



SMIA – S2

Corrigé série 4

Informatique 2
2020-2021

Exercice 1

```
1  #include <stdio.h>
2  #define N 20
3  int main(void){
4  int T[N], n, i;
5  /****** Q1******/
6  do{
7      printf("Dimension du tableau (max %d) : ",N);
8      scanf("%d", &n );
9  }while(n<1 || n>N);
10 for (i=0; i<n; i++){
11     printf("Element %d : ",i);
12     scanf("%d", &T[i]);
13 }
14 /****** Q2******/
15 int som, prod;
16 for (i=0,som=0,prod=1; i<n; i++){
17     som += T[i];
18     prod *= T[i];
19 }
20 printf("Somme des elements : %ld\n", som);
21 printf("Produits des elements : %ld\n", prod);
22 printf("La moyenne des elements est : %.2f\n", (double)(som)/n);
```

Exercice 1

```
23  /******* Q3 *****/
24  int nb_pairs;
25  for (nb_pairs=0, i=0; i<n; i++)
26      if(T[i]%2==0)
27          nb_pairs++;
28  printf("nombre des elements pairs est: %d\n", nb_pairs);
29  /******* Q4 *****/
30  int min,max,i_max,i_min;
31  max=min=T[0];
32  i_max=i_min=0;
33  for (i=0; i<n; i++){
34      if (T[i] > max){
35          max=T[i];
36          i_max=i;
37      }
38      if (T[i] < min){
39          min=T[i];
40          i_min=i;
41      }
42  }
43  printf("La valeur maximale est %d, son indice est : %d\n", max, i_max);
44  printf("La valeur minimale est %d, son indice est : %d\n", min, i_min);
```

Exercice 1

```
45  /***** Q5 *****/
46  int val, nb_occur;
47  printf("Donner la valeur a rechercher : ");
48  scanf("%d", &val);
49  nb_occur=0;
50  for (i=0; i<n; i++){
51      if (T[i] == val)
52          nb_occur++;
53  }
54  printf("Le nombre d'occurrence de %d dans le tableau est : %d\n", val, nb_occur);
55  /***** Q6 *****/
56  int j;
57  for(i=j=0; i<n-1; i++)
58      if(T[i]<= T[i+1])
59          j++;
60
61  if(j==0 || j==n-1)
62      printf("Le tableau est trie\n");
63  else
64      printf("Le tableau n'est pas trie\n");
```

Exercice 1

```
65  ***** Q7*****
66  /* Copier tous les éléments non nuls de i vers j et augmenter j pour les éléments non nuls. */
67  for (i=j=0; i<n ; i++){
68      T[j] = T[i];
69      if (T[i]!=0)
70          j++;
71  }
72  /* Nouvelle dimension du tableau ! */
73  n = j;
74  /* Edition des résultats */
75  printf("Tableau sans 0 :\n");
76  for (i=0; i<n; i++)
77      printf("%d ", T[i]);
78  printf("\n");
79  ***** Q8*****
80  /* Echange de T[i] et T[j] */
81  for (i=0,j=n-1 ; i<j ; i++,j--){
82      val = T[i];
83      T[i] = T[j];
84      T[j] = val;
85  }
86  printf("Tableau apres inversion :\n");
87  for (i=0; i<n; i++)
88      printf("%d ", T[i]);
89  printf("\n");
```


Exercice 1

```
90  /***** Q9 *****/
91  val=T[0];
92  for (i=0; i<n-1; i++)
93      T[i]=T[i+1];
94  T[n-1]=val;
95  printf("Tableau apres decalage : \n");
96  for (i=0; i<n; i++)
97      printf("%d ", T[i]);
98  printf("\n");
99  /***** Q10 *****/
100 int T2[N];
101 for (i=0 ; i<n ; i++)
102     T2[i] = 0 ;
103 for (i=j=0; i<n; i++)
104     if(T[i]%2==0)
105         T2[j++]=T[i];
106 printf("Tableau apres recopiage : \n");
107 for (i=0; i<n; i++)
108     printf("%d ", T2[i]);
109 printf("\n");
110 return 0;
111 }
```

Exercise2

```
1  #include <stdio.h>
2  #define N 20
3  int main(void){
4      int n,i,j;
5      int T[N];
6      int i_max1,i_max2;
7  do{
8      printf("Dimension du tableau (max.%d) : ",N);
9      scanf("%d", &n);
10 }while(n<2 || n>N);
11 for (i=0; i<n; i++){
12     printf("Element %d ", i);
13     scanf("%d", &T[i]);
14 }
15 if(T[0]>T[1]){
16     i_max1=0;
17     i_max2=1;
18 }
19 else{
20     i_max1=1;
21     i_max2=0;
22 }
23 for(i=2;i<n;i++){
24     if( T[i]>T[i_max1]){
25         i_max2=i_max1;
26         i_max1=i;
27     }
28     else if(T[i]>T[i_max2])
29         i_max2=i;
30 }
31 printf("Max 1 = %d indice 1= %d. Max 2= %d, indice 2= %d", T[i_max1],i_max1,T[i_max2],i_max2);
32 }
```

Exercice 3

```
1  #include <stdio.h>
2  #define N 20
3  int main(void){
4  int A[N][N],B[N][N], n, i,j;
5  /******Q1******/
6  do{printf("Nombre de lignes et de colonnes (max.%d) :\n ",N);
7  scanf("%d", &n );
8  }while(n<1 || n>N);
9  for (i=0; i<n; i++)
10     for (j=0; j<n; j++){
11         printf("A[%d][%d] : ", i,j);
12         scanf("%d", &A[i][j]);
13     }
14  for (i=0; i<n; i++)
15     for (j=0; j<n; j++){
16         printf("B[%d][%d] : ", i,j);
17         scanf("%d", &B[i][j]);
18     }
19  /******Q2******/
20  int C[N][N],k;
21  for (i=0;i<n;i++){
22     for (j=0;j<n;j++){
23         C[i][j]=0;
24         for(k=0;k<n;k++)
25             C[i][j]=C[i][j]+A[i][k]*B[k][j];
26     }
27 }
28 printf ("matrice produit\n");
29 for (i=0;i<n;i++){
30     for (j=0;j<n;j++)
31         printf("%d \t", C[i][j]);
32     printf("\n");
33 }
```



```

34  /*****Q3*****/
35  int diag1[N],diag2[N];
36  for(i=0,j=n-1;i<n;i++,j--){
37      diag1[i]=A[i][i];
38      diag2[i]=A[i][j];
39  }
40  printf("la premiere diagonal de A (de gauche a droite)\n");
41  for(i=0;i<n;i++)
42      printf("%d ",diag1[i]);
43  printf("\n la deuxieme diagonal de A (de droite a gauche)\n");
44  for(i=0;i<n;i++)
45      printf("%d ",diag2[i]);
46  /*****Q4*****/
47  int val=diag1[0],egal=1;
48  for(i=0;i<n && egal==1;i++)
49      if(diag1[i]!=val || diag2[i]!=val)
50          egal=0;
51  if(egal==0)
52      printf("\n les elements des diagonales ne sont pas egaux a une meme valeur %d \n",val);
53  else
54      printf("\n les elements des diagonales sont egaux a une meme valeur\n");
55  /*****Q5*****/
56  int trace=0;
57  for(i=0;i<n;i++)
58      trace+=diag1[i];
59  printf("la trace est %d",trace);

```

```

60  /*****Q6*****/
61  //for(i=0;i<=n/2;i++)
62  //  for(j=i+1;j<n;j++){
63  //      val=A[i][j];
64  //      A[i][j]=A[j][i];
65  //      A[j][i]=val;
66  //  }
67  int Ta[N][N];
68  for(i=0;i<n;i++)
69      for(j=0;j<n;j++)
70          Ta[i][j]=A[j][i];
71  printf("\n La matrice transposee\n");
72  for(i=0;i<n;i++){
73      for(j=0;j<n;j++)
74          printf("%d\t",Ta[i][j]);
75      printf("\n");
76  }
77  }

```

Exercice 4

```
1  #include <stdio.h>
2  #define nbl 5
3  #define nbC 4
4  int main(void){
5      int maxL[nbl],minC[nbC]; /* Tableaux pour max lignes et mincolonnes */
6      int compteur; /* Compteur Max-Min */
7      int i, j; /* indices courants */
8      int A[nbl][nbC]={1,3,2,3},{9,6,5,4},{11,13,8,7},{0,3,1,3},{10,9,13,12}};
9      printf("Matrice donnee :\n");
10     for (i=0; i<nbl; i++){
11         for (j=0; j<nbC; j++) printf("%d\t", A[i][j]);
12         printf("\n");
13     }
14     /* tableau maximum des lignes */
15     for (i=0; i<nbl; i++){
16         maxL[i]=A[i][0];
17         for (j=1; j<nbC; j++)
18             if (A[i][j]>maxL[i]) maxL[i]=A[i][j];
19     }
20     /* tableau minimum des colonnes */
21     for (j=0; j<nbC; j++){
22         minC[j]=A[0][j];
23         for (i=1; i<nbl; i++)
24             if (A[i][j]<minC[j]) minC[j]=A[i][j];
25     }
26     /* Recherche des Max-Min */
27     printf("Max-Min :\n");
28     for (compteur=0,i=0; i<nbl; i++)
29         for(j=0;j<nbC;j++)
30             if (maxL[i]==minC[j]){
31                 compteur++;
32                 printf("L element %d\t est max sur ligne %d\t et min sur colonne %d\n",A[i][j],i+1,j+1);
33             }
34     if (compteur==0)
35         printf("Le tableau ne contient pas de point Max-Min.\n");
36 }
```