**Lab no: 5 – 1D ARRAYS**

Q1. Find the largest and smallest element in a 1D array.

**Program:**

/\*Finding the largest and the smallest element in the array\*/

#include <stdio.h>

#include <stdlib.h>

int main()

{

printf("Name : MANOJ M MALLYA\n\n");

int a[100],i,n,min,max;

printf("Enter no of elements : ");

scanf("%d",&n);

printf("Enter the elements : \n");

for(i=0; i<n; i++) // input 1D array

{

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

}

min = a[0];

max = a[0];

for (i=1;i<n;i++)

{

if (a[i]<min)

{

min = a[i];

}

if (a[i]>max)

{

max = a[i];

}

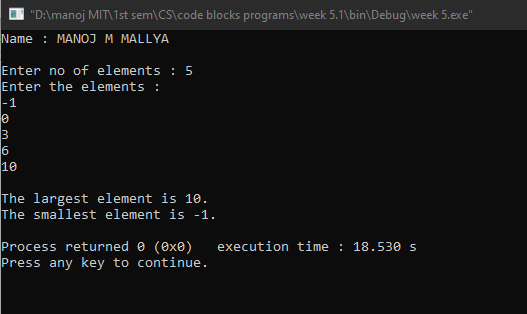
}

printf("\nThe largest element is %d.\nThe smallest element is %d.\n",max,min);

return 0;

}

**Output:**

****

Q2. Print all the prime numbers in a given 1D array.

**Program:**

/\*Printing the prime numbers in a given 1D array\*/

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

printf("Name : MANOJ M MALLYA\n\n");

int i,n,j,a[100],b[100],flag,k=0;

printf("Enter the number of elements : ");

scanf("%d",&n);

printf("Enter the elements : \n");

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

{

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

}

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

{

flag = 1;

for(j=2;j<=sqrt(a[i]);j++)

{

if(a[i]%j==0)

{

flag = 0;

break;

}

}

if((flag==1)&&(a[i]>1))

{

b[k]=a[i];

k++;

}

}

printf("\nThe prime numbers present in the array are : ");

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

{

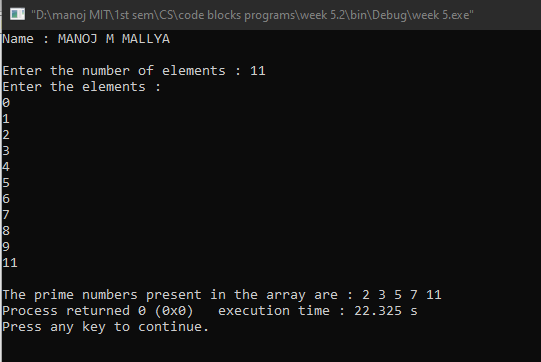
printf("%d ",b[j]);

}

return 0;

}

**Output:**

****

Q3. Arrange the given elements in a 1D array in ascending and descending order using bubble sort method. [Hint: use switch case (as case ‘a’ and case‘d’) to specify the order].

**Program:**

//Arranging the given elements in a 1D array in ascending and descending order using bubble sort and switch statement.

#include <stdio.h>

#include <stdlib.h>

int main()

{

printf("Name : MANOJ M MALLYA\n\n");

int arr[100],i,j,n,temp;

char order;

printf("Enter the number of elements : ");

scanf("%d",&n);

printf("Enter your elements : \n");

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

{

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

}

fflush(stdin);//to clear the input buffer

printf("\nEnter 'a' to sort in ascending order\nEnter 'd' to sort in descending order\n");

printf("Enter your choice : ");

scanf("%c",&order);

switch (order)

{

case 'a':

printf("\nThe ascending order of array elements : ");

for(i=0; i<n-1; i++)

{

for(j=0; j<n-i-1; j++)

{

if(arr[j]>arr[j+1])

{

temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

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

{

printf("%d ",arr[i]);

}

break;

case 'd':

printf("\nThe descending order of array elements : ");

for(i=0; i<n-1; i++)

{

for(j=0; j<n-i-1; j++)

{

if(arr[j]<arr[j+1])

{

temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

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

{

printf("%d ",arr[i]);

}

default:

printf("\nEnter proper choice.\n");

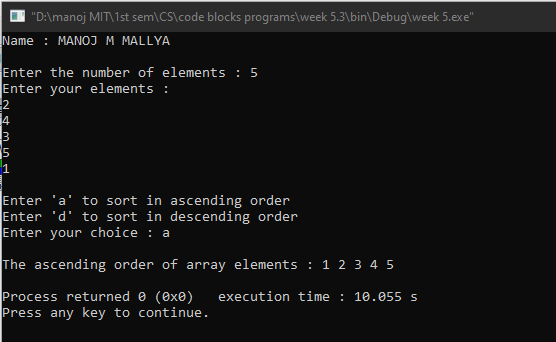
}

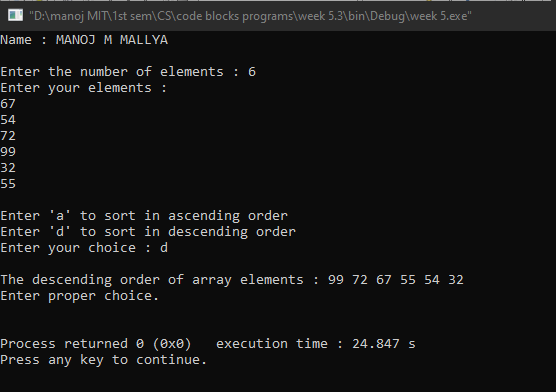
printf("\n");

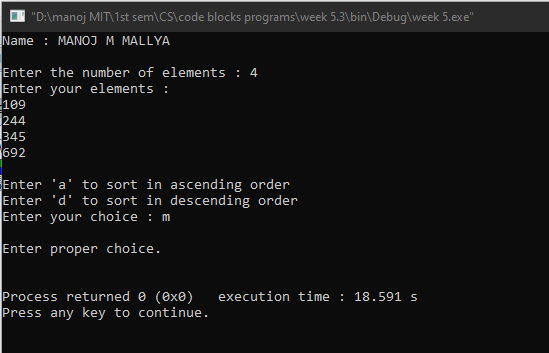
return 0;

}

**Output:**

****

****

****

Q4. Insert an element into a 1D array by getting an element and the position from the user.

**Program:**

//Inserting a new element in a 1D array by getting its value and position

#include <stdio.h>

#include <stdlib.h>

int main()

{

printf("Name : MANOJ M MALLYA\n\n");

int a[100],n,i, pos, ele;

printf("Enter the number of elements in the array : ");

scanf ("%d",&n );//getting number of elements

printf("\nEnter the elements of array : \n");

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

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

printf("\nEnter the element and position of insertion : \n");

scanf("%d %d",&ele,&pos);

for(i=n ; i>=pos ; i--)//shifting the elements to right

{

a[i]=a[i-1];

}

a[pos-1]=ele;//ele is inserted at the specified pos.

n=n+1;//increment the count of no of elements

printf("\n\nThe array after the insertion would be : \n\n");

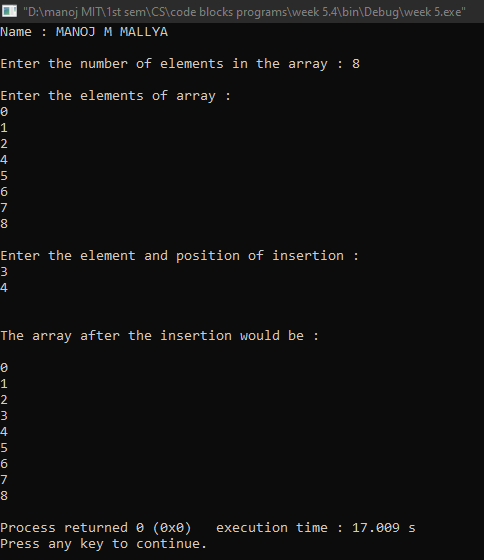
for(i=0;i<n;i++)

printf("%d \n",a[i]);

return 0;

}

**Output:**

****

Q5. Search the position of the number that is entered by the user and delete that number from the array and display the resultant array elements.

**Program:**

/\*Searching the position of the number that is entered by the user and deleting that number

from the array and displaying the resultant array elements\*/

#include <stdio.h>

#include <stdlib.h>

int main()

{

printf("Name : MANOJ M MALLYA\n\n");

int a[100],n,i,val,pos;

printf("Enter the number of elements : ");

scanf("%d",&n);

printf("\nEnter the elements : \n");

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

{

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

}

printf("\nEnter the value of the element to be deleted : ");

scanf("%d",&val);

printf("\nThe position of the %d in the array is ",val);

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

{

if(a[i]==val)

{

printf("%d\n",i+1);

pos = i+1;

}

}

for(i=pos-1;i<n-1;i++)//shifting the elements to left

{

a[i]=a[i+1];

}

n=n-1;

printf("\nThe resultant array would be : ");

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

{

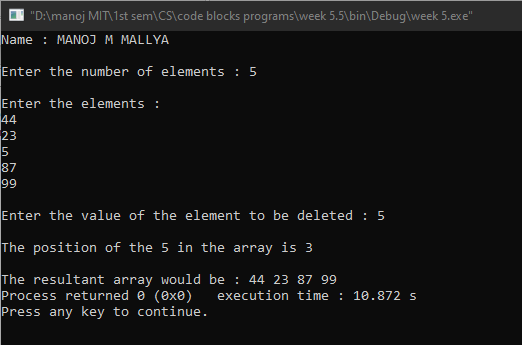
printf("%d ",a[i]);

}

return 0;

}

**Output:**

****

\*\*\*\*\*\*\*\*\*\*

**Lab no: 6 – 2D ARRAYS**

Q1. Find whether a given matrix is symmetric or not. [Hint: A = AT]

**Program:**

//Checking whether a matrix is symmetric or not.

#include <stdio.h>

#include <stdlib.h>

int main()

{

printf("Name : MANOJ M MALLYA\n\n");

int a[100][100],i,j,m,n,flag=1;

printf("Enter the dimension of the matrix : \n");

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

if(m!=n)

{

printf("\n\nIT IS NOT A SQUARE MATRIX => IT CAN NEVER BE A SYMMETRIC MATRIX.\n\n");

}

else

{

printf("\nEnter the elements of the matrix : \n");//getting the matrix

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

{

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

{

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

}

}

printf("\nThe current matrix is : \n");

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

{

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

{

printf("%d ",a[i][j]);

}

printf("\n");

}

//checking whether the matrix is equal to its transpose

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

{

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

{

if (a[i][j]!=a[j][i])

{

flag=0;

break;

}

}

}

if(flag==1)

{

printf("\nIts a symmetric matrix.\n");

}

else

{

printf("\nIts not a symmetric matrix.\n");

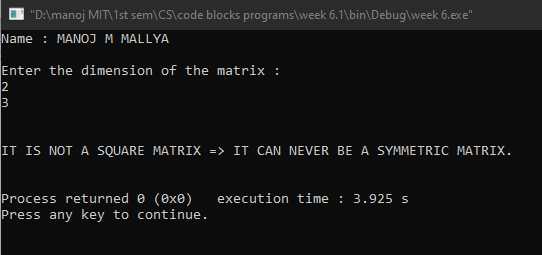
}

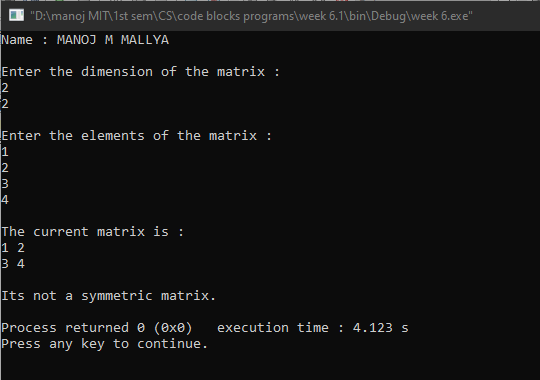
}

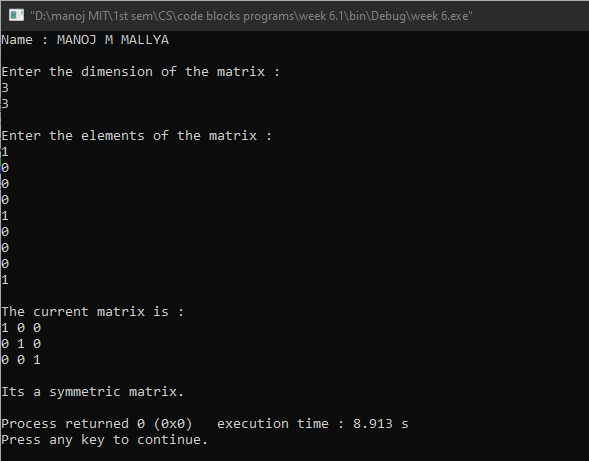
return 0;

}

**Output:**

****

****

****

Q2. Find the trace and norm of a given square matrix. [Hint: Trace= sum of principal diagonal elements Norm= SQRT (sum of squares of the individual elements of an array)]

**Program:**

//Finding trace and norm of a square matrix.

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main()

{

printf("Name : MANOJ M MALLYA\n\n");

int a[100][100],m,n,i,j,trace=0,sum=0;

float norm;

printf("Enter the dimension of the square matrix : \n");

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

if(m!=n)

{

printf("\n\nIT IS NOT A SQUARE MATRIX.\n\n");

}

else

{

printf("\nEnter the elements of the matrix : \n");

for (i=0; i<m; i++) //getting the matrix

{

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

{

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

}

}

printf("\nThe current matrix is : \n");//printing the matrix

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

{

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

{

printf("%d ",a[i][j]);

}

printf("\n");

}

//finding trace and norm

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

{

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

{

sum+=(a[i][j])\*(a[i][j]);

if(i==j)

{

trace+=a[i][j];

}

}

}

norm = sqrt(sum);

printf("\nThe trace of the matrix is %d\n",trace);

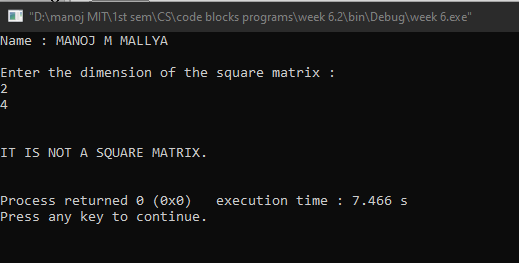
printf("The norm of the matrix is %f\n",norm);

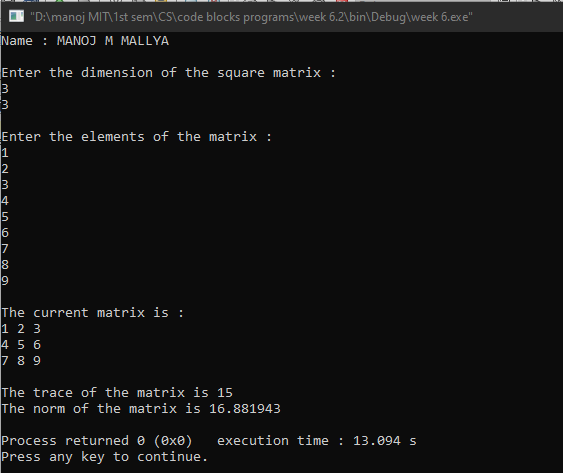
}

return 0;

}

**Output:**





Q3. Perform matrix multiplication.

**Program:**

//Performing matrix multiplication.

#include <stdio.h>

#include <stdlib.h>

int main()

{

printf("Name : MANOJ M MALLYA\n\n");

int a[100][100],b[100][100],c[100][100],i,j,k,m,n,p,q;

printf("Enter the dimensions of the 1st matrix : \n");

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

printf("Enter the dimensions of the 2nd matrix : \n");

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

if(n!=p)//if the condition for matrix multiplication is not satisfied

{

printf("\n\nMATRIX MULTIPLICATION IS NOT DEFINED FOR THESE TWO MATRICES (taken in order).\n\n");

}

else

{

printf("\nEnter the elements of the 1st matrix : \n");

for (i=0; i<m; i++) //getting the 1st matrix

{

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

{

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

}

}

printf("\nEnter the elements of 2nd matrix : \n");

for (i=0; i<p; i++) //getting the 2nd matrix

{

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

{

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

}

}

//Multiplying 2 matrices.This involves 3 nested for loops

for (i=0; i<m; i++) //traverses through every row of 1st matrix

{

for (j=0; j<q; j++) //traverses through every column of 2nd matrix

{

c[i][j]=0;

for(k=0; k<n; k++)//forms the sum of the products of corresponding elements

{

c[i][j]+=a[i][k]\*b[k][j];

}

}

}

//Printing the dimension of resultant matrix which is of the order 'm x q'

printf("\n\nThe matrix obtained after multiplication is : \n\n");

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

{

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

{

printf("%d ",c[i][j]);

}

printf("\n");

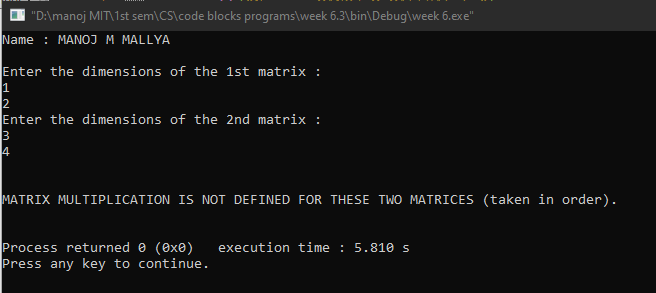
}

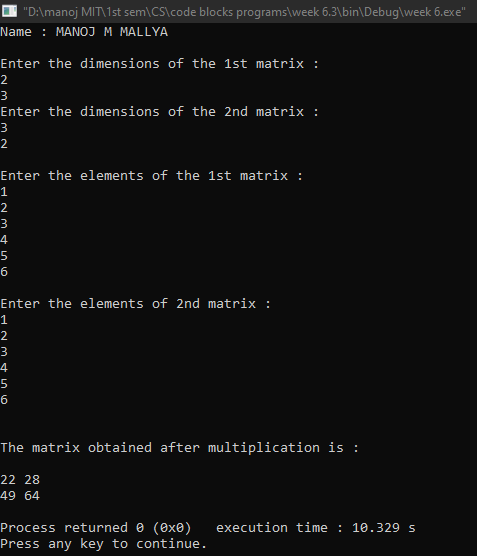
}

return 0;

}

**Output:**





Q4. To interchange the primary and secondary diagonal elements in the given Matrix.

**Program:**

// Interchanging the primary and secondary diagonal elements of a given matrix.

#include <stdio.h>

#include <stdlib.h>

int main()

{

printf("Name : MANOJ M MALLYA\n\n");

int a[100][100],m,n,i,j,temp;

printf("Enter the dimension of the square matrix : \n");

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

if(m!=n)

{

printf("\n\nIT IS NOT A SQUARE MATRIX.\n\n");

}

else

{

printf("\nEnter the elements of the matrix : \n");

for (i=0; i<m; i++) //getting the matrix

{

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

{

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

}

}

printf("\n\nTHE CURRENT MATRIX : \n");

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

{

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

{

printf("%d ",a[i][j]);

}

printf("\n");

}

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

{

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

{

if(i==j)

{

temp = a[i][i];

a[i][i]= a[i][n-i-1];

a[i][n-i-1]=temp;

}

}

}

printf("\n\nTHE MATRIX AFTER REQUIRED MODIFICATION : \n");

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

{

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

{

printf("%d ",a[i][j]);

}

printf("\n");

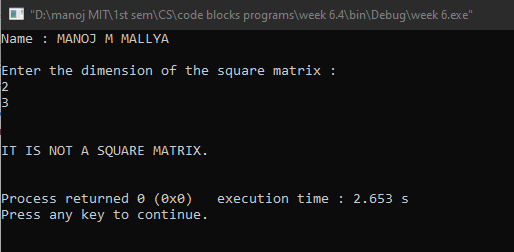
}

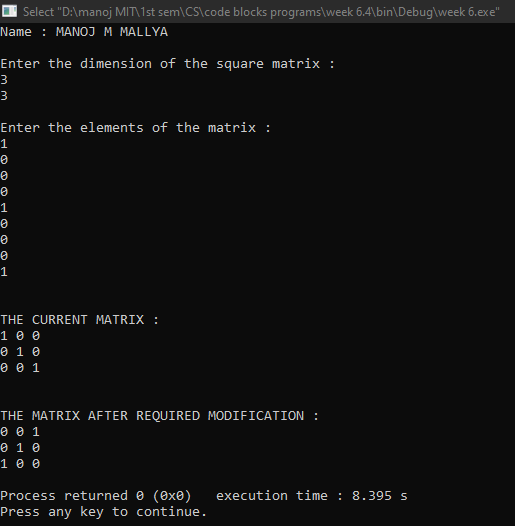
}

return 0;

}

**Output:**

****



Q5. Interchange any two Rows & Columns in the given Matrix.

**Program:**

//Interchanging any two rows & columns of a given matrix.

#include <stdio.h>

#include <stdlib.h>

int main()

{

printf("Name : MANOJ M MALLYA\n\n");

int a[100][100],m,n,i,j,c1,c2,r1,r2,temp;

char choice;

printf("Enter the dimensions of the matrix : \n");

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

printf("\nFill the matrix : \n");

for(i=0; i<m; i++) //getting the matrix

{

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

{

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

}

}

printf("\nThe current matrix is : \n");

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

{

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

{

printf("%d ",a[i][j]);

}

printf("\n");

}

fflush(stdin);

printf("\nEnter 'r' if rows are to be swapped\nEnter 'c' if columns are to be swapped");

printf("\nEnter your choice : ");

scanf("%c",&choice);

switch(choice)

{

case 'r':

printf("\nEnter the rows to be inter changed : ");

scanf("%d %d",&r1,&r2);

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

{

temp=a[r1-1][i];

a[r1-1][i]=a[r2-1][i];

a[r2-1][i]=temp;

}

printf("\nThe modified matrix is : \n");

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

{

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

{

printf("%d ",a[i][j]);

}

printf("\n");

}

break;

case 'c':

printf("\nEnter the columns to be inter changed : ");

scanf("%d %d",&c1,&c2);

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

{

temp=a[i][c1-1];

a[i][c1-1]=a[i][c2-1];

a[i][c2-1]=temp;

}

printf("\nThe modified matrix is : \n");

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

{

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

{

printf("%d ",a[i][j]);

}

printf("\n");

}

break;

default:

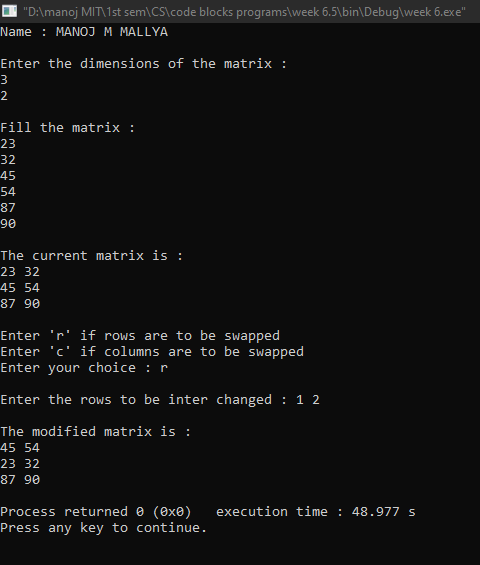
printf("\nEnter proper choice.\n");

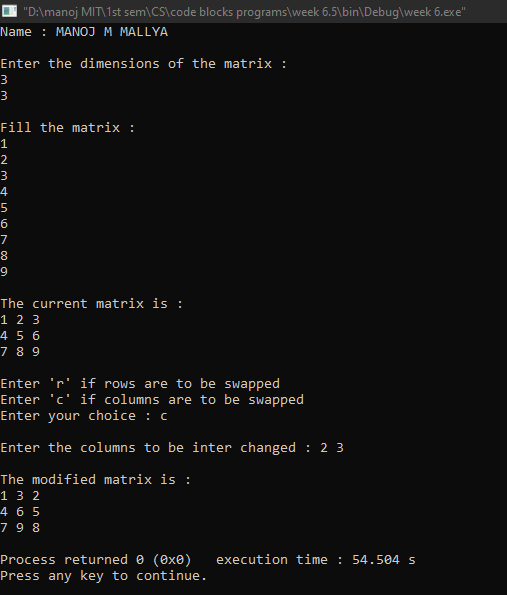
}

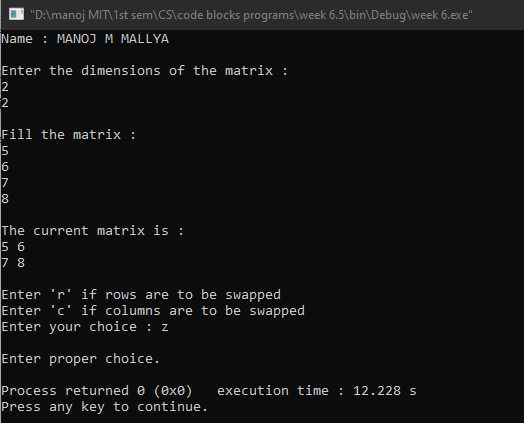
return 0;

}

**Output:**

****

****



Q6. Search for an element in a given matrix and count the number of its occurrences.

**Program:**

//Searching for an element in a given matrix and counting the number of its occurrences.

#include <stdio.h>

#include <stdlib.h>

int main()

{

printf("Name : MANOJ M MALLYA\n\n");

int a[100][100],m,n,i,j,count=0,ele,k=0;

printf("Enter the dimension of the matrix : \n");

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

printf("\nPopulate the matrix : \n");

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

{

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

{

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

}

}

printf("\nThe matrix is : \n");

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

{

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

{

printf("%d ",a[i][j]);

}

printf("\n");

}

printf("\nEnter the required element : ");

scanf("%d",&ele);

printf("\n%d is found in these positions in this matrix : ",ele);

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

{

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

{

if(a[i][j]==ele)

{

printf("(%d,%d) ",i,j);

count++;

}

}

}

if(count==0)

{

printf(" Element not found.\n");

exit(0);

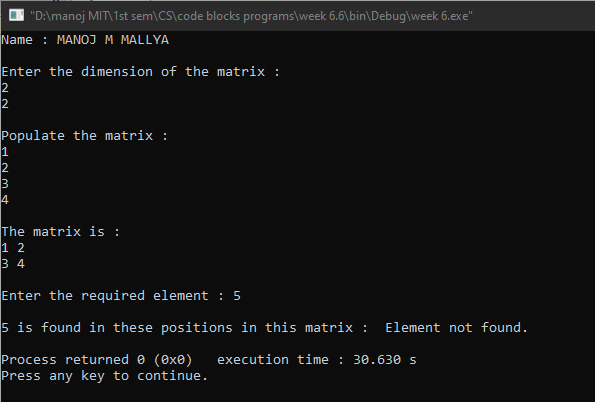
}

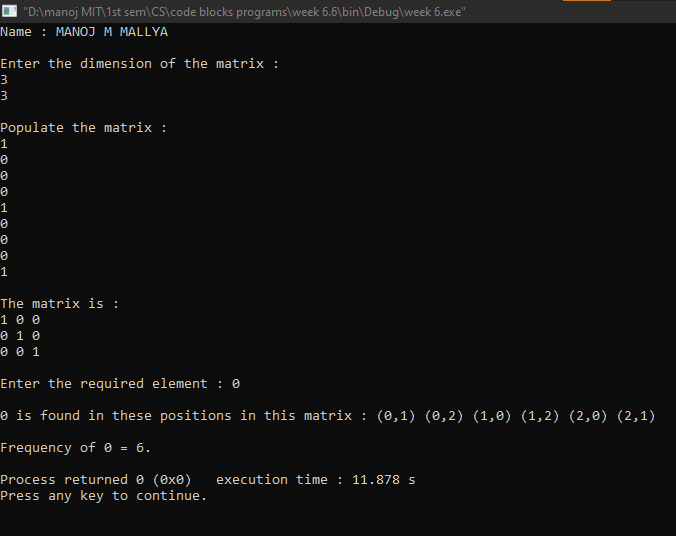
printf("\n\nFrequency of %d = %d. \n",ele,count);

return 0;

}

**Output:**





\*\*\*\*\*\*\*\*\*\*