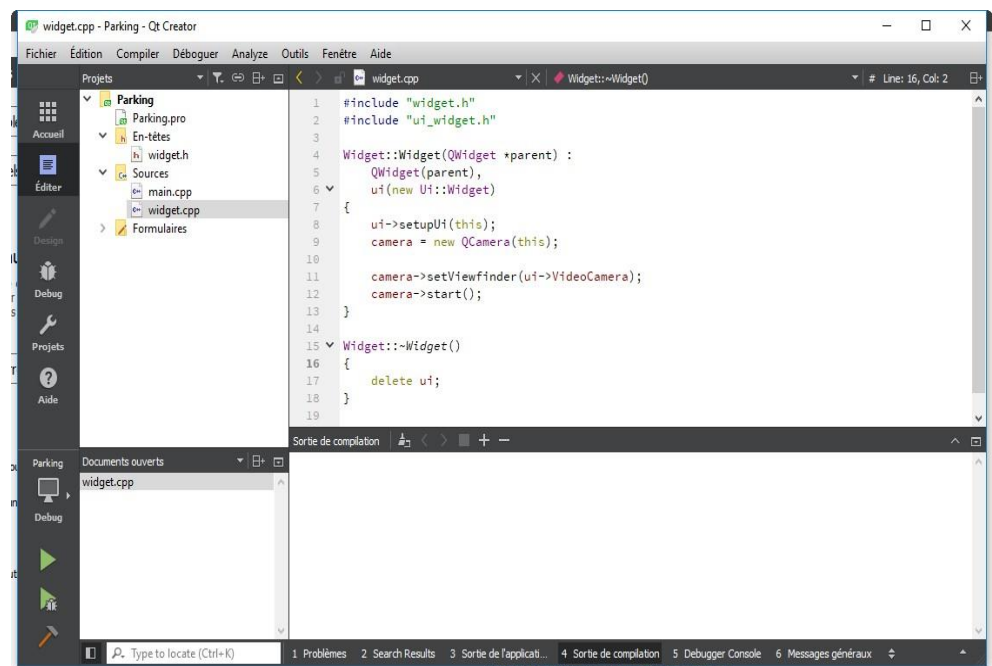


TP N°4



28/09/2020

Premiers algorithmes en C/C++

L'objectif de ce TP est de vous permettre de mettre en application des algorithmes vus en TD.

Vous utiliserez QtCreator pour tous les programmes qui suivent.

TP n°4

PREMIERS ALGORITHMES EN C/C++

1. Exercice 1- Saisie et calculs

Ecrire un programme qui lit le rayon d'un cercle et affiche le diamètre, la circonférence et l'aire de ce cercle. On utilisera la valeur constante 3.1415926 pour π .

2. Exercice 2 – Tests de valeur et calculs

On désire un programme effectuant la résolution d'une équation du second degré.

Rappel : soit une équation $f(x)$ telle que $f(x) = ax^2 + bx + c = 0$ avec $a \neq 0$

Avec a , b et c des coefficients entiers.

La résolution (trouver les valeurs de x qui annulent $f(x)$) passe par le calcul du discriminant Δ

$$\Delta = b^2 - 4ac$$

Si $\Delta > 0$ alors il y a 2 solutions (racines) x_1 et x_2 réels tels que :

$$x_1 = \frac{-b + \sqrt{\Delta}}{2a}$$

$$x_2 = \frac{-b - \sqrt{\Delta}}{2a}$$

Si $\Delta = 0$ alors il y a une racine double x :

$$x = \frac{-b}{2a}$$

Si $\Delta < 0$ alors il y a deux racines complexes conjuguées

$$x_1 = \frac{-b + i\sqrt{|\Delta|}}{2a}$$

$$x_2 = \frac{-b - i\sqrt{|\Delta|}}{2a}$$

Le programme lit les coefficients a , b et c et affiche les solutions de l'équation en précisant le signe du discriminant.

Algorithme (suggestion) :

- ➔ Déclarer les variables a , b et c de type entier (coefficients du polynôme $ax^2 + bx + c$)
- ➔ Déclarer la variable *delta* de type réel
- ➔ Déclarer les variables X_1 et X_2 de type réel
- ➔ Demander à l'utilisateur de saisir les valeurs de a , de b et de c
- ➔ Calculer le discriminant *delta*
- ➔ Tester le discriminant *delta*
 - Si $\text{delta} > 0$ alors
 - on calcule les deux solutions réelles X_1 et X_2
 - on affiche les solutions X_1 et X_2
 - Si $\text{delta} = 0$ alors
 - on calcule la solution double X_1 (on utilisera la variable X_1 au lieu de créer X)
 - on affiche la solution double X_1
 - Si $\text{delta} < 0$ alors

- on calcule séparément les parties réelles et imaginaires des deux solutions
 - on pourra utiliser X1 pour la partie réelle et X2 pour la partie imaginaire
- on affiche les deux solutions en deux parties
 - on affichera $X1 = [\text{partie réelle : } X1] + i*[\text{partie imaginaire : } X2]$
 - on affichera $X2 = [\text{partie réelle : } X1] - i*[\text{partie imaginaire : } X2]$

Éléments de cours :

Le test est effectué à l'aide de l'instruction `if(condition) ... else ...`

```
if (condition)
{
    //bloc exécuté si la condition est vraie
}
else
{
    //bloc exécuté si la condition est fausse
}
```

Par exemple :

```
if(z !=0)
{
    r = 1/z ;
}
else
{
    cout << "non autorisé" <<endl ;
}
```

Les tests à connaître :

- Lors des tests, la condition doit être vraie, c'est à dire égale à 1.
- Le test d'égalité : `==`
 - Par exemple : `if(a==0)...`
- Le test de différence : `!=`
 - Par exemple : `if(a !=0)...`
- Le test d'infériorité ou égale : `<=`
 - Par exemple : `if(a<=10)...`
- Le test de supériorité ou égale : `>=`
 - Par exemple : `if(a>=1)`
- Les tests à plusieurs propositions logiques :
 - Le ET logique
 - Par exemple : `if((a>0)&&(a<100))...`
 - Remarque : l'évaluation commence par le membre de gauche du `&&` S'il est d'entrée **faux** alors le second test n'est pas réalisé puisque c'est un ET
 - Le OU logique
 - Par exemple : `if((a==0) || (b==0))...`
 - Remarque : l'évaluation commence par le membre de gauche du `||` S'il est d'entrée **vrai** alors le second test n'est pas réalisé puisque c'est un OU

Il peut être intéressant de se souvenir des tables de vérité et du théorème de Morgan.