Série 4 (la solution)

Exercice 1 :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
int M[100][100],n,m,i,j,S,P;
printf("donner la taille de ligne\n");
scanf("%d",&n);
printf("donner la taille des
   colone\n");
scanf("%d",&m);
/* remplissage de la matrice */
for(i=0;i<n;i++){
  for(j=0;j<m;j++){
printf("donner un element\n");
scanf("%d",&M[i][j]);
}}
/* calcule de la somme */
S=0;
for(i=0;i<n;i++){
  for(j=0;j<m;j++){
  S = S + M[i][j];
}}/* affichage de la somme */
printf("la somme est %d",S);
/* calcule de produit */
P=1;
for(i=0;i<n;i++){
  for(j=0;j<m;j++){
  P= P * M[i][j];
}}
/* affichage de produit
printf("le produit est %d",P);
return 0;
}
```

Exercice 2 :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
int M[n][m],n,m,i,j,max;
int K,f;
/* déclaration de élément
nécessaire pour remplir la matrice */
scanf("%d %d",&n,&m);
for(i=0;i<m;i++){
  for(j=0;j<m;j++){
scanf("%d",&M[i][j]);
}}
/* recherche de maximum */
max = M[0][0];
K=0; f=0;
for(i=0;i<n;i++){
  for(j=0;j<m;j++){
if(max < M[i][j]){
   max = M[i][j];
```

k = i ; f = j ; }}

printf (" le maximum est %d et sa position est la ligne %d et la colone %d ", max, k+1, f+1);
return 0; }}

Exercice 3 :

1/ Algorithme parcour ;
Var : M : tableau (100, 100) ; entier
val, i, j, n, m, c : entrer
Debut :
lire (n, m) ;
// remplisage de la matrice
Pour i ← 1 à n faire
pour j ← 1 à m faire
lire (M(i, j)) ;
Fpour
Fpour
lire (val) ;
// Parcour ligne par ligne pour trouvé nbr d'occorence de val.

Pour i ← 1 à n faire
c ← 0 ;
Pour j ← 1 à m faire
si val = M(i, j) alors
c ← c+1 ;
Fsi
Fpour ;

(2)

Ecrire ('le nbr d'occorence est', c) ;
Fpour
Fin.

2/ - metter dans un tableau les nbr d'occorence.
on change la Boucle "(2)"
Var : T : tableau (100) ; entrer ;
Pour i ← 1 à n faire
c ← 0
Pour j ← 1 à m faire
si val = M(i, j) alors
c ← c+1 ;
Fsi
Fpour
T(i) ← c ;
Fpour
// affichage de tableau.
Pour i ← 1 à n faire
Ecrire (T(i)) ;
Fpour

Exercice 4 :

Pour nbr d'occorence pour chaque colone il suffit just de changer les Boucle (2) comme suit :
Pour j ← 1 à m faire
c ← 0 ;
pour i ← 1 à n faire
si val = M(i, j) alors
c ← c + 1 ;
Fsi
Fpour

Ecrire l'le nbr d'occurence et - ; d), 2/ parcour diagonale

Exercice 5 :

1/- parcour ligne par ligne

Algorithme Parcour ligne ;

Var : M : tableau (n,m) : entier ;

i, j, n : entier ;

Debut :

lire (n) ; ;

// remplissage de la matrice.

Pour i ← 1 à n faire

pour j ← 1 à n faire

lire (M(i,j)) ;

Fpour

Fpour

// metter des zero par parcour ligne par ligne

Pour i ← 1 à n faire

pour j ← 1 à n faire

si i = j alors

M(i,j) ← 0 ;

Fsi

Fpour

Fpour

(2)

// affichage de la matrice

Pour i ← 1 à n faire

Pour j ← 1 à n faire

Ecrire (M(i,j)) ;

Fpour Fpour Fin.

// changement de la boucle (2)

pour i ← 1 à n faire

M(i, i) ← 0 ;

Fpour

// affichage de la matrice après la modification.

Pour i ← 1 à n faire

pour j ← 1 à n faire

Ecrire (M(i,j)) ;

Fpour Fpour

Fin.

Exercice 6 :

matrice symitrique

M(i,j) = M(j,i)

exemple :

M(1,2) = M(2,1)

M(2,3) = M(3,2)

Algorithme Symitrique ;

Var : M : tableau (100,100) : entier ;

i, j, n : entier

Debut :

lire (n) ;

pour i ← 1 à n faire

pour j ← 1 à n faire

lire (M(i,j)), Fpour, Fpour,

// Vérifie si la matrice Symtrique
// Supposé que M est symtrique
S ← 1;
Pour i ← 1 à n faire
Pour j ← i+1 à n faire
Si M(i,j) <> M(j,i) alors
S ← 0;
// S ← 0 c-à-d Matrice symtrique
Fsi
Fpour
Fpour

Si S = 1 alors
Ecrire ("la matrice est Symtrique");
Sinon
Ecrire ("la matrice n'est pas Symtrique");
Fsi
Fin.

Exercice 7 :

```
#include <stdio.h>
#include <stdlib.h>
int main(){
int mat1[100][100], mat2[100][100];
int prod[100][100], i, j, n, m, l, k;
printf("donner le nombre de lignes et de colonne \n");
scanf("%d%d", &n, &m);
/* remplissage de la matrice 1 */
for(i=0; i<n; i++){
for(j=0; j<m; j++){
scanf("%d", &mat1[i][j]);
}}
/* remplissage de la matrice 2 */
for(i=0; i<n; i++){
for(j=0; j<m; j++){
scanf("%d", &mat2[i][j]);
}}
/* le produit */
for(i=0; i<n; i++){
for(j=0; j<l; j++){
prod[i][j]=0;
for(k=0; k<m; k++){
prod[i][j] = prod[i][j] + mat1[i][k] * mat2[k][j];
}}}
/* affichage de la matrice Prod */
for(i=0; i<n; i++){
for(j=0; j<l; j++){
printf("%d", Prod[i][j]);
}}
return 0;
}
```

Remarque :

le produit entre deux matrice Mat1 et Mat2 est une troisième matrice Prod telle que :

$$Prod(i,j) = \sum_{k=1}^{m} Mat1(i,k) * Mat2(k,j)$$

Exercice 8 :

```
#include <stdio.h>
#include <stdlib.h>
int main ()
{
int M [100][100], i, j, n, m, K, B[100];
/* demmande de taille*/
Printf(" donner le nbr de ligne et decolone \n");
Scanf( "%d%d", &n, &m );
/* remplissage de la matrice*/
for(i=0;i<n;i++)
 for(j=0;j<m;j++)
 Scanf ("%d", &M[i][j]);
}}
/* transforme M en tableau B */
 K=0;
for(i=0;i<n;i++)
for(j=0;j<m;j++)
 B[K] = M[i][j];
  K++;
 }}
/* affichage de tableau B*/
for(K=0;K<nam;K++)
 Printf ("%d", B[K]);
}
return0;
}
```