Name	Lekh Nayak
UID no.	2023800068
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. Start
2. Define a void function read with the arguments : dimensions m and n of a 2D array ,array.
  a. Declare variables i and j.
  b. Iterate i from 0 to m-1 and iterate j from 0 to n-1:
     - - Display "Enter the value of arr[i][j]: ".
     - - Read the value of arr[i][j] using scanf.
3. Define the function print with the arguments : dimensions m and n of a 2D array and the array itself.
   a. Declare variables i and j.
   b. Iterate i from 0 to m-1 and iterate j from 0 to n-1:
     - - Print the value of arr[i][j] followed by a tab.
     - - If j equals n-1, print a newline.
4. Define the function addMatrice with the arguments : dimensions m and n of two 2D arrays and their elements, and a third array for storing the result.
   a. Declare variables i and j.
   b. Iterate i from 0 to m-1 and iterate j from 0 to n-1:
     - - Calculate the sum of corresponding elements of arr1 and arr2 and store it in res[i][j].
   c. Display "result: ".
   d. Call the print function with parameters m, n, and res.
5. Define the function subtractMatrix with the arguments : dimensions m and n of two 2D arrays and their elements, and a third array for storing the result.
  a. Declare variables i and j.
  b. Iterate i from 0 to m-1 and iterate j from 0 to n-1:
    - - Calculate the difference of corresponding elements of arr1 and arr2 and store it in res[i][j].
  c. Display "result: ".
  d. Call the print function with parameters m, n, and res.
6. Define the main function.
  a. Declare integer variables m, n, and choice.
  b. Display "rows: " and read the value of m using scanf.
  c. Display "columns: " and read the value of n using scanf.
  d. Declare 2D arrays arr1, arr2, and res, each with dimensions m x n.
  e. Display "Enter elements first:" and call the read function with parameters m, n, and arr1.
  f. Display "Enter elements second:" and call the read function with parameters m, n, and arr2.
  g. Display "First matrix is:" and call the print function with parameters m, n, and arr1.
  h. Display "Second matrix is:" and call the print function with parameters m, n, and arr2.
   i. Display the menu options for addition and subtraction.
   j. Read the value of choice using scanf.
  k. Using a switch-case statement, perform the chosen operation based on the user's choice.
  l. Return 0 to indicate successful execution.
7. End
```

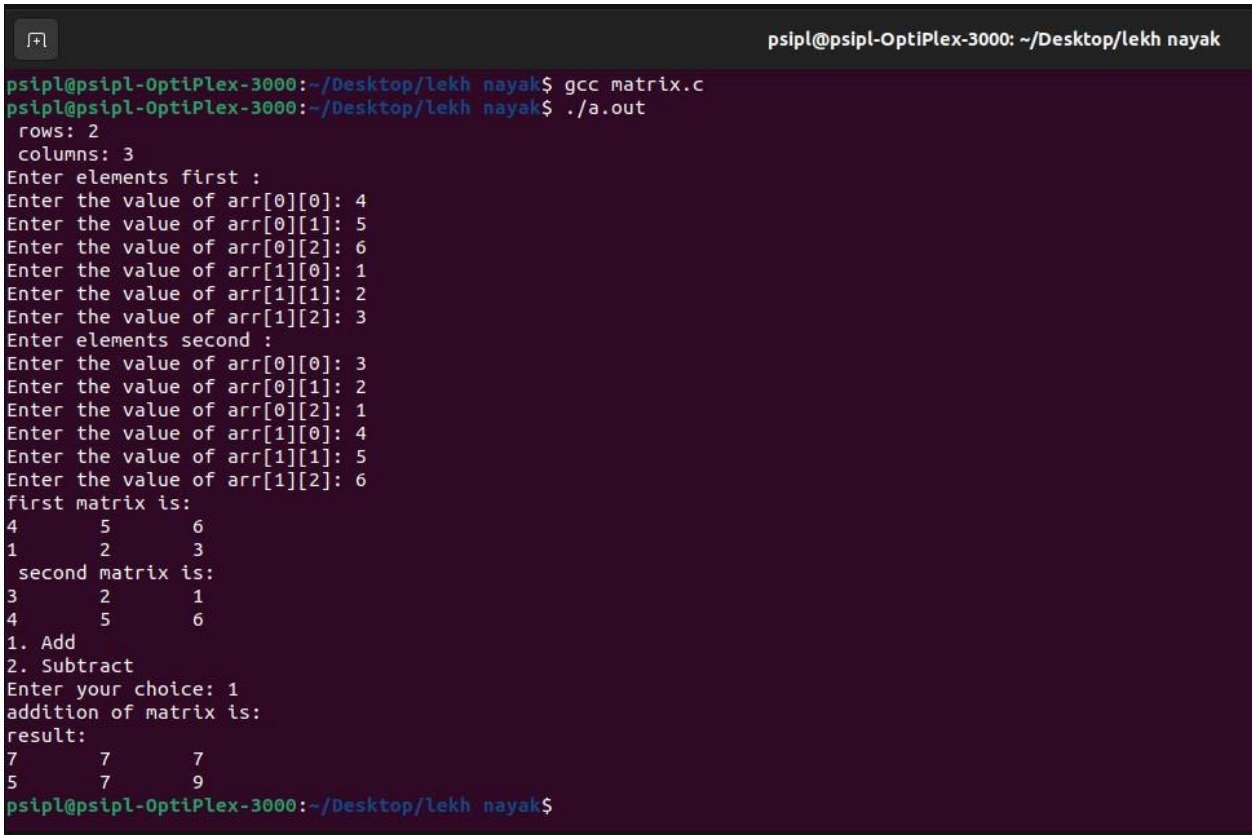
PROGRAM:

```
#include <stdio.h>
void read (int m,int n, int arr[m][n])
{
int i,j;
for (i=0;i<m;i++)
{
for (j=0;j<n;j++)
{
printf("Enter the value of arr[%d][%d]: ",i,j);
scanf("%d",&arr[i][j]);
}
}
}
void print(int m,int n, int arr[m][n])
{
```

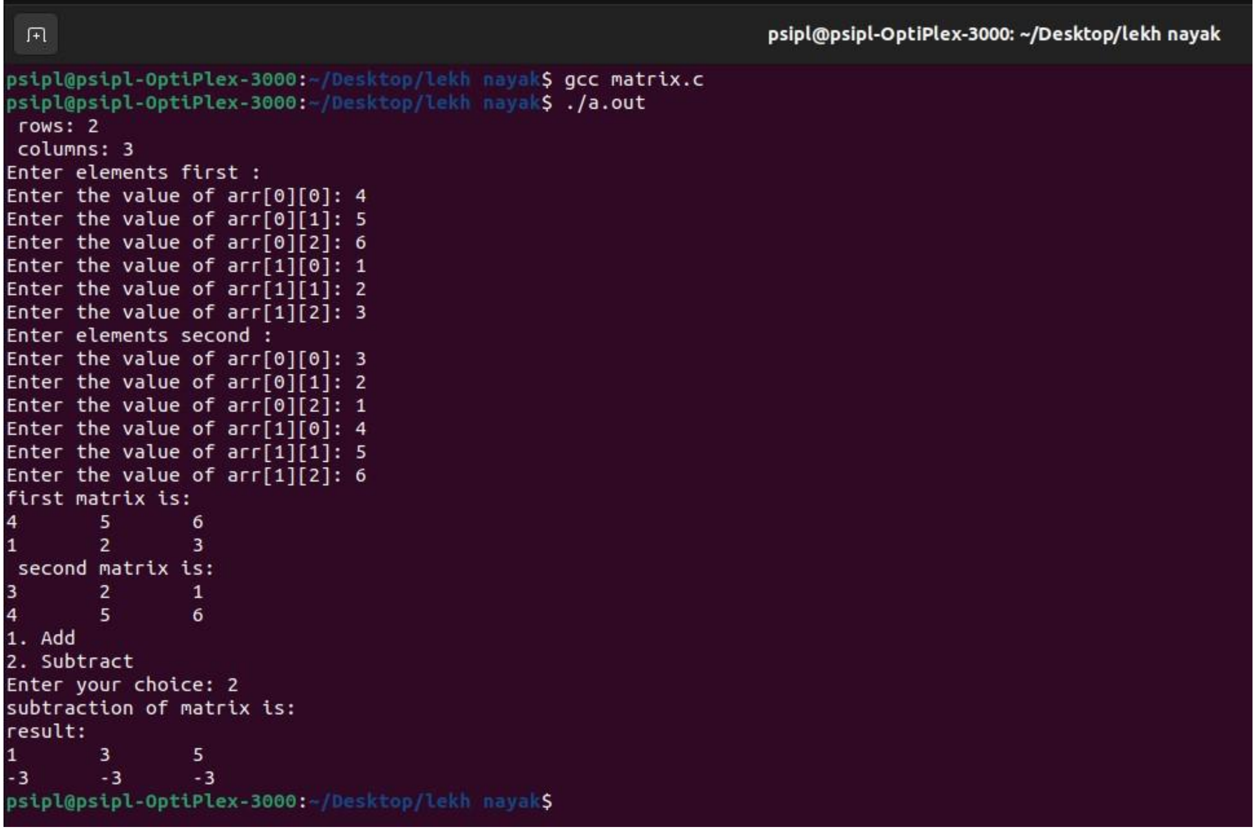
```
int i,j;
for (i=0;i<m;i++)
{
for (j=0;j<n;j++)
{
printf("%d\t",arr[i][j]);
}
printf("\n");
}
}
void addMatrice(int m, int n ,int arr1[m][n],int arr2[m][n],int res[m][n])
{
int i,j;
for (i=0;i<m;i++)
{
for (j=0;j<n;j++)
{
res[i][j]=arr1[i][j]+arr2[i][j];
}
}
printf("result: \n");print(m,n,res);
}
void subtractMatrix(int m, int n ,int arr1[m][n],int arr2[m][n],int res[m][n])
{
int i,j;
for (i=0;i<m;i++)
{
for (j=0;j<n;j++)
{
res[i][j]=arr1[i][j]-arr2[i][j];
}
}
printf("result: \n");
print(m,n,res);
}
int main()
{
int m,n;
```

```
printf(" rows: ");
scanf("%d",&m);
printf(" columns: ");
scanf("%d",&n);
int arr1[m][n],arr2[m][n],res[m][n];
printf("Enter elements first : \n");
read(m,n,arr1);
printf("Enter elements second : \n");
read(m,n,arr2);
printf("first matrix is: \n");
print(m,n,arr1);
printf(" second matrix is: \n");
print(m,n,arr2);
printf("1. Add\n2. Subtract\n");
printf("Enter your choice: ");
int choice;
scanf("%d",&choice);
switch(choice){
case 1: printf("addition of matrix is: \n");
addMatrice(m,n,arr1,arr2,res);
break;
case 2:printf("subtraction of matrix is: \n");
subtractMatrix(m,n,arr1,arr2,res);
break;
default: printf("Invalid choice");
}
return 0;
}
```

RESULT:



```
psipl@psipl-OptiPlex-3000:~/Desktop/lekh nayak$ gcc matrix.c
psipl@psipl-OptiPlex-3000:~/Desktop/lekh nayak$ ./a.out
 rows: 2
 columns: 3
Enter elements first :
Enter the value of arr[0][0]: 4
Enter the value of arr[0][1]: 5
Enter the value of arr[0][2]: 6
Enter the value of arr[1][0]: 1
Enter the value of arr[1][1]: 2
Enter the value of arr[1][2]: 3
Enter elements second :
Enter the value of arr[0][0]: 3
Enter the value of arr[0][1]: 2
Enter the value of arr[0][2]: 1
Enter the value of arr[1][0]: 4
Enter the value of arr[1][1]: 5
Enter the value of arr[1][2]: 6
first matrix is:
4       5       6
1       2       3
 second matrix is:
3       2       1
4       5       6
1. Add
2. Subtract
Enter your choice: 1
addition of matrix is:
result:
7       7       7
5       7       9
psipl@psipl-OptiPlex-3000:~/Desktop/lekh nayak$
```



```
psipl@psipl-OptiPlex-3000:~/Desktop/lekh nayak$ gcc matrix.c
psipl@psipl-OptiPlex-3000:~/Desktop/lekh nayak$ ./a.out
 rows: 2
 columns: 3
Enter elements first :
Enter the value of arr[0][0]: 4
Enter the value of arr[0][1]: 5
Enter the value of arr[0][2]: 6
Enter the value of arr[1][0]: 1
Enter the value of arr[1][1]: 2
Enter the value of arr[1][2]: 3
Enter elements second :
Enter the value of arr[0][0]: 3
Enter the value of arr[0][1]: 2
Enter the value of arr[0][2]: 1
Enter the value of arr[1][0]: 4
Enter the value of arr[1][1]: 5
Enter the value of arr[1][2]: 6
first matrix is:
4       5       6
1       2       3
 second matrix is:
3       2       1
4       5       6
1. Add
2. Subtract
Enter your choice: 2
subtraction of matrix is:
result:
1       3       5
-3      -3      -3
psipl@psipl-OptiPlex-3000:~/Desktop/lekh nayak$
```

Program 2	
PROBLEM STATEMENT :	Write a program to perform Matrix Multiplication
ALGORITHM:	1. Start 2. Define a void function read with the arguments : dimensions m and n of a 2D array , array. a. Declare variables i and j. b. Iterate i from 0 to m and iterate j from 0 to n,incrementing i and j by 1 each iteration: - - Read the value of arr[i][j] using scanf. 3. Define the function print with the arguments : dimensions m and n of a 2D array ,array. a. Declare variables i and j. b. Iterate i from 0 to m and iterate j from 0 to n,incrementing i and j by 1 each iteration: - - Print the value of arr[i][j] followed by a tab. - print a newline. 4. Define a void function multiplyMatrices with the arguments : dimensions m, n, p, and q of two 2D arrays and their elements, result array. a. Iterate i from 0 to m,incrementing i by 1 each iteration: b. Iterate j from 0 to n ,incrementing j by 1 each iteration: - - result[i][j] =0. - - Iterate k from 0 to n,incrementing k by1 each iteration: - - Update result[i][j] by adding the product of A[i][k] and B[k][j]. c. Print "Resultant matrix:". d. Call the print function with parameters m, n, and result. 5. Define the main function. a. Declare integer variables m, n, p, and q. b. Print "row column for 1st:" and read the values of m and n using scanf. c. Print "row column for 2:" and read the values of p and q using scanf. d. If the number of columns in the first matrix is not equal to the number of rows in the second matrix: - - Print "error" e. Declare 2D arrays A, B, and res, each with dimensions m x n, p x q, and n x p, respectively. f. Print "Enter elements first:" and call the read function with parameters m, n, and A. g. Print "Enter elements second:" and call the read function with parameters p, q, and B. h. Print "First matrix is:" and call the print function with parameters m, n, and A.

i. Print "Second matrix is:" and call the print function with parameters p, q, and B.

j. Call the multiplyMatrices function with parameters m, n, p, q, A, B, and res.

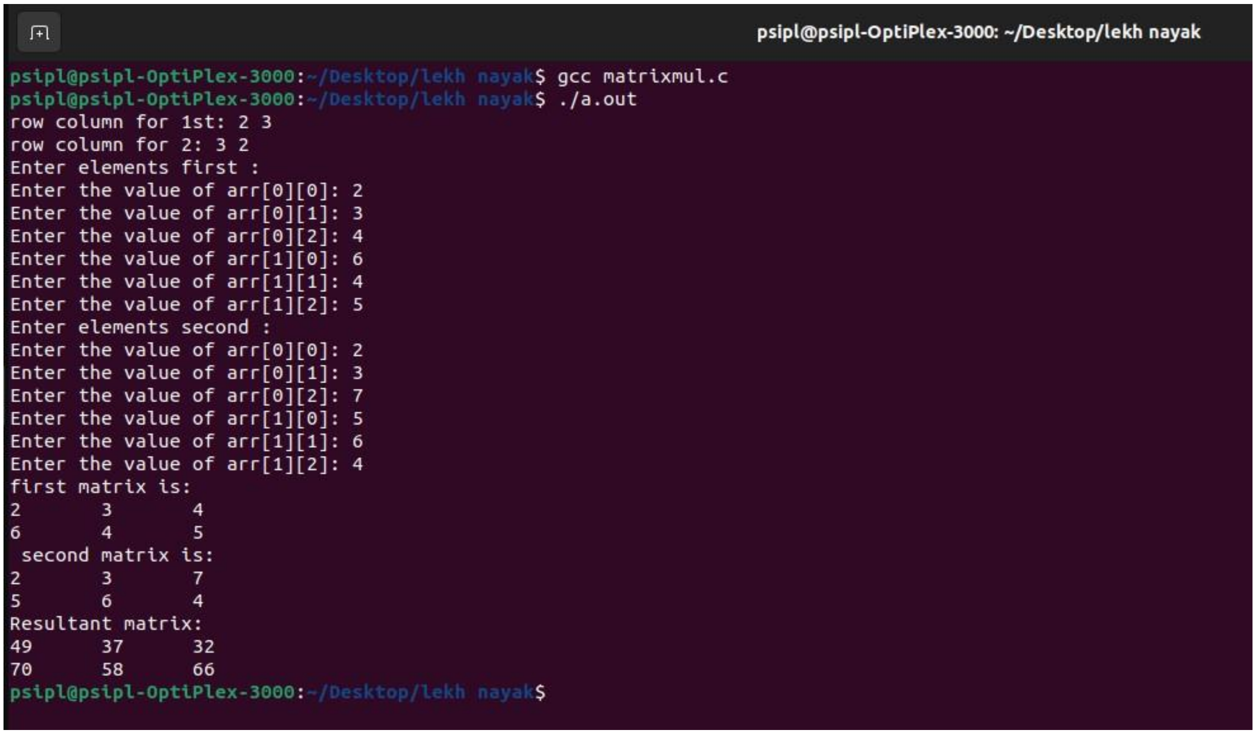
6. End

PROGRAM:

```
#include <stdio.h>
void read (int m,int n, int arr[m][n])
{
int i,j;
for (i=0;i<m;i++)
{
for (j=0;j<n;j++)
{
printf("Enter the value of arr[%d][%d]: ",i,j);
scanf("%d",&arr[i][j]);
}
}
}
void print(int m,int n, int arr[m][n])
{
int i,j;
for (i=0;i<m;i++)
{
for (j=0;j<n;j++)
{
printf("%d\t",arr[i][j]);
}
printf("\n");
}
}
void multiplyMatrices( int m, int n, int p,int q,int A[m][n],
int B[p][q], int result[n][p])
{
for (int i = 0; i < m; i++){
for (int j = 0; j < n; j++)
{
result[i][j] = 0;
for (int k = 0; k < n; k++)
{
```

```
result[i][j] += A[i][k] * B[k][j];
}
}
}
printf("Resultant matrix:\n");
print(m,n,result);
}
int main()
{
int m, n, p, q;
printf("row column for 1st: ");
scanf("%d %d", &m, &n);
printf("row column for 2: ");
scanf("%d %d", &p, &q);
if (n != p)
{
printf("error\n");
return 1;
}
int A[m][n], B[p][q], res[n][p];
printf("Enter elements first : \n");
read(m,n,A);
printf("Enter elements second : \n");
read(m,n,B);
printf("first matrix is: \n");
print(m,n,A);
printf(" second matrix is: \n");print(m,n,B);
multiplyMatrices(m,n,p,q,A,B,res);
return 0;
}
```

RESULT:



```
psipl@psipl-OptiPlex-3000:~/Desktop/lekh nayak$ gcc matrixmul.c
psipl@psipl-OptiPlex-3000:~/Desktop/lekh nayak$ ./a.out
row column for 1st: 2 3
row column for 2: 3 2
Enter elements first :
Enter the value of arr[0][0]: 2
Enter the value of arr[0][1]: 3
Enter the value of arr[0][2]: 4
Enter the value of arr[1][0]: 6
Enter the value of arr[1][1]: 4
Enter the value of arr[1][2]: 5
Enter elements second :
Enter the value of arr[0][0]: 2
Enter the value of arr[0][1]: 3
Enter the value of arr[0][2]: 7
Enter the value of arr[1][0]: 5
Enter the value of arr[1][1]: 6
Enter the value of arr[1][2]: 4
first matrix is:
2       3       4
6       4       5
 second matrix is:
2       3       7
5       6       4
Resultant matrix:
49      37      32
70      58      66
psipl@psipl-OptiPlex-3000:~/Desktop/lekh nayak$
```

Program 3

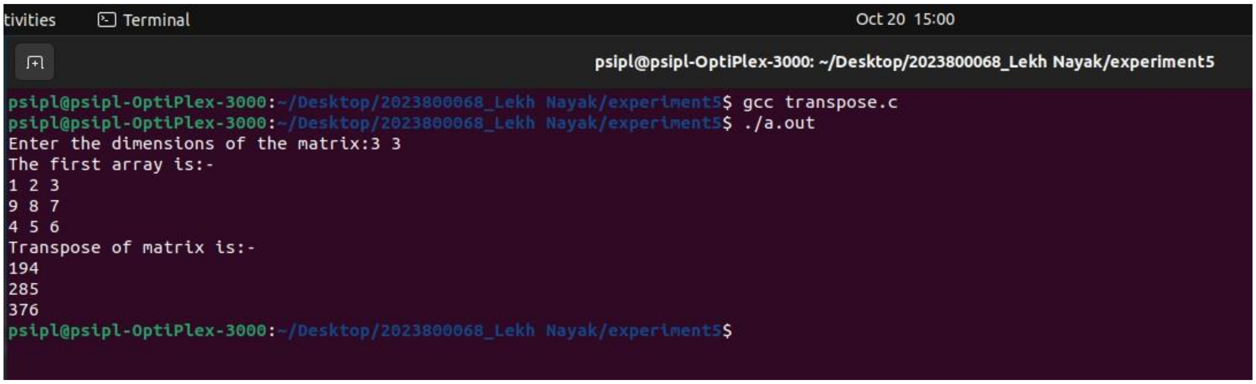
PROBLEM STATEMENT:	Write a program to perform Transpose of Matrix
ALGORITHM:	STEP 1: Start STEP 2: Declare row,col of the matrices and take the input from the user. STEP 3: Declare two arrays a[row][col] STEP 4: Call the function arreleinput() STEP 5: Declare array c[col][row] STEP 6: Call the transpose() function STEP 7: Call the arrprint() function STEP 8: Stop Algorithm for arreleinput(): STEP 1: Take arguments row ,col,a[row][col] [START OF OUTER FOR LOOP] STEP 2: int i=0;i<row [START OF INNER FOR LOOP] Int j=0;j<col Take a[i][j] as input

```
 j=j+1
 [END OF INNER FOR LOOP]
 i=i+1
 [END OF OUTER FOR LOOP]

Algorithm for arrprint():

STEP 1: Take arguments row , col ,a[row][col]
 [START OF OUTER FOR LOOP]
STEP 2: int i=0; i<row
 [START OF INNER FOR LOOP]
 Int j=0; j<col
 PRINT a[i][j]
 j=j+1
 [END OF INNER FOR LOOP]
 i=i+1
 [END OF OUTER FOR LOOP]
Algorithm for tranpose():
STEP 1:Take row ,col ,a[row][col], c[col][row] as arguments
 [START OF OUTER FOR LOOP]
STEP 2: : int i=0; i<row
 [START OF INNER FOR LOOP]
 Int j=0; j<col
 c[j][i]=a[i][j]
 j=j+1
 [END OF INNER FOR LOOP]
 i=i+1
 [END OF OUTER FOR LOOP
```

PROGRAM:

```
#include<stdio.h>

void read(int m,int n, int arr[m][n]){
 for(int i=0;i<m; i++){
   for(int j=0; j<n; j++){
   scanf("%d",&arr[i][j]);
    }
  }
}

void print(int m, int n,int arr[m][n]){
 for(int i=0; i<m; i++){
  for(int j=0; j<n; j++){
  printf("%d" , arr[i][j]);
   }
  printf("\n");
 }
}

void transpose(int m, int n, int arr1[m][n], int arr2[n][m]){
            for(int i=0;i<m;i++){
                        for(int j=0;j<n;j++){
                                    arr2[j][i]=arr1[i][j];
                        }
            }
}

int main(){
   int m,n;
   printf("Enter the dimensions of the matrix:");
   scanf("%d%d",&m,&n);
   int arr1[m][n];
   int arr2[n][m];
   printf("The first array is:-\n");
   read(m, n, arr1);
   transpose(m, n, arr1, arr2);
   printf("Transpose of matrix is:-\n");
   print(m, n, arr2);
}
```

RESULT:


tivities
Terminal
Oct 20 15:00
psipl@psipl-OptiPlex-3000: ~/Desktop/2023800068_Lekh Nayak/experiment5

```
psipl@psipl-OptiPlex-3000:~/Desktop/2023800068_Lekh Nayak/experiment5$ gcc transpose.c
psipl@psipl-OptiPlex-3000:~/Desktop/2023800068_Lekh Nayak/experiment5$ ./a.out
Enter the dimensions of the matrix:3 3
The first array is:-
1 2 3
9 8 7
4 5 6
Transpose of matrix is:-
194
285
376
psipl@psipl-OptiPlex-3000:~/Desktop/2023800068_Lekh Nayak/experiment5$
```

CONCLUSION:	In this experiment I learnt the use of two dimensional arrays to solve a given problem