

Réalisé par : LAMSADDEK DOUNIA

## Devoir Libre n° 1 - Langage C++

### Exercice1:

✚ Le fichier « Hotel.h »

```
1  #include<string>
2  #include <iostream>
3
4  using namespace std;
5
6  class Hotel{
7
8  private:
9      std::string nom;
10     int nbrChambre;
11     double prixChambre;
12
13 public:
14
15     Hotel( int nbrChambre, double prixChambre );
16     Hotel( int nbrChambre, double prixChambre, std::string nom);
17
18     void SetNomHotel(std::string nom);
19     void SetPrixChambre(double prixChambre);
20
21     std::string getNomHotel() const;
22     int getNbrChambre() const;
23     double getPrixChambre() const;
24
25     double ChiffreAffaires(int nbrChambreOccupe) const;
26     friend ostream &operator<<(ostream & sortie, const Hotel & hotel);
27
28 };
29
```

## ✚ Le fichier « Hotel.cpp »

```
1  #include "Hotel.h"
2  #include<string>
3  #include <iostream>
4  using namespace std;
5
6  Hotel::Hotel(int nbrChambre, double prixChambre){
7      this->prixChambre=prixChambre;
8      this->nbrChambre=nbrChambre;
9      this->nom="undefined";
10 }
11
12 Hotel::Hotel(int nbrChambre, double prixChambre, std::string nom){
13     this->prixChambre=prixChambre;
14     this->nbrChambre=nbrChambre;
15     this->nom=nom;
16 }
17
18 void Hotel::SetNomHotel(std::string nom){
19     this->nom=nom;
20 }
21
22 void Hotel::SetPrixChambre(double prixChambre){
23     this->prixChambre=prixChambre;
24 }
25
26 std::string Hotel::getNomHotel() const{
27     return this->nom;
28 }
29
30 double Hotel::getPrixChambre() const{
31     return this->prixChambre;
32 }
33
34 double Hotel::ChiffreAffaires(int nbrChambreOccupe) const{
35     return nbrChambreOccupe* this->prixChambre;
36 }
37
38 ostream & operator << (ostream & sortie, const Hotel & hotel){
39     sortie<<" Le nom de l'hotel est : " <<hotel.nom<<endl;
40     sortie<<"Le nombre de chambre de l'hotel est : "<<hotel.nbrChambre<<endl;
41     sortie<<"Le prix des chambres est : "<<hotel.prixChambre<<"Dhs";
42     return sortie;
43 }
44
```

## Le fichier « main.cpp »

```
1  #include <iostream>
2  #include "Hotel.h"
3
4  using namespace std;
5
6  int main()
7  {
8      Hotel hotel_1(2, 13.4);
9      Hotel hotel_2(2, 16.7, "panoramique");
10
11     cout<<hotel_1.getNbrChambre()<<endl;
12     cout<<hotel_2.getNomHotel()<<endl;
13
14     cout<<hotel_1.ChiffreAffaires(20)<<endl;
15
16     cout<<hotel_2<<endl;
17     cout<<hotel_1<<endl;
18
19     return 0;
20 }
21
```

## Son exécution :

```
2
panoramique
268
Le nom de l'hotel est : panoramique
Le nombre de chambre de l'hotel est : 2
Le prix des chambres est :16.7Dhs
Le nom de l'hotel est : undefined
Le nombre de chambre de l'hotel est : 2
Le prix des chambres est :13.4Dhs

Process returned 0 (0x0)   execution time : 0.166 s
Press any key to continue.
```

## Exercice2:

🚦 Le fichier « Hotel.h »

```

1  #include<string>
2  using namespace std;
3  #include <iostream>
4  class Hotel{
5  private:
6      std::string nom;
7      int nbrChambre;
8      double prixChambre;
9      int * Tableau;
10 public:
11     Hotel( int nbrChambre,double prixChambre,int * Tableau );
12     Hotel( int nbrChambre,double prixChambre,int * Tableau,std::string nom);
13     Hotel(const Hotel &HotelCopie);// constructeur de copie:
14
15     void SetNomHotel(std::string nom);
16     void SetPrixCahmmbre(double prixChambre);
17
18     std::string getNomHotel() const;
19     int getNbrChambre() const;
20     double getPrixChambre() const;
21     int* getTableau()const;
22
23     ~Hotel();// Le destructeur
24     Hotel& operator=(const Hotel &Hotel);// L'opérateur d'affectation:
25     double ChiffreAffaires(int nbrChambreOccupe) const;
26
27     friend ostream &operator<<(ostream & sortie, const Hotel & hotel);
28     int operator[](int i) const ;
29     int & operator[](int i) ;
30 };

```

📁 Le fichier « Hotel.cpp »

```

1  #include "Hotel.h"
2  #include<iostream>
3  using namespace std;
4
5  Hotel::Hotel( int nbrChambre, double prixChambre, int * Tableau, std::string nom){
6      this->nom = nom;
7      this->nbrChambre = nbrChambre;
8      this->prixChambre = prixChambre;
9      this->Tableau = new int[nbrChambre];
10     for(int i=0; i<nbrChambre; i++){
11         this->Tableau[i]=Tableau[i];
12     }
13 }
14 Hotel::Hotel(int nbrChambre, double prixChambre, int * Tableau){
15     this->nom = "undefined";
16     this->nbrChambre=nbrChambre;
17     this->prixChambre=prixChambre;
18     this->Tableau = new int[nbrChambre];
19     for(int i=0; i<nbrChambre; i++){
20         this->Tableau[i]=Tableau[i];
21     }
22 }
23 Hotel::Hotel(const Hotel &hotel){
24     this->nom = hotel.nom;
25     this->nbrChambre = hotel.nbrChambre;
26     this->prixChambre= hotel.prixChambre;
27     this->Tableau = new int[nbrChambre];
28     for(int i=0; i<nbrChambre; i++){
29         Tableau[i]=hotel.Tableau[i];
30     }
31
32
33
34     std::string Hotel::getNomHotel() const{
35         return this->nom;
36     }
37     int Hotel::getNbrChambre() const{
38         return this->nbrChambre;
39     }
40     double Hotel::getPrixChambre() const{
41         return this->prixChambre;
42     }
43     int* Hotel::getTableau() const{
44         return this->Tableau;
45     }
46
47
48     void Hotel::SetNomHotel(std::string nom){
49         this->nom = nom;
50     }
51
52     void Hotel::SetPrixCahmmbre(double prix){
53         this->prixChambre= prixChambre;
54     }
55
56
57     double Hotel::ChiffreAffaires(int nbrChambreOccupe) const{
58         return this->prixChambre * nbrChambreOccupe;
59     }
60
61
62

```

```

63 ostream & operator << (ostream & sortie, const Hotel & hotel){
64     sortie<<" Le nom de l'hotel est : " <<hotel.nom<<endl;
65     sortie<<"Le nombre de chambre de l'hotel est : " <<hotel.nbrChambre<<endl;
66     sortie<<"Le prix des chambres est : "<<hotel.prixChambre<<"Dhs"<<endl;
67     for(int i = 0 ; i < hotel.nbrChambre ; i++){
68         sortie<< "La chambre numero : "<<i+1<<" Contient : "<<hotel.Tableau[i]<<" occupants.";
69         sortie<<endl;
70     }
71     return sortie;
72 }
73
74
75 Hotel::~Hotel(){
76     delete Tableau;
77 }
78
79
80 Hotel& Hotel::operator=(const Hotel & hotel){
81     if(this != &hotel){
82         this->nom = hotel.nom;
83         this->prixChambre=hotel.prixChambre;
84         this->nbrChambre=hotel.nbrChambre;
85         delete [] this->Tableau;
86         Tableau=new int[nbrChambre];
87         for(int i=0 ; i<this->nbrChambre ; i++){
88             Tableau[i] = hotel.Tableau[i];
89         }
90     }
91     return (*this);
92 }
93
94
95 int Hotel::operator[] (int i)const{
96     if( i<1 | i>this->nbrChambre){
97         cout<<"indeice intouvable";
98     }
99     else{
100         return this->Tableau[i-1];
101     }
102 }
103 int& Hotel::operator[](int i){
104     if( i<1 | i>this->nbrChambre){
105         cout<<"indeice intouvable";
106     }
107     else{
108         return this->Tableau[i-1];
109     }
110 }
111

```

## Le fichier « main.cpp »

```
1  #include <iostream>
2  #include "Hotel.h"
3  using namespace std;
4  int main()
5  {
6      int *t;
7      t = new int[4];
8      t[0]=12;
9      t[1]=13;
10     t[2]=3;
11     t[3]=5;
12
13     Hotel h1(4,15.6,t);
14     Hotel h2(2,16.7,t,"panoramique");
15     Hotel h3(2,111.6,t);
16
17     cout<<h1.getTableau()[0]<<endl;
18     cout<<h1<<endl;
19     cout<<h2<<endl;
20     cout<<h3<<endl;
21     h1=h2;
22     cout<<h1<<endl;
23     cout<<h2<<endl;
24
25     cout<<h1[9]<<endl;
26     cout<<h2[1]<<endl;
27     return 0;
28 }
29
```



## Son exécution :

```
12
  Le nom de l'hotel est : undefined
Le nombre de chambre de l'hotel est : 4
Le prix des chambres est :15.6Dhs
La chambre numero : 1 Contient : 12 occupants.
La chambre numero : 2 Contient : 13 occupants.
La chambre numero : 3 Contient : 3 occupants.
La chambre numero : 4 Contient : 5 occupants.

  Le nom de l'hotel est : panoramique
Le nombre de chambre de l'hotel est : 2
Le prix des chambres est :16.7Dhs
La chambre numero : 1 Contient : 12 occupants.
La chambre numero : 2 Contient : 13 occupants.

  Le nom de l'hotel est : undefined
Le nombre de chambre de l'hotel est : 2
Le prix des chambres est :111.6Dhs
La chambre numero : 1 Contient : 12 occupants.
La chambre numero : 2 Contient : 13 occupants.

  Le nom de l'hotel est : panoramique
Le nombre de chambre de l'hotel est : 2
Le prix des chambres est :16.7Dhs
La chambre numero : 1 Contient : 12 occupants.
La chambre numero : 2 Contient : 13 occupants.

  Le nom de l'hotel est : panoramique
Le nombre de chambre de l'hotel est : 2
Le prix des chambres est :16.7Dhs
La chambre numero : 1 Contient : 12 occupants.
La chambre numero : 2 Contient : 13 occupants.

indeice intouvable4992084
12

Process returned 0 (0x0)   execution time : 0.219 s
Press any key to continue.
```

# Devoir Libre n° 2 - Langage C++

## Exercice1:

✚ Le fichier « Vect.h »

```
1  #include <string>
2  using namespace std;
3  typedef double Real;
4  #include <iostream>
5  class Vect {
6  public :
7      int n ;
8      Real *v ;
9
10     Vect() ;
11     Vect(int) ;
12     Vect(const Vect&) ;
13     Vect& operator=(const Vect &) ;
14     ~Vect() ;
15
16     Real & operator[](int) const ;
17     Vect operator+(const Vect &) const ;
18     Vect operator*(const Real &) const ;
19
20     void init_v(double nbre){
21         for (int i=0; i < n; i++)
22             v[i] = nbre;
23     }
24     friend istream & operator>>(istream &, Vect&) ;
25     friend ostream & operator<<(ostream&, const Vect&) ;
26
27 private :
28     Vect(int , Real*) ;
29     friend Vect operator*(const Real&, const Vect&) ;
30 };
```

DS1-2\_c+

## Le fichier « Vect.cpp »

```
1  #include "Vect.h"
2  #include <iostream>
3  #include <ostream>
4  using namespace std;
5  Vect::Vect(){
6      n=0 ;
7      v=NULL;
8  }
9  Vect::Vect(int n){
10     this->n=n;
11     this->v = new Real[n];
12     for(int j=0;j<n;j++)
13         v[j]=0;
14 }
15 Vect::Vect(const Vect & vect){
16     this->n=vect.n;
17     this->v=new Real[n];
18     for(int i=0;i<n;i++){
19         v[i]=vect.v[i];
20     }
21 }
22 Vect& Vect::operator=(const Vect & vect){
23     this->n = vect.n;
24     delete [] this->v;
25     this->v=new Real[n];
26     for(int i=0 ; i < vect.n ; i++){
27         this->v[i] = vect.v[i];
28     }
29     return *this;
30 }
```

```

30 }
31 }
32 Vect::~Vect(){
33     delete[] v;
34 }
35 Real & Vect::operator[](int i) const{
36     if(i>=this->n || i<0){
37         cout<<"Indice introuvable "<<endl;
38     }
39     else{
40         return this->v[i];
41     }
42 }
43 Vect::Vect(int n , Real * p){
44     this->n=n;
45     v=p;
46 }
47 Vect Vect::operator+(const Vect & vect) const {
48     if(vect.n==this->n){
49         Real * p=new Real[vect.n];
50         Vect SommeVecteurs(vect.n,p);
51         for(int i=0 ; i < vect.n ; i++){
52             SommeVecteurs.v[i] = this->v[i] + vect.v[i];
53         }
54         return SommeVecteurs;
55     }
56     else{
57         cout<<"impossible de les additionnées"<<endl;
58     }
59 }

```

```

63 Vect Vect::operator*(const Real & scalaire) const{
64     Vect v1(scalaire*(this));
65     return v1;
66 }
67 std::istream & operator>>(std::istream & sortie ,Vect& vecteur)
68 {
69     cout<<"entrer le nombre d'elements de vecteur : "<<endl;
70     sortie>>vecteur.n;
71     vecteur.v = new double[vecteur.n];
72     for(int i = 0 ; i < vecteur.n ; i++){cout<<"entrer la valeur de l'element "<<i+1<<" : "<<endl;
73         sortie>>vecteur.v[i];}
74     return sortie;
75 }
76 std::ostream & operator<<(std::ostream& sortie , const Vect& vecteur)
77 {
78     sortie << "(" ;
79     for(int i = 0 ; i < vecteur.n-1 ; i++){
80         sortie<< vecteur.v[i];
81         sortie<< " ,";
82     }
83     sortie<< vecteur.v[vecteur.n-1];
84     sortie<< " )";
85     return sortie;
86 }
87 Vect operator*(const Real& scalaire,const Vect& vecteur ){
88     Real * p=new Real[vecteur.n];
89     Vect ProduitScalair(vecteur.n,p);
90     for(int i=0 ; i < vecteur.n ; i++){ProduitScalair.v[i] = vecteur.v[i] * scalaire;}
91     return ProduitScalair;
92 }

```

## Le fichier « main.cpp »

```
1  #include <iostream>
2  #include "Vect.h"
3  typedef double Real;
4  using namespace std;
5  int main ()
6  {
7
8      Vect v1;
9      Vect v2(3);
10     Vect v3(v2);
11
12     cout<<"Le vecteur v1: "<<endl;
13     cin>>v1;
14     cout<<"Le vecteur v1:"<<v1<<endl;
15     cout<<"Le vecteur v3:"<<v3<<endl;
16     v3.init_v(3);
17     cout<<"Le vecteur v3 apres l'initialisation:"<<v3<<endl;
18     cout<<"Le vecteur v2:"<<v2<<endl;
19
20     v2=v1;
21     cout<<"Apres l'operation de l'affectation v2=v1 , Le vecteur v2 egal : "<<"v2="<<v2<<endl;
22     cout<<"la valeur de la 2 eme case c ad l'indice 0 du vecteur v2 est : "<<v2[1]<<endl;
23
24     cout<<"La somme des vecteurs v1 et v2 est "<<v1+v2<<endl;
25
26     cout<<"Le produit d'un vecteur v1 et un scalaire qui egal a 5 est egal:"<<v1*5<<endl;
27
28     return 0;
29 }
30
```

## Son execution :

```
Le vecteur v1:
entrer le nombre d'elements de vecteur :
3
entrer la valeur de l'element 1 :
2
entrer la valeur de l'element 2 :
3
entrer la valeur de l'element 3 :
4
Le vecteur v1:( 2 ,3 ,4 )
Le vecteur v3:( 0 ,0 ,0 )
Le vecteur v3 apres l'initialisation:( 3 ,3 ,3 )
Le vecteur v2:( 0 ,0 ,0 )
Apres l'operation de l'affectation v2=v1 , Le vecteur v2 egal : v2=( 2 ,3 ,4 )
la valeur de la 2 eme case c ad l'indice 0 du vecteur v2 est :3
La somme des vecteurs v1 et v2 est ( 4 ,6 ,8 )
Le produit d'un vecteur v1 et un scalaire qui egal a 5 est egal:( 10 ,15 ,20 )

Process returned 0 (0x0) execution time : 6.426 s
Press any key to continue.
```

## Exercice2:

### Le fichier «vecteur.h »

```
1  using namespace std;
2  class matrice;
3
4  class vecteur{
5  public:
6
7      int nbelem;
8      int *adv;
9
10     vecteur(int nbelem);
11     ~vecteur() ;
12     void saisir();
13     void affiche()const;
14
15     friend void produit(matrice &, vecteur &);
16
17 };
18
19
```

### Le fichier « vecteur.cpp »

```

1  #include<iostream>
2  #include "vecteur.h"
3  using namespace std;
4  vecteur::vecteur(int n){
5      nbelem=n;
6      adv = new int[nbelem];
7  }
8  vecteur::~vecteur(){
9      delete[] adv;
10 }
11 void vecteur::saisir(){
12     for(int i = 0 ; i < nbelem ; i++){
13         cout<<"entrer la valeur du "<<i+1<<" element: "<<endl;
14         cin>>adv[i];
15     }
16 }
17 void vecteur::affiche()const{
18     cout<<"Affichage du vecteur:"<<endl;
19     cout<<"(";
20     for(int i = 0 ; i < nbelem-1 ; i++){
21         cout << adv[i];
22         cout << ",";
23     }
24     cout << adv[nbelem-1];
25     cout << ")";
26 }
27

```

#### Le fichier « matrice.h »

```

1  using namespace std;
2
3  class vecteur;
4
5  class matrice{
6  public:
7      int nblig;
8      int nbcol;
9      int **adm;
10     matrice(int,int);
11     ~matrice();
12     void saisir();
13     void affiche()const;
14     friend void produit(matrice &, vecteur &);
15 };
16
17

```

## Le fichier « matrice.cpp »

```
1  #include "matrice.h"
2  #include "vecteur.h"
3  #include<iostream>
4  using namespace std;
5  matrice::matrice(int ligne,int colone){
6      nbcol = colone;
7      nblig = ligne;
8      adm = new int*[ligne];
9      for(int i = 0 ; i < ligne ; i++){
10         adm[i] = new int[colone];
11     }
12 }
13 matrice::~matrice(){
14     for(int i = 0 ; i < nblig ; i++){
15         delete[] adm[i];
16     }
17     delete[] adm;
18 }
19 void matrice::saisir(){
20     for(int i = 0 ; i < nblig; i++){
21         for(int j = 0 ; j < nbcol ; j++){
22             cout<<"entrer la valeur de ["<<i<<","<<j<<"] : ";
23             cin>>adm[i][j];
24         }
25     }
26 }
27
28 void matrice::affiche()const{
29     cout<<"Affichage de la matrice : "<<endl;
30     for(int i = 0 ; i < nblig; i++){
31         cout<< "[";
32         for(int j = 0 ; j < nbcol-1 ; j++){
33             cout<<adm[i][j]<<",";
34         }
35         cout<<adm[i][nbcol-1];
36         cout<< "]"<<endl;
37     }
38 }
39 void produit(matrice &mat, vecteur &vect){
40     if(mat.nbcol == vect.nbelem){
41         int n = mat.nblig;
42         vecteur prod(n);
43         for(int i = 0; i < n; i++){
44             prod.adv[i]=0;
45         }
46         for(int i = 0; i < n; i++) {
47             for(int j = 0; j < vect.nbelem; j++){
48                 prod.adv[i] += mat.adm[i][j] * vect.adv[j];
49             }
50         }
51         cout<<"Le Produit egal: "<<endl;
52         prod.affiche();
53     }else {
54         cout<<"impossible de calculer le produit";
55     }
56 }
57 }
```



## Le fichier « main.cpp »

```
1  #include <iostream>
2  #include "vecteur.h"
3  #include "matrice.h"
4
5  using namespace std;
6
7  int main()
8  {
9      vecteur v1(2);
10     v1.saisir();
11     v1.affiche();
12     matrice m1(2,2);
13     m1.saisir();
14     m1.affiche();
15     produit(m1,v1);
16
17     return 0;
18 }
19
```

## Son execution :

```
entrer la valeur du 1 element:
1
entrer la valeur du 2 element:
1
Affichage du vecteur:
( 1 ,1 )
entrer la valeur de [0,0] : 1
entrer la valeur de [0,1] : 1
entrer la valeur de [1,0] : 1
entrer la valeur de [1,1] : 1
Affichage de la matrice :
[1,1]
[1,1]
Le Produit egal:
Affichage du vecteur:
( 2 ,2 )
Process returned 0 (0x0)   execution time : 5.947 s
Press any key to continue.
```