



SMIA – S2

Corrigé série 5

Informatique 2
2019-2020

Ex 1

```
1  #include<stdio.h>
2  int main(){
3      void PNM(int,int);
4      int N,M;
5      printf("entrer N et M \n");
6      scanf("%d %d",&N,&M);
7      PNM(N,M);
8      return 0;
9  }
10
11
12  void PNM(int N,int M){
13      if (N>=M)
14          printf("pas d'affichage\n");
15      else{
16          printf("Les nombres paires entre %d et %d sont:\n",N,M);
17          for (;N<=M;N++)
18              if(N%2==0)
19                  printf("%d \t",N);
20      }
21  }
```

Ex 2

```
1  #include <stdio.h>
2  #define N 20
3  /***** Programme principal *****/
4  int main(void){
5      int min(int[], int);
6      int max(int[], int);
7      int occur(int[], int,int );
8      void echangeTab(int[], int);
9      int T[N], n, i, val;
10     do{ printf("Dimension du tableau (max %d) : ",N);
11         scanf("%d", &n );
12     }while(n<1 || n>N);
13     for (i=0; i<n; i++){
14         printf("Element %d : ",i);
15         scanf("%d", &T[i]);
16     }
17     printf("La valeur maximale est %d\n", max(T,n));
18     printf("La valeur minimale est %d\n", min(T,n));
19     printf("Donner la valeur a rechercher : ");
20     scanf("%d", &val);
21     printf("Le nombre d'occurrence de %d dans le tableau est : %d\n",val, occur(T,n,val));
22     echangeTab(T,n);
23     printf("Tableau apres inversion :\n");
24     for (i=0; i<n; i++)
25         printf("%d ", T[i]);
26     printf("\n");
27     return 0;
28 }
```

```

30  /***** Q1 *****/
31  int min(int T[], int n){
32      int min,i;
33      min=T[0];
34      for (i=0; i<n; i++)
35          if (T[i] < min)
36              min=T[i];
37      return min;
38  }

```

```

39  /***** Q2 *****/
40  int max(int T[], int n){
41      int max,i;
42      max=T[0];
43      for (i=0; i<n; i++)
44          if (T[i] > max)
45              max=T[i];
46      return max;
47  }

```

```

48  /***** Q3 *****/
49  int occur(int T[], int n,int val){
50      int nb_occur,i;
51      nb_occur=0;
52  for (i=0; i<n; i++){
53      if (T[i] == val)
54          nb_occur++;
55  }
56  return nb_occur;
57  }

```

```

58  /***** Q4 *****/
59  /* Echange de T[i] et T[j] */
60  void echangeTab(int T[], int n){
61      int i,j,val;
62  for (i=0,j=n-1 ; i<j ; i++,j--){
63      val = T[i];
64      T[i] = T[j];
65      T[j] = val;
66  }
67
68  }

```


Ex 3

```
1  #include <stdio.h>
2  #define N 20
3  /****** programme principal******/
4  int main(void){
5      int trace(int A[][N], int taille);
6      int appartenance(int A[][N], int taille, int x);
7      void permutationLignes(int A[][N], int taille, int k, int l);
8      void remplace(int A[][N], int taille, int x, int y);
9      int A[N][N],taille,i,j,val;
10     int Li1, Li2,Co1, Co2,ancien,nouveau;
11     do{ printf("Nombre de lignes et de colonnes (max.%d) : ",N);
12         scanf("%d", &taille );
13     }while(taille<0 || taille>N);
14     for (i=0; i<taille; i++)
15     {
16         for (j=0; j<taille; j++){
17             printf("A[%d][%d] : \t", i,j);
18             scanf("%d", &A[i][j]);
19             printf("\n");
20         }
21     }
22     /* Question 1*/
23     printf("La trace de la matrice: %d \n",trace(A, taille));
24     /*question 2*/
25     printf("Entrer la valeur a rechercher dans la matrice:\n");
26     scanf("%d",&val);
27     if(appartenance(A, taille, val)==1) printf("La matrice contient la valeur %d \n", val);
28     else printf("La matrice ne contient pas la valeur %d \n", val);
```

Ex 3

```
27  /* question 3*/
28  do{ printf("Entrer l'indice de la premiere ligne a permuter (premier indice=0):\n");
29      scanf("%d",&Li1);
30      printf("Entrer l'indice de la deuxieme ligne a permuter (premier indice=0):\n");
31      scanf("%d",&Li2);
32  }while(Li1<0 || Li1>=taille || Li2<0 || Li2>=taille);
33  permutationLignes(A, taille, Li1, Li2);
34  printf("Matrice avec lignes permutees :\n");
35  afficheMatrice(A,taille);
36  /* Question 4*/
37  printf("Entrer l'ancienne valeur et la nouvelle valeur':\n");
38  scanf("%d %d",&ancien,&nouveau);
39  remplace(A, taille, ancien, nouveau);
40  printf("Matrice avec element remplace :\n");
41  afficheMatrice(A,taille);
42  return 0;
43  }
44  /*procédure pour affichage de La matrice*/
45  void afficheMatrice(int mat[][N], int taille){
46      int i,j;
47      for (i=0; i<taille; i++){
48          for (j=0; j<taille; j++)
49              printf("%d\t", mat[i][j]);
50          printf("\n");
51      }}
```

Ex 3

```
52  /*question 1*/
53  int trace(int mat[][N], int taille){
54      int i, compt=0;
55      for (i=0; i<taille; i++)
56          compt+=mat[i][i];
57      return compt;  }
58  /*question 2*/
59  int appartenance(int mat[][N], int taille, int x){
60      int i, j;
61      for (i=0; i<taille; i++){
62          for (j=0; j<taille; j++)
63              if (mat[i][j] == x)
64                  return 1;
65      }
66      return 0;}
67  /*quetsion 3*/
68  void permutationLignes(int A[][N], int taille, int k, int m){
69      int i, j, temp;
70      for(i=0; i<taille; i++){
71          temp=A[k][i];
72          A[k][i] = A[m][i];
73          A[m][i] = temp;
74      }}
75  /*question 4*/
76  void remplace(int A[][N], int taille, int x, int y){
77      int i, j;
78      for (i=0; i<taille; i++)
79          for (j=0; j<taille; j++)
80              if (A[i][j] == x)
81                  A[i][j] = y;
82  }
```


Ex 4

```
1  #include<stdio.h>
2  #define N 20
3  int main(void){
4      void triSelection(int [], int );
5      void triBulle(int [], int );
6      int T[N], n, i,c;
7  do{ 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 printf("choisir 1 pour tri par selection et 2 pour tri a bulle \n");
15 scanf("%d",&c);
16 switch(c){
17     case 1: {triSelection(T, n);break;}
18     case 2: {triBulle(T, n);break;}
19 }
20 for(i=0;i<n;i++)
21     printf("%d \t", T[i]);
22 return 0;
23 }
```


Ex 4

```
24 void triSelection(int tab[],int taille){
25     int i,temp,j,Imin;
26     for(i =0; i <taille-1; i++){
27         Imin=i;
28         for(j=i+1;j<taille;j++)
29             if(tab[j] < tab[Imin])
30                 Imin=j;
31         if(Imin != i){
32             temp= tab[i];
33             tab[i] = tab[Imin];
34             tab[Imin] = temp;
35         }
36     }
37 }
```

```
38 void triBulle(int tab[],int taille){
39     int i, temp, n, desordre;
40     n = taille;
41     desordre =1;
42     while (desordre){
43         desordre=0;
44         for(i=0;i<n-1;i++){
45             if(tab[i] > tab[i+1]){
46                 temp= tab[i+1];
47                 tab[i+1] = tab[i];
48                 tab[i] = temp;
49                 desordre = 1;
50             }
51         }
52         n--;
53     }
54 }
```