer le carré?

Afin de calculer le carré d'un nombre flottant, vous pouvez utiliser l'opérateur `**` intégré ou la fonction `pow` de la bibliothèque `math`.

Afin d'utiliser les fonctions de cette bibliothèque, il faut insérer dans les premières lignes du programme l'expression: `import math`.

Cette fonction est définie de la façon suivante:

```
math.pow(base, exponent)
```

et donc elle prend comme argument deux nombres, `base` et `exponent`, et renvoi comme résultat "base puissance exponent". Par exemple, pour calculer le carré de la variable `m` et le stocker dans la variable `result`, on peut écrire `m**2` ou appeler la fonction `math.pow` comme suit:

```
result = math.pow(m, 2);
```

**Littérature:** Solutions d'une équation quadratique : [wiki/Équation du second degré](#)  
La bibliothèque `math` : [wiki/Python\\_math\\_library](#)