

Introduction au langage C++

Travaux pratiques

Énoncé

Youssef EL ALLIOUI

yelalloui@uhp.ac.ma

Notions de base

Exercice 1.

Ecrire un programme qui demande à l'utilisateur de taper la largeur et la longueur d'un champ et qui en affiche le périmètre et la surface.

```
#include<iostream>
using namespace std;
int main()
{
    double largeur, longueur, surface, perimetre;

    cout << "Tapez la largeur du champ : "; cin >> largeur;
    cout << "Tapez la longueur du champ : "; cin >> longueur;

    surface = largeur * longueur;
    perimetre = 2 * (largeur + longueur);

    cout << "La surface vaut : " << surface << endl;
    cout << "Le perimetre vaut : " << perimetre << endl;

    cout << "Appuyez sur une touche pour continuer." << endl;
    cin.ignore();
}
```

```
cin.get();  
  
return EXIT_SUCCESS;  
}
```

Exercice 2.

Ecrire un programme qui demande à l'utilisateur de taper 5 entiers et qui affiche leur moyenne. Le programme ne devra utiliser que 2 variables.

Exercice 3.

Ecrire un programme qui demande à l'utilisateur de saisir 2 entiers A et B, qui échange le contenu des variables A et B puis qui affiche A et B.

Exercice 4.

Ecrire un programme qui demande à l'utilisateur de taper le prix **HT** d'un kilo de tomates, le **nombre de kilos** de tomates achetés, le taux de **TVA** (Exemple 10%,20%,...). Le programme affiche alors le prix **TTC** des marchandises.

Structures de contrôles

Exercice 5.

Ecrire un programme qui demande à l'utilisateur de taper un entier et qui affiche GAGNE si l'entier est entre 56 et 78 bornes incluses PERDU sinon.

Exercice 6.

Ecrire un programme qui demande à l'utilisateur de taper 10 entiers et qui affiche leur somme.

Exercice 7.

Ecrire un programme qui demande à l'utilisateur de taper 10 entiers et qui affiche le plus petit de ces entiers.

Exercice 8.

Ecrire un programme qui demande à l'utilisateur de taper un entier N et qui calcule u(N) défini par :

$$u(0)=3$$

$$u(n+1)=3.u(n)+4$$

Exercice 9.

Ecrire un programme qui demande à l'utilisateur de taper un entier N et qui calcule u(N) défini par :

$$u(0)=1$$

$u(1)=1$
 $u(n+1)=u(n)+u(n-1)$

Exercice 10.

Ecrire un programme qui permet de faire des opérations sur un entier (valeur initiale à 0). Le programme affiche la valeur de l'entier puis affiche le menu suivant :

1. Ajouter 1
2. Multiplier par 2
3. Soustraire 4
4. Quitter

Le programme demande alors de taper un entier entre 1 et 4. Si l'utilisateur tape une valeur entre 1 et 3, on effectue l'opération, on affiche la nouvelle valeur de l'entier puis on réaffiche le menu et ainsi de suite jusqu'à ce qu'on tape 4. Lorsqu'on tape 4, le programme se termine.

Exercice 11.

Ecrire un programme qui demande à l'utilisateur de taper des entiers strictement positifs et qui affiche leur moyenne. Lorsqu'on tape une valeur négative, le programme affiche ERREUR et demande de retaper une valeur. Lorsqu'on tape 0, cela signifie que le dernier entier a été tapé. On affiche alors la moyenne. Si le nombre d'entiers tapés est égal à 0, on affiche PAS DE MOYENNE.

Exercice 12.

Ecrire un programme qui demande de saisir un entier N et qui affiche N!.

Exercice 13.

Ecrire un programme qui demande à l'utilisateur de saisir un entier N et qui affiche la figure suivante.

```
N=1
*

N=2
* *
*

N=3
* * *
* *
*
```

Exercice 14.

Ecrire un programme qui demande à l'utilisateur de saisir un entier N et qui affiche la figure suivante.

```
N=1
*
```

N=2

* *
*

N=3

* * *
* *
*

et ainsi de suite.

Les tableaux

Exercice 15.

Écrire un programme qui demande à l'utilisateur de saisir 10 entiers stockés dans un tableau. Le programme doit afficher le nombre d'entiers supérieurs ou égaux à 10.

Exercice 16.

Ecrire un programme qui demande à l'utilisateur de saisir 10 entiers stockés dans un tableau ainsi qu'un entier V. Le programme doit rechercher si V se trouve dans le tableau et afficher "V se trouve dans le tableau" ou "V ne se trouve pas dans le tableau".

Exercice 17.

Ecrire un programme qui demande à l'utilisateur de saisir 10 entiers stockés dans un tableau. Le programme doit ensuite afficher l'indice du plus grand élément.

Exercice 18.

Ecrire un programme qui demande à l'utilisateur de saisir 10 entiers stockés dans un tableau ainsi qu'un entier V. Le programme doit rechercher si V se trouve dans le tableau et doit supprimer la première occurrence de V en décalant d'une case vers la gauche les éléments suivants et en rajoutant un 0 à la fin du tableau. Le programme doit ensuite afficher le tableau final.

Exercice 19.

Ecrire un programme qui demande à l'utilisateur de taper 10 entiers qui seront stockés dans un tableau. Le programme doit trier le tableau par ordre croissant et doit afficher le tableau.

Algorithme suggéré :

- On cherche l'indice du plus petit élément parmi les indices de 0 à 9 et on échange cet élément avec t[0].
- On cherche l'indice du plus petit élément parmi les indices de 1 à 9 et on échange cet élément avec t[1].
- On cherche l'indice du plus petit élément parmi les indices de 2 à 9 et on échange cet élément avec t[2].
- ...

- On cherche l'indice du plus petit élément parmi les indices de 8 à 9 et on échange cet élément avec $t[8]$.

Exercice 20.

Ecrire un programme qui demande à l'utilisateur de taper 10 entiers qui seront stockés dans un tableau. Le programme doit trier le tableau par ordre croissant et doit afficher le tableau.

Algorithme suggéré (tri bulle) :

- On parcourt le tableau en comparant $t[0]$ et $t[1]$ et en échangeant ces éléments s'ils ne sont pas dans le bon ordre.
- on recommence le processus en comparant $t[1]$ et $t[2]$,
- ... et ainsi de suite jusqu'à $t[8]$ et $t[9]$.
- On compte lors de ce parcours le nombre d'échanges effectués.
- On fait autant de parcours que nécessaire jusqu'à ce que le nombre d'échanges soit nul : le tableau sera alors trié.

Ecrire un programme qui saisit 2 tableaux de 10 entiers a et b. c est un tableau de 20 entiers. Le programme doit mettre dans c la fusion des tableaux a et b. On copiera dans les 10 premières cases de c le tableau a, dans les dix dernières le tableau b. Le programme affiche ensuite le tableau c.

Exercice 21.

Ecrire un programme qui gère une liste d'entiers grâce au menu suivant :

1. Ajouter un entier
2. Afficher la liste des entiers
3. Supprimer dernier entier de la liste.
4. Afficher la dernière note tapée
5. Quitter

Il y aura au maximum 10 entiers. Lorsqu'on rajoute un entier, il sera rajouté à la fin de la liste.

Exercice 22.

Ecrire un programme qui gère une liste d'entiers grâce au menu suivant :

- 1 - Ajouter un entier
- 2 - Afficher la liste des entiers
- 3 - Supprimer le premier entier ayant une valeur donnée.
- 4 - Supprimer tous les entiers ayant une valeur donnée
- 5 - Quitter

Il y aura au maximum 10 entiers. La liste devra être en permanence triée : lorsqu'on rajoute un entier, il sera inséré au bon endroit dans la liste pour que celle-ci reste triée.

Les fonctions

Exercice 23.

Ecrire une fonction distance ayant comme paramètres 4 doubles (x_a , y_a) et (x_b , y_b) qui représentent les coordonnées de deux points A et B et qui renvoie la distance AB. Tester cette fonction.

Exercice 24.

Ecrire une fonction f ayant en paramètre un entier et qui renvoie par un return un booléen : true si l'entier est premier false sinon. Tester cette fonction.

Exercice 25.

Ecrire une fonction f ayant comme paramètre un entier n et qui renvoie le **nième** nombre premier : cette fonction utilisera la fonction du 3). Tester cette fonction.

Exercice 26.

Ecrire une fonction swap ayant en paramètres 2 entiers a et b et qui échange les contenus de a et de b . Tester cette fonction.

Exercice 27.

Ecrire une fonction f ayant en paramètres un tableau t de taille quelconque et un entier n indiquant la taille du tableau. f doit renvoyer par un return un entier égal à l'indice de la première case du tableau (parmi les n premières) comprise entre 0 et 10. S'il n'existe pas de telle valeur, la fonction renvoie -1. Tester cette fonction.

Exercice 28.

Réalisez un programme C++ permettant de calculer les solutions de l'équation

$$(E) : ax^2 + bx + c = 0 \quad a, b \text{ et } c \text{ sont des réels}$$

On prendra garde à bien tester tous les cas possibles :

- a est nul, et l'équation est en fait une équation du premier degré. Exemple : $4x-2=0$ donne une unique solution $x=0.5$.
- le discriminant $\Delta=(b^2- 4ac)$ est nul, et il n'y a qu'une seule solution, appelée racine double, au problème. Exemple : $2x^2+4x+2=0$ donne $x=-1$.
- le discriminant Δ est positif, et deux solutions existent : $x_1=(-b-\Delta^{1/2})/2a$ et $x_2=(-b+\Delta^{1/2})/2a$. Exemple : $2x^2+x-6=0$ donne $x_1=1.5$ et $x_2=-2$.
- le discriminant est négatif, et il n'existe pas de solutions (réelles) au problème.

NB : Afin d'utiliser la fonction prédéfinie racine carrée `sqrt()` de la bibliothèque C, vous prendrez garde à bien rajouter `#include <cmath>` en tête de votre programme.