

---

## Labo 3 - Equation quadratique

---

### 1 Objectifs pédagogiques

- Récupérer des valeurs saisies par l'utilisateur.
- Faire un diagramme de flux.
- Utilisation de l'instruction « if ».
- Introduire la notion de test automatique.
- Editer, corriger et **déboguer** un programme.

### 2 Cahier des charges

- Créer un programme qui résout l'équation quadratique.

$$A * x^2 + B * x + C = 0$$

- Le programme doit demander d'entrer les paramètres de l'équation (A,B,C).
- Le programme doit afficher la valeur des 3 paramètres passés en entrée.
- Le programme doit afficher les solutions de l'équation.
- Le programme ne contrôle pas si autre chose que des nombres sont entrés par l'utilisateur.
- Faire un schéma bloc et un diagramme de flux du programme. Avec ces diagrammes, on doit comprendre comment le programme fonctionne. (Vous pouvez utiliser Viso par exemple).
- S'il n'y a pas de solution valide, le programme retourne 1 sinon 0.
- Ajouter un test #5 dans le fichier `t.json` pour tester les valeurs A=4, B=4, C=1

## 3 Aides

### 3.1 Affichage et formatage d'une chaîne de caractère

Une chaîne de caractère est représentée entre guillemet double

`"Une chaîne de caractère\n"`

Pour afficher une chaîne de caractère, nous allons utiliser « printf ». Cette fonction prend en entrée un format et les variables.

```
int x = 4;
double r = sqrt(x);
printf("La racine de %d est : %.3f\n", x, r);
```

Après le signe %, on retrouve le type de format que l'on souhaite. Vous pouvez regarder sur la carte jaune pour trouver différentes manières de formater une chaîne de caractère. Dans le cas ci-dessus, la deuxième valeur passée en paramètre sera interprétée comme un float avec 3 chiffres après la virgule.

Vous remarquez dans cet exemple « sqrt » qui peut vous être utile pour la suite.

### 3.2 Test d'une valeur

Il se peut que dans votre programme vous deviez faire des tests pour comparer une valeur à une autre.

L'instruction « if » vous permet de faire un test. Voici un exemple qui fait la valeur absolue de la variable « test »

```
double test = -1.23;
double res = 0.0;

if (test > 0) {
    // je fais quelque chose si test > 0
    res = test;
}
else {
    // sinon je fais autre chose
    res = -test;
}

printf("Le resultat est : %lf\n", res);
```

### 3.3 Compiler le code

Vous pouvez compiler le code avec plusieurs commandes qui donnent le même résultat.

- <ctrl> + <shift> + <b>
- Menu « Terminal » -> « Run build task »
- Dans le terminal entrer « make »

Votre code source sera compilé, ce qui va créer le fichier exécutable `app`.

### 3.4 Déboguer

Le programme doit être compilé.

- Démarrez le programme en mode débogage (F5) ou menu « Run » -> « Start debug ».

### 3.5 Exécuter le programme

- En bas de l'écran, vous devez avoir une fenêtre « Terminal »  
Si ce n'est pas le cas, cliquer sur menu « Terminal » -> « New terminal ».
- Aller dans le terminal et entrer ./app

La fenêtre « terminal » est une console Linux qui permet de lancer des commandes. C'est l'équivalent d'une fenêtre « cmd » sous Window.

### 3.6 Valeur de retour

C'est l'instruction `return` qui permet de choisir la valeur que le programme retourne à la fin de l'exécution.

Par exemple pour retourner `0` si tout se passe bien

```
return 0
```

## 4 Tâches à effectuer

- Copier le dossier « labo3 » qui se trouve dans le répertoire ci-dessous, sur votre ordinateur.

<\\eistore1\profs\TMZ\info\info1\labos\labo3>

- Ouvrir Visual Code
- Dans « File » -> « Open Folder »
- Sélectionner le dossier sur votre ordinateur que vous avez copié.
- Cliquer sur l'icône vert en bas à gauche et choisissez « Reopen in Container ».
- Modifier le fichier « main.c ».

## 5 Affichage du programme

Voici 3 exemples de ce que le programme doit afficher à l'écran

<pre>\$ ./app Entrer A : 1 Entrer B : 2 Entrer C : -3  La valeur de A = 1 La valeur de B = 2 La valeur de C = -3 Solution 1 = -3.0 Solution 2 = 1.0</pre>	<pre>\$ ./app Entrer A : 1 Entrer B : 2 Entrer C : 3  La valeur de A = 1 La valeur de B = 2 La valeur de C = 3 Pas de solution</pre>	<pre>\$ ./app Entrer A : -2 Entrer B : 4 Entrer C : -2  La valeur de A = -2 La valeur de B = 4 La valeur de C = -2 Solution = 1.0</pre>
---	--	---

Pour la partie orange (la saisie) vous êtes complètement libre des textes et du format. Par contre pour la partie bleue (l'affichage des résultats) **il faut respecter ce qui est dans le tableau ci-dessus.**

## 6 Tests

Pour tester le programme, vous allez utiliser un système de test automatique.

Pour voir comment le programme sera testé, il faut regarder le fichier `t.json` qui est dans le répertoire du laboratoire.

### 6.1 Tester le programme

**Le programme doit compiler sans erreur et être exécutable**

Dans le terminal entrer `make test`

## 7 Conseils

- Commenter le code si nécessaire.

## 8 Travaux à rendre

### 8.1 Liste des livrables

Créer un zip avec :

- Le fichier exécutable (app)
- Le ou les diagrammes dans un fichier PDF
- Le fichier source du programme (main.c)
- Le fichier test du programme (t.json)

L'archive doit être déposée dans le répertoire "Labo03" de Cyberlearn (à la date demandée sur le site INFO1 de Cyberlearn).