Name	Debjit Ghosal
UID no.	2023300065
Experiment No.	5

AIM:	Demonstrate the use of two-dimensional arrays to solve a given problem
Program 1	
PROBLEM STATEMENT :	Write a program to perform Matrix Addition, Subtraction
ALGORITHM:	1. Declare and define the `read_mat` function: a. Take the number of rows (r) and columns (c) as input. b. Create a 2D array 'a' with 'r' rows and 'c' columns. c. Use nested loops to read 'r' * 'c' integers and store them in the 'a' array. 2. Declare and define the `print_mat` function: a. Take the number of rows (r) and columns (c) as input. b. Display the elements of the 2D array 'a' in a matrix format using nested loops. 3. Declare and define the `add_matrix` function: a. Take the number of rows (r) and columns (c) as input. b. Create a 2D array 'result' with 'r' rows and 'c' columns. c. Use nested loops to add corresponding elements of arrays 'a' and 'b' and store the result in the 'result' array. d.Call the `print_mat` function to print the result matrix. 4. Declare and define the `sub_matrix` function: a. Take the number of rows (r) and columns (c) as input. b. Create a 2D array 'result' with 'r' rows and 'c' columns. c. Use nested loops to subtract corresponding elements of array 'b' from array 'a' and store the result in the 'result' array. d. Call the `print_mat` function to print the result matrix. 5. In the `main` function: a. Prompt the user to enter the number of rows and columns. b. Read the values of 'r' and 'c' from the user. c. Create two 2D arrays 'mat1' and 'mat2' with 'r' rows and 'c' columns. d. Prompt the user to enter 'r' * 'c' integers for the first array and call the `read_mat` function to store the values in 'mat1'.

	e. Prompt the user to enter 'r' * 'c' integers for the second array and call the `read_mat` function to store the values in 'mat2'. f. Display "The addition of the given matrices is:" and call the `add_matrix` function with 'mat1' and 'mat2'. g. Display "The subtraction of the given matrices is:" and call the `sub_matrix` function with 'mat1' and 'mat2'.

PROGRAM:	#include<stdio.h>
	void mat_addition(int r,int c,int mat1[r][c],int mat2[r][c],int mat3[r][c])
	{
	int i,j;
	for(i=0;i<r;i++)
	for(j=0;j<c;j++)
	mat3[i][j]=mat1[i][j]+mat2[i][j];
	}
	void mat_subtraction(int r,int c,int mat1[r][c],int mat2[r][c],int mat4[r][c])
	{
	int i,j;
	for(i=0;i<r;i++)
	for(j=0;j<c;j++)
	mat4[i][j]=mat1[i][j]-mat2[i][j];
	}
	void read_mat(int r,int c,int a[][c])
	{
	for(int i=0;i<r;i++)
	for(int j=0;j<c;j++)
	scanf("%d",&a[i][j]);
	}
	void print_mat(int r,int c,int a[][c])
	{
	for(int i=0;i<r;i++)
	{
	for(int j=0;j<c;j++)
	printf(" %d ",a[i][j]);
	printf("\n");
	}
	}
	int main ()
	{
	int r,c;

```
            printf("Enter the Number of Rows : ");
            scanf("%d",&r);
                printf("Enter the Number of Columns : ");
                 scanf("%d",&c);
           int mat1[r][c],mat2[r][c],mat3[r][c],mat4[r][c];
             printf("Start Entering elements for matrix 1 : \n");
             read_mat(r,c,mat1);
                printf("Start Entering elements for matrix 2 : \n");
               read_mat(r,c,mat2);
           mat_addition(r,c,mat1,mat2,mat3);
              printf("\n Addition of two matrices : \n");
              print_mat(r,c,mat3);
                    mat_subtraction(r,c,mat1,mat2,mat4);
           printf("\n Subtraction of two matrices : \n");
                  print_mat(r,c,mat4);
return 0;
}
```

RESULT:

```
Enter the Number of Rows : 2
Enter the Number of Columns : 2
Start Entering elements for matrix 1 :
10
20
30
40
Start Entering elements for matrix 2 :
10
15
20
25

 Addition of two matrices :
 20  35
 50  65

 Subtraction of two matrices :
 0  5
 10  15
```

Program 2	
PROBLEM STATEMENT :	Write a program to perform Matrix Multiplication
ALGORITHM:	1. Declare and define the `read_mat` function: a. Take the number of rows (r) and columns (c) as input and a 2D array 'a'. b. Use nested loops to read 'r' * 'c' integers and store them in the 'a' array. 2. Declare and define the `print_mat` function: a. Take the number of rows (r) and columns (c) as input and a 2D array 'a'. b. Display the elements of the 2D array 'a' in a matrix format using nested loops. 3. Declare and define the `mul_mat` function: a. Take the number of rows (r1), columns (c1) for the first matrix, and the first matrix 'mat1'. b. Take the number of rows (r2), columns (c2) for the second matrix, and the second matrix 'mat2'. c. Check if the multiplication is possible by comparing 'c1' with 'r2'. If not, display an error message and return. d. Create a 2D array 'result' with 'r1' rows and 'c2' columns. e. Use nested loops to perform matrix multiplication: f. Initialize elements of 'result' to 0. g. Use nested loops to perform the matrix multiplication calculation. h. Call the `print_mat` function to print the result matrix. 4. In the `main` function: a. Set up an infinite loop to allow the user to enter matrix dimensions until a valid multiplication is possible. b. Prompt the user to enter the number of rows and columns for the first and second matrices (r1, c1, r2, c2). c. Check if the multiplication is possible by comparing 'c1' with 'r2'. If not, display an error message and ask the user to enter dimensions again. d. If multiplication is possible, break out of the loop. e. Create two 2D arrays 'mat1' and 'mat2' with the specified dimensions. f. Prompt the user to enter 'r1' * 'c1' integers for the first matrix and call the `read_mat` function to store the values in 'mat1'. g. Prompt the user to enter 'r2' * 'c2' integers for the second matrix and call the `read_mat` function to store the values in 'mat2'. h. Display "The multiplication is:" and call the `mul_mat` function with the two matrices and their dimensions.

PROGRAM:

```
#include<stdio.h>
void read_mat(int r, int c, int a[][c])
{
	for (int i=0;i<r;i++)
		for (int j=0;j<c;j++)
			scanf ("%d", &a[i][j]);
}
void print_mat(int r, int c, int a[][c])
{
	for (int i=0;i<r;i++)
	{
		for (int j=0;j<c;j++)
			printf ("%d ", a[i][j]);
		printf("\n");
	}
}
void mul_mat(int r1, int c1, int mat1[][c1], int r2, int c2, int mat2[][c2])
{
	int result[r1][c2];
	for (int i=0;i<r1;i++)
	{
		for (int j=0;j<c2; j++)
		{
		result[i][j]=0;
			for (int k=0;k<c1;k++)
			{
				result[i][j]+=mat1[i][k]*mat2[k][j];
			}
		}
	}
	print_mat(r1,c2,result);
}
int main()
{
	int r1, c1, r2, c2;
	while(1)
	{
		printf ("Enter the number of rows of first matrix:");
		scanf ("%d", &r1);
		printf ("Enter the number of columns of first matrix:");
```

```
                scanf ("%d", &c1);
                printf ("Enter the number of rows of second matrix:");
                scanf ("%d", &r2);
                printf ("Enter the number of columns of second matrix:");
                scanf ("%d", &c2);
                if(c1!=r2)
                {
                        printf("multiplication is not possible.Enter again.\n");
                }
                else
                {
                        break;
                }
        }
        int mat1 [r1][c1], mat2 [r2][c2];
        printf("Enter %d numbers of first array:", r1*c1);
        read_mat(r1,c1,mat1);
        printf("Enter %d numbers of second array:", r2*c2);
        read_mat(r2,c2,mat2);
        printf("The multiplication is:\n");
        mul_mat(r1, c1, mat1, r2, c2, mat2);
        return 0;
}
```

RESULT:

```
Enter the number of rows of first matrix:2
Enter the number of columns of first matrix:2
Enter the number of rows of second matrix:2
Enter the number of columns of second matrix:2
Enter 4 numbers of first array:5
6
4
3
Enter 4 numbers of second array:2
8
9
10
The multiplication is:
64 100
35 62
```

Program 3	
PROBLEM STATEMENT:	Write a program to perform Transpose of Matrix
ALGORITHM:	1. Declare and define the `read_mat` function: a. Take the number of rows (r) and columns (c) as input and a 2D array 'a'. b. Use nested loops to read 'r' * 'c' integers and store them in the 'a' array. 2. Declare and define the `print_mat` function: a. Take the number of rows (r) and columns (c) as input and a 2D array 'a'. b. Display the elements of the 2D array 'a' in a matrix format using nested loops. 3. Declare and define the `transpose` function: a. Take the number of rows (r) and columns (c) of the original matrix 'm'. b. Create a new 2D array 'trans' with 'c' rows and 'r' columns for the transposed matrix. c. Use nested loops to calculate the transpose by swapping rows and columns. d. Call the `print_mat` function to print the transposed matrix. 4. In the `main` function: a. Prompt the user to enter the number of rows and columns for the matrix. b. Read the values of 'r' and 'c' from the user. c. Create a 2D array 'mat' with the specified dimensions. d. Prompt the user to enter 'r' * 'c' integers for the matrix and call the `read_mat` function to store the values in 'mat'. e. Display "The transpose is:" and call the `transpose` function with the matrix and its dimensions.

PROGRAM:	#include <stdio.h>
	int main()
	{
	int a[10][10], transpose[10][10], r, c;
	printf("Enter rows and columns: ");
	scanf("%d %d", &r, &c);
	printf("\nEnter matrix elements:\n");
	for (int i = 0; i < r; ++i)
	for (int j = 0; j < c; ++j)
	{
	printf("Enter element a%d%d: ", i + 1, j + 1);
	scanf("%d", &a[i][j]);

```
    }

  printf("\nEntered matrix: \n");
      for (int i = 0; i < r; ++i)
         for (int j = 0; j < c; ++j)
  {
   printf("%d  ", a[i][j]);
   if (j == c - 1)
   printf("\n");
   }
    for (int i = 0; i < r; ++i)
        for (int j = 0; j < c; ++j)
      {
          transpose[j][i] = a[i][j];
      }
  printf("\nTranspose of the matrix:\n");
  for (int i = 0; i < c; ++i)
  for (int j = 0; j < r; ++j)
  {
   printf("%d  ", transpose[i][j]);
   if (j == r - 1)
   printf("\n");
   }
  return 0;
}
```

RESULT:

```
Enter rows and columns: 2
3

Enter matrix elements:
Enter element a11: 1
Enter element a12: 2
Enter element a13: 3
Enter element a21: 4
Enter element a22: 5
Enter element a23: 6

Entered matrix:
1  2  3
4  5  6

Transpose of the matrix:
1  4
2  5
3  6
```

CONCLUSION:	Froom this experiment I learnt the use of two-dimentional arrays by addition, subtraction, multiplication and transpose of matrices.