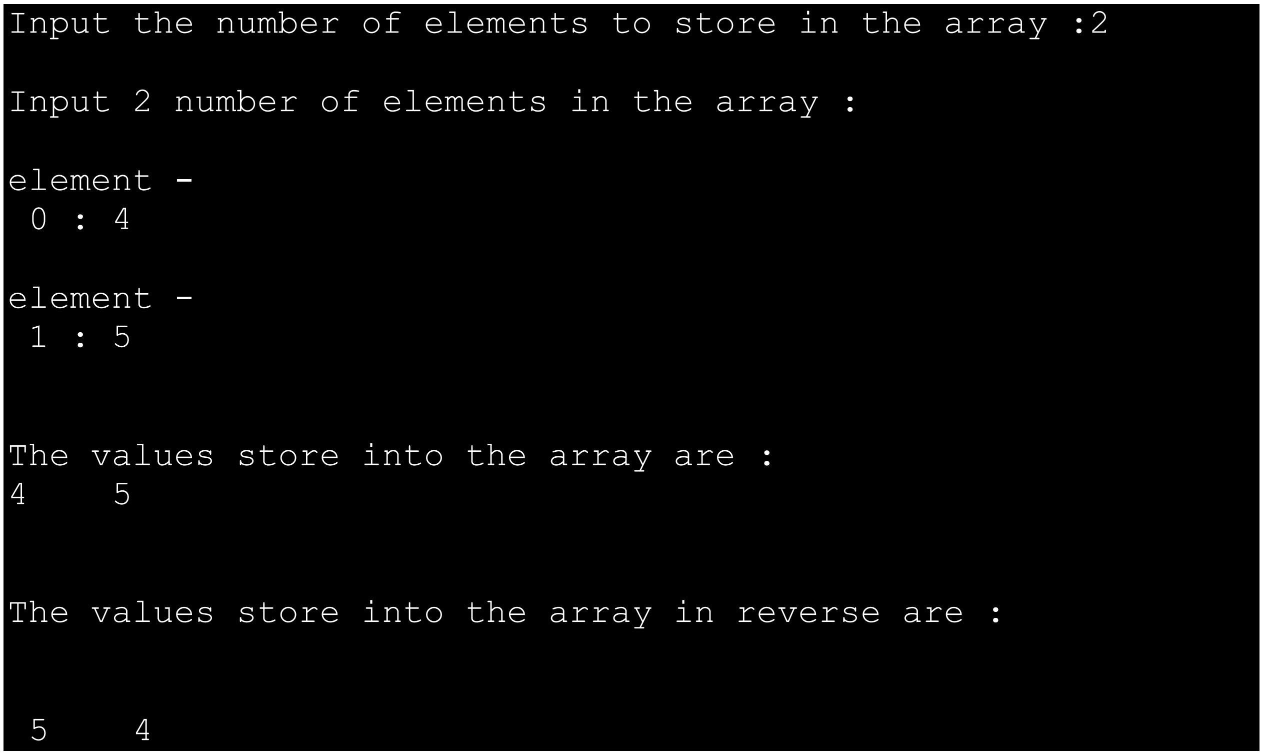
Assingment-7

Ankit Kumar patel

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

```
#include <stdio.h>

void main()
{
  int i,n,a[100];

  printf("Input the number of elements to store in the array :");
  scanf("%d",&n);

  printf("Input %d number of elements in the array :\n",n);
  for(i=0;i<n;i++)
    {
        printf("element - %d : ",i);
        scanf("%d",&a[i]);
        }

  printf("\nThe values store into the array are : \n");
  for(i=0;i<n;i++)
   {
         printf("% 5d",a[i]);
       }
  printf("\n\nThe values store into the array in reverse are :\n");
  for(i=n-1;i>=0;i--)
    {
         printf("% 5d",a[i]);
        }
  printf("\n\n");
}
```

Output:

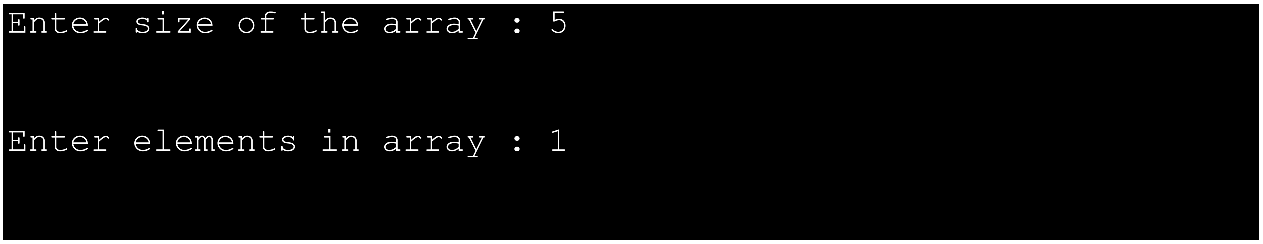


```
Input the number of elements to store in the array :2

Input 2 number of elements in the array :

element -
 0 : 4

element -
 1 : 5


The values store into the array are :
4    5


The values store into the array in reverse are :


 5    4
```

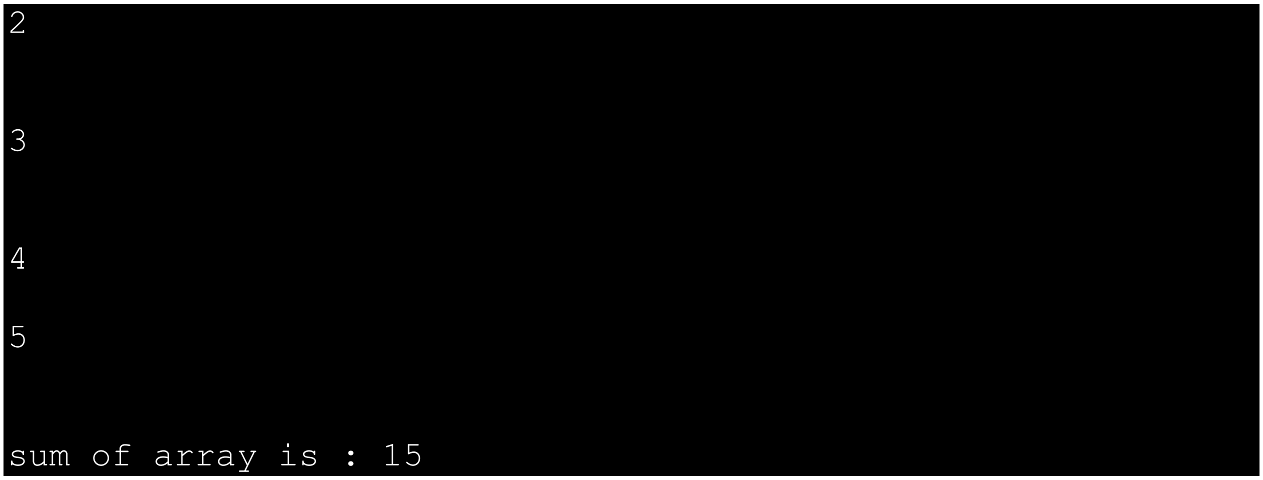
2. Find the sum of all elements of the array.

```
#include <conio.h>

int main()

{
    int a[1000],i,n,sum=0;
    printf("Enter size of the array : ");
    scanf("%d",&n);

    printf("Enter elements in array : ");
    for(i=0; i<n; i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=0; i<n; i++)
    {
        sum+=a[i];
    }
     printf("sum of array is : %d",sum);
    return 0;
}
```

Output:



```
Enter size of the array : 5


Enter elements in array : 1
```



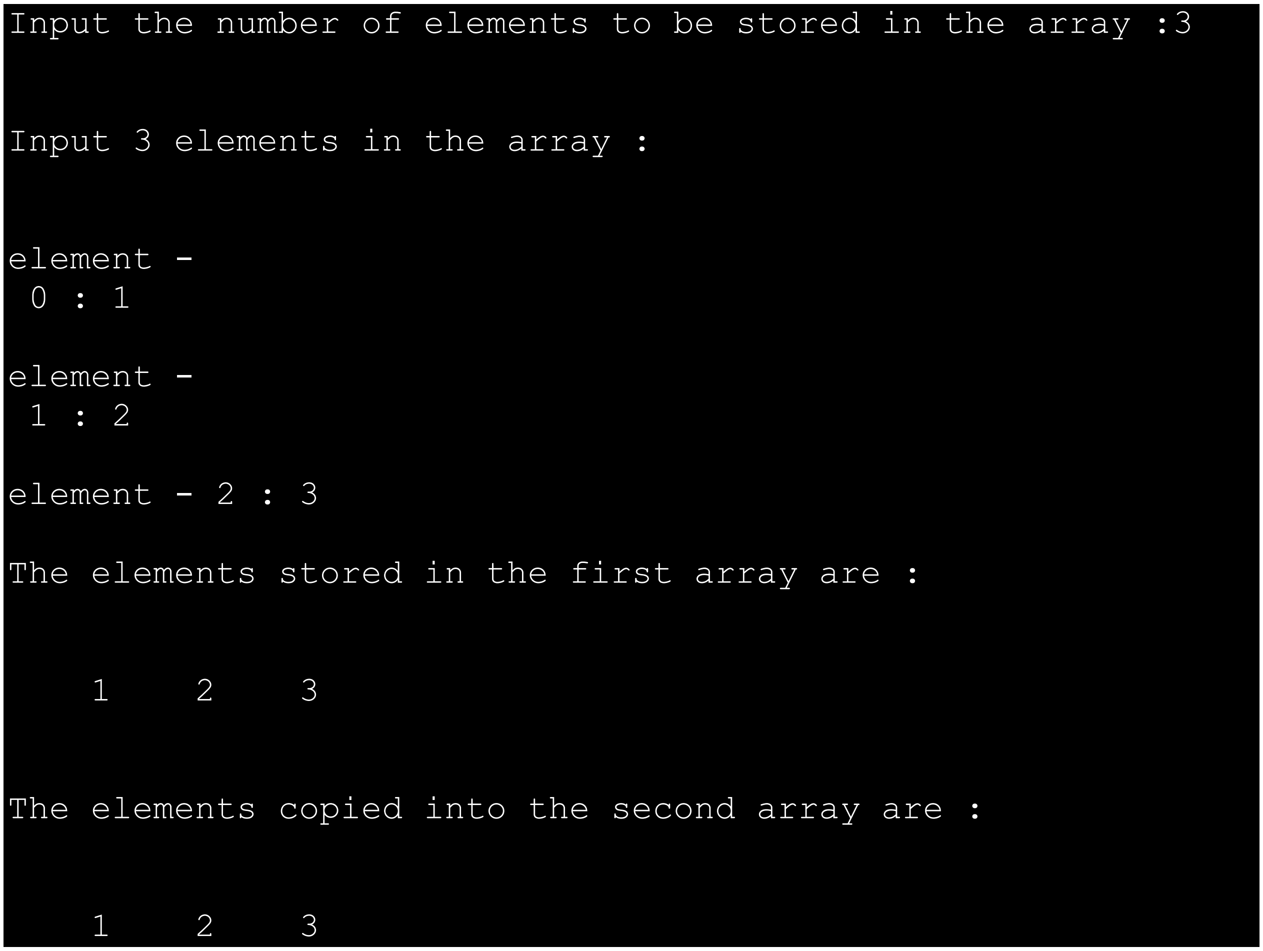
```
2


3


4

5


sum of array is : 15
```

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

```
#include <stdio.h>

void main()

{

   int arr1[100], arr2[100];

   int i, n;

      printf("Input the number of elements to be stored in the array :");

      scanf("%d",&n);

      printf("Input %d elements in the array :\n",n);

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

       {

             printf("element - %d : ",i);

             scanf("%d",&arr1[i]);
```

```
            }

    for(i=0; i<n; i++)
    {
        arr2[i] = arr1[i];
    }
    printf("\nThe elements stored in the first array are :\n");
    for(i=0; i<n; i++)
    {
        printf("% 5d", arr1[i]);
    }
    printf("\n\nThe elements copied into the second array are :\n");
    for(i=0; i<n; i++)
    {
        printf("% 5d", arr2[i]);
    }
                printf("\n\n");
}
```

Output:



```
Input the number of elements to be stored in the array :3


Input 3 elements in the array :


element -
 0 : 1

element -
 1 : 2

element - 2 : 3

The elements stored in the first array are :


    1    2    3


The elements copied into the second array are :


    1    2    3
```

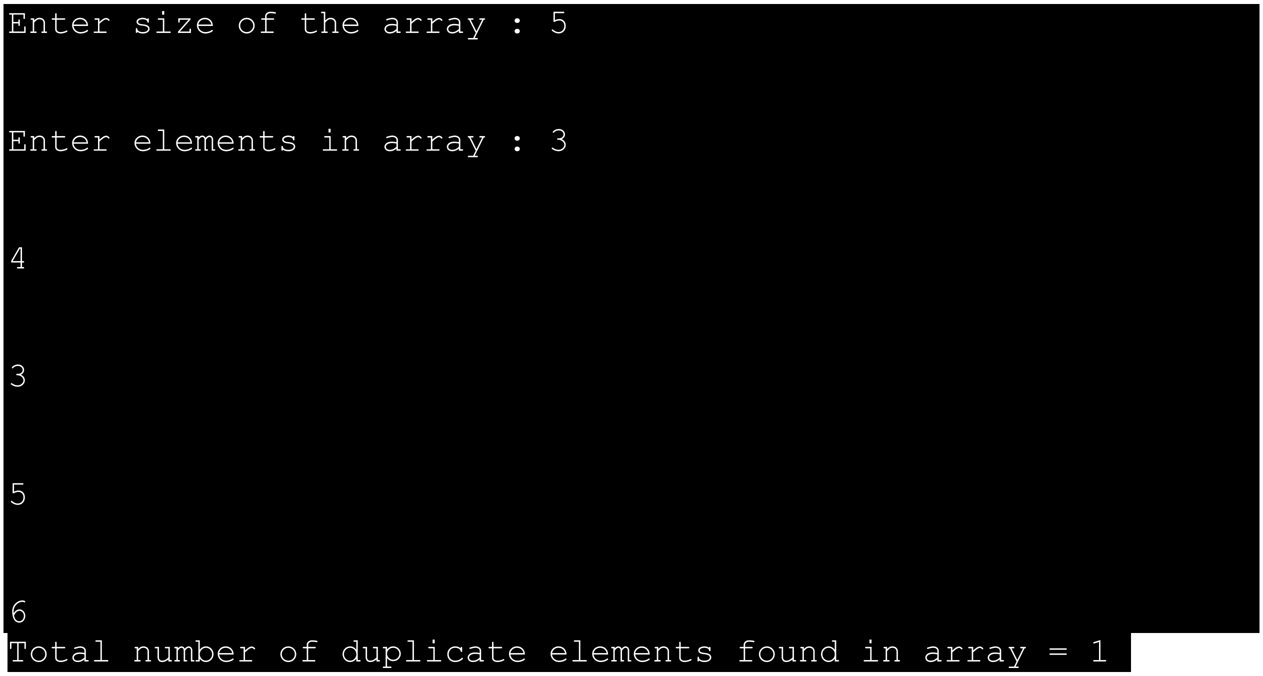
4. Count a total number of duplicate elements in an array.

```
#include <stdio.h>


#define MAX_SIZE 100

int main()

{

   int arr[MAX_SIZE];

   int i, j, size, count = 0;
```

```
printf("Enter size of the array : ");

scanf("%d", &size);



printf("Enter elements in array : ");

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

{

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

}



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

{

    for(j=i+1; j<size; j++)

    {

        if(arr[i] == arr[j])

        {

            count++;
```

```
                break;
            }
        }
    }

    printf("\nTotal number of duplicate elements found in array = %d", count);

    return 0;
}
```

Output:



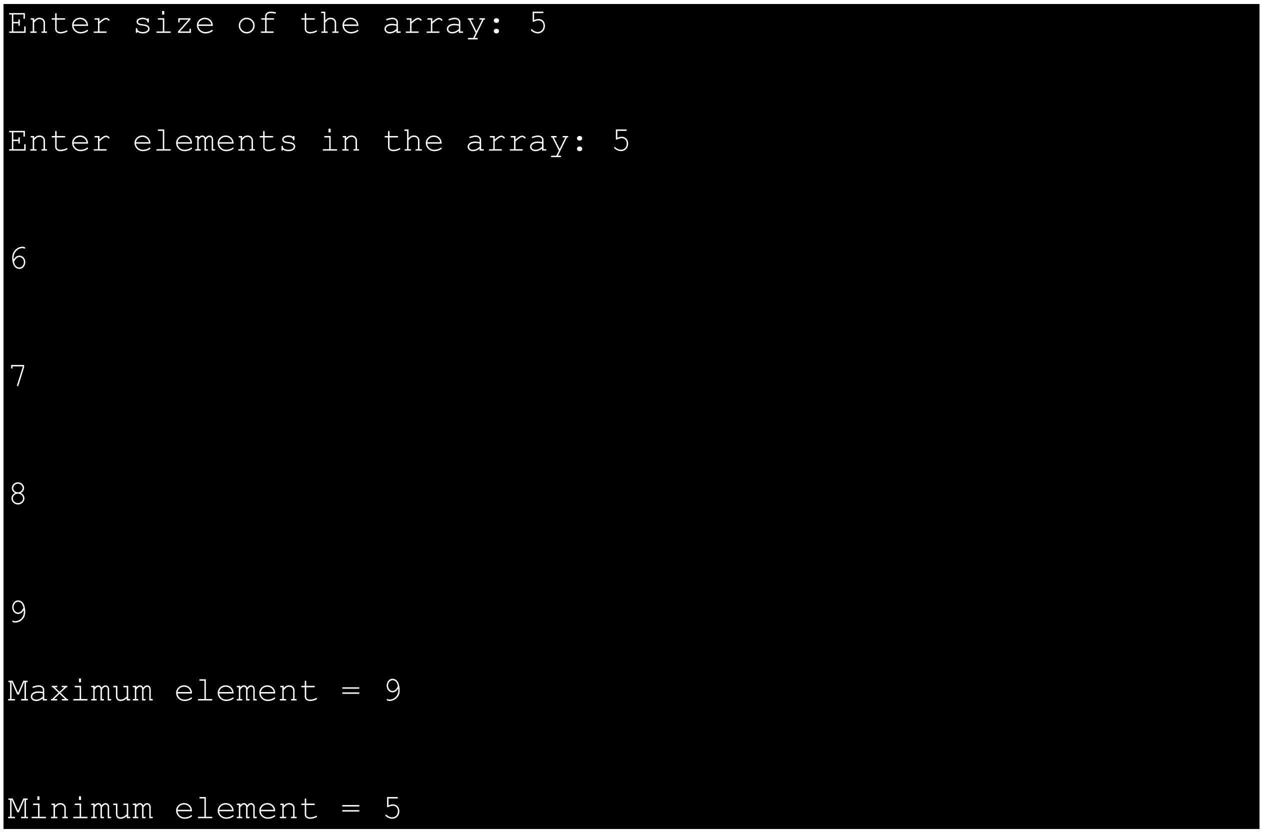
```
Enter size of the array : 5

Enter elements in array : 3

4

3

5

6
Total number of duplicate elements found in array = 1
```

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

```
#include <stdio.h>


#define MAX_SIZE 100

int main()

{

   int arr[MAX_SIZE];

   int i, max, min, size;


   printf("Enter size of the array: ");

   scanf("%d", &size);


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

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

   {

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

   }


   max = arr[0];

   min = arr[0];
```

```
    for(i=1; i<size; i++)
    {

        if(arr[i] > max)
        {
            max = arr[i];
        }

        if(arr[i] < min)
        {
            min = arr[i];
        }
    }
    printf("Maximum element = %d\n", max);
    printf("Minimum element = %d", min);

    return 0;
}
```

Output:

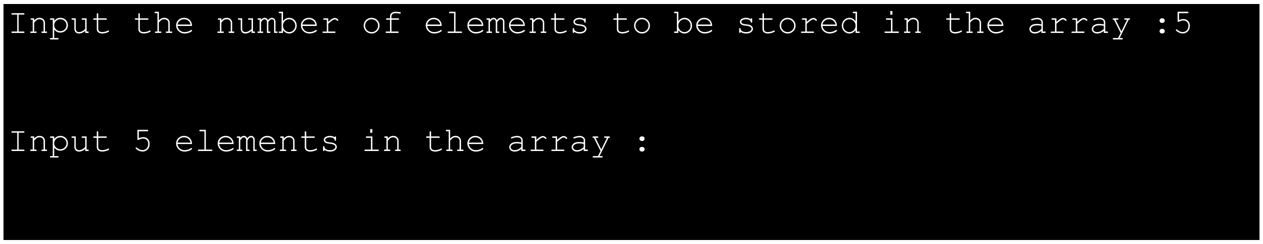


```
Enter size of the array: 5

Enter elements in the array: 5

6

7

8

9
Maximum element = 9

Minimum element = 5
```

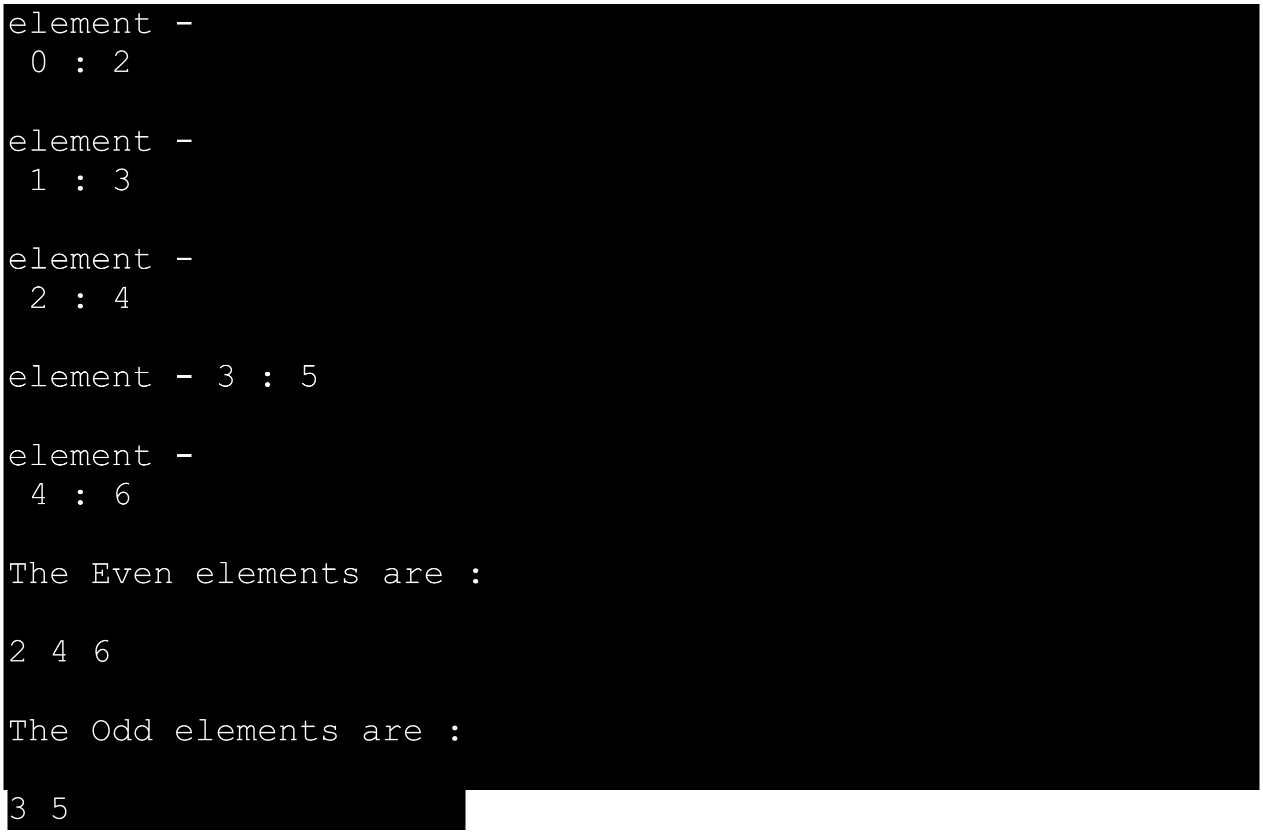
6. Separate odd and even integers in separate arrays.

```
#include <stdio.h>


void main()
 {
   int arr1[10], arr2[10], arr3[10];
   int i,j=0,k=0,n;


      printf("Input the number of elements to be stored in the array :");
   scanf("%d",&n);

   printf("Input %d elements in the array :\n",n);
   for(i=0;i<n;i++)
        {
            printf("element - %d : ",i);
            scanf("%d",&arr1[i]);
          }

for(i=0;i<n;i++)
{
      if (arr1[i]%2 == 0)
      {
         arr2[j] = arr1[i];
         j++;
      }
      else
      {
         arr3[k] = arr1[i];
         k++;
        }
    }

    printf("\nThe Even elements are : \n");
    for(i=0;i<j;i++)
    {
        printf("%d ",arr2[i]);
    }

    printf("\nThe Odd elements are :\n");
    for(i=0;i<k;i++)
    {
        printf("%d ", arr3[i]);
    }
    printf("\n\n");
 }
```

Output:



```
Input the number of elements to be stored in the array :5


Input 5 elements in the array :
```



```
element -
 0 : 2

element -
 1 : 3

element -
 2 : 4

element - 3 : 5

element -
 4 : 6

The Even elements are :

2 4 6

The Odd elements are :

3 5
```

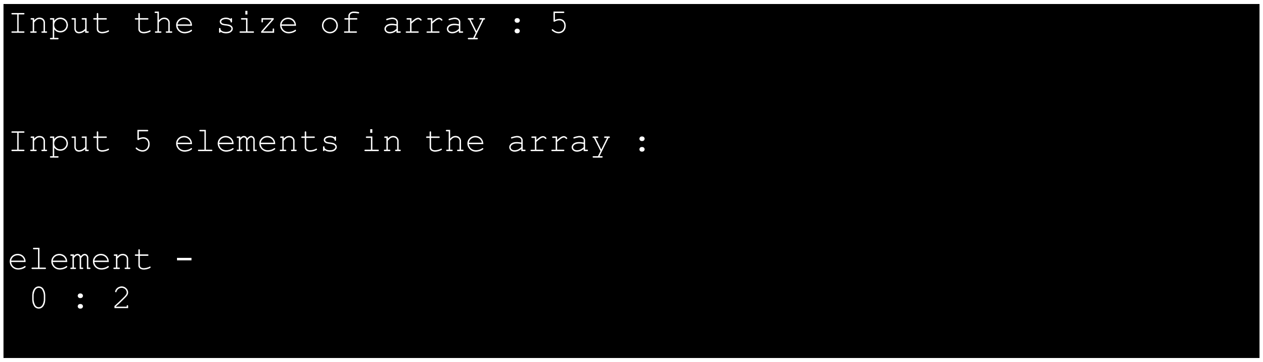
9. Find the second largest element in an array.

```
#include <stdio.h>


void main(){
  int arr1[50],n,i,j=0,lrg,lrg2nd;
       printf("Input the size of array : ");
       scanf("%d", &n);


       printf("Input %d elements in the array :\n",n);
```

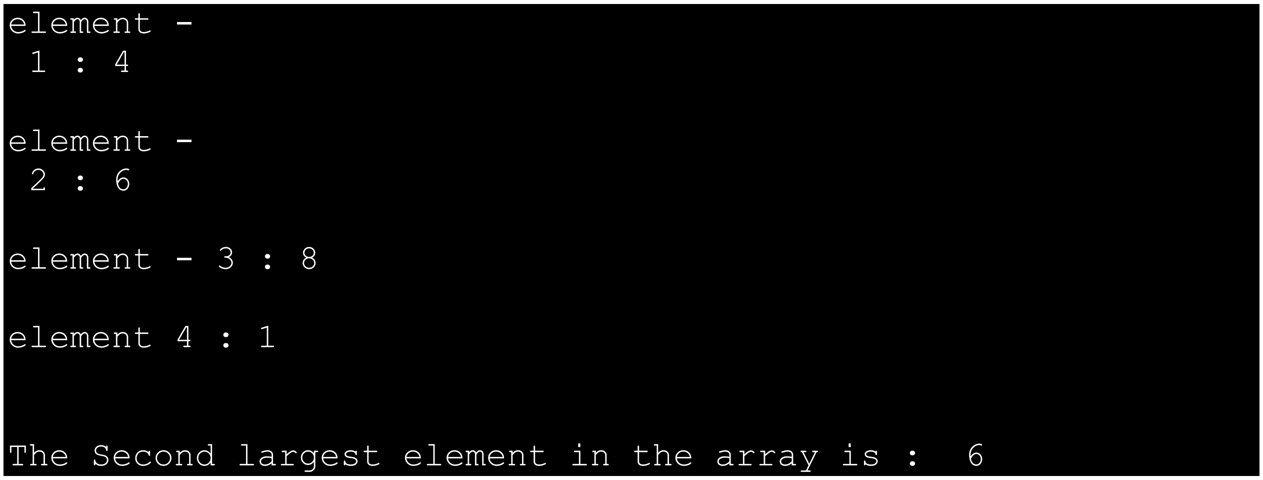
```
     for(i=0;i<n;i++)
          {
              printf("element - %d : ",i);
              scanf("%d",&arr1[i]);
            }
 lrg=0;
for(i=0;i<n;i++)
{
    if(lrg<arr1[i])
          {
         lrg=arr1[i];
         j = i;
    }
}

 lrg2nd=0;
for(i=0;i<n;i++)
{
   if(i==j)
      {
```

```
      i++;

            i--;

     }

   else

     {

      if(lrg2nd<arr1[i])

         {

         lrg2nd=arr1[i];

        }

     }

  }


  printf("The Second largest element in the array is :  %d \n\n", lrg2nd);

}
```

Output:



```
Input the size of array : 5


Input 5 elements in the array :


element -
 0 : 2
```



```
element -
 1 : 4

element -
 2 : 6

element - 3 : 8

element 4 : 1


The Second largest element in the array is :  6
```

11. Multiplication of two square Matrices.

```
#include <stdio.h>


int main()
{
  int m, n, p, q, c, d, k, sum = 0;
  int first[10][10], second[10][10], multiply[10][10];


  printf("Enter the number of rows and columns of first matrix\n");
  scanf("%d%d", &m, &n);
  printf("Enter the elements of first matrix\n");
  for (  c = 0 ; c < m ; c++ )
   for ( d = 0 ; d < n ; d++ )
    scanf("%d", &first[c][d]);

  printf("Enter the number of rows and columns of second matrix\n");
  scanf("%d%d", &p, &q);

  if ( n != p )
   printf("Matrices with entered orders can't be multiplied with each
 other.\n");
  else
  {
   printf("Enter the elements of second matrix\n");

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

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

```
  for ( d = 0 ; d < q ; d++ )
  {
    for ( k = 0 ; k < p ; k++ )
    {
      sum = sum + first[c][k]*second[k][d];
    }

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

printf("Product of entered matrices:-\n");

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

  printf("\n");
```

```
    }

  }


  return 0;

}
```

Output:

```
Enter the number of rows and columns of first matrix 3
3
Enter the elements of first matrix
1 2 0
0 1 1
2 0 1
Enter the number of rows and columns of second matrix 3
3
Enter the elements of second matrix
1 1 2
2 1 1
1 2 1
Product of entered matrices:-
5     3     4
3     3     2
3     4     5
```

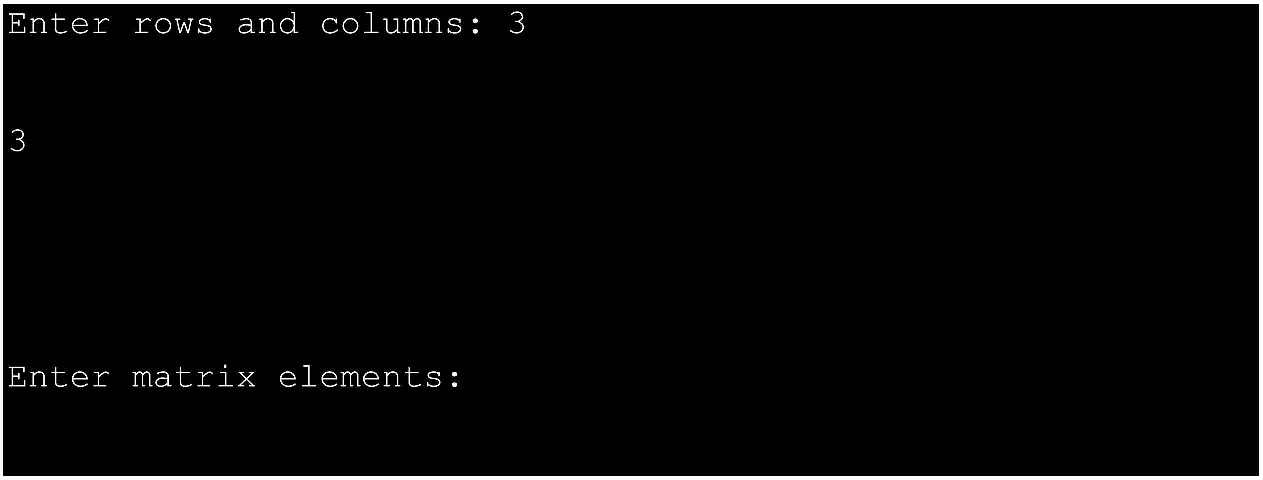
12. Find transpose of a given matrix.

```
#include <stdio.h>

int main()

{
```

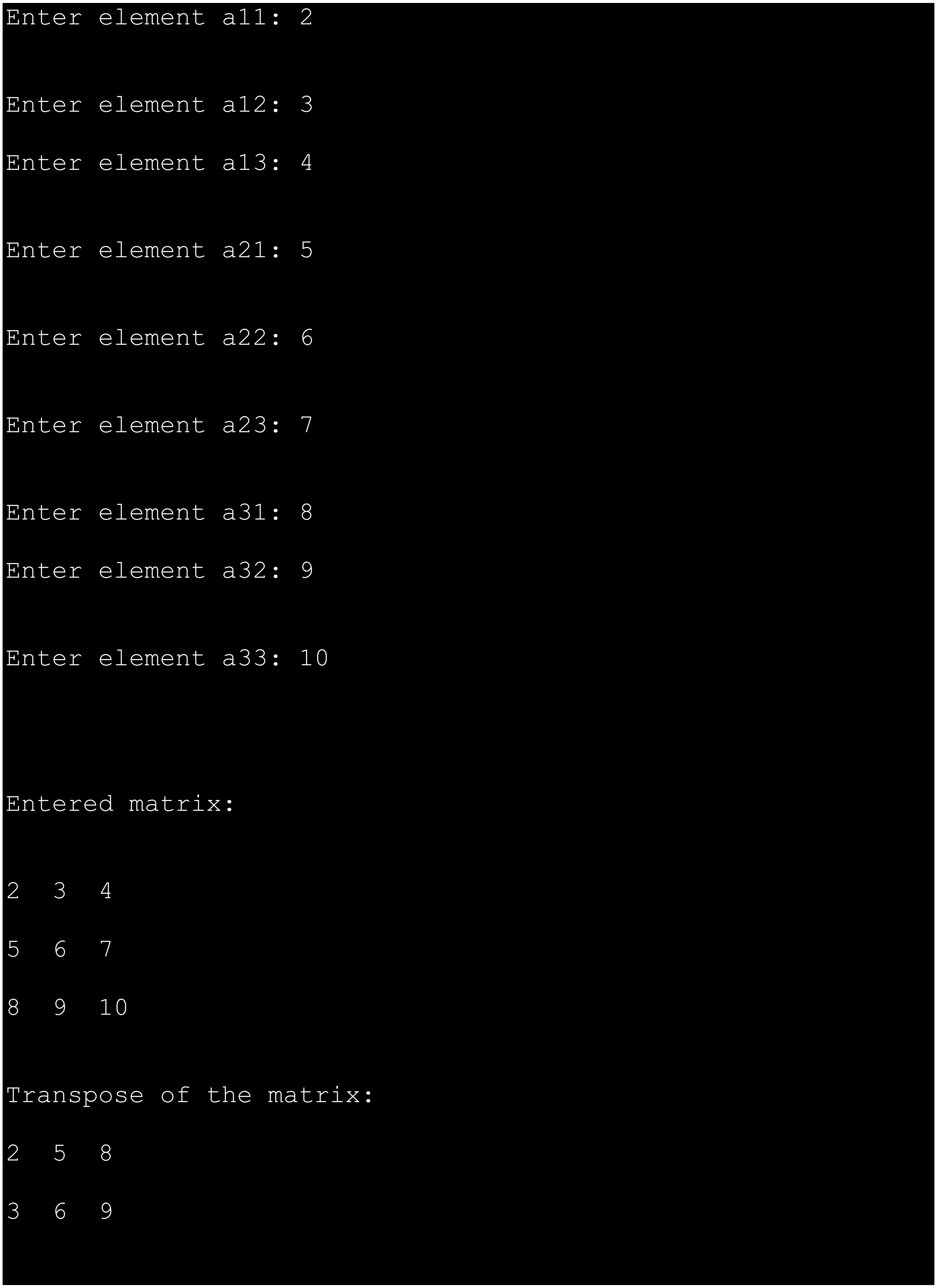
```
int a[10][10], transpose[10][10], r, c, i, j;
printf("Enter rows and columns: ");
scanf("%d %d", &r, &c);

printf("\nEnter matrix elements:\n");
for (i = 0; i < r; ++i)
    for (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 (i = 0; i < r; ++i)
    for (j = 0; j < c; ++j) {
        printf("%d  ", a[i][j]);
        if (j == c - 1)
            printf("\n");
    }


for (i = 0; i < r; ++i)
```

```
        for (j = 0; j < c; ++j) {
            transpose[j][i] = a[i][j];
        }

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

Output:



```
Enter rows and columns: 3

3



Enter matrix elements:
```



```
Enter element a11: 2


Enter element a12: 3

Enter element a13: 4


Enter element a21: 5


Enter element a22: 6


Enter element a23: 7


Enter element a31: 8

Enter element a32: 9


Enter element a33: 10




Entered matrix:


2  3  4

5  6  7

8  9  10


Transpose of the matrix:

2  5  8

3  6  9
```

```
4   7   10
```

13. Find the sum of left diagonals of a matrix.

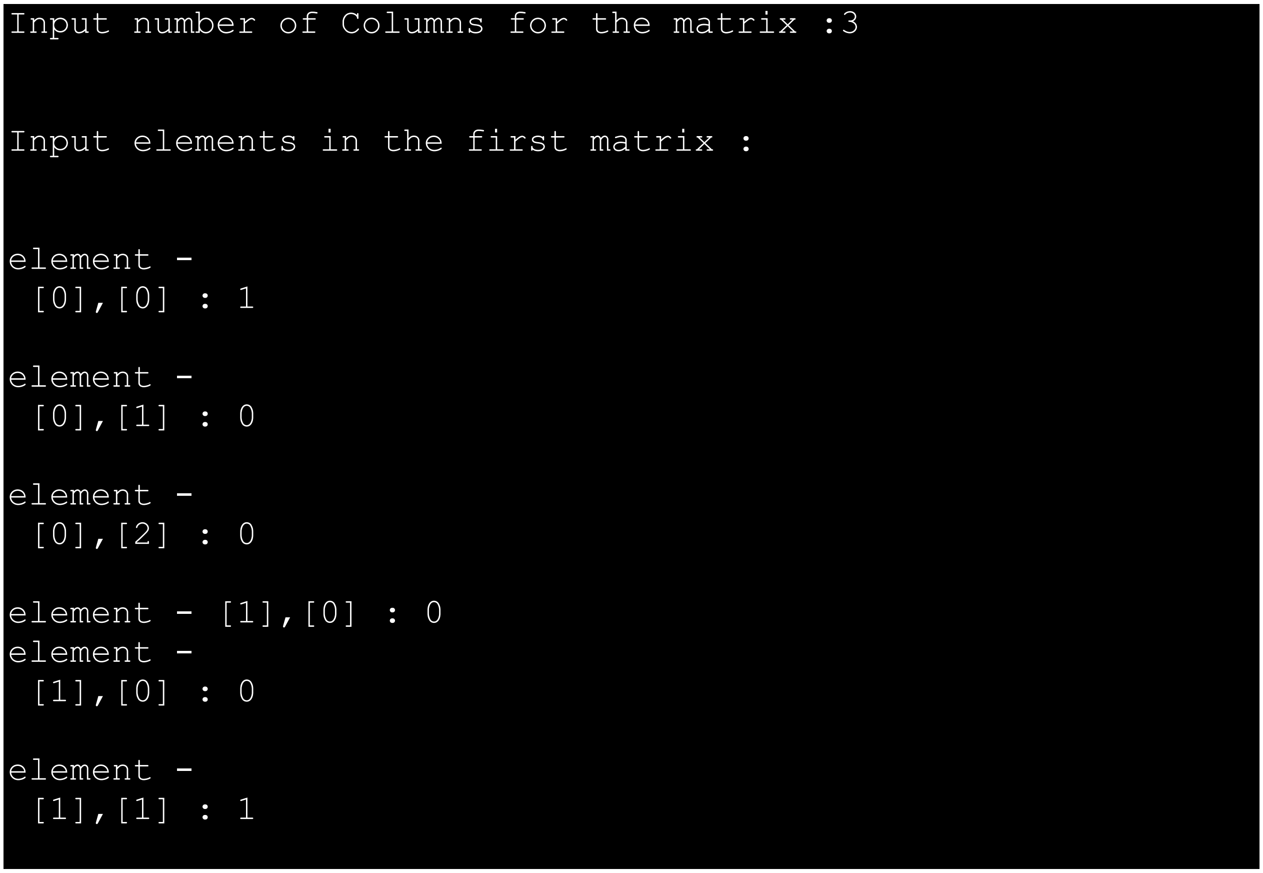
14. Check whether a given matrix is an identity matrix

```
#include <stdio.h>

void main()

{

  int arr1[10][10];

  int r1,c1;

  int i, j, yn =1;

  printf("Input number of Rows for the matrix :");

  scanf("%d", &r1);

  printf("Input number of Columns for the matrix :");

  scanf("%d",&c1);

        printf("Input elements in the first matrix :\n");

     for(i=0;i<r1;i++)

      {

         for(j=0;j<c1;j++)

         {

                printf("element - [%d],[%d] : ",i,j);
```

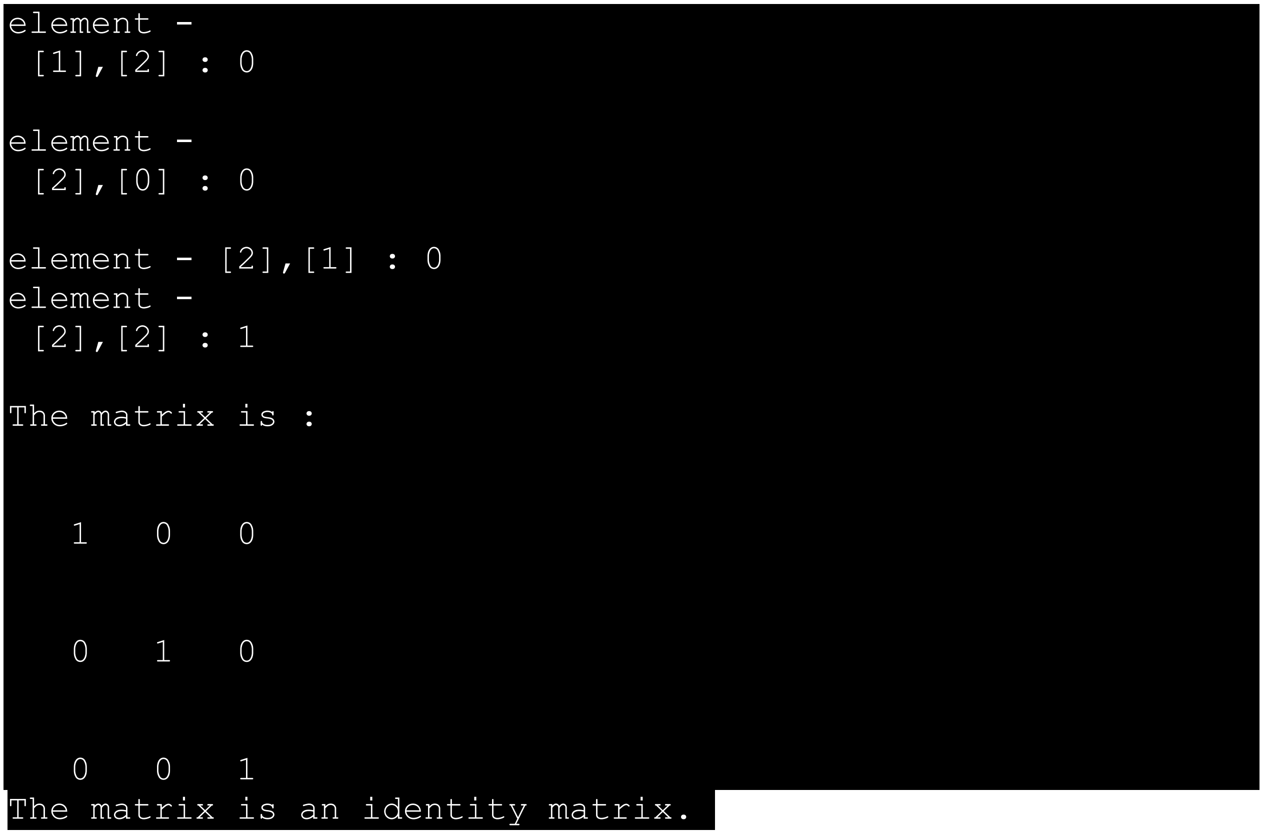
```
                 scanf("%d",&arr1[i][j]);
        }
    }
       printf("The matrix is :\n");
       for(i=0;i<r1;i++)
       {
         for(j=0;j<c1 ;j++)
           printf("% 4d",arr1[i][j]);
          printf("\n");
       }

for(i=0; i<r1; i++)
{
 for(j=0; j<c1; j++)
 {
       if(arr1[i][j] != 1 && arr1[j][i] !=0)
       {
         yn = 0;
         break;
       }
```

```
      }

  }


  if(yn == 1 )

      printf(" The matrix is an identity matrix.\n\n");

  else

      printf(" The matrix is not an identity matrix.\n\n");

}
```

Output:



```
Input number of Columns for the matrix :3


Input elements in the first matrix :


element -
 [0],[0] : 1

element -
 [0],[1] : 0

element -
 [0],[2] : 0

element - [1],[0] : 0
element -
 [1],[0] : 0

element -
 [1],[1] : 1
```



```
element -
 [1],[2] : 0

element -
 [2],[0] : 0

element - [2],[1] : 0
element -
 [2],[2] : 1

The matrix is :


   1   0   0


   0   1   0


   0   0   1
The matrix is an identity matrix.
```