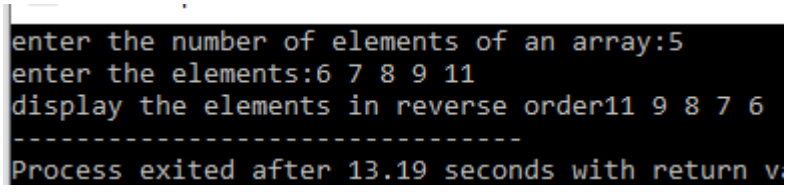
Q1. Read n number of values in an array and display it in reverse order.

PROGRAM:-

```
#include<stdio.h>
int main()
{
    int arr[100];
    int i=0,n;
    printf("enter the number of elements of an array:");
    scanf("%d",&n);
    printf("enter the elements:");
    for(i=0;i<n;i++)
    {
        scanf("%2d",&arr[i]);
    }
    printf("display the elements in reverse order");
    for(i=n-1;i>=0;i--)
    {
        printf("%2d",arr[i]);


    }
    return 0;
}
```

OUTPUT:-

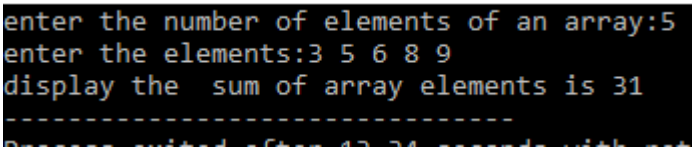


Q2. Find the sum of all elements of the array.

PROGRAM:-

```
#include<stdio.h>
int main()
{
    int arr[100];
    int sum=0,i,n;
    printf("enter the number of elements of an array:");
    scanf("%d",&n);
    printf("enter the elements:");
    for(i=0;i<n;i++)
    {
        scanf("%2d",&arr[i]);
        sum=sum+arr[i];
    }

    printf("display the  sum of array elements is %d",sum);
    return 0;
}
```

OUTPUT:-



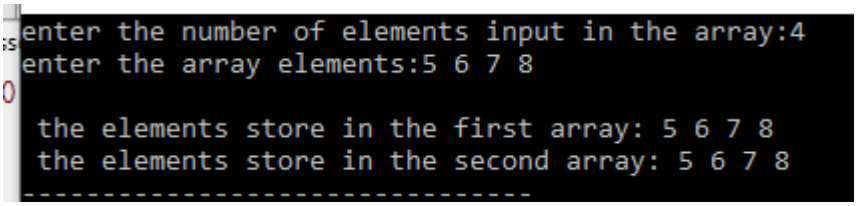
Q3. Copy the elements of one array into another array.

PROGRAM:-

```
#include<stdio.h>
int main()
{
    int arr[5];
```

```
        int c[5];
        int i,n;
        printf("enter the number of elements input in the array:");
        scanf("%d",&n);
        printf("enter the array elements:");
        for(i=0;i<n;i++)
        {
        scanf("%d",&arr[i]);
        }
for(i=0;i<n;i++)
{
        c[i]=arr[i];
}
printf("\n the elements store in the first array:");
for(i=0;i<n;i++)
{
        printf("%2d",arr[i]);
}
printf("\n the elements store in the second array:");
for(i=0;i<n;i++)
{
        printf("%2d",c[i]);
}



        return 0;
}
```

OUTPUT:-



Q4. Count a total number of duplicate elements in an array.

PROGRAM:-

```
#include<stdio.h>
int main()
{
int arr[100];
int i,j,n,count=0;
printf("enter the number of elements input in the array:");
scanf("%d",&n);
printf("enter the array elements:");
for(i=0;i<n;i++)
{
scanf("%d",&arr[i]);
}

for(i=0;i<n;i++)
{
for(j=i+1;j<n;j++)
{
if(arr[i]==arr[j])
{
count++;
break;
```

```
                    }
                }
        }
        printf("Total number of duplicate elements found in array = %d",count);




        return 0;
}
```

OUTPUT:-

```
enter the number of elements input in the array:8
enter the array elements:2 3 4 2 5 6 3 8
Total number of duplicate elements found in array = 2
--------------------------------
Process exited after 25.85 seconds with return value 0
Press any key to continue . . .
```

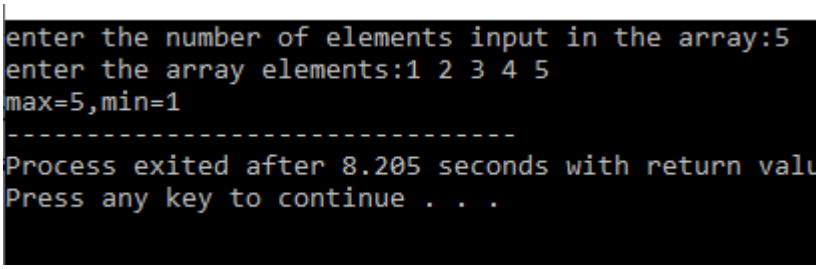
Q5. Find the maximum and minimum element in an array.

PROGRAM:-

```
#include<stdio.h>
int main()
{
        int arr[100];
        int i,n,min,max;
        printf("enter the number of elements input in the array:");
        scanf("%d",&n);
        printf("enter the array elements:");
        for(i=0;i<n;i++)
        {
        scanf("%d",&arr[i]);
```

```
        }
        min=max=arr[0];

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

                        if(min>arr[i])

                        min=arr[i];
                        if(max<arr[i])
                        max=arr[i];

        }
        printf("max=%d,min=%d",max,min);
        return 0;
}
```

OUTPUT:-



Q6. Separate odd and even integers in separate arrays.

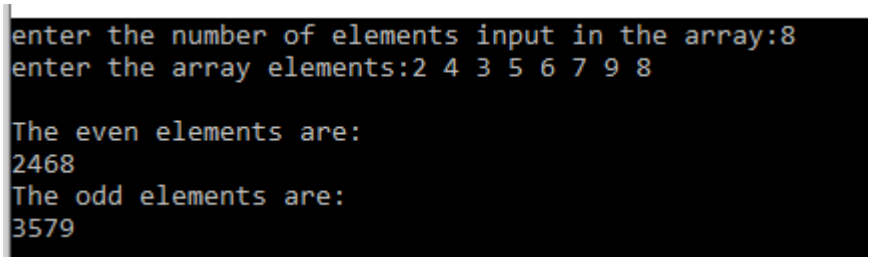
PROGRAM:-

```
#include<stdio.h>
int main()
```

```
{
        int arr[100],even[100],odd[100];
        int i,j=0,k=0,n;
        printf("enter the number of elements input in the array:");
        scanf("%d",&n);
        printf("enter the array elements:");
        for(i=0;i<n;i++)
        {
        scanf("%d",&arr[i]);
        }
        for(i=0;i<n;i++)
        {
                if(arr[i]% 2 == 0)
                {
                        even[j]=arr[i];
                        j++;



                }
                else
                {
                        odd[k]=arr[i];
                        k++;
                }
        }
        printf("\nThe even elements are:\n");
        for(i=0;i<j;i++)
```

```
        {
                printf("%d",even[i]);
        }
        printf("\nThe odd elements are:\n");
        for(i=0;i<k;i++)
        {
                printf("%d",odd[i]);
        }
        return 0;


}
```

OUTPUT:-



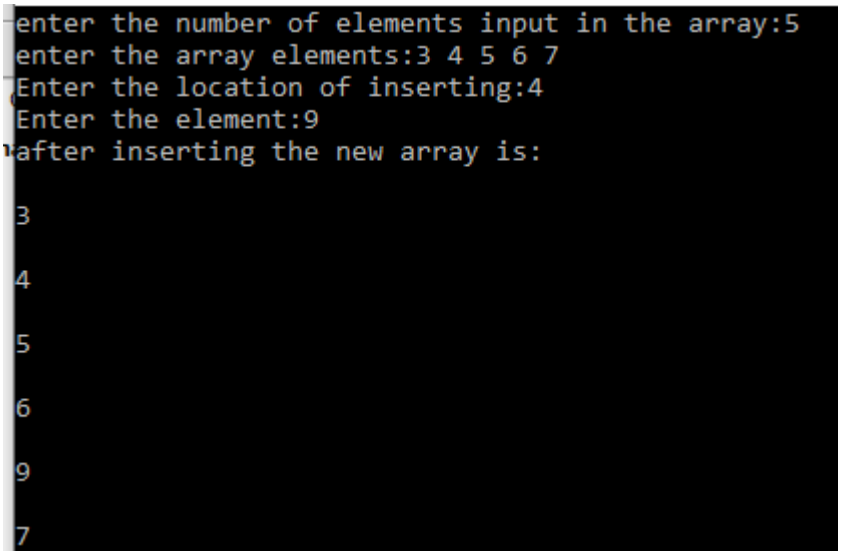
Q7.Insert New value in the array.

PROGRAM:- #include<stdio.h>

```
int main()
{
        int arr[100];
        int i,n,pos,ele;
        printf("enter the number of elements input in the array:");
        scanf("%d",&n);
        printf("enter the array elements:");
```

```
	for(i=0;i<n;i++)
	{
	scanf("%d",&arr[i]);
	}
	printf("Enter the location of inserting:");
	scanf("%d",&pos);
	printf("Enter the element:");
	scanf("%d",&ele);
	for(i=n-1;i>=pos;i--)
	{
		arr[i+1]=arr[i];
	}
	arr[pos]=ele;
	printf("after inserting the new array is:\n");
	for(i=0;i<n+1;i++)
	{
		printf(" \n%d\n",arr[i]);
	}




	return 0;
}
```

OUTPUT:-



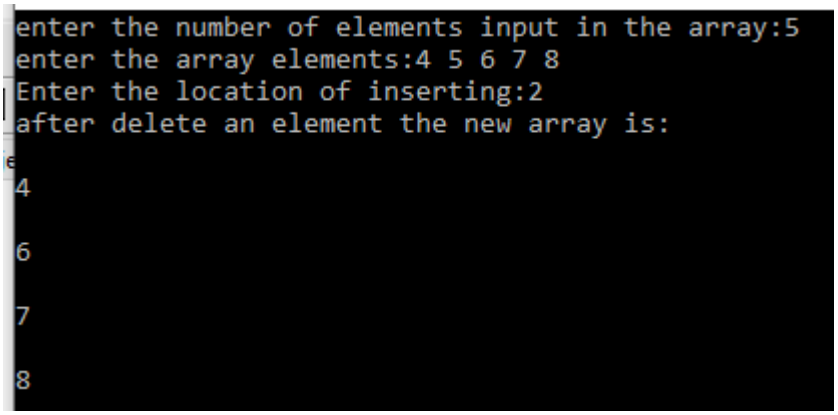
Q8. Delete an element at desired position from an array.

PROGRAM:-

```
#include<stdio.h>
int main()
{
    int arr[100];
    int i,n,pos,ele;
    printf("enter the number of elements input in the array:");
    scanf("%d",&n);
    printf("enter the array elements:");
    for(i=0;i<n;i++)
    {
    scanf("%d",&arr[i]);
    }
    printf("Enter the location of inserting:");
    scanf("%d",&pos);
    if(pos<0 || pos>n)
    {
```

```
            printf("Invalid Postion!please enter position between 1 to %d",n);
        }
        for(i=pos-1;i<n-1;i++)
        {
            arr[i]=arr[i+1];
        }
        n--;

        printf("after delete an element the new array is:\n");
        for(i=0;i<n;i++)
        {
            printf(" \n%d\n",arr[i]);
        }




        return 0;


}
```

OUTPUT:-



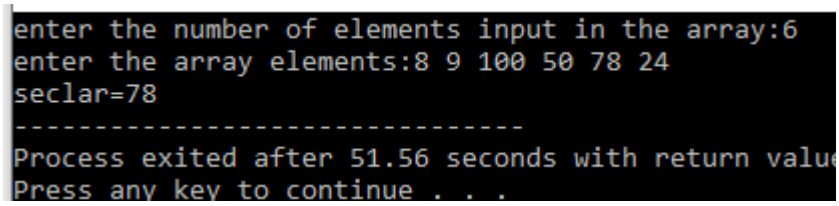
Q9. find the second largest element in an array.

PROGRAM:-

```
#include<stdio.h>
int main()
{
    int arr[100];
    int i,n,lar,seclar;
    printf("enter the number of elements input in the array:");
    scanf("%d",&n);
    printf("enter the array elements:");
    for(i=0;i<n;i++)
    {
    scanf("%d",&arr[i]);
    }
    lar=arr[0];
    seclar=arr[1];
    for(i=0;i<n;i++)
    {
        if(arr[i]>lar)
        {
            seclar=lar;
```

```
                lar=arr[i];
            }
            else if(arr[i]>seclar && arr[i]!=lar)
            {
                seclar=arr[i];
            }
        }
        printf("seclar=%d",seclar);
        return 0;
        }
```

OUTPUT:-



Q10. Find the median of two sorted arrays of same size.

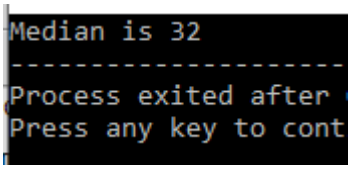
PROGRAM:-

```
#include <stdio.h>
int findMedian(int ar1[], int ar2[], int n)
{
   int i = 0;
   int j = 0;
   int count;
   int m1 = -1, m2 = -1;


   for (count = 0; count <= n; count++)
```

```
{

    if (i == n)
    {
        m1 = m2;
        m2 = ar2[0];
        break;
    }


    else if (j == n)
    {
        m1 = m2;
        m2 = ar1[0];
        break;
    }

    if (ar1[i] <= ar2[j])
    {
        m1 = m2;
        m2 = ar1[i];
        i++;
    }
    else
    {
        m1 = m2;
        m2 = ar2[j];
```

```
            j++;
        }
    }

    return (m1 + m2)/2;
}
int main()
{
    int ar1[] = {10, 20, 30, 40, 50};
    int ar2[] = {12, 22, 34, 46, 58};

    int n1 = sizeof(ar1)/sizeof(ar1[0]);
    int n2 = sizeof(ar2)/sizeof(ar2[0]);
    if (n1 == n2)
        printf("Median is %d", findMedian(ar1, ar2, n1));
    else
        printf("Doesn't work for arrays of unequal size");

    return 0;
}
```

OUTPUT:-



Q11. . multiplication of two square Matrices.

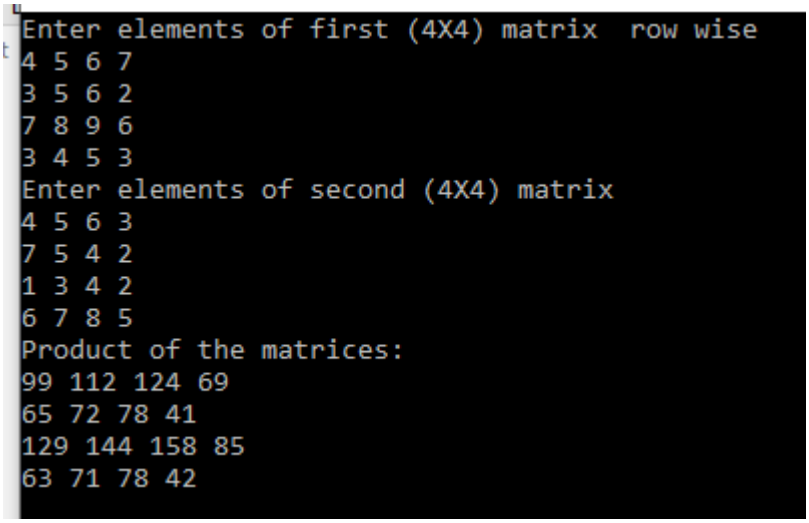
PROGRAM:-

```
#include <stdio.h>
```

```
int main()
{
  int m, n, p, q, i, j, k, sum = 0;
  int mat1[4][4], mat2[4][4], multiply[4][4];

  printf("Enter elements of first (4X4) matrix  row wise\n");

  for (i = 0; i < 4; i++)
    for (j = 0; j < 4; j++)
      scanf("%d", &mat1[i][j]);

    printf("Enter elements of second (4X4) matrix\n");

    for (i = 0; i < 4; i++)
      for (j = 0; j < 4; j++)
        scanf("%d", &mat2[i][j]);

    for (i = 0; i < 4; i++) {
      for (j = 0; j < 4; j++) {
        for (k = 0; k < 4; k++) {
          sum = sum + mat1[i][k]*mat2[k][j];
        }

        multiply[i][j] = sum;
        sum = 0;
      } }
```

```
 printf("Product of the matrices:\n");
 for (i = 0; i < 4; i++) {
     for (j = 0; j < 4; j++)
       printf("%d ", multiply[i][j]);


     printf("\n");
   }
 return 0;
}
```

OUTPUT:-



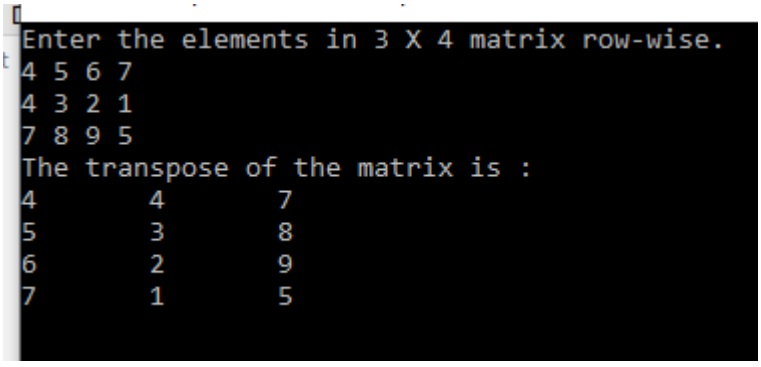
Q12. find transpose of a given matrix.

PROGRAM:

```
#include<stdio.h>
#define ROW 3
#define COL 4
int main(){
int mat1[ROW][COL],mat2[COL][ROW],i,j;
printf("Enter the elements in 3 X 4 matrix row-wise.\n");
for(i=0;i<ROW;i++){
```

```
    for(j=0;j<COL;j++){
    scanf("%d",&mat1[i][j]);
    }
}
for(i=0;i<ROW;i++){
    for(j=0;j<COL;j++){
    mat2[j][i]=mat1[i][j];
    }
}
printf("The transpose of the matrix is : \n");
for(i=0;i<COL;i++){
    for(j=0;j<ROW;j++){
    printf("%d\t",mat2[i][j]);
    }
    printf("\n");
}
return 0;
}
```

OUTPUT:-

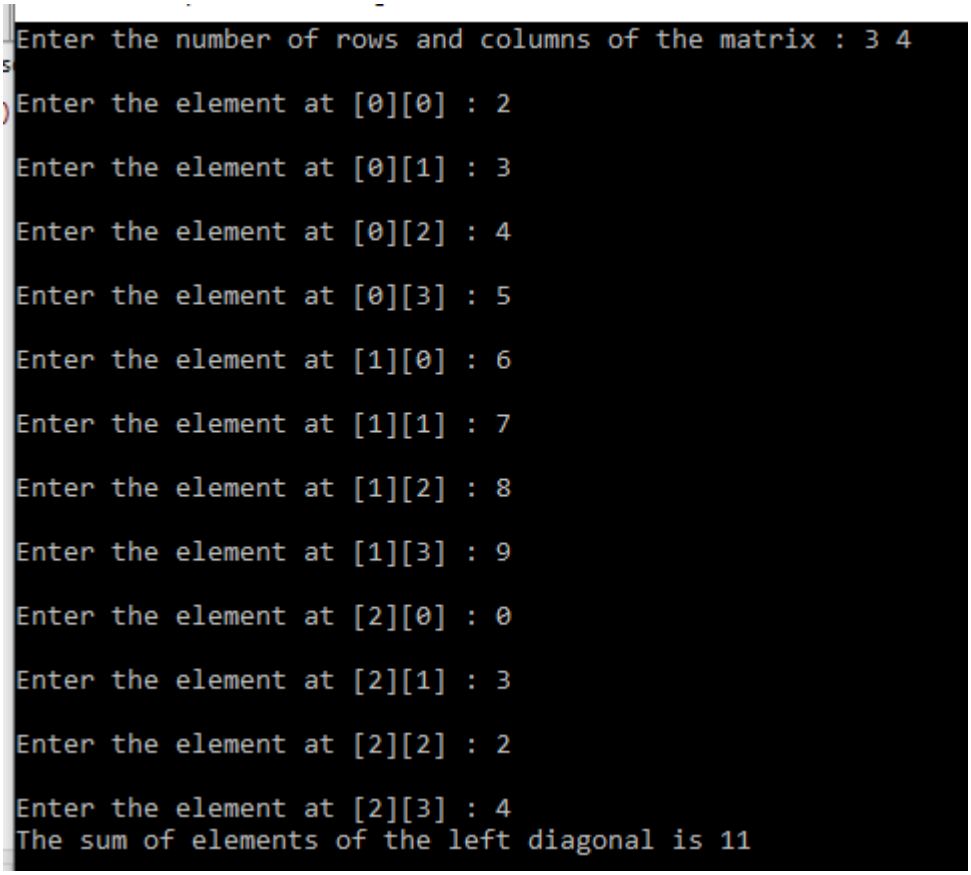


Q13. find the sum of left diagonals of a matrix.

PROGRAM:-

```
#include <stdio.h>
int main(){
int mat[100][100],i,j,sum=0,x,y;
printf("Enter the number of rows and columns of the matrix : ");
scanf("%d %d",&x,&y);
for(i=0;i<x;i++){
   for(j=0;j<y;j++){
      printf("\nEnter the element at [%d][%d] : ",i,j);
      scanf("%d",&mat[i][j]);
      if(i==j){
         sum+=mat[i][j];
      }
   }
}
printf("The sum of elements of the left diagonal is %d ",sum);
return 0;

}
```

OUTPUT:-



Q14. check whether a given matrix is an identity matrix.

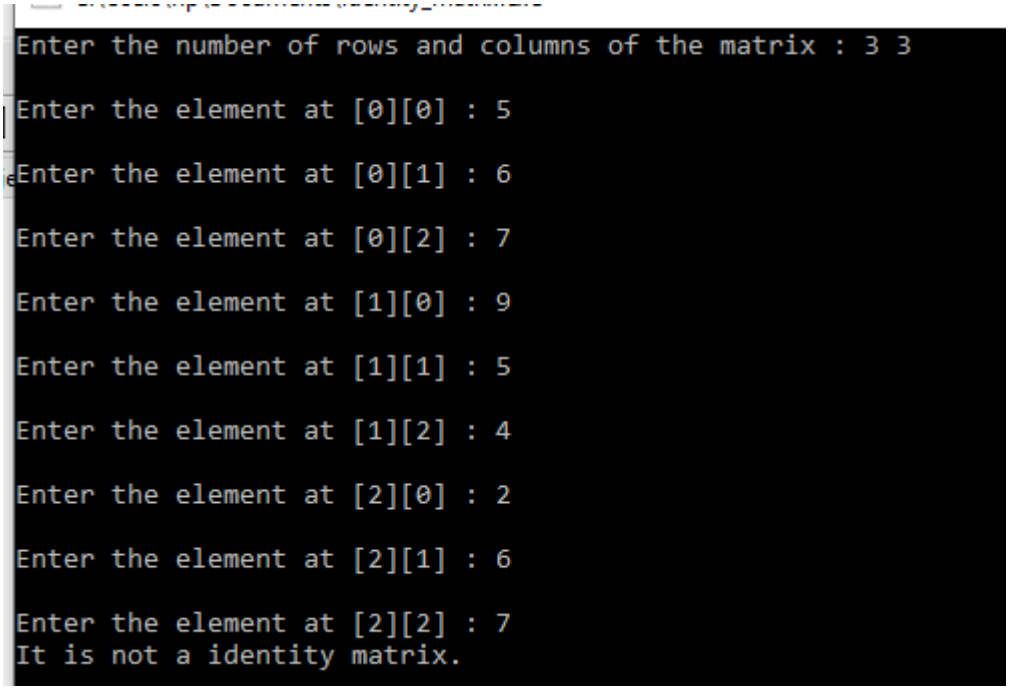
PROGRAM:-

```
#include <stdio.h>
int main(){
int mat[100][100],i,j,x,y;
printf("Enter the number of rows and columns of the matrix : ");
scanf("%d %d",&x,&y);

if(x==y){
   for(i=0;i<x;i++){
      for(j=0;j<x;j++){
        printf("\nEnter the element at [%d][%d] : ",i,j);
        scanf("%d",&mat[i][j]);
```

```
        }
    }
     for(i=0;i<x;i++){
        for(j=0;j<x;j++){
          if(i==j){
            if(mat[i][j]!=1){
                printf("It is not a identity matrix.");
                return 0;
            }
          }
          else{
            if(mat[i][j]!=0){
                printf("It is not a identity matrix.");
                return 0;
          }
        }
    }


}
printf("It is a identity matrix.");
return 0;
}
else{
    printf("The number of rows and columns must be same in an identity matrix.");
    return 0;
}
```

}

OUTPUT:-



Q15. . search an element in a row wise and column wise sorted matrix.

PROGRAM:-

```
#include<stdio.h>

int main(){

int mat[5][5]={{10,20,30,40,50},

        {11,22,33,44,55},

        {12,23,34,45,56},

        {13,24,35,46,57},

        {14,25,36,47,58}};

int x,y=0,i,j;

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

for(i=0;i<5;i++){

for(j=0;j<5;j++){

printf("%d\t",mat[i][j]);

}
```

```
printf("\n");

}

printf("Enter the element to be searched : ");

scanf("%d",&x);

for(i=0;i<5;i++){

   for(j=0;j<5;j++){

      if(x==mat[i][j]){

         printf("%d is found at position [%d][%d]\n",x,i,j);

         y++;

      }

   }

}

if(y==0){

   printf("%d is not found in the matrix",x);

}

return 0;

}
```

OUTPUT:-