

## Travaux pratiques de la Programmation Orientée Objet en C++

### Solutions des exercices du TP N° 03: Tableaux statiques et dynamiques

#### Exercice 01 : compétences visées --> maîtrise les tableaux unidimensionnels statiques

Ecrire un programme permet de chercher et localiser (par indice) le maximum et le minimum d'un tableau entré au clavier (**exiger de l'utilisateur de déclarer un tableau <=100**). Si le tableau contient plusieurs maximums/minimums, le programme doit garder l'indice du premier maximum/minimum rencontré.

#### Solution

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      float Tab[100]; //déclaration de deux tableaux
6      unsigned int N(0);
7      do
8      {
9          cout<<"Entrer la taille des tableaux (max 100)"<<endl;
10         cin >> N;
11     }
12     while (N<=0 || N>100);
13     // Saisie des composantes du tableau
14     for(int i=1;i<N;i++)
15     {
16         cout << "Entrer la valeur de TabA["<<i<<"]=" ";
17         cin >> Tab[i];
18     }
19     int maxTab=Tab[0],minTab=Tab[0],indceMax(0),indceMin(0);
20     //Cherche la valeur maximale et minimale dans le tableau
21     for(int i=1;i<N;i++)
22     {
23         if (Tab[i]>maxTab) //Pour trouver la valeur maximale
24         {
25             maxTab=Tab[i];
26             indceMax=i;
27         }
28         if (Tab[i]<minTab) //Pour trouver la valeur minimale
29         {
30             minTab=Tab[i];
31             indceMin=i;
32         }
33     }
34     //Affichage des résultats
35     cout<<"le maximum du tableau est Tab["<<indceMax<<"]=" << Tab[indceMax]<<endl;
36     cout<<"le minimum du tableau est Tab["<<indceMin<<"]=" << Tab[indceMin]<<endl;
37     return 0;
```

### Exemple d'exécution :

```
Entrer la taille des tableaux (max 100)
150
Entrer la taille des tableaux (max 100)
140
Entrer la taille des tableaux (max 100)
10
Entrer la valeur de TabA[1]= 15
Entrer la valeur de TabA[2]= 20
Entrer la valeur de TabA[3]= 28
Entrer la valeur de TabA[4]= 36
Entrer la valeur de TabA[5]= -45
Entrer la valeur de TabA[6]= 12
Entrer la valeur de TabA[7]= 0
Entrer la valeur de TabA[8]= 12
Entrer la valeur de TabA[9]= -48
le maximum du tableau est Tab[4]= 36
le minimum du tableau est Tab[9]= -48
```

### Exercice 02 : compétences visées --> maîtrise les tableaux unidimensionnels statiques

Ecrire un programme permettant de calculer la somme et le produit scalaire des deux tableaux (de même taille) entrés au clavier TabA, TabB (exiger de l'utilisateur de déclarer un tableau  $\leq 100$ ). Afficher ensuite le tableau  $\text{TabSom} = \text{TabA} + \text{TabB}$ , ainsi que le résultat du produit scalaire (ProSca).

### Solution :

```
1 | #include <iostream>
2 | using namespace std;
3 | int main()
4 | {
5 |     float TabA[100], TabB[100]; //déclaration de deux tableaux
6 |     unsigned int N(0);
7 |     double ProSca(0);
8 |     do
9 |     {
10 |         cout<<"Entrer la taille des tableaux (max 100)"<<endl;
11 |         cin >> N;
12 |     }while (N<=0 || N>100);
13 |
14 |     float TabSom[N]={0}; //Tableau vide
15 |
16 |     //Saisie du tableau TabA
17 |     cout << "Saisie du tableau TabA"<<endl;
18 |     for(int i=0; i<N; i++) // saisie des éléments du tableau TabA
19 |     {
20 |         cout << "Entrer la valeur de TabA["<<i<<"]="= ";
21 |         cin >> TabA[i];
22 |     }
23 |     //Saisie du tableau TabB
24 |     cout << "Saisie du tableau TabB"<<endl;
25 |     for(int i=0; i<N; i++) // saisie des éléments du tableau TabB
26 |     {
27 |         cout << "Entrer la valeur de TabB["<<i<<"]="= ";
28 |         cin >> TabB[i];
29 |     }
30 |     for(int i=0; i<N; i++) // saisie des éléments du tableau TabB
```

```

31 {
32     TabSom[i]=TabA[i]+TabB[i]; // calcul la somme de deux tableaux
33     ProSca+= TabA[i]*TabB[i]; //calcul le produit scalaire de deux tableaux
34 }
35 // Affichage du tableau TabSom
36 cout << "TabSom=[ ";
37 for(int i=0;i<N;i++) // saisie des éléments du tableau TABB
38 {
39     cout<< TabSom[i]<< ' ';
40 }
41 cout << "]"<<endl;
42 // Affichage du résultat de produit scalaire de deux tableaux
43 cout << "le produit scalaire de deux tableaux = "<<ProSca<<endl;
44 return 0;
45 }

```

### Exemple d'exécution :

```

Entrez la taille des tableaux (max 100)
4
Saisie du tableau TabA
Entrez la valeur de TabA[0]= 1
Entrez la valeur de TabA[1]= 2
Entrez la valeur de TabA[2]= 3
Entrez la valeur de TabA[3]= 4
Saisie du tableau TabB
Entrez la valeur de TabB[0]= -1
Entrez la valeur de TabB[1]= -2
Entrez la valeur de TabB[2]= -3
Entrez la valeur de TabB[3]= -4
TabSom=[ 0 0 0 0 ]
le produit scalaire de deux tableaux = -30

```

### Exercice 03 : compétences visées --> maîtrise les tableaux unidimensionnels statiques

Ecrire un programme permet de lire les notes des étudiants en module POO et les mémoriser dans un tableau (exiger que le nombre de notes ne dépasse pas 200). Le programme doit réaliser les tâches suivantes :

- trier le tableau des notes par ordre croissant ;
- Afficher la moyenne des notes ;
- Afficher ( $14 < \text{nombre des notes} \leq 20$ ), ( $10 \leq \text{nombre des notes} \leq 14$ ), ( $\text{nombre des notes} < 10$ ).

### Solution :

```

1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      float TabNotes[200]; //tableau vide
6      unsigned int nbrNote(0), nbrNote14_20(0),
7      nbrNote10_14(0), nbrNoteInf10(0);

```

```

8 | double moyenne_notes(0);
9 | do
10 | {
11 | cout<<"Combien des notes vous aurez saisies (max 200) ?"<<endl;
12 | cin>>nbrNote;
13 | }while (nbrNote<=0 || nbrNote>200);
14 |
15 | // Saisie les notes des étudiants dans le tableau
16 | cout << "Saisie des notes dans le tableau"<<endl;
17 | for(int i=0;i<nbrNote;i++)
18 | {
19 |     cout << "Entrer une note= ";
20 |     cin >> TabNotes[i];
21 |     moyenne_notes+=TabNotes[i]/nbrNote;
22 | }
23 | // classement des Notes par ordre croissant
24 | for( int i=0;i<nbrNote;i++)
25 | {
26 |     for(int j=i+1;j<nbrNote;j++)
27 |     {
28 |         if (TabNotes[j]<TabNotes[i])
29 |         {
30 |             double VarAide=TabNotes[i];
31 |             TabNotes[i]=TabNotes[j];
32 |             TabNotes[j]=VarAide;
33 |         }
34 |     }
35 | }
36 | // Statistique des notes
37 | for( int i=0;i<nbrNote;i++)//
38 | {
39 |     if ((TabNotes[i]>14)&&(TabNotes[i]<=20))
40 |     {
41 |         nbrNote14_20++;
42 |     }
43 |     else if ((TabNotes[i]>=10)&&(TabNotes[i]<=14))
44 |     {
45 |         nbrNote10_14++;
46 |     }
47 |     else if (TabNotes[i]<10)
48 |     {
49 |         nbrNoteInf10++;
50 |     }
51 | }

```

```

52 | //Affichage du tableau trié
53 | cout <<" Tableau trie= [";
54 | for( int i=0;i<nbrNote;i++)
55 | {
56 |     cout << TabNotes[i] <<' ';
57 | }
58 | cout <<"]"<<endl;
59 | //Affichage la statistique des notes
60 | cout << "14 < Nombre des notes   <= 20 = "<< nbrNote14_20 <<endl;
61 | cout << "10 <= Nombre des notes   <= 14 = "<< nbrNote10_14 <<endl;
62 | cout << "Nombre des notes   < 10 = "<< nbrNoteInf10 <<endl;
63 | cout <<"la moyenne de notes= "<< moyenne_notes<<endl;
64 | return 0;
65 | }

```

#### Exemple d'exécution:

```

Combien des notes vous aurez saisies (max 200) ?
300
Combien des notes vous aurez saisies (max 200) ?
10
Saisie des notes dans le tableau
Entrer une note= 0
Entrer une note= 1
Entrer une note= 10
Entrer une note= 12
Entrer une note= 12.5
Entrer une note= 14
Entrer une note= 15
Entrer une note= 18
Entrer une note= 17
Entrer une note= 0.5
Tableau trie= [0 0.5 1 10 12 12.5 14 15 17 18 ]
14 < Nombre des notes   <= 20 = 3
10 <= Nombre des notes   <= 14 = 4
Nombre des notes   < 10 = 3
la moyenne de notes= 10

```

#### Exercice 04 : compétences visées --> maîtrise les tableaux bidimensionnels

Ecrire un programme qui calcule le transposé d'une matrice carrée entrée au clavier (taille maximale =20x20). *Remarque : On exige l'affichage de la matrice entrée et son transposée.*

Solution :

```

1 | #include <iostream>
2 | using namespace std;
3 | int main()
4 | {
5 |     unsigned int L(0);
6 |     float M [20][20]; //matrice carrée vide

```

```

7   do
8   {
9   cout<<"Entrer la dimension de la matrice (max 20 ) "<<endl;
10  cin >>L;
11  }while (L<=0 || L>20);|
12
13  double VarAide(0);// Variable
14  //Saisie de la matrice M
15  for(int i=0;i<L;i++)
16  {
17  for(int j=0;j<L;j++)
18  {
19  cout <<" M["<<i<<"]["<<j<<"]=" ? " ;
20  cin >> M[i][j];
21  }
22  }
23  //Affichage de la matrice M
24  cout << "Matrice entree M= "<<endl;
25  for( int i=0;i<L;i++)
26  {
27  for(int j=0;j<L;j++)
28  {
29  cout << M[i][j]<<"\t";
30  }
31  cout <<"\n";
32  }
33  //Calcul le transposé de la matrice M
34  for( int i=0;i<L;i++)
35  {
36  for( int j=0;j<L;j++)
37  {
38  if (j>i)
39  {// permutation des elements
40  VarAide=M[i][j];
41  M[i][j]=M[j][i];
42  M[j][i]=VarAide;
43  }
44  }
45  }
46  //Affichage de la matrice transposée
47  cout << "Matrice transposee Mt= "<<endl;
48  for(int i=0;i<L;i++)
49  {
50  for(int j=0;j<L;j++)
51  {
52  cout << M[i][j]<<"\t";

```

```

53     }
54     cout << "\n";
55     }
56     return 0;
57     }

```

**Exemple d'exécution:**

```

Entrez la dimension de la matrice (max 20 )
4
M[0][0]= ? 1
M[0][1]= ? 1
M[0][2]= ? 1
M[0][3]= ? 1
M[1][0]= ? 2
M[1][1]= ? 2
M[1][2]= ? 2
M[1][3]= ? 2
M[2][0]= ? 3
M[2][1]= ? 3
M[2][2]= ? 3
M[2][3]= ? 3
M[3][0]= ? 4
M[3][1]= ? 4
M[3][2]= ? 4
M[3][3]= ? 4
Matrice entree M=
1      1      1      1
2      2      2      2
3      3      3      3
4      4      4      4
Matrice transposee Mt=
1      2      3      4
1      2      3      4
1      2      3      4
1      2      3      4

```

**Exercice 05 : compétences visées --> maîtrise les tableaux bidimensionnels statiques**

Ecrire un programme qui calcul la somme et la multiplication de deux matrices carrée (A et B) entrées au clavier (taille maximale de chaque matrice =20x20). Affiche ensuite les deux matrices (SomMat=A+B et ProMat=A.B).

Solution

```

1   #include <iostream>
2   using namespace std;
3   int main()
4   {
5       unsigned int L(0);
6       float MatA [20][20],MatB[20][20],
7       ProMat[20][20],SomMat[20][20]; //04 matrices carrées vides

```

```

8   do
9   {
10  cout<<"Entrer la dimension de deux matrices (max 20 ) "<<endl;
11  cin >>L;
12  }while (L<=0 || L>20);
13  //Saisie de la première matrice
14  cout << "Saisie de la premiere matrice MatA= "<<endl;
15  for(int i=0;i<L; i++)
16  {
17      for(int j=0;j<L;j++)
18      {
19          cout <<" MatA["<<i<<"] ["<<j<<"]= ? " ;
20          cin >> MatA[i][j];
21      }
22  }
23  //Saisie de la deuxième matrice
24  cout << "Saisie de la deuxieme matrice MatB= "<<endl;
25  for(int i=0;i<L; i++)
26  {
27      for(int j=0;j<L;j++)
28      {
29          cout <<" MatB["<<i<<"] ["<<j<<"]= ? " ;
30          cin >> MatB[i][j];
31      }
32  }
33  //Calcul de la somme et le produit des deux matrices
34  for(int i=0;i<L; i++)
35  {
36      for(int j=0;j<L;j++)
37      {
38          SomMat[i][j]=MatA[i][j]+MatB[i][j];
39          ProMat[i][j]=0;//Initialisation
40          for(int k=0;k<L;k++)
41          {
42              ProMat[i][j]+= MatA[i][k]*MatB[k][j];
43          }
44      }
45  }
46  //Affichage de la matrice SomMat= MatA + MatB
47  cout<<"la matrice SomMat=MatA+MatB= "<<endl;
48  for(int i=0;i<L; i++)
49  {
50      for(int j=0;j<L;j++)
51      {
52          cout << SomMat[i][j]<<"\t";
53      }
54  cout<< endl;

```

```

55     }
56     //Affichage de la matrice ProMat= MatA * MatB
57     cout<<"la matrice ProMat=MatA*MatB= "<<endl;
58     for(int i=0;i<L; i++)
59     {
60         for(int j=0;j<L;j++)
61         {
62             cout << ProMat[i][j]<<"\t";
63         }
64         cout<< endl;
65     }
66
67     return 0;
68 }

```

### Exemple d'exécution

```

Entrez la dimension de deux matrices (max 20 )
3
Saisie de la premiere matrice MatA=
MatA[0][0]= ? 1
MatA[0][1]= ? 1
MatA[0][2]= ? 1
MatA[1][0]= ? 2
MatA[1][1]= ? 2
MatA[1][2]= ? 2
MatA[2][0]= ? 3
MatA[2][1]= ? 3
MatA[2][2]= ? 3
Saisie de la deuxieme matrice MatB=
MatB[0][0]= ? 1
MatB[0][1]= ? 1
MatB[0][2]= ? 1
MatB[1][0]= ? 2
MatB[1][1]= ? 2
MatB[1][2]= ? 2
MatB[2][0]= ? 3
MatB[2][1]= ? 3
MatB[2][2]= ? 3
la matrice SomMat=MatA+MatB=
2      2      2
4      4      4
6      6      6
la matrice ProMat=MatA*MatB=
6      6      6
12     12     12
18     18     18

```

### Exercice 06 : compétences visées --> maîtrise les tableaux dynamiques

Ecrire un programme qui lit un tableau statique Tab (taille maximale =20) et le séparer ensuite en deux tableaux dynamiques, le 1<sup>er</sup> contient les valeurs positifs (TabValPos) et le 2<sup>ème</sup> contient les valeurs négatives (TabValNeg). Afficher ensuite les trois tableaux. *Remarque : On exige que les tailles des tableaux dynamiques et correspondent exactement aux valeurs qu'ils contiennent.*

### Solution

```
1  #include <iostream>
2  #include <vector>//pour utiliser les tableaux dynamiques
3  using namespace std;
4  int main()
5  {
6  unsigned int taille(0);
7  float Tab [20];// Tableau statique vide
8  do
9  {
10 cout<<"Entrer la taille du tableau (max 20 ) "<<endl;
11 cin >>taille;
12 }while (taille<=0 || taille>20);
13
14 vector <double> TabValPos;//Tableau dynamique pour stocker
15 // les valeurs positives
16 vector <double> TabValNeg;// Tableau dynamique pour stocker
17 // les valeurs négatives
18 //saisie les éléments du tableau Tab
19 for(int i=0;i<taille;i++)
20 {
21 cout << " Tab ["<<i<<"]=" ? ";
22 cin>>Tab[i];
23 }
24 //Séparation des valeurs positives, négatives et nulles
25 for(int i=0;i<taille;i++)//saisie des composantes du tableau
26 {
27     if(Tab[i]>0)
28     {
29         TabValPos.push_back(Tab [i]);
30     }
31     else if(Tab[i]<0)
32     {
33         TabValNeg.push_back(Tab [i]);
34     }
35 }
36 // Affichage du tableau entré au clavier 'Tab'
37 cout <<" Tableau entre=[";
38 for(int i=0;i<taille;i++)
39 {
40 cout<< Tab [i]<<' ';
41 }
```

```

42     cout << "]"<<endl;
43     // Affichage du tableau dynamique TabValPos
44     cout<<" Tableau dynamique contient les valeurs positives=[";
45     for(int i=0;i<TabValPos.size();i++)
46     {
47         cout<< TabValPos[i]<<' ';
48     }
49     cout << "]"<<endl;
50
51     // Affichage du tableau dynamique TabValNeg
52     cout<<" Tableau dynamique contient les valeurs negatives=[";
53     for(int i=0;i<TabValNeg.size();i++)
54     {
55         cout<< TabValNeg[i]<<' ';
56     }
57     cout << "]"<<endl;
58     return 0;
59 }

```

### Exemple d'exécution

```

Entrez la taille du tableau (max 20 )
10
Tab [0]= ? 01
Tab [1]= ? 12
Tab [2]= ? -8
Tab [3]= ? 15
Tab [4]= ? 26
Tab [5]= ? -45
Tab [6]= ? -78
Tab [7]= ? 145
Tab [8]= ? -158
Tab [9]= ? 12
Tableau entre=[1 12 -8 15 26 -45 -78 145 -158 12 ]
Tableau dynamique contient les valeurs positives=[1 12 15 26 145 12 ]
Tableau dynamique contient les valeurs negatives=[-8 -45 -78 -158 ]

```