BÀI THỰC HÀNH SỐ 11

MẢNG HAI CHIỀU- TWO-DIMENSIONAL (2D) ARRAY

For example

```
1.  float x[3][4];
```

	Column 1	Column 2	Column 3	Column 4
Row 1	x[0][0]	x[0][1]	x[0][2]	x[0][3]
Row 2	x[1][0]	x[1][1]	x[1][2]	x[1][3]
Row 3	x[2][0]	x[2][1]	x[2][2]	x[2][3]

Initialization of a two dimensional array

```
// Different ways to initialize two dimensional array

int c[2][3] = {{1, 3, 0}, {-1, 5, 9}};

int c[][3] = {{1, 3, 0}, {-1, 5, 9}};

int c[2][3] = {1, 3, 0, -1, 5, 9};
```

Duyệt ma trận:

```
for ( i =0; i<row; i++)
{
        for ( j=0; j< column; j++)
                [if (condition)] Access m[i][j];
}
```

Example 1: Two Dimensional Array to store and print values

```c
// C program to store temperature of two cities for a week and display it.

#include <stdio.h>

const int CITY = 2;
const int WEEK = 7;

int main()
{
    int temperature[CITY][WEEK];
    for (int i = 0; i < CITY; ++i) {
        for(int j = 0; j < WEEK; ++j) {
            printf("City %d, Day %d: ", i+1, j+1);
            scanf("%d", &temperature[i][j]);
        }
    }

    printf("\nDisplaying values: \n\n");
    for (int i = 0; i < CITY; ++i) {
        for(int j = 0; j < WEEK; ++j)
        {
            printf("City %d, Day %d = %d\n", i+1, j+1, temperature[i][j]);
        }
    }
    return 0;
}
```

Output

```
City 1, Day 1: 33
City 1, Day 2: 34
City 1, Day 3: 35
City 1, Day 4: 33
City 1, Day 5: 32
City 1, Day 6: 31
City 1, Day 7: 30
City 2, Day 1: 23
City 2, Day 2: 22
City 2, Day 3: 21
City 2, Day 4: 24
City 2, Day 5: 22
City 2, Day 6: 25
City 2, Day 7: 26
```

```
Displaying values:

City 1, Day 1 = 33
City 1, Day 2 = 34
City 1, Day 3 = 35
City 1, Day 4 = 33
City 1, Day 5 = 32
City 1, Day 6 = 31
City 1, Day 7 = 30
City 2, Day 1 = 23
City 2, Day 2 = 22
City 2, Day 3 = 21
City 2, Day 4 = 24
City 2, Day 5 = 22
City 2, Day 6 = 25
City 2, Day 7 = 26
```

Example 2: Sum of two matrices

```c
// C program to find the sum of two matrices of order 2*2

#include <stdio.h>
int main()
{
   float a[2][2], b[2][2], result[2][2];

   // Taking input using nested for loop
   printf("Enter elements of 1st matrix\n");
   for (int i = 0; i < 2; ++i)
      for (int j = 0; j < 2; ++j)
      {
         printf("Enter a%d%d: ", i + 1, j + 1);
         scanf("%f", &a[i][j]);
      }

   // Taking input using nested for loop
   printf("Enter elements of 2nd matrix\n");
   for (int i = 0; i < 2; ++i)
      for (int j = 0; j < 2; ++j)
      {
         printf("Enter b%d%d: ", i + 1, j + 1);
         scanf("%f", &b[i][j]);
      }

   // adding corresponding elements of two arrays
   for (int i = 0; i < 2; ++i)
      for (int j = 0; j < 2; ++j)
      {
         result[i][j] = a[i][j] + b[i][j];
      }

   // Displaying the sum
   printf("\nSum Of Matrix:");

   for (int i = 0; i < 2; ++i)
      for (int j = 0; j < 2; ++j)
      {
         printf("%.1f\t", result[i][j]);

         if (j == 1)
            printf("\n");
      }
   return 0;
}
```

Output

```
Enter elements of 1st matrix
Enter a11: 2;
Enter a12: 0.5;
Enter a21: -1.1;
Enter a22: 2;
Enter elements of 2nd matrix
Enter b11: 0.2;
Enter b12: 0;
Enter b21: 0.23;
Enter b22: 23;

Sum Of Matrix:
2.2       0.5
-0.9      25.0
```

Example 3:

```c
/* Static Matric Demo.*/
#include <stdio.h>
#define MAXR 20
#define MAXC 20
/* Input a mtrix of ints, num of rows and column are known */
void input(int m[][MAXC], int r, int c);
int max (int m[][MAXC], int r, int c);
void print (int m[][MAXC], int r, int c);
int main()
{  int m[MAXR][MAXC]; /* Declare a static matrix*/
   int r, c; /* real used number of rows and columns */
   int maxVal;
   do
   {  printf("Enter number of rows and columns of the matrix:");
      scanf("%d%d", &r, &c);
   }
   while (r<1 || r >MAXR || c<1 || c > MAXC);
   printf("Enter a matrix %d x %d\n", r, c);
   input(m, r, c);
   maxVal  = max (m, r, c);
   printf("Max value:%d\n", maxVal);
   printf("\nInputted matrix:\n");
   print(m, r, c);
   while (getchar()!='\n');getchar();
   return 0;
}

void input(int m[][MAXC], int r, int c)
{ int i, j;
  for (i=0;i<r; i++) /* Enter values to each row */
  {  for (j=0; j<c; j++) /* Enter value to each column */
      { printf("Value at [%d][%d]:", i, j);
        scanf("%d", &m[i][j]);
      }
  }
}
```

```
36 int max(int m[][MAXC], int r, int c)
37 {  int result = m[0][0];
38    int i, j;
39    for (i=0;i<r; i++)
40       for (j=0; j<c; j++)
41         if (result < m[i][j]) result=m[i][j];
42    return result;
43 }
44 void print (int m[][MAXC], int r, int c)
45 {  int i, j;
46    for (i=0;i<r; i++)
47    { for (j=0; j<c; j++) printf("%7d", m[i][j]);
48      printf("\n");
49    }
50 }
```

Bài 1

Viết chương trình thực hiện các yêu cầu sau:

- Tạo ma trận A vuông bậc n (n nhập từ bàn phím) với các phần tử được nhập từ bàn phím, xuất ma trận.
- Tính tổng các phần tử trên đường chéo chính (vết - trace) của ma trận A.

```
Nhap bac ma tran: 3 ↵
a[0][0] = 1 ↵
...
a[2][2] = 9 ↵
    1    2    3
    0    5    6
    0    0    9
Trace = 15
```

Bài 2

Viết chương trình để trừ hai Ma trận có cùng kích thước.

Test Data :
Input the size of the square matrix (less than 5): 2
Input elements in the first matrix :
element - [0],[0] : 5
element - [0],[1] : 6
element - [1],[0] : 7
element - [1],[1] : 8
Input elements in the second matrix :
element - [0],[0] : 1
element - [0],[1] : 2
element - [1],[0] : 3
element - [1],[1] : 4

Expected Output :
The First matrix is :

5 6
7 8
The Second matrix is :

1 2
3 4
The Subtraction of two matrix is :

4 4
4 4

Bài 3

Viết chương trình để tìm chuyển vị của một ma trận đã cho.

Test Data :
Input the rows and columns of the matrix : 2 2
Input elements in the first matrix :
element - [0],[0] : 1
element - [0],[1] : 2
element - [1],[0] : 3
element - [1],[1] : 4
Expected Output :
The matrix is :

1 2
3 4

The transpose of a matrix is :
1 3
2 4