Algorithmique et Programmation C++

TD ❷

1. Tri à bulle

```
#include <iostream.h>
#include<math.h>
#include<stdlib.h>


// Prototypes :

long NbrAleat(long limite);
void TabAleat(long Tab[],int,long,long);
void TabDisp(long Tab[],int,int,int);

// Definitions des fonctions :

long NbrAleat(long max)
{
        long a=rand();
        while (a>10000)
                a = a - 10000;
        return static_cast<long>(max*a/10000);
}

void TabAleat(long Tableau[],int taille, long max=99,long min=0)
{
        for (int i=0; i<taille; i++)
                Tableau[i]=NbrAleat(max-min)+min;
}

void TabDisp(long Tableau[], int taille, int min=0, int max=0)
{
        if (min<0 || min>taille) min=0;
        if (max<=min || max>taille) max=taille;
        for (int a=min; a<max;a++)
                cout<<Tableau[a]<<";";
}

// Fin de l'entete...
// Debut du programme : Exo 01 - TD2

int main()
{
cout << "> Methode de tri a bulle en C++\n";
cout << "  ------------------------------\n";
const int N=50;
int k,n;
long T[N];
cout << "\nCombien d'essais voulez-vous ? ";
cin >> n;
if (n<1) n=1;
while (k<n)
{
        cout << "\n\n> Essai "<< ++k << "\n";
        cout << "  -----\n";
        TabAleat(T,N);
        cout << "\n  - AVANT : ";TabDisp(T,N);
        long mem;
        bool trie=false;
        while (!trie)
        {
                trie=true;
                for (int i=0; i<N-1;i++)
                {
                        if (T[i]>T[i+1])
                        {
                                mem=T[i];T[i]=T[i+1];T[i+1]=mem;
                                trie=false;
                        }
                }
        }
        cout << "\n\n  - APRES : ";TabDisp(T,N);
}
cout << "\n";
}
```

2. Tri par insertion

```
#include<iostream.h>
#include<math.h>
#include<stdlib.h>


// Prototypes :

long NbrAleat(long limite);                    // Generateur de nombres
pseudo-aleatoires
void TabAleat(long Tab[],int,long,long);  // Remplissage aleatoire
d'un tableau
void TabDisp(long Tab[],int,int,int);          // Affichage d'un
tableau

// Definitions des fonctions :

long NbrAleat(long max)
{
        long a=rand();
        while (a>10000)
                a = a - 10000;
        return static_cast<long>(max*a/10000);
}

void TabAleat(long Tableau[],int taille, long max=99,long min=0)
{
        for (int i=0; i<taille; i++)
                Tableau[i]=NbrAleat(max-min)+min;
}


void TabDisp(long Tableau[], int taille, int min=0, int max=0)
{
        if (min<0 || min>taille) min=0;
        if (max<=min || max>taille) max=taille;
        for (int a=min; a<max;a++)
                cout<<Tableau[a]<<";";
}


// Programme : Exo 02 - TD2

int main()
{
        const int N=100; long T[N];
        TabAleat(T,N);
        int j;
        long aux;
        cout<<"Tableau initial :\n";
        TabDisp(T,N);

        ////////////////////////////////////////
        for (int i=1; i<N; i++)
        {
                j=i-1;
                while(j>=0 && T[j]>T[i]) j--;
                aux=T[i];
                for (int k=i-1;k>=j+1;k--)
                        T[k+1]=T[k];
                T[j+1]=aux;
        }
        ////////////////////////////////////////

        cout<<"\nTableau trie :\n";
        TabDisp(T,N);
        cout<<"\n";
}
```

3. Comptage et compactage

```
#include <iostream.h>

int main()
{
        const int N=20;

        int T[N]={1,1,1,3,3,6,8,8,8,8,4,4,4,4,4,4,2,2,9,9};
        int T2[N],EFF[N],TF[2*N];
        int value, idx=0, occ=0;

        // Boucle principale
        value=T[0];
        for (int i=1; i<=N; i++)
        {
                if (value==T[i] && i!=N)
                {
                        occ++;
                }
                else
                {
                        //Ecrit la valeur dans T2 et EFF :
                        T2[idx]=value;
                        EFF[idx]=occ+1;
                        //Ecrit la valeur dans TF :
                        TF[2*idx]=occ+1;
                        TF[2*idx+1]=value;
                        //Reinitialise les compteurs :
                        occ=0;
                        idx++;
                        if (i<N) value=T[i];
                }
        }

        //Affichage du resultat
        cout << "Resultat du comptage-compactage...\n";
        cout << "T:   ";
        for (int a=0; a<N; a++)
                cout<<T[a]<<";";
        cout << "\nT2:  ";
        for (int b=0; b<idx; b++)
                cout<<T2[b]<<";";
        cout << "\nEFF: ";
        for (int c=0; c<idx; c++)
                cout<<EFF[c]<<";";
        cout << "\nTF:  ";
        for (int d=0; d<2*idx; d++)
                cout<<TF[d]<<";";
        cout << "\n";
}
```

4. Puissance

```
#include <iostream.h>

float puiss(float X, int N)
{
        float R=1;
        while (N!=0)
        {
                if (N%2==1) R=R*X;
                N=N/2;
                X=X*X;
        }
        return R;
}

int main()
{
        float X;
        cout << "> Test de la fontion puiss(float,int)\n";
        cout << "Entrez une valeur pour X : ";
        cin >> X;
        for(int i=2; i<=20; i++)
                cout << "X puissance " << i << " : " << puiss(X,i) <<
"\n";
}
```

5. Racine d'une fonction

```
#include <iostream.h>
#include <math.h>

float f(float x)
{
        // Fonction quelconque pour tester...
        return x*x*x-2.34;
}

float zero(float eps, float A, float B)
{
        float Z=(A+B)/2;
        while(abs(f(Z))>=eps)
        {
                if (f(Z)*f(A)>0) A=Z; else B=Z;
                Z=(A+B)/2;
        }
        return Z;
}

int main()
{
        // Exo 05 : Test de la fonction zero(float,float,float)
        float A,B,e,z;
        cout << "Entrez deux valeurs differentes pour A et B telles
que A<B :\n";
        cout << "A="; cin >> A;
        cout << "B="; cin >> B;
        cout << "Entrez un epsilon suffisamment petit :\n";
        cout << "Epsilon="; cin >> e;
        cout << "Recherche dichotomiqe de la racine de f en cours...
(ca le fait non ?) :\n";
        z=zero(e,A,B);
        cout << "Zero trouve a " << e << " pres : "<<z;
        cout << "\nNous vous remercions de votre visite. Au
revoir.\n";
}
```

6. Carré magique

```
#include <iostream.h>
#include <iomanip.h>

int main()
{
        // Titre JOLI :
        cout << "------------ GENERATEUR DE CARRES MAGIQUES";

        // Declarations :
        int A[100][100];
        bool bascule;

        // Entree de la taille du tableau :
        int n=0;
        while (!(n%2==1 && n>2 && n<100))
        {
                cout << "\nEntrez la taille du carre : ";
                cin >> n;
                if (n%2==0) cout << "Vous devez entrer un nombre
impair...";
                if (n<2) cout << "Taille trop petite...";
                if (n>100) cout << "Taille trop grande...";
        }
        int x=(n-1)/2+1, y=(n-1)/2;


        // Initialisation du carre a zero :
        for (int i=0;i<n;i++)
                for (int j=0;j<n;j++)
                        A[i][j]=0;

        // Boucle principale :
        for (i=1; i<=n*n; i++)
        {
                while (A[x][y]!=0)
                {
                        if (bascule) {x=(x+1)%n;y=(n+y-1)%n;}
                        else {x=(x+1)%n;y=(y+1)%n;}
                        bascule=!bascule;
                }
                A[x][y]=i;
                x=(x+1)%n;
                y=(y+1)%n;
                bascule=true;
        }

        // Affichage du resultat :
        for (int a=0; a<n; a++)
        {
                for (int b=0; b<n; b++)
                        cout << setw(4) << A[a][b];
                cout << "\n";
        }
}
```

7. Drapeau

```
/*  EXERCICE 7 : Le drapeau francais.....................  *\
 *  Avec affichage detaille de l'avancement du rangement  *
\*  ................................Par Jeremie Osmont  */


#include <iostream.h>

void ech(int,int);   // Echange de deux valeurs du drapeau
char c(int);         // Donne la couleur d'un element du drapeau

char T[]="brwwrbbwrwrbbrww"; // Par exemple !

int main()
{
        int n=sizeof(T)/sizeof(T[0])-1;
        int b,w,r,k,i;
        b=0;
        w=r=n;
        while(b<w)
        {
                cout << "\n\nIteration "<<++k<<" : c(b)="<<c(b);
                switch (c(b))
                {
                        case 'b' : b++;
                                break;
                        case 'r' :  ech(r-1,w-1);ech(b,r-1);r--;w--;
                                break;
                        case 'w' : ech(b,w-1);w--;
                                break;
                }
                cout << "\n   => Etat des variables :  b=" << b << "
w=" << w << "  r="<<r;
                cout << "\n   => Avancement : ";
                for(i=0;i<n;i++)
                        cout<<" "<<T[i];
        }
        cout<<"\n\n";
}

// Fonctions (non demande dans l'enonce) :

void ech(int x,int y)
{
         cout<<"\n   => Appel de la fonction ech...
(x="<<x<<",y="<<y<<")";
        char aux=T[x];
        cout<<"\n        1) aux="<<aux;
        T[x]=T[y];
        cout<<"     2) T[x]="<<T[x];
        T[y]=aux;
        cout<<"     3) T[y]="<<T[y];
}

char c(int i)
{
        return T[i];
}
```

8. Produit de matrices

```
/*************************************************\
 * Exercice 8 TD2 : Produit de deux matrices        *
 * Par Jeremie Osmont  -  http://www.jeyland.fr.st  *
\*************************************************/


#include <iostream.h>
#include <iomanip.h>
#include <stdlib.h>
#include <math.h>


// Generateur de nombres pseudo-aleatoires :
long NbrAleat(long limite);

// Remplissage aleatoire d'un tableau :
void TabAleat(long Tab[],int,long,long);

// Definitions des fonctions :

long NbrAleat(long max)
{
        long a=rand();
        while (a>10000)
                a = a - 10000;
        return static_cast<long>(max*a/10000);
}

void TabAleat(long Tableau[],int taille, long max=99,long min=0)
{
        for (int i=0; i<taille; i++)
                Tableau[i]=NbrAleat(max-min)+min;
}


// Debut du programme :

int main()
{

        // Declarations :

        const int nligA=5,  // par exemple...

                  ncolA=4,  // Veillez a ce que ces
                  nligB=4,  // deux valeurs soient egales !

                  ncolB=7;  // au hasard...

        long A[nligA][ncolA];
        long B[nligB][ncolB];
        long C[nligA][ncolB];
        long r;

        // Remplissage aleatoire des matrices A et B :

        srand(0);

        for (int a=0; a<nligA; a++)
                TabAleat(A[a],ncolA,5,-5);

        for (int b=0; b<nligB; b++)
                TabAleat(B[b],ncolB,5,-5);

        // Debut du calcul du produit :

        if(nligB==ncolA)
        {
                for (int i=0;i<nligA;i++)
                        for (int j=0; j<ncolB;j++)
                        {
                                r=0;
                                for (int k=0;k<nligB;k++)
                                        r=r+A[i][k]*B[k][j];
                                C[i][j]=r;
                        }
        }
        else
                cout<<"Erreur de dimension...\n";

        // Affichage du resultat :

        cout <<"\nMatrice A :\n";
        for (a=0; a<nligA; a++)
                {for (b=0; b<ncolA; b++)
                        cout << setw(5) << A[a][b];
                cout<<"\n";}

        cout <<"\nMatrice B :\n";
        for (a=0; a<nligB; a++)
                {for (b=0; b<ncolB; b++)
                        cout << setw(5) << B[a][b];
                cout<<"\n";}

        cout <<"\nMatrice C=AxB:\n";
        for (a=0; a<nligA; a++)
                {for (b=0; b<ncolB; b++)
                        cout << setw(5) << C[a][b];
                cout<<"\n";}

}
```