MMDS LAB ASSINGMENT 1

VENNELA G
20BDS0146

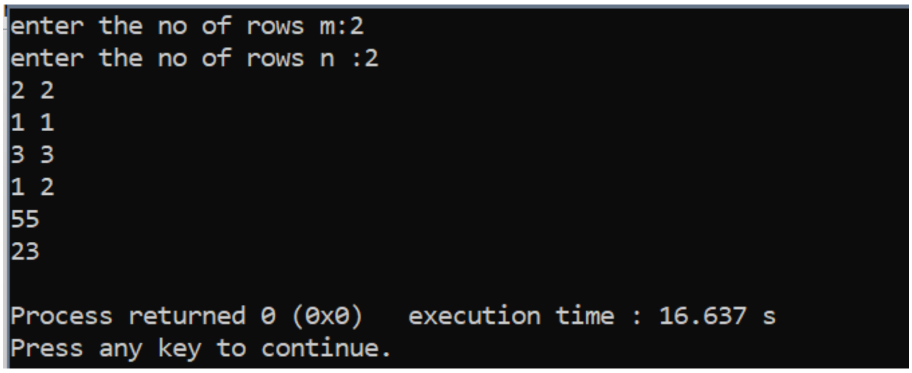
1.Code to add two matrices:

Code:

```
#include <stdio.h>
int main()
{
int m, n;
printf("enter the no of rows m:");
scanf("%d",&m);
printf("enter the no of rows n :");
scanf("%d",&n);
int i, j;
int mat1[m][n], mat2[m][n], mat3[m][n];
for(i = 0; i < m; i++)
{
for(j = 0; j < n; j++)
scanf("%d",&mat1[i][j]);
}
for(i = 0; i < n; i++)
{
```

```
for(j = 0; j < n; j++)
scanf("%d",&mat2[i][j]);
 }

 for(i = 0; i < m; i++)
 {
 for(j = 0; j < n; j++)
 {
 mat3[i][j] = mat1[i][j] + mat2[i][j];
 }
 }

 for(i = 0; i < m; i++)
 {
 for(j = 0; j < n; j++)
 printf("%d", mat3[i][j]);
 printf("\n");
 }

 return 0;
 }
```

Output:



```
enter the no of rows m:2
enter the no of rows n :2
2 2
1 1
3 3
1 2
55
23

Process returned 0 (0x0)   execution time : 16.637 s
Press any key to continue.
```

code for multiplication of two matrix:

Code:

```
#include <stdio.h>


int main()

{

int m, n, p, q, c, d, k, sum = 0;

int mat1[10][10], mat2[10][10], mat3[10][10];


printf("Enter number of rows and columns of mat1 matrix\n");

scanf("%d%d", &m, &n);

printf("Enter elements of matrix 1\n");


for (c = 0; c < m; c++)

for (d = 0; d < n; d++)

scanf("%d", &mat1[c][d]);


printf("\nEnter number of rows and columns of mat2 matrix\n");

scanf("%d%d", &p, &q);
```

```
if (n != p)
printf("\nThe matrices can’t be multiplied with each other.\n");
else
{
printf("\nEnter elements of matrix2\n");

for (c = 0; c < p; c++)
for (d = 0; d < q; d++)
scanf("%d", &mat2[c][d]);

for (c = 0; c < m; c++) {
for (d = 0; d < q; d++) {
for (k = 0; k < p; k++) {
sum = sum + mat1[c][k]*mat2[k][d];
}

mat3[c][d] = sum;
sum = 0;
}
}

printf("\nProduct of the matrices:\n");

for (c = 0; c < m; c++) {
for (d = 0; d < q; d++)
printf("%d\t", mat3[c][d]);
```

printf("\n");

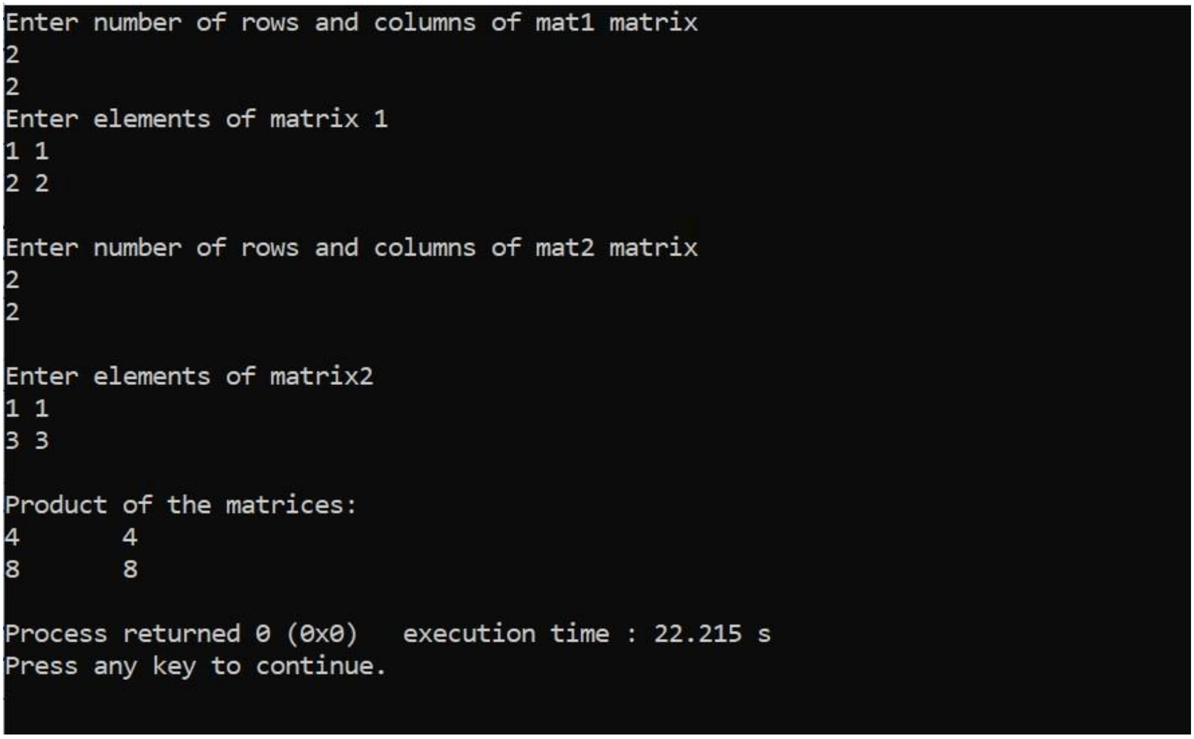
}

}

return 0;

}

Output:



```
Enter number of rows and columns of mat1 matrix
2
2
Enter elements of matrix 1
1 1
2 2

Enter number of rows and columns of mat2 matrix
2
2

Enter elements of matrix2
1 1
3 3

Product of the matrices:
4       4
8       8

Process returned 0 (0x0)   execution time : 22.215 s
Press any key to continue.
```

3.code for finding transpose of a matrix:

Code:

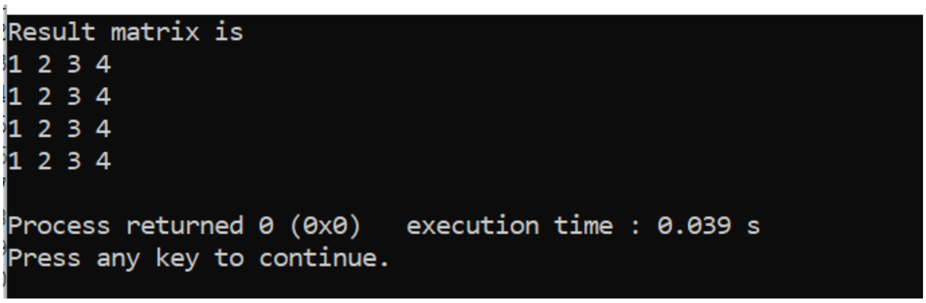
#include <stdio.h>

#define N 4

```
void transpose(int A[][N], int B[][N])
 {
        int i, j;
        for (i = 0; i < N; i++)
                for (j = 0; j < N; j++)
                        B[i][j] = A[j][i];
 }

 int main()
 {
        int A[N][N] = { {1, 1, 1, 1},
                                        {2, 2, 2, 2},
                                        {3, 3, 3, 3},
                                        {4, 4, 4, 4}};

        int B[N][N], i, j;

        transpose(A, B);

        printf("Result matrix is \n");
        for (i = 0; i < N; i++)
        {
                for (j = 0; j < N; j++)
                printf("%d ", B[i][j]);
                printf("\n");
```

```
    }


    return 0;

}
```

OUTPUT:

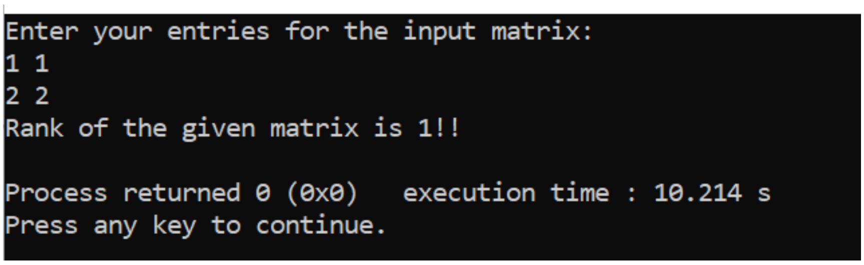


```
Result matrix is
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4

Process returned 0 (0x0)   execution time : 0.039 s
Press any key to continue.
```

Code:

```
#include <stdio.h>

#define ORDER 2

int main() {

int i, j, n = ORDER, det, matrix[2][2];

printf("Enter your entries for the input matrix:\n");

for (i = 0; i < n; i++) {

for (j = 0; j < n; j++) {

scanf("%d", &matrix[i][j]);

}

}

det = (matrix[0][0] * matrix[1][1]) -

(matrix[0][1] * matrix[1][0]);

if (det) {

printf("Rank of the given matrix is 2!!\n");
} else {

printf("Rank of the given matrix is 1!!\n");

}

return 0;

}
```

OUTPUT:



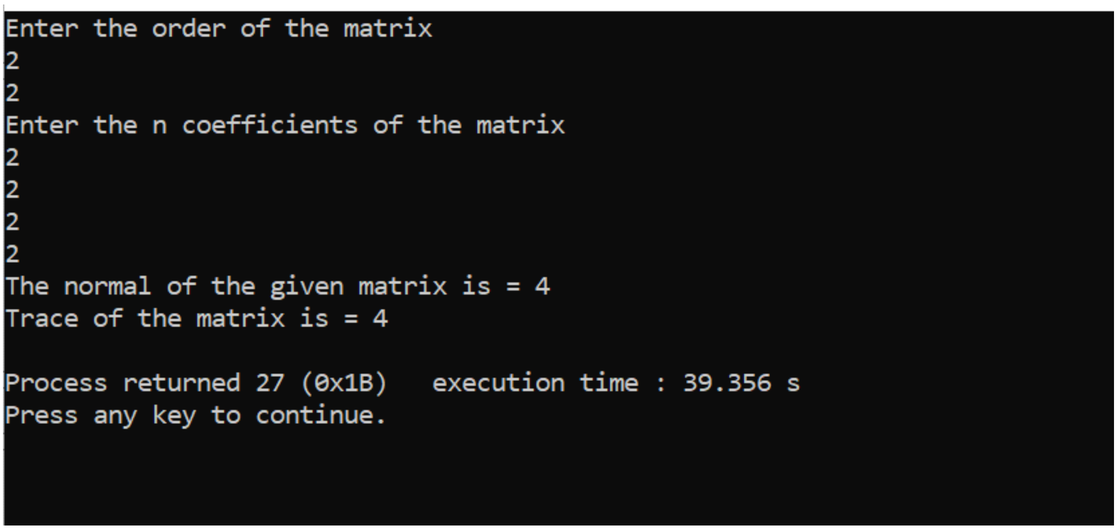
```
Enter your entries for the input matrix:
1 1
2 2
Rank of the given matrix is 1!!

Process returned 0 (0x0)   execution time : 10.214 s
Press any key to continue.
```

Find the trace of a Matrix.

Code:

```
#include <stdio.h>

    #include <math.h>



    void main ()

    {



        static int array[10][10];

        int i, j, m, n, sum = 0, sum1 = 0, a = 0, normal;



        printf("Enter the order of the matrix\n");

        scanf("%d %d", &m, &n);



        printf("Enter the n coefficients of the matrix \n");

        for (i = 0; i < m; ++i)
        {
            for (j = 0; j < n; ++j)
            {
                scanf("%d", &array[i][j]);
                a = array[i][j] * array[i][j];
                sum1 = sum1 + a;
            }
        }

        normal = sqrt(sum1);
        printf("The normal of the given matrix is = %d\n", normal);
        for (i = 0; i < m; ++i)
        {
            sum = sum + array[i][i];
        }

        printf("Trace of the matrix is = %d\n", sum);

    }
```

OUTPUT:



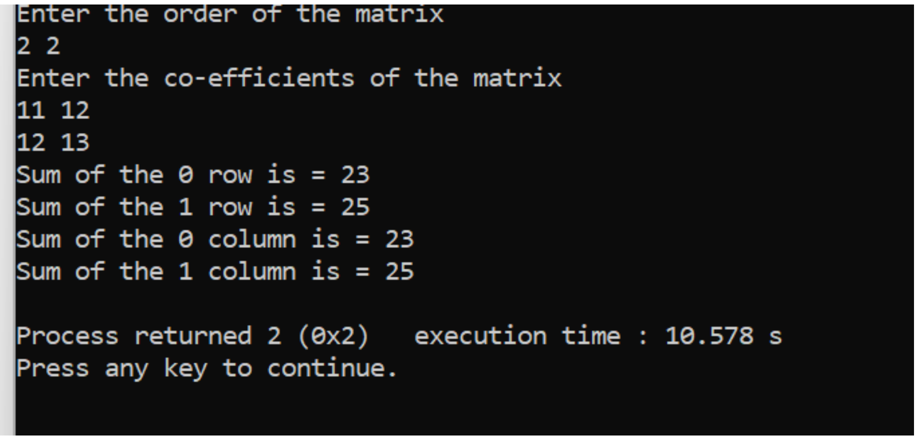
```
Enter the order of the matrix
2
2
Enter the n coefficients of the matrix
2
2
2
2
The normal of the given matrix is = 4
Trace of the matrix is = 4

Process returned 27 (0x1B)   execution time : 39.356 s
Press any key to continue.
```

QUES 2. (i) Find the sum of each row of a matrix (ii) Find the sumof each column of a matrix:

Code:

```
#include <stdio.h>

void main ()

{


  static int array[10][10];

  int i, j, m, n, sum = 0;


  printf("Enter the order of the matrix\n");

  scanf("%d %d", &m, &n);


  printf("Enter the co-efficients of the matrix\n");

  for (i = 0; i < m; ++i)

  {

    for (j = 0; j < n; ++j)
    {
        scanf("%d", &array[i][j]);
    }
}

for (i = 0; i < m; ++i)
{
    for (j = 0; j < n; ++j)
    {
        sum = sum + array[i][j] ;
    }

    printf("Sum of the %d row is = %d\n", i, sum);
    sum = 0;

}
sum = 0;
for (j = 0; j < n; ++j)
{
    for (i = 0; i < m; ++i)
    {
        sum = sum + array[i][j];
    }

    printf("Sum of the %d column is = %d\n", j, sum);
    sum = 0;
    }

  }
```

OUTPUT:



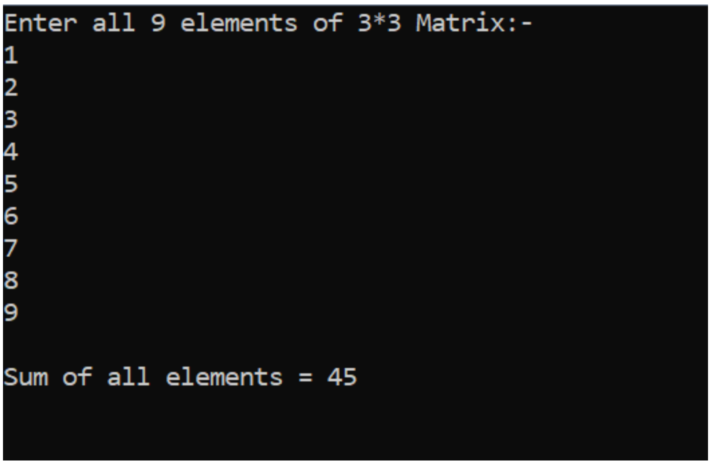
```
Enter the order of the matrix
2 2
Enter the co-efficients of the matrix
11 12
12 13
Sum of the 0 row is = 23
Sum of the 1 row is = 25
Sum of the 0 column is = 23
Sum of the 1 column is = 25

Process returned 2 (0x2)   execution time : 10.578 s
Press any key to continue.
```

Ques 2.3) ***Find the sum of all the elements of a matrix.***

Code;

```
#include<stdio.h>

#include<conio.h>

int main()

{

   int mat[3][3], i, j, sum;

   sum = 0;

   printf("Enter all 9 elements of 3*3 Matrix:-\n");

   for(i=0; i<3; i++)

   {

      for(j=0; j<3; j++)

      {
         scanf("%d", &mat[i][j]);

         sum = sum + mat[i][j];

      }

   }

   printf("\nSum of all elements = %d", sum);

   getch();

   return 0;

}
```

Output:



```
Enter all 9 elements of 3*3 Matrix:-
1
2
3
4
5
6
7
8
9

Sum of all elements = 45
```

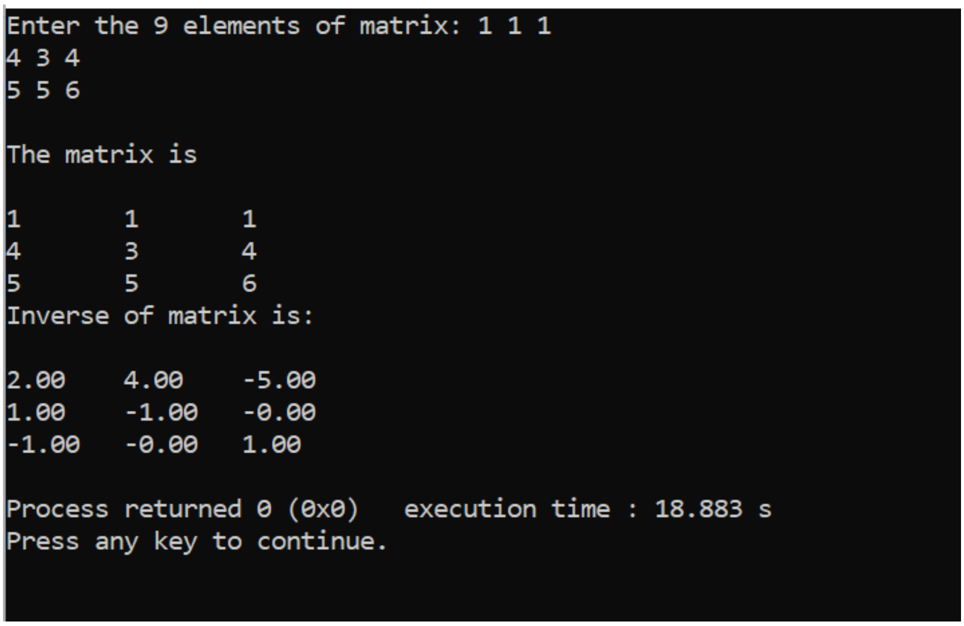
Ques 2.4) Find the inverse of a Matrix:

Code:

```
#include<stdio.h>


int main(){


 int a[3][3],i,j;

 float determinant=0;
```

```
  printf("Enter the 9 elements of matrix: ");
  for(i=0;i<3;i++)
     for(j=0;j<3;j++)
          scanf("%d",&a[i][j]);

  printf("\nThe matrix is\n");
  for(i=0;i<3;i++){
     printf("\n");
     for(j=0;j<3;j++)
          printf("%d\t",a[i][j]);
  }

  for(i=0;i<3;i++)
      determinant = determinant + (a[0][i]*(a[1][(i+1)%3]*a[2][(i+2)%3] -
a[1][(i+2)%3]*a[2][(i+1)%3]));

   printf("\nInverse of matrix is: \n\n");
   for(i=0;i<3;i++){
     for(j=0;j<3;j++)
          printf("%.2f\t",((a[(i+1)%3][(j+1)%3] * a[(i+2)%3][(j+2)%3]) -
(a[(i+1)%3][(j+2)%3]*a[(i+2)%3][(j+1)%3]))/ determinant);
       printf("\n");
   }

   return 0;
}
```

Output:



```
Enter the 9 elements of matrix: 1 1 1
4 3 4
5 5 6

The matrix is

1       1       1
4       3       4
5       5       6
Inverse of matrix is:

2.00    4.00    -5.00
1.00    -1.00   -0.00
-1.00   -0.00   1.00

Process returned 0 (0x0)   execution time : 18.883 s
Press any key to continue.
```

2.5) (v) Find the sum of all the diagonal elements (Both Diagonals) of a matrix:

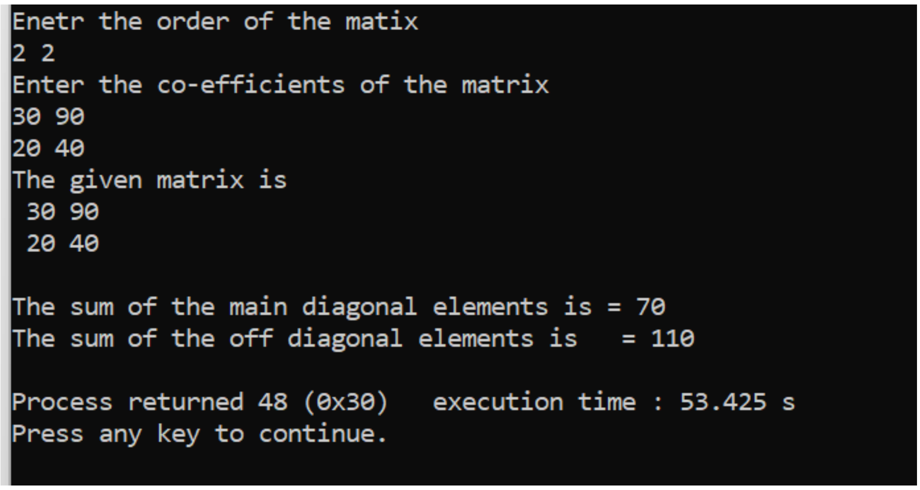
Code:

```
#include <stdio.h>

  void main ()

  {



    static int array[10][10];

    int i, j, m, n, a = 0, sum = 0;



    printf("Enetr the order of the matix \n");

    scanf("%d %d", &m, &n);



    if (m == n )

    {



      printf("Enter the co-efficients of the matrix\n");
```

```
for (i = 0; i < m; ++i)
{
    for (j = 0; j < n; ++j)
    {
        scanf("%d", &array[i][j]);
    }
}

printf("The given matrix is \n");
for (i = 0; i < m; ++i)
{
    for (j = 0; j < n; ++j)
    {
        printf(" %d", array[i][j]);
    }
    printf("\n");
}

for (i = 0; i < m; ++i)
{
    sum = sum + array[i][i];
    a = a + array[i][m - i - 1];
}

printf("\nThe sum of the main diagonal elements is = %d\n", sum);
printf("The sum of the off diagonal elements is  = %d\n", a);
```

```
    }

    else

      printf("The given order is not square matrix\n");

  }
```

Output:



```
Enetr the order of the matix
2 2
Enter the co-efficients of the matrix
30 90
20 40
The given matrix is
 30 90
 20 40

The sum of the main diagonal elements is = 70
The sum of the off diagonal elements is   = 110

Process returned 48 (0x30)   execution time : 53.425 s
Press any key to continue.
```