

# Introduction au langage C++

## Travaux pratiques Correction

Youssef EL ALLIOUI

[y.elalloui@usms.ma](mailto:y.elalloui@usms.ma)

### Notions de base

#### Exercice 1.

```
#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.

Cet exercice a pour but de vérifier les points techniques suivants :

- La notion de variables et leur déclaration.
- Calcul du moyenne.
- Utilisation des types int et double.
- L'utilisation de cin et de cout.
- L'affectation.

Voici le fichier source :

```
#include<iostream>
using namespace std;
int main()
{
    int a;double s=0;

    cout<<"Tapez la valeur numero 1 : ";cin>>a; s=s+a;
    cout<<"Tapez la valeur numero 2 : ";cin>>a; s=s+a;
    cout<<"Tapez la valeur numero 3 : ";cin>>a; s=s+a;
    cout<<"Tapez la valeur numero 4 : ";cin>>a; s=s+a;
    cout<<"Tapez la valeur numero 5 : ";cin>>a; s=s+a;

    s=s/5.0;
    cout<<"La moyenne vaut : "<<s<<endl;

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

    return EXIT_SUCCESS;
}
```

## Exercice 3.

Cet exercice a pour but de vérifier les points techniques suivants :

- La notion de variables et leur déclaration.
- L'utilisation de cin et de cout.
- L'affectation.
- Un "algorithme" rudimentaire : échanger le contenu de 2 variables.

Voici le fichier source :

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

int main()
{
    int a,b,temp;

    cout<<"Tapez la valeur de a : ";cin>>a;
    cout<<"Tapez la valeur de b : ";cin>>b;
```

```

temp=a;
a=b;
b=temp;

cout<<"La valeur de a est "<<a<<endl;
cout<<"La valeur de b est "<<b<<endl;

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

return EXIT_SUCCESS;
}

```

## Exercice 4.

Cet exercice a pour but de vérifier les points techniques suivants :

- La notion de variables et leur déclaration.
- Le choix d'identificateurs pertinents et explicites.
- L'utilisation de cin et de cout.
- L'affectation.
- Modélisation d'un problème "économique".

Voici le fichier source :

```

#include<iostream>
using namespace std;
int main()
{
double prixht,poids,tva,total;

cout<<"Tapez le prix HT d'un kilo de tomates : "; cin>>prixht;
cout<<"Combien de kilos avez-vous achetés : "; cin>>poids;
cout<<"Quel est le taux de TVA : "; cin>>tva;

total=(1+tva/100)*prixht*poids;

cout<<"Le prix TTC est : "<<total<<endl;

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

return EXIT_SUCCESS;;
}

```

## Structures de contrôles

## Exercice 5.

Cet exercice a pour but de vérifier les points techniques suivants :

- La notion de variables et leur déclaration.

- L'utilisation de cin et de cout.
- Le choix d'une structure de contrôle adaptée au problème !

Voici le fichier source :

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

int main()
{
    int a;
    cout<<"Tapez un entier : ";cin>>a;
    if ((a>=56) && (a<=78)) cout<<"GAGNE"<<endl; else cout<<"PERDU"<<endl;

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

## Exercice 6.

Cet exercice a pour but de vérifier les points techniques suivants :

- Utilisation d'un for.
- Etude d'un algorithme usuel : calcul d'une somme.

Voici le fichier source :

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

int main()
{
    int i,s=0,x;

    for(i=0;i<10;i++)
    {
        cout<<"Tapez un entier : ";cin>>x;
        s=s+x;
    }

    cout<<"La somme vaut : "<<s<<endl;

    return 0;
}
```

## Exercice 7.

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

int main()
{
    int i,ppt,x;
```

```

for(i=0;i<10;i++)
{
    cout<<"Tapez un entier : ";cin>>x;
    if(i==0)ppt=x;else if(x<ppt)ppt=x;
}

cout<<"Le plus petit vaut vaut : "<<ppt<<endl;

return 0;
}

```

## Exercice 8.

Cet exercice a pour but de vérifier les points techniques suivants :

- Utilisation simple du for.
- Etude d'un algorithme usuel : calcul des termes d'une suite récurrente.
- Modélisation d'un problème issu des mathématiques.

Voici le fichier source :

```

#include<iostream>
using namespace;
int main()
{
    int i,u=3,N;

    cout<<"Tapez N : ";cin>>N;

    for(i=0;i<N;i++)
        u=u*3+4;

    cout<<"u ("<<N<<")="<<u<<endl;

    return 0;
}

```

## Exercice 9.

Cet exercice a pour but de vérifier les points techniques suivants :

- Utilisation simple du for.
- Etude d'un algorithme usuel : calcul d'une suite récurrente.
- Modélisation d'un problème simple issu des mathématiques.

Voici le fichier source :

```

#include<iostream>
using namespace std;

int main()
{
    int i,u=1,v=1,w,N;

    cout<<"Tapez N : ";cin>>N;

    w=1;

```

```

for (i=2; i<=N; i++)
{
    w=u+v;
    u=v;
    v=w;
}

cout<<"u ("<<N<<")="<<w<<endl;

return 0;
}

```

## Exercice 10.

Cet exercice a pour but de vérifier les points techniques suivants :

- Utilisation du while.
- Utilisation du switch.
- Gestion d'un programme à l'aide d'un menu.
- Modélisation d'un problème simple sous forme informatique.

Voici le fichier source :

```

#include<iostream>
using namespace std;

int main()
{
    int x=0, choix;

    do
    {
        cout<<"x vaut "<<x<<endl;
        cout<<"1 : Ajouter 1"<<endl;
        cout<<"2 : Multiplier par 2"<<endl;
        cout<<"3 : Soustraire 4"<<endl;
        cout<<"4 : Quitter"<<endl;
        cout<<"Votre choix : "; cin>>choix;

        switch(choix)
        {
            case 1 : x++; break;
            case 2 : x=x*2; break;
            case 3 : x=x-4; break;
        }
    }while(choix!=4);

    cout<<"La valeur finale de x vaut : "<<x<<endl;

    return 0;
}

```

## Exercice 11.

Cet exercice a pour but de vérifier les points techniques suivants :

- Utilisation d'un while de difficulté moyenne.
- Etude d'un algorithme usuel : calcul d'une moyenne.

Voici le fichier source :

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

int main()
{
    int x, s=0,nb=0;
    double moyenne;

    do{
        cout<<"Tapez un entier :";cin>>x;
        if(x>0){s=s+x;nb++;}
        else if(x<0)cout<<"ERREUR ";

        }while(x!=0);

    if(nb==0)cout<<"AUCUN ENTIER TAPE ";<<endl<<"PAS DE MOYENNE";<<endl;
    else {
        moyenne=(double) s/nb;
        cout<<"La moyenne vaut : ";<<moyenne<<endl;
    }

    return 0;
}
```

## Exercice 12.

Cet exercice a pour but de vérifier les points techniques suivants :

- Utilisation du for.
- Etude d'un algorithme usuel : calcul d'une factorielle.
- Modélisation d'un problème issu des mathématiques.

Voici le fichier source :

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

int main()
{
    int N,i,f=1;

    cout<<"Tapez un entier : ";cin>>N;
    for(i=2;i<=N;i++) f=f*i;
    cout<<N<<"! vaut ";<<f<<endl;

    return 0;
}
```

## Exercice 13.

Cet exercice a pour but de vérifier les points techniques suivants :

- Utilisation des boucles for.
- Imbrication de boucles assez complexe.

Voici le fichier source :

```

#include <iostream>

using namespace std;

int main()
{
    int N=0;
    cout<<"Saisissez une valeur pour N: ";
    cin>>N;
    cout<<endl<<endl;
    cout<<"N= "<<n<<endl;
    for(int i=0;i<N;i++)
    {
        for(int j=0;j<(N-i);j++)
        {
            cout<<"*";
        }
        cout<<endl;
    }
    cout << "Appuyez sur une touche pour continuer ..." << endl;
    cin.ignore();
    cin.get();
    return 0;
}

```

## Exercice 14.

Cet exercice a pour but de vérifier les points techniques suivants :

- Utilisation des boucles for.
- Imbrication de boucles.

Voici le fichier source :

```

#include<iostream>
using namespace std;

int main()
{
    int i,j,N;

    cout<<"Tapez la valeur de N : ";cin>>N;

    for(i=1;i<=N;i++)
    {
        for(j=1;j<i;j++)cout<<" ";
        for(j=1;j<=N+1-i;j++)cout<<"*";
        cout<<endl;
    }
    return 0;
}

```

## Les tableaux

## Exercice 15.



Cet exercice a pour but de vérifier les points techniques suivants :

- Utilisation simple de tableaux.
- Un algorithme simple sur un tableau : recherche du nombre d'éléments vérifiant une propriété.

Voici le fichier source :

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

const int N=10;

int main()
{
    int t[10],i,nb=0;
    for(i=0;i<N;i++){cout<<"Tapez un entier ";cin>>t[i];}
    for(i=0;i<N;i++)if(t[i]>=10)nb++;
    cout<<"Le nombre d'entiers supérieurs ou égaux à 10 est : "
        <<nb<<endl;
    return 0;
}
```

## Exercice 16.

Cet exercice a pour but de vérifier les points techniques suivants :

- Utilisation simple de tableaux.
- Un algorithme simple sur un tableau : recherche d'un élément dans un tableau.

Voici le fichier source :

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

const int N=10;

int main()
{
    int t[N],i,V;
    bool trouve;
    for(i=0;i<N;i++){cout<<"Tapez un entier ";cin>>t[i];}
    cout<<"Tapez la valeur de V : ";cin>>V;

    trouve=false;
    i=0;
    while(!trouve && i<N)
        if(t[i]==V)trouve=true; else i++;
    if(trouve) cout<<"La valeur V se trouve dans le tableau"<<endl;
    else cout<<"La valeur V ne se trouve pas dans le tableau"<<endl;
    return 0;
}
```

## Exercice 17.

Cet exercice a pour but de vérifier les points techniques suivants :

- Utilisation simple de tableaux.
- Un algorithme simple sur un tableau : recherche de l'indice du plus grand élément.

Voici le fichier source :

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

```

const int N=10;
int main()
{
int t[N],i,indice;

for(i=0;i<N;i++){cout<<"Tapez un entier ";cin>>t[i];}
indice=0;
for(i=1;i<N;i++)
    if(t[indice]<t[i])indice=i;

cout<<"L'indice du plus grand élément est : "<<indice<<endl;
return 0;
}

```

## Exercice 18.

Cet exercice a pour but de vérifier les points techniques suivants :

- Utilisation simple de tableaux.
- Un algorithme simple sur un tableau : suppression d'un élément avec décalage des suivants.

Voici le fichier source :

```

#include<iostream>
using namespace std;

const int N=10;

int main()
{
int t[N],i,j,V;
bool trouve;
for(i=0;i<N;i++){cout<<"Tapez un entier ";cin>>t[i];}
cout<<"Tapez la valeur de V : ";cin>>V;

trouve=false;
i=0;
while(!trouve && i<N)
    if(t[i]==V)trouve=true; else i++;

if(trouve)
{
    for(j=i;j<N-1;j++)t[j]=t[j+1];
    t[N-1]=0;
}
for(i=0;i<N;i++)cout<<t[i]<<endl;

return 0;
}
}

```

## Exercice 19.

Cet exercice a pour but de vérifier les points techniques suivants :

- Utilisation simple de tableaux.
- Un algorithme simple sur un tableau : tri d'un tableau.

Voici le fichier source :

```

#include<iostream>
using namespace std;

const int N=10;

int main()
{
    int a[N],i,j,min,imin,tmp;

    for(i=0;i<N;i++)
    {
        cout<<"Veuillez taper l'entier numero "<<i<<" : ";cin>>a[i];
    }

    for(i=0;i<N-1;i++)
    {
        imin=i;min=a[i];
        for(j=i+1;j<N;j++) if(a[j]<min){min=a[j];imin=j;}

        tmp=a[imin];a[imin]=a[i];a[i]=tmp;
    }
    cout<<"VOICI LE TABLEAU TRIE : "<<endl;
    for(i=0;i<N;i++) cout<<"a["<<i<<"]="<<a[i]<<endl;

    return 0;
}

```

## Exercice 20.

Cet exercice a pour but de vérifier les points techniques suivants :

- Utilisation simple de tableaux.
- Un algorithme simple sur un tableau : tri d'un tableau.

Voici le fichier source :

```

#include<iostream>
using namespace std;

const int N=10;

int main()
{
    int a[N],i,nb,tmp;

    for(i=0;i<N;i++)
    {cout<<"Veuillez taper l'entier numero "<<i<<" : ";cin>>a[i];}

    do
    {
        nb=0;
        for(i=0;i<N-1;i++)
            if(a[i]>a[i+1])
            {
                tmp=a[i];a[i]=a[i+1];a[i+1]=tmp;
                nb++;
            }
    }while(nb!=0);

    cout<<"VOICI LE TABLEAU TRIE : "<<endl;
    for(i=0;i<N;i++) cout<<"a["<<i<<"]="<<a[i]<<endl;
}

```

```
return 0;
}
```

## Exercice 21.

Cet exercice a pour but de vérifier les points techniques suivants :

- Utilisation simple de tableaux.
- Gestion d'une liste simple grâce à un tableau statique.

Voici le fichier source :

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

const int N=10;
int main()
{
    int t[N],nb=0,choix,e,i;
    bool fini=false;

    while(fini==false)
    {
        cout<<"1. Ajouter un entier"<<endl;
        cout<<"2. Afficher la liste des entiers"<<endl;
        cout<<"3. Supprimer le dernier entier de la liste"<<endl;
        cout<<"4. Afficher le dernier entier de la liste"<<endl;
        cout<<"5. Quitter"<<endl;
        cout<<"Votre choix : ";cin>>choix;
        switch(choix)
        {
            case 1 : cout<<"Tapez un entier : ";cin>>e;
                     if(nb<N){t[nb]=e; nb++; cout<<"ENTIER
AJOUTE"<<endl;}
                     else cout<<"IMPOSSIBLE LE TABLEAU EST
PLEIN"<<endl;
                     break;

            case 2 : if(nb==0)cout<<"LA LISTE EST VIDE"<<endl;
                     else {
LISTE"<<endl;
for(i=0;i<nb;i++)cout<<t[i]<<" ";
cout<<endl;
                     }
                     break;

            case 3 : if(nb>0){nb--; cout<<"ENTIER SUPPRIME"<<endl;}
                     else cout<<"LA LISTE EST
VIDE"<<endl;
                     break;

            case 4 : if(nb>0)cout<<"LE DERNIER ENTIER EST "<<t[nb-
1]<<endl;
                     else cout<<"LA LISTE EST
VIDE"<<endl;
                     break;

            case 5 : fini=true;

```

```

        break;
    }

    }
return 0;
}

```

## Exercice 22.

Cet exercice a pour but de vérifier les points techniques suivants :

- Utilisation simple de tableaux.
- Gestion d'une liste triée grâce à un tableau statique.

Voici le fichier source :

```

#include<iostream>
using namespace std;

const int N=10;
int main()
{

int t[N],nb=0,choix,e,V,i,j,trouve;
bool fini=false;

while(fini==false)
    {
        cout<<"1. Ajouter un entier"<<endl;
        cout<<"2. Afficher la liste des entier"<<endl;
        cout<<"3. Supprimer le premier entier ayant une valeur
donnée"<<endl;
        cout<<"4. Supprimer tous les entiers ayant une valeur
donnée"<<endl;
        cout<<"5. Quitter"<<endl;
        cout<<"Votre choix : ";cin>>choix;
        switch(choix)
            {
                case 1 : if(nb<N)
                    {
                        cout<<"Tapez un entier : ";cin>>e;
                        i=0;
                        while(i!=nb && t[i]<e)i++;
                        for(j=nb;j>i;j--)t[j]=t[j-1];
                        t[i]=e;
                        nb++;
                    }
                else cout<<"IMPOSSIBLE LE TABLEAU EST
PLEIN"<<endl;
                break;

                case 2 : if(nb==0)cout<<"LA LISTE EST VIDE"<<endl;
                    else {
                        cout<<"VOICI LA
LISTE"<<endl;
                        for(i=0;i<nb;i++)cout<<t[i]<<" ";

                        cout<<endl;
                    }
                break;
            }
    }
}

```

```

        case 3 : cout<<"Tapez la valeur à supprimer :";cin>>V;
                  trouve=false;
                  i=0;
                  while(!trouve &&
i<nb)if(t[i]==V)trouve=true; else i++;
                  if(trouve)
                  {
                      for(j=i;j<nb-1;j++)t[j]=t[j+1];
                      nb--;
                  }

        break;

        case 4 : cout<<"Tapez la valeur à supprimer :";cin>>V;
                  j=0;
                  for(i=0;i<nb;i++)
                      if(t[i]!=V){t[j]=t[i];j++;}
                  nb=j;

        break;

        case 5 : fini=true;
        break;
    }

    return 0;
}

```

## Les fonctions

### Exercice 23.

Cet exercice a pour but de vérifier les points techniques suivants :

- Création de fonction simple.
- Passage de paramètres par valeur.
- Utilisation de return.
- Appel d'une fonction.

Voici le fichier source :

```

#include<iostream>
using namespace std;
#include<cmath>

double distance(double xa, double ya, double xb, double yb)
{
    double dx,dy;
    dx=xa-xb;
    dy=ya-yb;
    return sqrt(dx*dx+dy*dy);
}

int main()
{
    double x1,y1,x2,y2,d;

    cout<<"Tapez l'abscisse de A : ";cin>>x1;

```

```

cout<<"Tapez l'ordonnée de A : ";cin>>y1;
cout<<"Tapez l'abscisse de B : ";cin>>x2;
cout<<"Tapez l'ordonnée de B : ";cin>>y2;

d=distance(x1,y1,x2,y2);

cout<<"La distance AB vaut : "<<d<<endl;
return 0;
}

```

## Exercice 24.

Cet exercice a pour but de vérifier les points techniques suivants :

- Création de fonction simple.
- Appel d'une fonction.
- Validation des données avant l'appel d'une fonction.
- Fonction renvoyant un booléen.

Voici le fichier source :

```

#include<iostream>
using namespace std;
#include<cmath>

bool f(int x)
{
    bool r=true;
    int d=2;
    while(r && d*d<=x) if(x%d==0)r=false; else d++;
    return r;
}

int main()
{
    int x;
    bool premier;
    do{
        cout<<"Tapez x :";cin>>x;
    }while(x<=0);
    premier=f(x);
    if(premier)cout<<"x est premier"<<endl;
        else cout<<"x n'est pas premier"<<endl;
    return 0;
}

```

## Exercice 25.

Cet exercice a pour but de vérifier les points techniques suivants :

- Création de fonctions simples.
- Appel de fonction.
- Fonction qui appelle une autre fonction.

Voici le fichier source :

```

#include<iostream>
using namespace std;

```

```

bool premier(int x)
{
    bool r=true;
    int d=2;
    while(r && d*d<=x) if(x%d==0) r=false; else d++;
    return r;
}

int Npremier(int N)
{
    int nb=0;
    int i=2;

    while(nb!=N)
    {
        if(premier(i)) nb++;
        i++;
    }
    return i-1;
}

int main()
{
    int N,p;
    cout<<"Tapez la valeur de N : ";cin>>N;
    p=Npremier(N);
    cout<<"Le N-ième nombre premier est : "<<p<<endl;
    return 0;
}

```

## Exercice 26.

Voir solution donnée dans le cours

## Exercice 27.

Cet exercice a pour but de vérifier les points techniques suivants :

- Ecriture d'une fonction ayant comme paramètre un tableau de taille quelconque.
- Recherche d'un élément dans un tableau.
- Utilisation de return.

Voici le fichier source :

```

#include<iostream>
using namespace std;

void saisir(int t[],int n)
{
    int i; for(i=0;i<n;i++)
    {
        cout<<"Tapez la valeur numero "<<i<<" : ";
        cin>> t[i];
    }
}

int f(int t[], int n)
{
    int i=0,ind=-1;

```



```
while(ind==-1 && i<n)
if(t[i]>=0 && t[i]<=10)ind=i;else i++;
return ind;
}

int main()
{
int a[10];
int w;

saisir(a,10);
w=f(a,10);
if(w!=-1)cout<<"Il existe une valeur entre 0 et 10. "
<<"l'indice de la première case est "<<w<<endl;
else cout<<"Il n'existe pas de valeurs entre 0 et 10"<<endl;
return 0;
}
```

## Exercice 28.

Voir solution donnée dans le cours