

TP6 POO - C++

gilles.scarella@univ-cotedazur.fr, simon.girel@univ-cotedazur.fr

Consignes:

- Pour tester vos codes sur les stations du PV, vous pouvez, par exemple:
 - Utiliser Visual Studio Code (commande *code*), en travaillant dans une fenêtre pour écrire le code et dans une autre, pour le terminal
 - Ou bien écrire votre code dans un éditeur (*gedit*) et le compiler et l'exécuter dans le terminal
- L'aide pour installer/configurer C++ sur votre machine se trouve dans Moodle, dans les fichiers *procedure_cpp_win.pdf* ou *procedure_cpp_mac.pdf*. Vous pouvez contacter les enseignants en cas de problème.

1 Superficie d'un champ

Dans le fichier *champ.cpp*, écrire un programme qui demande à l'utilisateur d'entrer la largeur et la longueur d'un champ et qui affichera sa superficie. On considérera des valeurs réelles.

NB: on utilisera respectivement *cin* et *cout* pour récupérer les entrées clavier et pour afficher une valeur à l'écran.

2 Moyenne (I)

Dans le fichier *moyenne1.cpp*, écrire un programme qui demande à l'utilisateur d'entrer 5 valeurs au clavier. Ce programme calculera la moyenne de ces valeurs et l'affichera. On devra choisir le type le plus général (le type entier étant trop restrictif).

NB: on ne demande pas d'utiliser un tableau.

3 Moyenne (II)

Exercice semblable à l'exercice 2, à la différence qu'ici on demande, au début, le nombre n de valeurs à entrer. Dans le fichier *moyenne2.cpp*, on affichera ensuite "Entrer $\langle n \rangle$ valeurs" en remplaçant $\langle n \rangle$ par sa valeur et on calculera la moyenne des n valeurs et on l'affichera.

4 Racines d'un polynôme

Dans le fichier *poly.cpp*, écrire un programme qui demande les 3 coefficients a , b et c du polynôme du second degré $ax^2 + bx + c$ et qui écrira à l'écran s'il a des racines réelles et, si c'est le cas, affichera leur valeur.

NB: Vous devrez inclure le header *cmath*, pour utiliser en C++ la librairie math de C, afin d'utiliser ici la fonction *sqrt*.

Voir plus d'info sur www.cplusplus.com pour connaître les fonctions de la librairie *cmath*.

5 Fonction puissance

Dans le fichier *myPow.cpp*, écrire une fonction *myPow* prenant deux paramètres et qui renvoie le premier paramètre élevé à la puissance du deuxième paramètre.

Cette fonction prendra deux paramètres, x et m , respectivement de type double et entier, et son type de retour sera double. Il conviendra de gérer tous les cas possibles, notamment le signe de l'exposant. Le cas où $x=0$ et $m<0$ ne doit pas être pris en compte.

Ecrire ensuite un programme qui testera la fonction sur plusieurs exemples avec différentes valeurs de x et de m . On comparera le résultat par rapport à la fonction *pow*, nécessitant le header *cmath*.

6 Fonction factorielle (I)

Dans le fichier *myFact.cpp*, écrire une fonction *myFactIt* qui renvoie la factorielle d'un nombre passé en paramètre, à l'aide d'une boucle. Tester votre fonction sur quelques exemples.

7 Fonction factorielle (II)

Dans le même fichier *myFact.cpp*, le but est d'écrire une fonction *myFactRec* qui renvoie la factorielle d'un nombre passé en paramètre à l'aide d'une récursion.

En C++, il est possible de définir des fonctions récursives, c'est-à-dire qui s'appellent elles-mêmes à condition que la récursion termine. Par exemple, si on veut calculer le n -ième terme de la suite (u_n) définie par $u_0 = 1$ et $u_{n+1} = 3u_n$, en C++ on peut écrire:

```
int computeU(int n)
{
    int U;
    if(n==0) U = 1;
    else
        U = 3*computeU(n-1);
    return U;
}
```

Ecrire la fonction récursive *myFactRec* toujours dans le fichier *myFact.cpp* et tester votre fonction.

8 Fonction racine carrée

Dans le fichier *mySqrt.cpp*, écrire la fonction *mySqrt* qui renvoie la racine carrée réelle du nombre réel passé en paramètre. On utilisera la méthode de Héron - voir sur Wikipédia "méthode de Héron", au lien suivant,

https://fr.wikipedia.org/wiki/M%C3%A9thode_de_H%C3%A9ron

Il conviendra de n'utiliser aucune des fonctions de la librairie C/C++ !

Pour obtenir un affichage précis avec plusieurs chiffres après la virgule, consulter la fonction *setprecision* de *iomanip* (utiliser dans l'entête `#include<iomanip>`)

Tester la fonction sur des exemples.

9 Fonction *estPremier* [Facultatif]

Pour un nombre entier inférieur à 10^6 , une manière de déterminer si ce nombre est premier est de tester s'il est divisible par n'importe quel entier plus petit que sa racine carrée.

- Dans le fichier *estPremier.cpp*, définir la fonction ***estPremier*** prenant comme seul argument un entier *n* non signé et renvoyant un booléen indiquant si *n* est premier ou non.
- Tester la fonction sur les trois entiers suivants en faisant afficher le résultat de l'appel à ***estPremier***, dans la fonction ***main***, écrite dans *estPremier.cpp*: 71, 2427, 87109

Indications:

- On pourra utiliser l'opérateur modulo, cité dans le cours, qui fournit le reste de la division entière.
- Il est nécessaire d'inclure *cmath* au début du fichier *estPremier.cpp* pour que la fonction racine carrée soit reconnue par le compilateur.