

Labo1/Devoir1 – CSI2772A
Mardi/Vendredi 19/22 Septembre 2023
SITE - Université d'Ottawa
Dû EN LIGNE le vendredi 6 Octobre à 22:00
En groupes d'au plus deux étudiant(e)s.
/10

- **Objectifs**

- i) Mots réservés, Opérateurs, Types de donnée de base, Les spécificateurs de type, Les constantes, Conversion explicite, Entrées/sorties (cin, cout , cerr et clog), Structures de contrôle
- ii) Principes des fonctions, Passage des arguments par valeur et par référence, et Portée des variables.
- iii) Manipuler les tableaux et utiliser la classe **vector**

I. Rappels

I.1. Rappels sur les tableaux

I.1.a. Les tableaux à 1 dimension

Un tableau permet à partir d'un seul nom d'accéder à plusieurs données de même type (vecteurs). La définition générale d'un tableau est la suivante: **classe type tableau[taille];**

où **classe** spécifie la classe de mémorisation (extern, auto, static),
type le type des éléments (int, char, float, double, ...),
tableau le nom et
taille le nombre d'éléments composant ce tableau.

La définition d'un tableau nommé **tab** de 10 entiers se fera alors de la manière suivante:
int tab[10];

L'initialisation des éléments du tableau peut se faire lors de la définition:

int tab [10] = {10,9,8,7,6,5,4,3,2,1};

Dans le cas d'un tableau de caractères, cette initialisation est peu différente:

char chaine[20] = "Bonjour";

Le huitième élément de ce tableau sera automatiquement initialisé par le caractère de fin de chaîne **\0**.

Une fois défini, chacun des éléments peut être affecté par une valeur. L'accès aux différents éléments se fait en précisant son indice. Il faut noter que le premier élément possède un indice égal à **0**. L'affectation des 3 premiers éléments s'écrit alors:

```
tab[0]=10;  
tab[1]=9;  
...  
1
```

I.1.b. Les tableaux à plusieurs dimensions

La définition d'un tableau à **n** ($n > 1$) dimensions respecte la syntaxe générale suivante:

classe type tableau [taille 1][taille 2].....[taille n];

Il existe plusieurs manières d'initialiser ce type de tableau. Avec un tableau de 3 lignes et 4 colonnes, on peut initialiser celui-ci sous forme d'une liste ou ligne par ligne:

int tab[3][4]={1,2,3,4,5,6,7,8,9,10,11,12};

ou

```
int tab[3][4] = {{1,2,3,4},  
                  {5,6,7,8},  
                  {9,10,11,12}  
};
```

Cette initialisation est équivalente à l'ensemble des affectations suivantes:

```
tab[0][0]=1;tab[0][1]=2;tab[0][2]=3;tab[0][3]=4;  
tab[1][0]=5;tab[1][1]=6;tab[1][2]=7;tab[1][3]=8;  
tab[2][0]=9;tab[2][1]=10;tab[2][2]=11;tab[2][3]=12;
```

I.2. Algorithmes de tri

Comment trier un tableau d'entiers ?

rechercher le plus grand élément et le placer à la fin du tableau

recherche le deuxième plus grand et le placer en avant-dernière position

etc.

Algorithmes de tri

Comment trier un tableau d'entiers Tab ?

Pour IndFin = Tab.size()-1 ... 1 faire rechercher l'indice IMAX du plus grand élément de Tab entre les indices 0 à IndFin.

Permuter les éléments placés en IMAX et IndFin.

On sélectionne d'abord le plus grand élément et on le place à la fin du tableau ; puis on trouve le deuxième plus grand et on le place à l'avant-dernière place etc.

Fonction de sélection : renvoie l'indice de la valeur maximale d'un tableau passé en paramètre (T).

```
int rechercheIndex(vector<int> T, int imax) {  
    if (T.size() < imax - 1) {  
        cout << "Erreur ! Tableau trop petit !" << endl;  
        return -1;  
    }  
    int res = 0;  
    for (int i = 1; i <= imax; i++)  
        if (T[i] > T[res]) res = i;  
    return res;  
}
```

II. Devoir 1

À noter : Dans toutes les questions suivantes, vous n'êtes pas autorisés à modifier le programme principal (`main`) si ce n'est pas demandé (Ex 2 et 3), ni les signatures des fonctions demandées. Vous pouvez par contre ajouter des fonctions si vous désirez et les appeler dans les fonctions demandées.

Exercice 1 (2 POINTS) :

- Écrire un programme principal (`main`) dans un fichier source nommé "**monfichier1.cpp**" afin que votre programme vous retourne à l'écran le nombre d'octets utilisé par les types de donnée **char, int, float, double, unsigned int et short int**. Vous obtiendrez ces informations grâce à l'opérateur `sizeof`.
- A la suite de ces informations, déclarez un entier, un réel et un caractère que le programme vous demandera de saisir.
- Faites afficher le nombre entier en décimal, en octal, et en hexadécimal. Proposez 2 manières de réaliser cet affichage.
- Faites afficher le nombre réel avec et sans puissance de 10 (notation scientifique ou non) avec 3 chiffres significatifs.
- Faites afficher le caractère ainsi que sa valeur décimale.
- Soumettez votre fichier "**monfichier1.cpp**" ainsi qu'un fichier "**monfichier1.h**" qui doit contenir les fichiers en-tête nécessaires :

```
#include <iostream>
#include <iomanip>
using namespace std;

/*Exemple de SORTIE*/
```

Taille en octets d'un caractere: 1
Taille en octets d'un entier: 4
Taille en octets d'un reel: 4
Taille en octets d'un double: 8
Taille en octets d'un entier court: 2
Taille en octets d'un entier non signe: 4

Saisissez un entier: 50
nombre en decimal : 50
nombre en octal : 62
nombre en hexa : 32

nombre en decimal : 50
nombre en octal : 62
nombre en hexa : 32

Saisissez un reel: 12.354
12.354
0x1.8b5p+3

Saisissez un caractere: Hello
H
48

Exercice 2 (4 POINTS : 0.8 pts pour chaque) :

- a) Expliquer en quelques mots ce que fait la fonction *main* donnée en Annexe A2.
- b) Écrire les deux fonctions suivantes appelées dans la fonction *main*:
 - i) La fonction nommée *surface* qui calcule la surface d'un disque, et
 - ii) La fonction nommée *volume* qui calcule le volume d'une cylindre.
- c) Les définition des deux fonctions doivent être faites à l'extérieur de la fonction principale *main*.
 - i) Compléter le fichier en tête, *monfichier2.h*, donné en Annexe A2, en incluant la déclaration de ces deux fonctions manquantes.
 - ii) Donner le programme final à exécuter.

/*Exemple de SORTIE*/

Que souhaitez-vous faire ?:

- Calculer la surface du disque de rayon (Tapez 1)
- Calculer le volume du cylindre de rayon et hauteur (Tapez 2)
- Quittez le programme (Tapez 3)

Votre choix: 1

Calcul de la surface :

Donnez le rayon: 2

La surface est: 12.5664

Que souhaitez-vous faire ?:

- Calculer la surface du disque de rayon (Tapez 1)
- Calculer le volume du cylindre de rayon et hauteur (Tapez 2)
- Quittez le programme (Tapez 3)

Votre choix: 2

Calcul du volume :

Donnez le rayon: 1

Donnez la hauteur: 3

Le volume est: 9.42477

Que souhaitez-vous faire ?:

- Calculer la surface du disque de rayon (Tapez 1)
- Calculer le volume du cylindre de rayon et hauteur (Tapez 2)
- Quittez le programme (Tapez 3)

Votre choix: 3

Sortie du programme

La fonction volume a été lancée 1 fois

La fonction surface a été lancée 1 fois

Exercice 3 (4 POINTS : 2 pts pour chaque) :

- a) Définir la fonction *trier* dans le fichier *monfichier3a.cpp* donné ci-joint (dans la place indiquée), pour trier en ordre croissant un tableau passé en argument. Le tableau est passé par valeur. La fonction renvoie le tableau trié.

La fonction *trier* doit appeler la fonction de sélection *rechercheIndex* donnée ci-dessus dans le paragraphe I.2.

```
vector<int> trier(vector<int> T);
```

- b) Définir la fonction *trier* dans le fichier *monfichier3b.cpp* donné ci-joint, pour trier un tableau passé en argument par référence cette fois-ci, en ordre croissant. La fonction ne renvoie rien.

```
void trier(vector<int>& T);
```

/*Exemple de SORTIE pour les deux programmes a) et b)*/

Entrer la taille de votre tableau: 5

Saisir les valeurs de votre tableau:

```
val[ 0] =12  
val[ 1] =3  
val[ 2] =0  
val[ 3] =98  
val[ 4] =1
```

Les valeurs de mon tableau sont :

```
12  
3  
0  
98  
1
```

Les valeurs de mon tableau trié sont :

```
0  
1  
3  
12  
98
```

III. Annexes

III.1. Annexe A1

/*Fichier d'entête *monfichier2.h/**

```
#include <iostream>  
#include <iomanip>  
#include <process.h>  
  
using namespace std;  
  
const double Pi = 3.14159;  
  
char menu(void);  
int volume(double const& ray, double const& haut);  
int surface(double const& ray);
```

III.2 Annexe A2

```
/* Fonction principale main */
#include "monfichier2.h"

int main(void) {
    char choix;
    int nvolume = 0;           //nombre de fois que la fonction volume a été lancée
    int nsurface = 0;          //nombre de fois que la fonction surface a été lancée
    double rayon, hauteur;

    while (1)
    {
        choix = menu();
        switch (choix)
        {
            case '1': cout << endl << "Calcul de la surface :" << endl;
                        cout << "Donnez le rayon: ";
                        cin >> rayon;
                        nsurface = surface(rayon);
                        break;

            case '2': cout << endl << "Calcul du volume :" << endl;
                        cout << "Donnez le rayon: ";
                        cin >> rayon;
                        cout << "Donnez la hauteur: ";
                        cin >> hauteur;
                        nvolume = volume(rayon, hauteur);
                        break;

            case '3': cout << endl << "Sortie du programme" << endl;
                        cout << "La fonction volume a été lancée " << nvolume << " fois" << endl;
                        cout << "La fonction surface a été lancée " << nsurface << " fois" << endl;
                        exit(0);

            default: break;
        }
    }
}
```

III.3 Annexe A3

```
/*Fonction menu() : Fonction qui affiche un menu d'option et retourne l'option choisie */

char menu(void){
    char choix;

    cout << endl;
    cout << "Que souhaitez-vous faire ?: " << endl;
    cout << "\t-Calculer la surface du disque de rayon (Tapez 1)" << endl;
    cout << "\t-Calculer le volume du cylindre de rayon et hauteur (Tapez 2)" << endl;
    cout << "\t-Quitez le programme (Tapez 3)" << endl ;
    cout << "Votre choix: ";
    cin >> choix;

    return(choix);
}
```

IV. Créer et soumettre un seul fichier zip

Directives

- Créez un répertoire que vous nommerez *Devoir1_ID*, où vous remplacerez ID par votre numéro d'étudiant (celui qui soumet le devoir).
Mettez tous les fichiers suivants dans votre répertoire compressé *Devoir1_ID.zip* pour soumission dans le campus virtuel Brightspace.

Fichiers :

- ✓ *README.txt*
 - ✓ *monfichier1.cpp*
 - ✓ *monfichier1.h*
 - ✓ *monfichier2.cpp*
 - ✓ *monfichier2.h*
 - ✓ *monfichier3a.cpp*
 - ✓ *monfichier3b.cpp*

- N'oubliez pas d'ajouter des commentaires dans chaque programme pour expliquer le but du programme, la fonctionnalité de chaque méthode et le type de ses paramètres ainsi que le résultat.
- Dans le répertoire *Devoir1_ID*, créez un fichier texte nommé *README.txt*, qui devra contenir **les noms des deux étudiant(e)s**, ainsi qu'une brève description du contenu :

Nom étudiant :

Numéro d'étudiant :

Code du cours : CSI2772A

Fraude scolaire :

Cette partie du devoir a pour but de sensibiliser les étudiants face au problème de fraude scolaire (plagiat). Consulter les liens suivants et bien lire les deux documents:

<https://www.uottawa.ca/administration-et-gouvernance/reglement-scolaire-14-autres-informations-importantes>

https://www.uottawa.ca/administration-et-gouvernance/sites/www.uottawa.ca.administration-et-gouvernance/files/processus_de_traitement_des_cas_de_fraude_academique - nov 2019.pdf

Les règlements de l'université seront appliqués pour tout cas de plagiat.

En soumettant ce devoir :

1. vous témoignez avoir lu les documents ci-haut ;
2. vous comprenez les conséquences de la fraude scolaire.