Programmation 2

# Cours 1

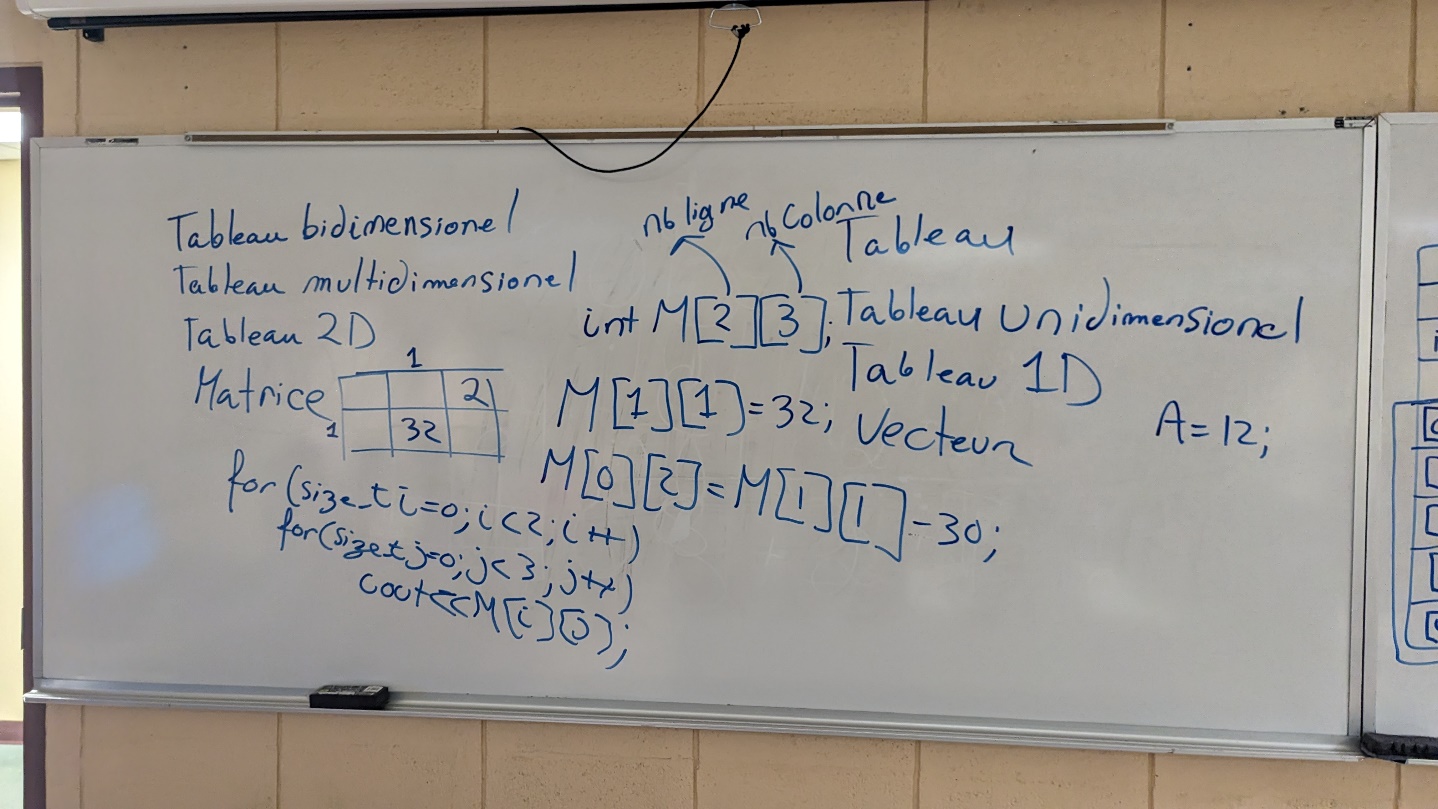
## Tableau:

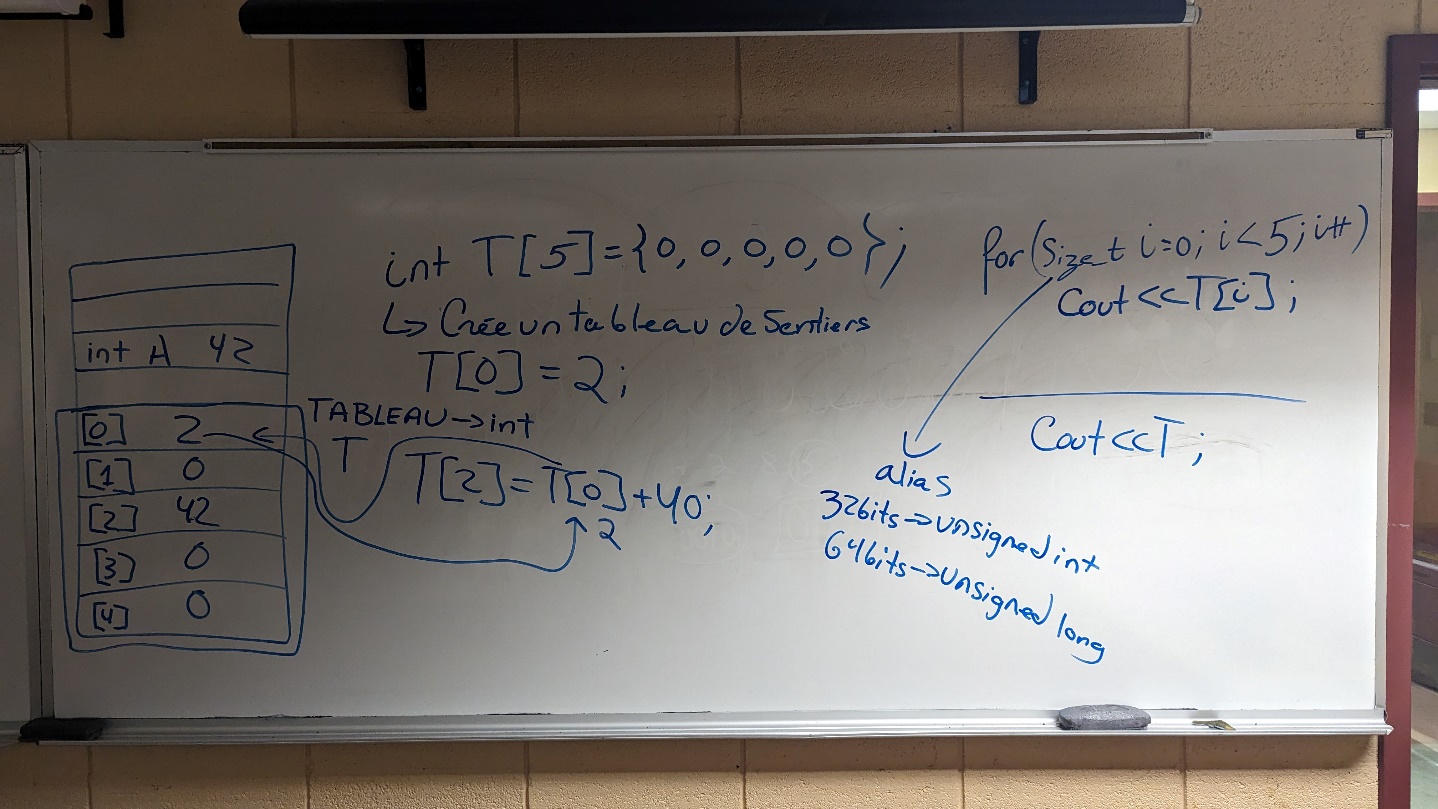
* Unidimensionnel
* 1D
* Vecteur
* Multidimensionnel
* Bidimensionnel
* 2D
* Matrice

Veulent tous dire la même chose

Une collection de valeur unique, ne sont que des strings, ou ne sont que des int, etc.

Size\_t : alias, sur 32 bits unsigned int, sur 64 bit unsigneed long. Créer un entier non signer positif le system va prendre le meilleur int





Cours 2 addition tableau

Tableau

* Afficher le contenue d<une case

type nom[taille1][taille2];

cout << nom[pos][pos] = valeur;

Mat1[i][j] = i+j

Mat2[i][j] = 2\*i+3\*j

Mat3[i][j] = mat1[i][j]+2\*mat2[i][j]

\*voir visual studio\*

Même code, on ne fait que changer l’opération

# Cours 2

## Les structures

Exemple pour un examen

* Nb question
* Pondération
* Note
* Nom de l’examen
* Le nom du cours
* Type de réponse
* Date
* Les questions à l’examen

Tout ceci est la structure, c’est décrire l’examen. Nous allons devoir qu’ajouter le type(float,String,int,etc.) Pas nécessairement besoin de tous les éléments il faut les tirer. Dans notre Example nous avons besoin de la liste des étudiants et de la note. C’est un nouveau type de variables.

La pondération, le type de réponse t la question vont souvent se répéter. On peut créer une nouvelle structure avec ces informations.

# Cours 3

## Les fonctions

* Les fonctions sont : int main() , sqrt(), pow(), setw(),. La plupart ont des parenthèse.
* Les fonctions sont faites pour être réutiliser un peu comme le fichier cvm.23
* Getch() ne prend pas de paramètre mais renvoie une valeur, setw prend un paramètre mais pas ne renvoie pas de valeur et clrsc ne prend aucun paramètre et ne revoit aucune valeur

Comment créer une fonction

Fonctionne sur les principes de déclaration et de définition

<Type retour> <nom>(<type><nom> <type><nom){

Int

For

Switch

Return<variable> (elle renvoie la reponse)

} ~~; (pas de point virgule)~~

Les fonction peuvent être appres le main

Signature : type de retour nom (et paramettre) {

Corps

} (c’est la définition)

Il peut se placer n’importe où, mais ne va pas fonction si la définition est après l’appel dans le main

## Prototype

* Déclaration
* Est la signature <type de retour> <nom> (paramètre);
* Ce place Au début avant le main

Définition

* La fonction elle meme
* Signature + le corps { … }
* Pet ce placer n<importe ou

Le passage

Passe les valeur depuis un parametre

* Passage par valeur
* Ex :

Foo(int a)

Main{

Int b =42;

Poo (b);

}

Si on met void on n’est pas obliger de mettre de return

Ex void passage(int a){}

# Surcharge

Different opateurs

+-\*/%

&&, | !

< <= == != > >=

(operateur de flu)

<<, >>

Iostream permet d’appeler cout et cin ( Ostream et Istream

* Permet de dire que je vais faire une surcharge

Void operator << (std ::osteamv & stream, const Vecteur{}& a)

* Stream est une variable, un flux appeler stream
* Osteam est un type de sortie

# [**Les pointeurs et les tableaux dynamiques**](https://cvm.moodle.decclic.qc.ca/course/view.php?id=1422#section-7)

L’adresse d’une case mémoire

C’est un valeur ou se qu’on garde l’adresse d’une autre variable ( ou elle se trouve)

Trouver l’adresse mémoire d’une variable

Ex :

Int main()^

Int base = 3;

Cout << &base;

Le & permet de montrer l’adresse mémoire

Pointeur

ex

Int base = 3;

Int\* pointeur = &base;

Cout << &base << \n << pointeur;

Les deux doivent être le même type de base

Ex

Int a = 0

Float b =0

Int\* pint =&a

Float pfloat = &b

Cout a va afficher 0

Cout b affiche 0

Cout pint va afficher l’adresse de a dans la memorie

Cout pfloat va fficher l’adresse de b dans al meoire

Cout \*pint va fficher 0 (a l’adresse de)

Cout\*pfloat va affich0er 0

Dans une autre fonction

Void Foo (int \* a){

\*a = 50

/si pas de \* alors se sont deux different a

}

Cout a va afficher 50

Deferencement

Permet d<aller chercher la valeur avec le pointeur

Int base = 3;

Int\* pointeur = &base;

Cout << base << \n << \*pointeur;

Haritmetique des pointeur

Cout << pointeur +1’

Le pointeur fera plus un donc va augmenter l<adresse de 4 octet comme c<est un float

Faire attention car il faut pas modoifier certain adress mémoire

Malloc et free

Int main () {

Int taille,

Cout << «combien ? » ;

Cin >> taille;

//alloue la taille correct

Void\*variabe =malloc(taille);

//liberer la memoire

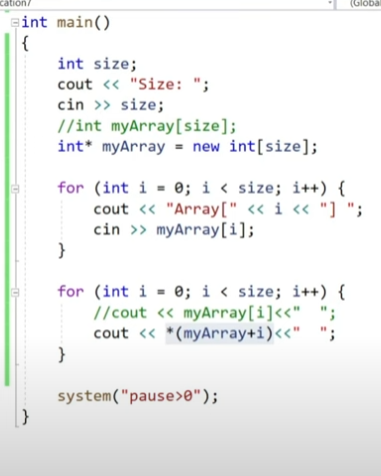
Free(variable);

Prend des octeets donc on peut rentrer 10000000 -🡪 10octet

AUTRE

Int\*variable = new int;

Cout << \*variable;

Delete variable; // même chose que new permet de libérer la mémoire 

Delete[]myArray

MyArray = NULL

(devant system(pause) )