

## Séance 9 : Jeudi 07/03/2019

CM5 (première partie) : Tableau 1D algo / C

Télécharger les slides du cours sur la page web de l'UE

Fonctions de manipulation de base des tableaux 1D en C/C++

Initialisation / Affichage / Remplissage avec un tableau d'entiers de taille 4. Découverte de la structure de données tableau, utilisation dans les sous-programmes.

```
#include<iostream>
#include<math.h>
using namespace std;

void initialisation (int T[4]) // procédure d'initialisation
{
    int i;
    for (i=0;i<4;i++)
    {
        T[i] = 0;
    }
}

void affichage (int T[4])
{
    int i;
    for (i=0;i<4;i++)
    {
        cout<<T[i]<<" | ";
    }
    cout<<endl;
}

void remplissage (int T[4])
{
    int i;
    for (i=0;i<4;i++)
    {
        cout<<"Donnez un entier"<<endl;
        cin>>T[i];
    }
}

int main (void)
{
    int T[4], i;
    affichage(T);
    initialisation(T);
    affichage(T);
    remplissage(T);
    affichage(T);
    return 0;
}
```

Taille des tableaux définie par une constante dans le programme => adaptation rapide et simple aux différents problèmes.

```

#include<iostream>
#include<math.h>
using namespace std;
const int MAX = 4 ;
void initialisation (int T[MAX]) // procédure d'initialisation
{
    int i;
    for (i=0;i<MAX;i++)
    {
        T[i] = 0;
    }
}

void affichage (int T[MAX])
{
    int i;
    for (i=0;i<MAX;i++)
    {
        cout<<T[i]<<" | ";
    }
    cout<<endl;
}

void remplissage (int T[MAX])
{
    int i;
    for (i=0;i<MAX;i++)
    {
        cout<<"Donnez un entier"<<endl;
        cin>>T[i];
    }
}

int main (void)
{
    int T[MAX], i;
    affichage(T);
    initialisation(T);
    affichage(T);
    remplissage(T);
    affichage(T);
    return 0;
}

```

## Exercices TD / TP sur les tableaux

Écrire un sous-programme RechercheValMin qui permet de retourner la plus petite valeur contenue dans le tableau.

```

int RechercheValMin (int T[MAX])
{
    int mini = T[0], i;
    for (i=1;i<MAX;i++)
    {
        if (T[i]<mini)
            mini = T[i];
    }
    return mini ;
}

```

Écrire un sous-programme RechercheMin qui permet de retourner l'indice de la plus petite valeur contenue dans le tableau.

```
int RechercheIndMin (int T[MAX])
{
    int mini = T[0], i, indmin = 0;
    for (i=1; i<MAX; i++)
    {
        if (T[i]<mini)
        {
            mini = T[i];
            indmin = i ;
        }
    }
    return indmin;
}
```

Ou en utilisant uniquement les indices :

```
int RechercheIndMin (int T[MAX])
{
    int i, indmin = 0;
    for (i=1; i<MAX; i++)
    {
        if (T[i]<T[indmin])
        {
            indmin = i ;
        }
    }
    return indmin;
}
```

Écrire un sous-programme SommeTableau qui calcule et retourne la somme des valeurs du tableau précédent.

```
int SommeTab(int T[MAX])
{
    // calculer la somme de tous les elements du tableau
    int i, som = 0;
    for (i=0; i<MAX ; i++)
    {
        som += T[i];
    }
    return som ;
}
```

Transformer la fonction précédente en procédure

```
void SommeTab(int T[MAX], int &som )
{
    // calculer la somme de tous les elements du tableau
    int i;
    som = 0 ;
    for (i=0; i<MAX ; i++)
    {
        som += T[i];
    }
}
int main (void)
{
    int T[MAX], i, s;
    affichage(T);
    initialisation(T);
    affichage(T);
}
```

```

    remplissage(T);
    affichage(T);
    SommeTab(T,s);
    cout<<"La somme des elements du tableau est : "<<s<<endl;
    return 0;
}

```

Écrire l'algorithme d'un sous-programme permettant de renverser le contenu d'un tableau

Dans un autre tableau :

```

void RenverseTab(int T[MAX], int Tbis[MAX])
{
    int i;
    for (i=0;i<MAX;i++)
    {
        Tbis[i] = T[MAX-1 -i];
    }
}

```

Dans le même tableau

```

void RenverseMemeTab(int T[MAX])
{
    int i,t;
    for (i=0;i<(MAX)/2;i++)
    {
        cout<<"on va permuter la case "<<i<< " avec la case "<<MAX-1 -
i<<endl;
        t = T[i];
        T[i] = T[MAX-1 - i];
        T[MAX-1 - i] = t;
    }
}
int main (void)
{
    int T[MAX], i, s,T2[MAX];
    initialisation(T);
    remplissage(T);
    affichage(T);
    RenverseMemeTab(T);
    affichage(T);
    return 0;
}

```

Écrire un sous-programme `Cumul` qui remplacera les valeurs de chaque case par la somme des valeurs des cases précédentes.

```

void CumulTab(int T[MAX], int TCumul[MAX])
{
    int i;
    TCumul[0] = T[0] ;
    for (i=1;i<MAX;i++)
    {
        TCumul[i] = TCumul[i-1] + T[i];
    }
}
int main (void)
{
    int T[MAX], i, s,T2[MAX];
    initialisation(T);
    remplissage(T);
    affichage(T);
}

```

```

    CumulTab(T, T2);
    affichage(T2);
    return 0;
}

```

### Avec une double boucle !

```

void CumulTab(int T[MAX], int TCumul[MAX])
{
    int i,j;
    for (i=0;i<MAX;i++)
    {
        for (j=0;j<=i;j++)
        {
            TCumul[i] += T[j];
        }
    }
}

```

On dispose de deux tableaux de tailles identiques T1 et T2. On souhaite à partir de ces deux tableaux constituer un troisième tableau (noté T3) contenant les moyennes pondérées par les coefficients suivants : T1 coefficient 1, et T2 coefficient 2. Écrire un sous-programme `CalculMoyenneTableau` permettant à partir des deux premiers tableaux de remplir un troisième tableau T3 en utilisant les coefficients donnés.

```

void CalculeMoyenneTableau(int T1[MAX], int T2[MAX], float T3[MAX])
{
    int i;
    for (i=0;i<MAX;i++)
    {
        T3[i] = (T1[i] + 2*T2[i]) / 3.0 ;
    }
}
int main (void)
{
    int T1[MAX],T2[MAX];
    float T3[MAX];
    remplissage(T1);
    remplissage(T2);
    CalculeMoyenneTableau(T1,T2,T3);
    affichage(T3);
    return 0;
}

```

### Et en paramétrant le nombre de cases utilisées ....

```

#include<iostream>
#include<math.h>
using namespace std;
const int MAX = 100 ;
void initialisation (int T[MAX], int nb) // procédure d'initialisation
{
    int i;
    for (i=0;i<nb;i++)
    {
        T[i] = 0;
    }
}

void affichage (int T[MAX], int nb)
{
    int i;
    for (i=0;i<nb;i++)
    {
        cout<<T[i]<<" | ";
    }
}

```

```

        cout<<endl;
    }
void affichage (float T[MAX], int nb)
{
    int i;
    for (i=0;i<nb;i++)
    {
        cout<<T[i]<<" | ";
    }
    cout<<endl;
}

void remplissage (int T[MAX], int nb)
{
    int i;
    for (i=0;i<nb;i++)
    {
        cout<<"Donnez un entier"<<endl;
        cin>>T[i];
    }
}

int RechercheValMin (int T[MAX], int nb)
{
    int mini = T[0],i;
    for (i=1;i<nb;i++)
    {
        if (T[i]<mini)
            mini = T[i];
    }
    return mini ;
}

int RechercheIndMin (int T[MAX], int nb)
{
    int i, indmin = 0;
    for (i=1;i<nb;i++)
    {
        if (T[i]<T[indmin])
        {
            indmin = i ;
        }
    }
    return indmin;
}

void SommeTab(int T[MAX], int &som, int nb )
{
    // calculer la somme de tous les elements du tableau
    int i;
    som = 0 ;
    for (i=0;i<nb ; i++)
    {
        som += T[i];
    }
}

void RenverseTab(int T[MAX], int Tbis[MAX], int nb)
{
    int i;
    for (i=0;i<nb;i++)
    {
        Tbis[i] = T[nb-1 -i];
    }
}

void RenverseMemeTab(int T[MAX], int nb)
{

```

```

    int i,t;
    for (i=0;i<(nb)/2;i++)
    {
        cout<<"on va permuter la case "<<i<<" avec la case "<<nb-1 -
i<<endl;
        t = T[i];
        T[i] = T[nb-1 - i];
        T[nb-1 - i] = t;
    }
}

void CumulTab(int T[MAX], int TCumul[MAX], int nb)
{
    int i,j;
    for (i=0;i<nb;i++)
    {
        for (j=0;j<=i;j++)
        {
            TCumul[i] += T[j];
        }
    }
}

void CalculeMoyenneTableau(int T1[MAX], int T2[MAX], float T3[MAX], int nb)
{
    int i;
    for (i=0;i<nb;i++)
    {
        T3[i] = (T1[i] + 2*T2[i]) / 3.0 ;
    }
}

int saisie_valeur()
{
    int n;
    do
    {
        cout<<"Combien de cases veux-tu utiliser sur les 100 possibles
?"<<endl;
        cin >>n ;
    }
    while (n<1 || n>MAX);
    return n;
}

int main (void)
{
    int T1[MAX],T2 [MAX], nb;
    float T3[MAX];

    nb=saisie_valeur();
    remplissage(T1,nb);
    RechercheValMin(T1,nb);
    affichage(T1,nb);
    return 0;
}

```

Écrire un sous-programme qui calcule et "retourne" un tableau contenant les N premiers termes de la suite  $U_n$  définie par :

```

-- - - -
#include<iostream>
#include<math.h>
using namespace std;
const int MAX = 100 ;
void affichage (float T[MAX], int nb)
{

```

```

    int i;
    for (i=0;i<nb;i++)
    {
        cout<<T[i]<<" | ";
    }
    cout<<endl;
}
int saisie_valeur()
{
    int n;
    do
    {
        cout<<"Combien de cases veux-tu utiliser sur les 100 possibles
?"<<endl;
        cin >>n ;
    }
    while (n<1 || n>MAX);
    return n;
}

void RemplitSuite(float T[MAX], int nb)
{
    int i;
    T[0] = 1;
    for (i=1 ; i<nb;i++)
    {
        T[i] = T[i-1] / i ;
    }
}
int main (void)
{
    float Tab_Suite[MAX];
    int nb_val = saisie_valeur();
    RemplitSuite(Tab_Suite,nb_val);
    affichage(Tab_Suite,nb_val);
    return 0;
}

```