KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

Deemed to be University

School of Computer Science and Engineering

Assignment-I

Submitted By:-

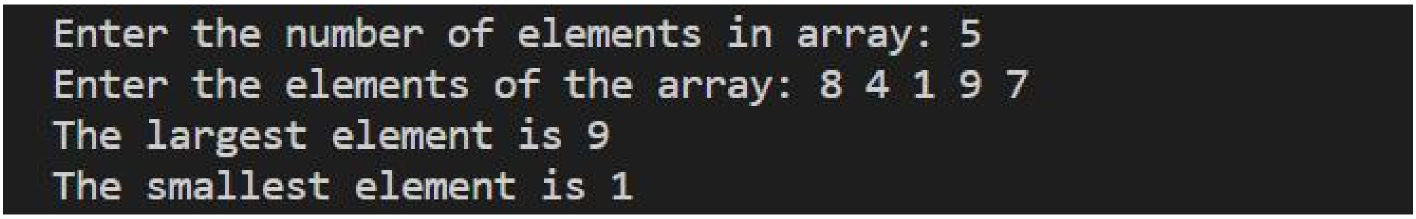
Name: Abhishikt Mohanty
Roll No: 21052991
Section: CSE-40
Branch: Computer Science and Engineering
Subject: DSA LAB

Assignment 1 | 18/07/2022 | DSA Lab

Q1. WAP to find out the smallest and largest element stored in an array of n integers.
Code:-

```
#include<stdio.h>

int main(){
    int a[50],i,n,large,small;
    printf("Enter the number of elements in array: ");
    scanf("%d",&n);
    printf("Enter the elements of the array: ");
    for(i=0;i<n;++i){
        scanf("%d",&a[i]);
        large=small=a[0];
    }
    for(i=1;i<n;++i){
        if(a[i]>large){
            large=a[i];
        }
        if(a[i]<small){
            small=a[i];
        }
    }
    printf("The largest element is %d",large);
    printf("\nThe smallest element is %d\n",small);
    return 0;
}
```

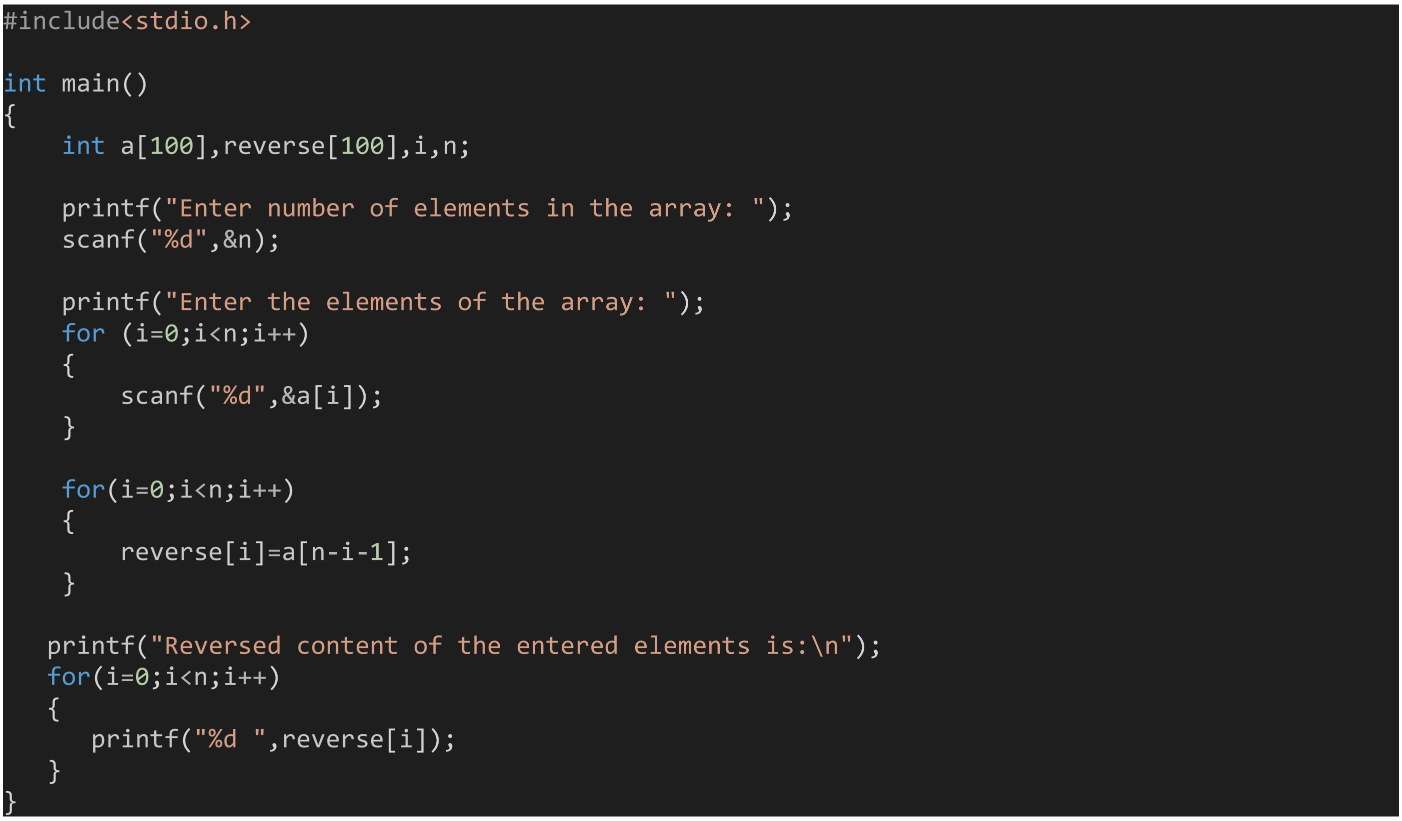
Output:-



```
Enter the number of elements in array: 5
Enter the elements of the array: 8 4 1 9 7
The largest element is 9
The smallest element is 1
```

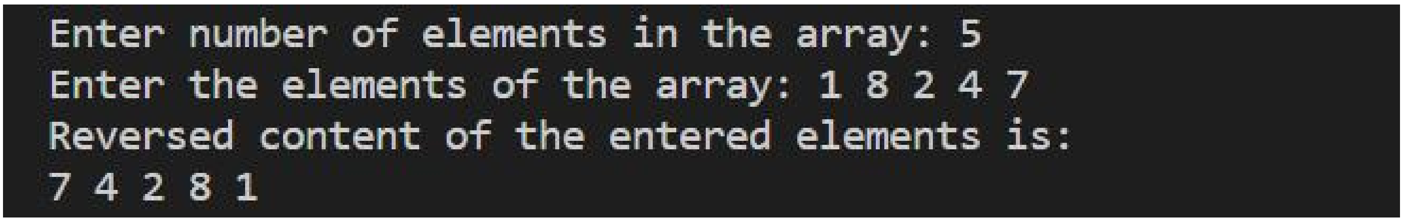
Q2. WAP to reverse the contents of an array of n elements.

Code :-



```c
#include<stdio.h>

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

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

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

    for(i=0;i<n;i++)
    {
        reverse[i]=a[n-i-1];
    }

   printf("Reversed content of the entered elements is:\n");
   for(i=0;i<n;i++)
   {
      printf("%d ",reverse[i]);
   }
}
```

Output:-



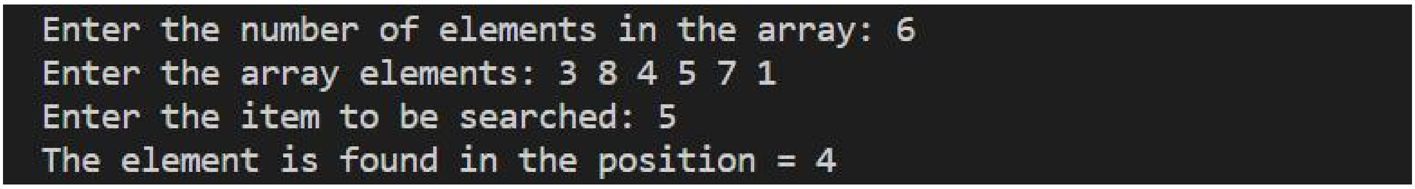
```
Enter number of elements in the array: 5
Enter the elements of the array: 1 8 2 4 7
Reversed content of the entered elements is:
7 4 2 8 1
```

Q3. WAP to search an element in an array of n numbers. 4. WAP to sort an array of n numbers.

Code:-

```
#include <stdio.h>

int main() {
   int n, i, r, arr[30];

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

   printf("Enter the array elements: ");
   for (i = 0; i < n; i++) {
      scanf("%d", &arr[i]);
   }

   printf("Enter the item to be searched: ");
   scanf("%d", &r);

   i = 0;
   while (i < n && r != arr[i]) {
      i++;
   }

   if (i < n) {
      printf("The element is found in the position = %d", i + 1);
   } else {
      printf("Element not found!");
   }

   return 0;
}
```

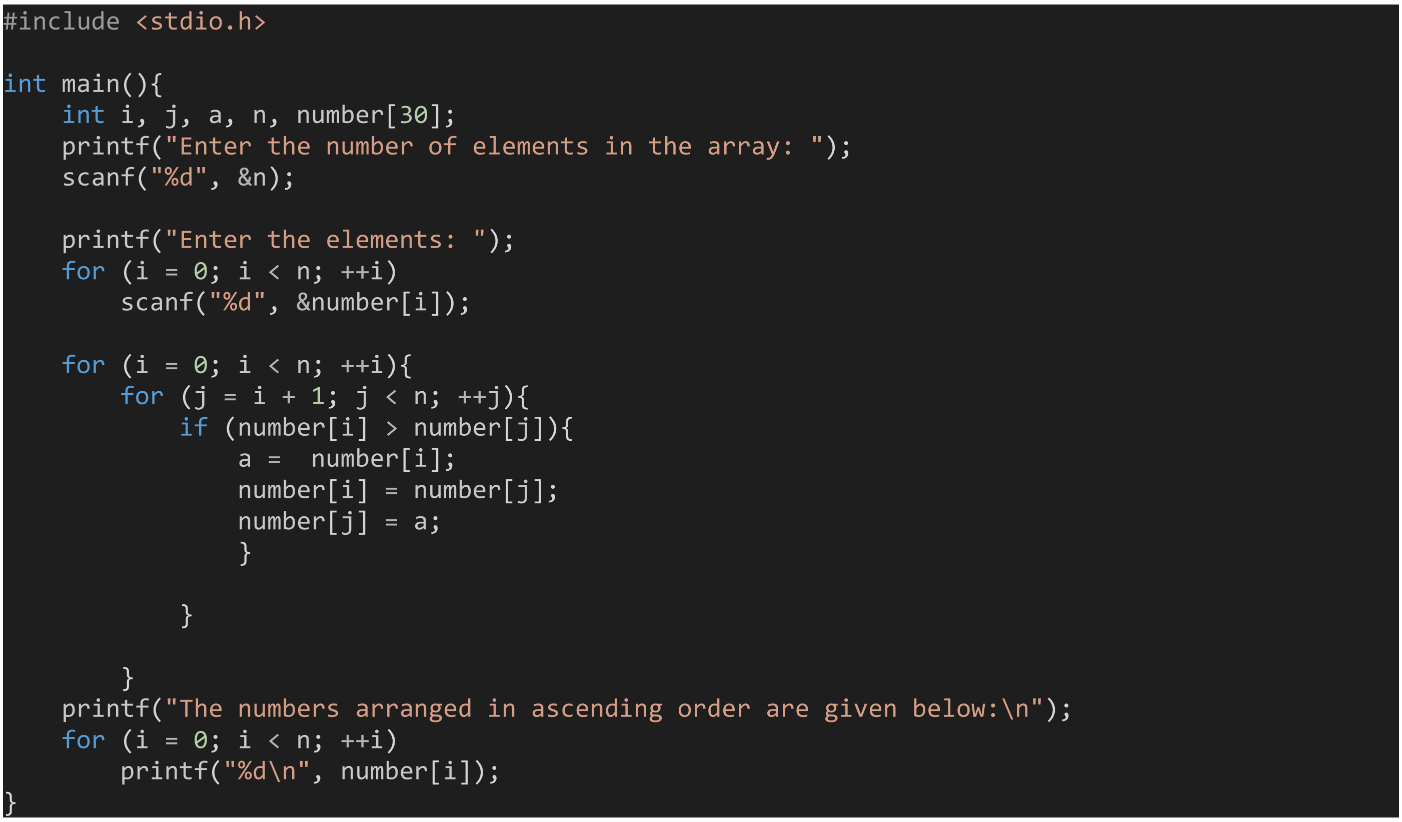
Output:-



```
Enter the number of elements in the array: 6
Enter the array elements: 3 8 4 5 7 1
Enter the item to be searched: 5
The element is found in the position = 4
```

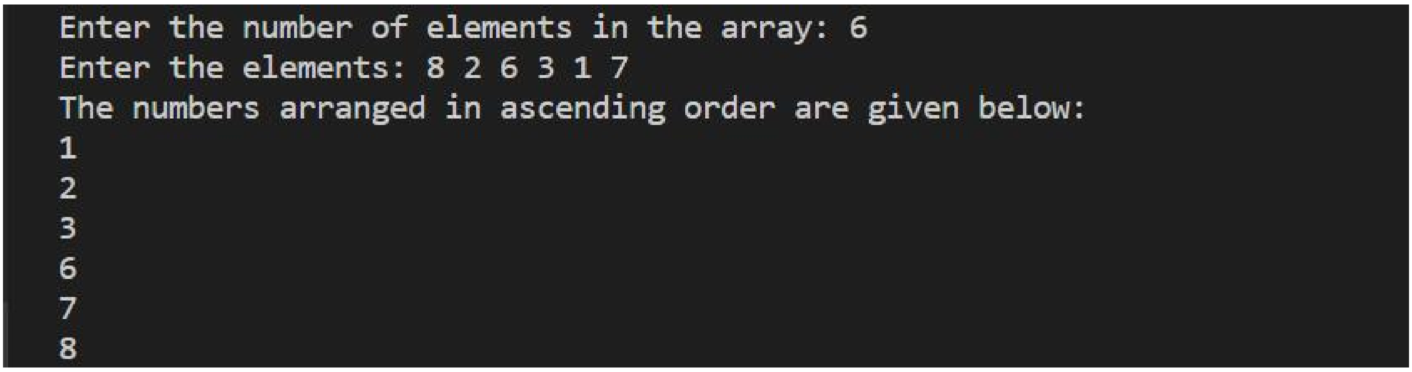
Q4. WAP to sort an array of n numbers.

Code:-



```
#include <stdio.h>

int main(){
    int i, j, a, n, number[30];
    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);

    printf("Enter the elements: ");
    for (i = 0; i < n; ++i)
        scanf("%d", &number[i]);

    for (i = 0; i < n; ++i){
        for (j = i + 1; j < n; ++j){
            if (number[i] > number[j]){
                a =  number[i];
                number[i] = number[j];
                number[j] = a;
                }

            }

        }
    printf("The numbers arranged in ascending order are given below:\n");
    for (i = 0; i < n; ++i)
        printf("%d\n", number[i]);
}
```

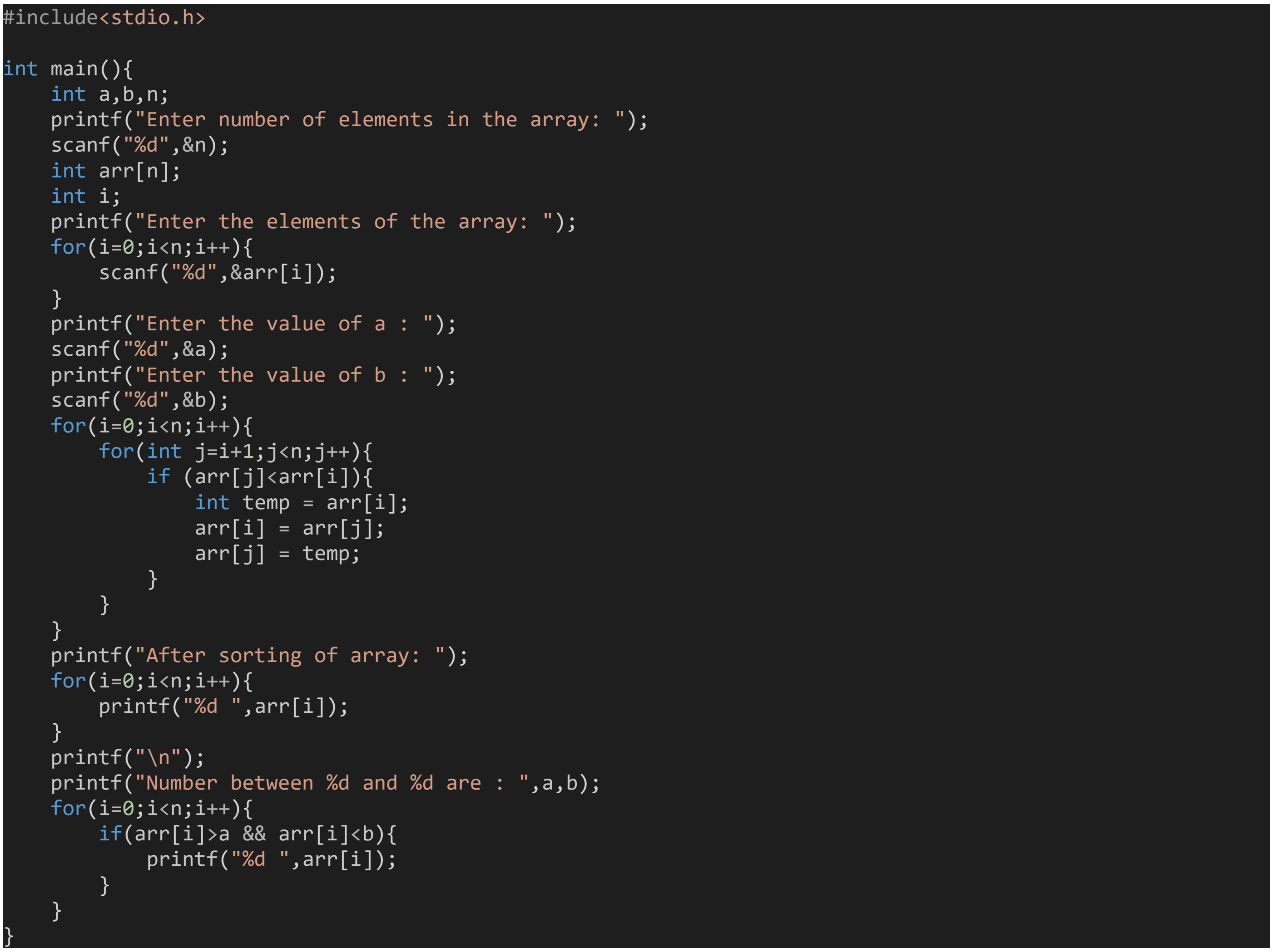
Output:-



```
Enter the number of elements in the array: 6
Enter the elements: 8 2 6 3 1 7
The numbers arranged in ascending order are given below:
1
2
3
6
7
8
```

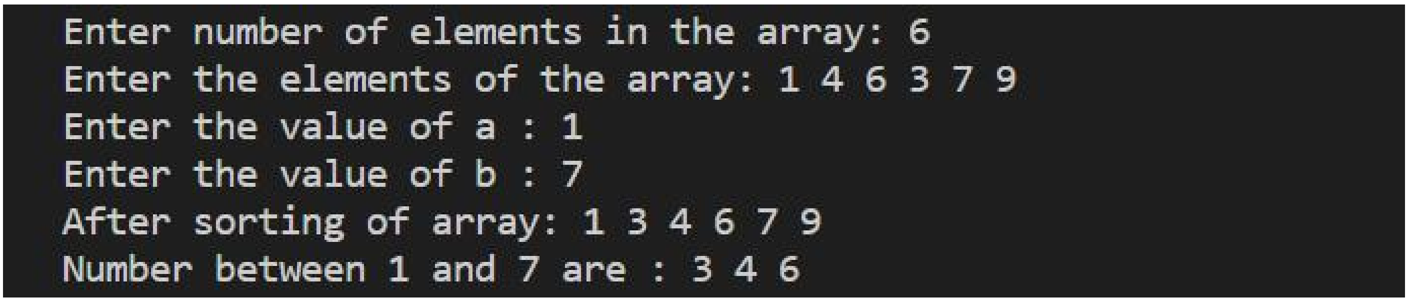
Q5. Given an unsorted array of size n, WAP to find number of elements between two elements a and b (both inclusive).

Code:-



```c
#include<stdio.h>

int main(){
    int a,b,n;
    printf("Enter number of elements in the array: ");
    scanf("%d",&n);
    int arr[n];
    int i;
    printf("Enter the elements of the array: ");
    for(i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    printf("Enter the value of a : ");
    scanf("%d",&a);
    printf("Enter the value of b : ");
    scanf("%d",&b);
    for(i=0;i<n;i++){
        for(int j=i+1;j<n;j++){
            if (arr[j]<arr[i]){
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
    printf("After sorting of array: ");
    for(i=0;i<n;i++){
        printf("%d ",arr[i]);
    }
    printf("\n");
    printf("Number between %d and %d are : ",a,b);
    for(i=0;i<n;i++){
        if(arr[i]>a && arr[i]<b){
            printf("%d ",arr[i]);
        }
    }
}
```

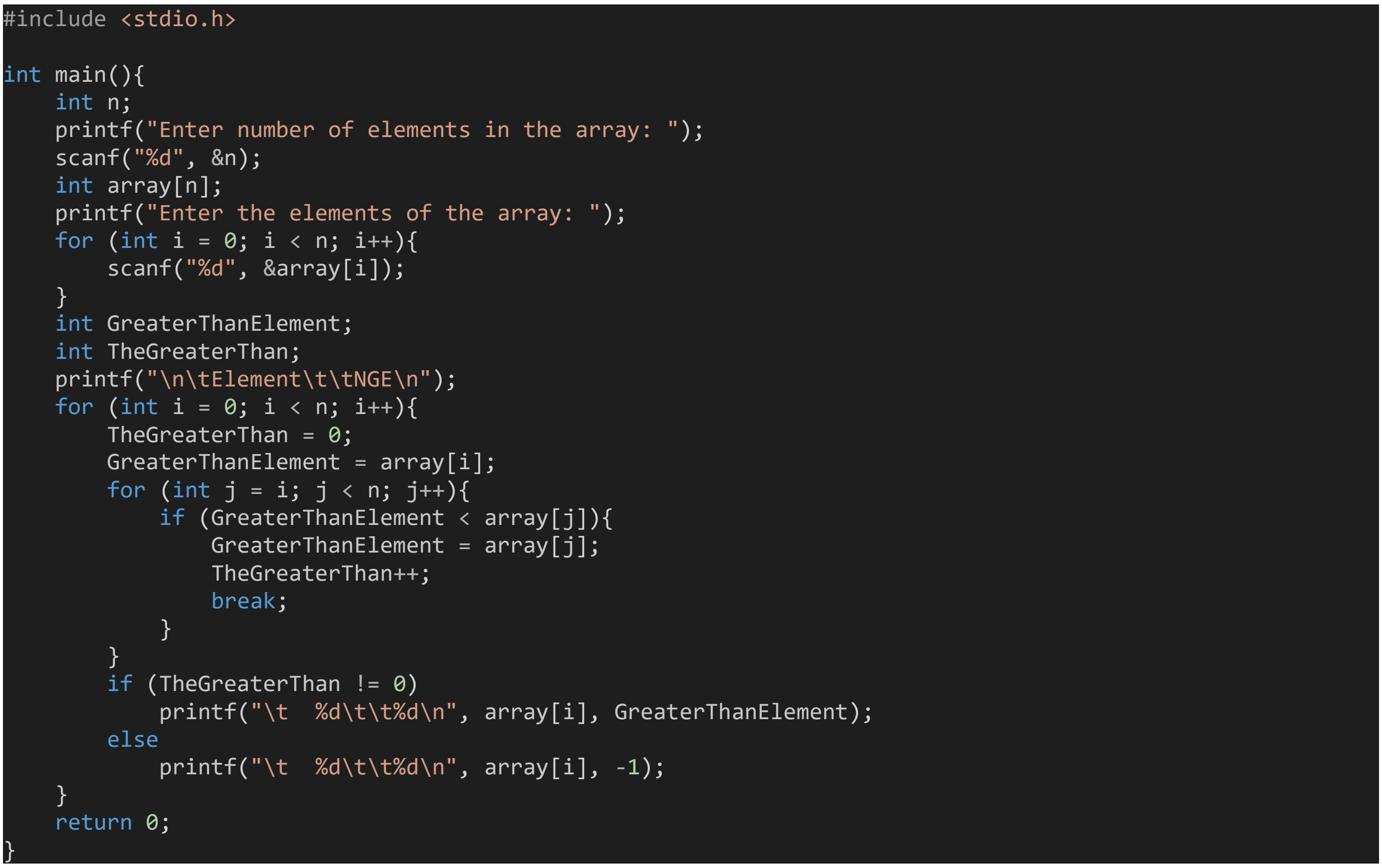
Output:-



```
Enter number of elements in the array: 6
Enter the elements of the array: 1 4 6 3 7 9
Enter the value of a : 1
Enter the value of b : 7
After sorting of array: 1 3 4 6 7 9
Number between 1 and 7 are : 3 4 6
```

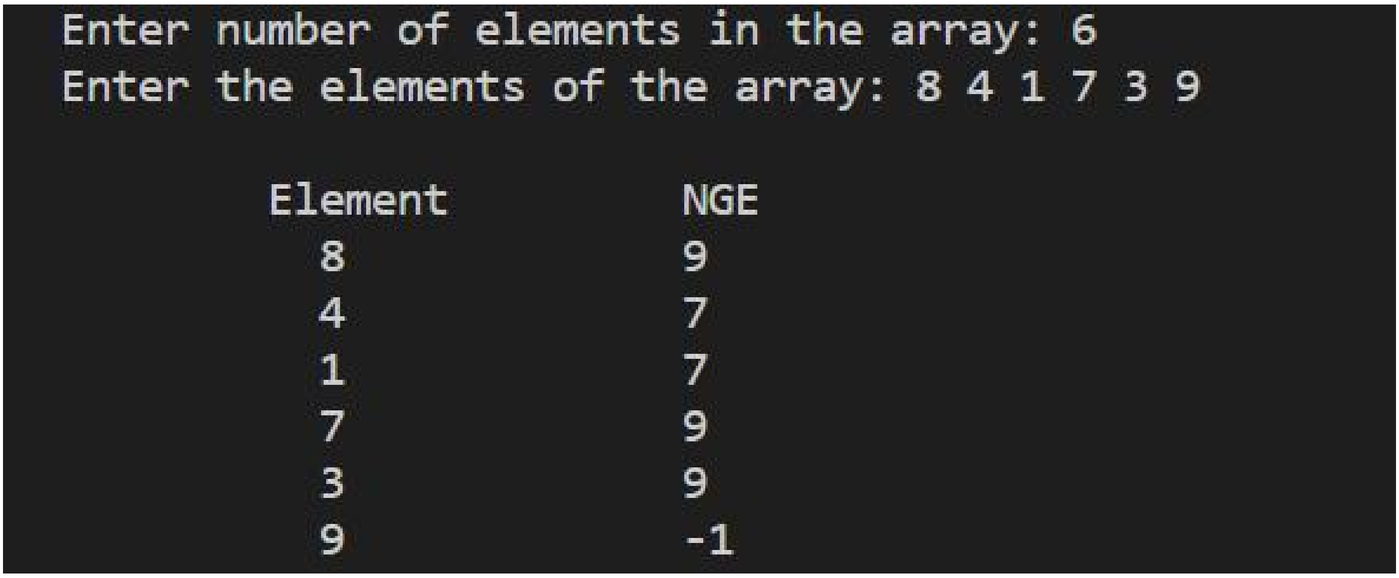
Q6. Given an array, WAP to print the next greater element (NGE) for every element. The next greater element for an element x is the first greater element on the right side of x in array. Elements for which no greater element exist, consider next greater element as -1.

Code:-



```c
#include <stdio.h>

int main(){
    int n;
    printf("Enter number of elements in the array: ");
    scanf("%d", &n);
    int array[n];
    printf("Enter the elements of the array: ");
    for (int i = 0; i < n; i++){
        scanf("%d", &array[i]);
    }
    int GreaterThanElement;
    int TheGreaterThan;
    printf("\n\tElement\t\tNGE\n");
    for (int i = 0; i < n; i++){
        TheGreaterThan = 0;
        GreaterThanElement = array[i];
        for (int j = i; j < n; j++){
            if (GreaterThanElement < array[j]){
                GreaterThanElement = array[j];
                TheGreaterThan++;
                break;
            }
        }
        if (TheGreaterThan != 0)
            printf("\t  %d\t\t%d\n", array[i], GreaterThanElement);
        else
            printf("\t  %d\t\t%d\n", array[i], -1);
    }
    return 0;
}
```

Output:-



```
Enter number of elements in the array: 6
Enter the elements of the array: 8 4 1 7 3 9

        Element         NGE
          8             9
          4             7
          1             7
          7             9
          3             9
          9             -1
```

Q7. Let A be nXn square matrix array. WAP by using appropriate user defined functions for the following:

a) Find the number of nonzero elements in A

b) Find the sum of the elements above the leading diagonal.

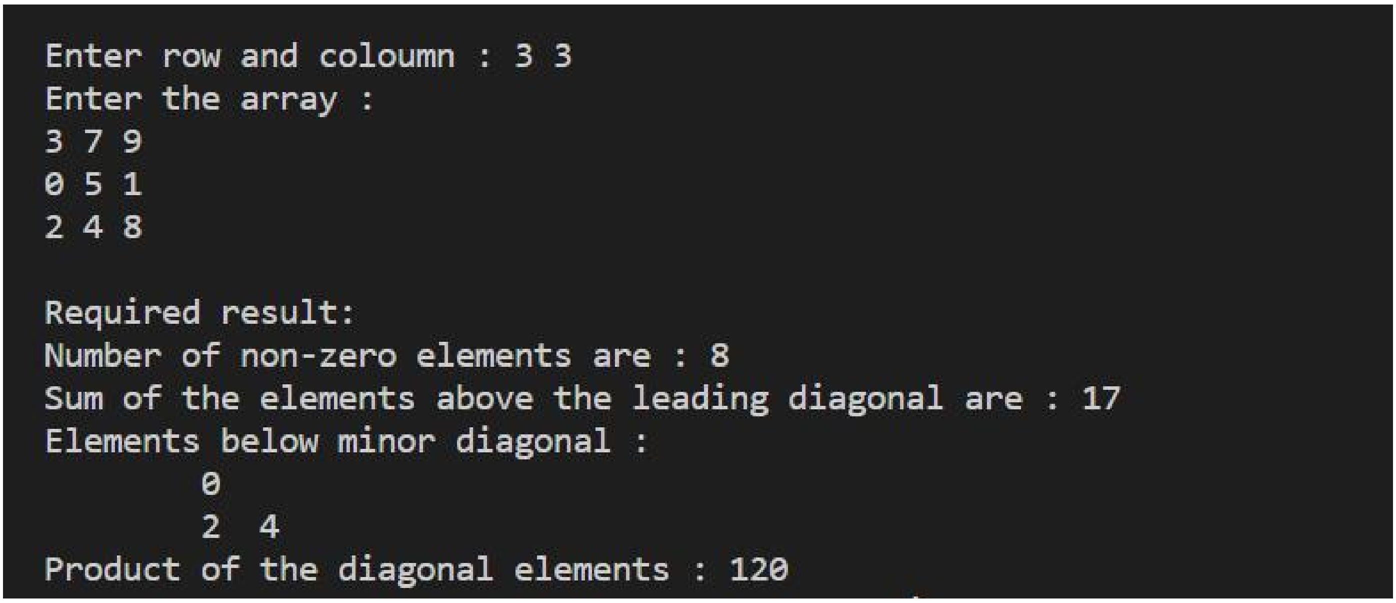
c) Display the elements below the minor diagonal.

d) Find the product of the diagonal elements.

Code:-

```
#include<stdio.h>

void non_zero(int row,int coloumn,int a[row][coloumn]){
    int count = 0;
    int i,j;
    for (i = 0; i < row; i++){
        for (j = 0; j < coloumn; j++){
            if (a[i][j] != 0){
                count++;
            }
        }
    }
    printf("Number of non-zero elements are : %d\n", count);
}
void above_leading_diagonal(int row,int coloumn,int a[row][coloumn]){
    int sum = 0;
    int i,j;
    for (i = 0; i < row - 1; i++){
        for (j = i + 1; j < coloumn; j++){
            sum += a[i][j];
        }
    }
    printf("Sum of the elements above the leading diagonal are : %d\n", sum);
}
void minor_diagonal_elements(int row,int coloumn,int a[row][coloumn]){
    printf("Elements below minor diagonal : \n");
    int i,j;
    for (i = 1; i < row; i++){
        printf("\t");
        for (j = 0; j < i; j++){
            printf("%d  ", a[i][j]);
        }
        printf("\n");
    }
}
void product_of_diagonal_elements(int row,int coloumn,int a[row][coloumn]){
    int product = 1;
    int i,j;
    for (i = 0; i < row; i++){
        for (j = 0; j < coloumn; j++){
            if (i == j){
                product = product * a[i][j];
            }
        }
    }
    printf("Product of the diagonal elements : %d", product);
}
int main(){
    int row,coloumn;
    printf("Enter row and coloumn : ");
    scanf("%d%d",&row,&coloumn);
    int a[row][coloumn];
    int i,j;
    printf("Enter the array : \n");
    for (i = 0; i < row; i++){
        for(j = 0; j < coloumn; j++){
            scanf("%d",&a[i][j]);
        }
    }
    printf("\nRequired result:\n");
    //check for nonzero
    non_zero(row,coloumn,a);
    //Sum of elements above leading diagonal
    above_leading_diagonal(row,coloumn,a);
    // display elements below minor diagonal elements
    minor_diagonal_elements(row,coloumn,a);
    //product of diagonal elements
    product_of_diagonal_elements(row,coloumn,a);
    return 0;
}
```

Output:-



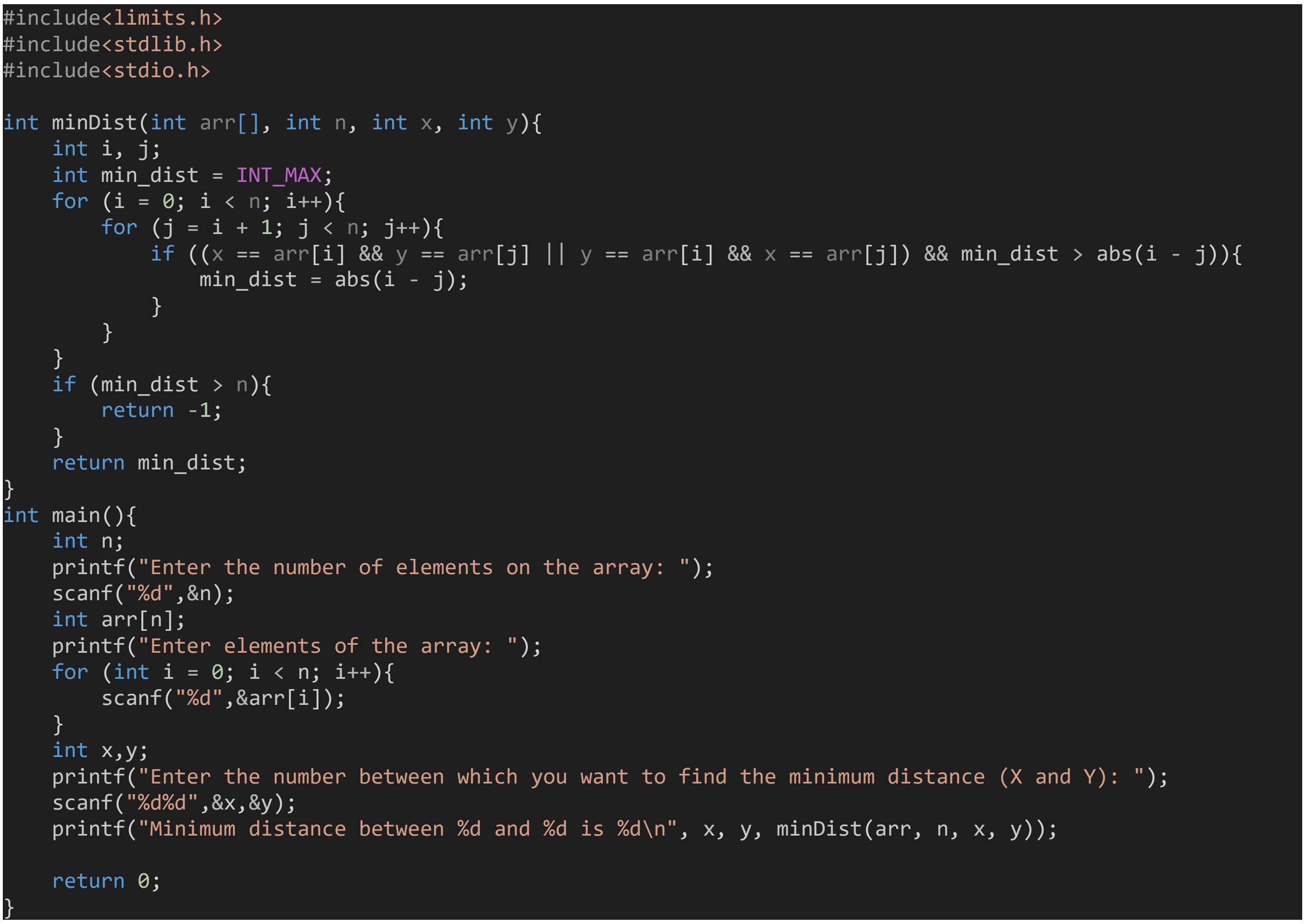
```
Enter row and coloumn : 3 3
Enter the array :
3 7 9
0 5 1
2 4 8

Required result:
Number of non-zero elements are : 8
Sum of the elements above the leading diagonal are : 17
Elements below minor diagonal :
        0
        2  4
Product of the diagonal elements : 120
```

Q8. Given an unsorted array arr[] and two numbers x and y. find the minimum distance between x and y in arr[]. The array might also contain duplicates. You may assume that both x and y are different and present in arr[].

Input: arr[]= (3, 5, 4, 2, 6, 5, 6, 6, 5, 4, 8, 3), x = 3, y=6

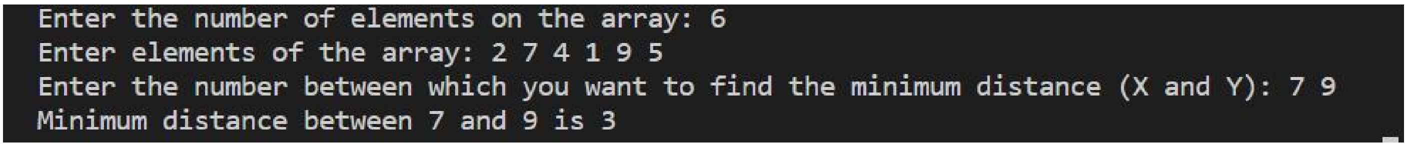
Output: Minimum distance between 3 and 6 is 4.

Code:-



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

int minDist(int arr[], int n, int x, int y){
    int i, j;
    int min_dist = INT_MAX;
    for (i = 0; i < n; i++){
        for (j = i + 1; j < n; j++){
            if ((x == arr[i] && y == arr[j] || y == arr[i] && x == arr[j]) && min_dist > abs(i - j)){
                min_dist = abs(i - j);
            }
        }
    }
    if (min_dist > n){
        return -1;
    }
    return min_dist;
}
int main(){
    int n;
    printf("Enter the number of elements on the array: ");
    scanf("%d",&n);
    int arr[n];
    printf("Enter elements of the array: ");
    for (int i = 0; i < n; i++){
        scanf("%d",&arr[i]);
    }
    int x,y;
    printf("Enter the number between which you want to find the minimum distance (X and Y): ");
    scanf("%d%d",&x,&y);
    printf("Minimum distance between %d and %d is %d\n", x, y, minDist(arr, n, x, y));

    return 0;
}
```

Output:-



```
Enter the number of elements on the array: 6
Enter elements of the array: 2 7 4 1 9 5
Enter the number between which you want to find the minimum distance (X and Y): 7 9
Minimum distance between 7 and 9 is 3
```

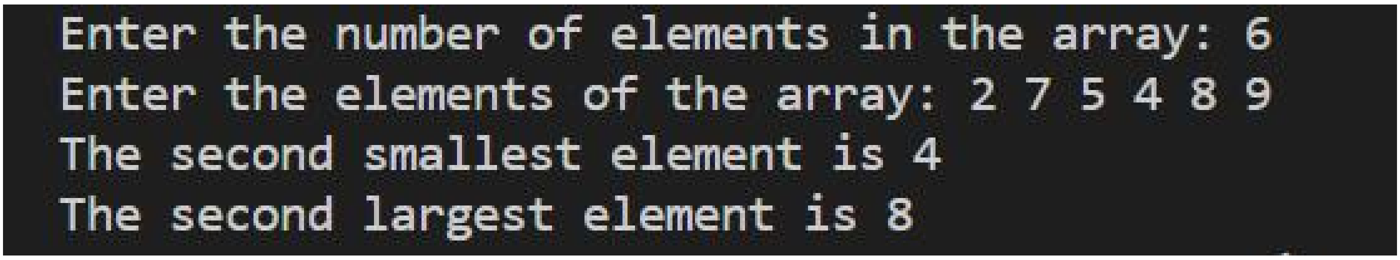
Home-Assignment 1 | 18/07/2022 | DSA Lab

Q1. WAP to find out the second smallest and second largest element stored in an array.

Code:-

```c
#include <stdio.h>

int main(){
    int n;
    printf("Enter the number of elements in the array: ");
    scanf("%d",&n);
    printf("Enter the elements of the array: ");
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    for(int i=0;i<n;i++){
        int temp;
        for(int j=i+1; j<n ;j++){
            if(a[i]<a[j]){
                temp=a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        }
    }
    printf("The second smallest element is %d",a[n-2]);
    printf("\n");
    printf("The second largest element is %d",a[1]);
    return 0;
}
```

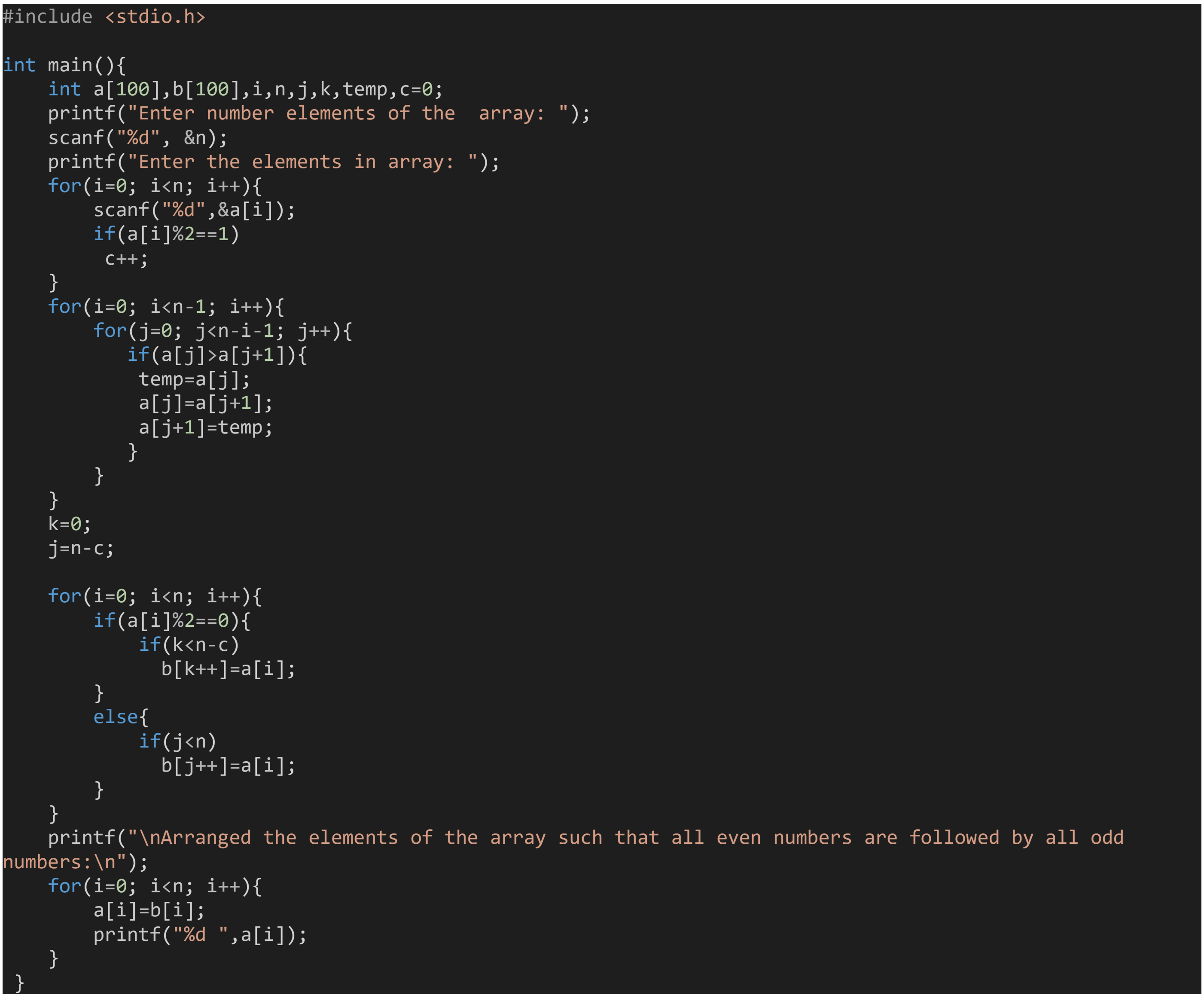
Output:-



```
Enter the number of elements in the array: 6
Enter the elements of the array: 2 7 5 4 8 9
The second smallest element is 4
The second largest element is 8
```

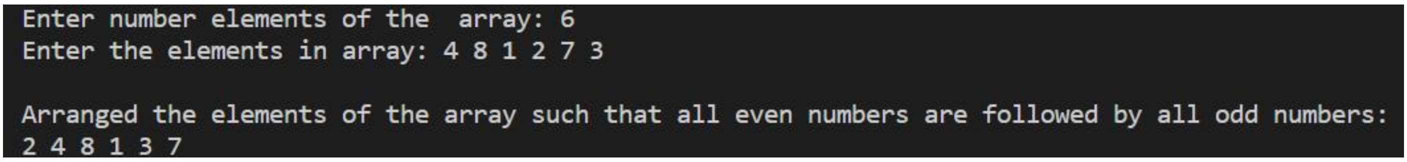
Q2. WAP to arrange the elements of an array such that all even numbers are followed by all odd numbers

Code:-



```
#include <stdio.h>

int main(){
    int a[100],b[100],i,n,j,k,temp,c=0;
    printf("Enter number elements of the  array: ");
    scanf("%d", &n);
    printf("Enter the elements in array: ");
    for(i=0; i<n; i++){
        scanf("%d",&a[i]);
        if(a[i]%2==1)
         c++;
    }
    for(i=0; i<n-1; i++){
        for(j=0; j<n-i-1; j++){
           if(a[j]>a[j+1]){
            temp=a[j];
            a[j]=a[j+1];
            a[j+1]=temp;
           }
        }
    }
    k=0;
    j=n-c;

    for(i=0; i<n; i++){
        if(a[i]%2==0){
            if(k<n-c)
              b[k++]=a[i];
        }
        else{
            if(j<n)
              b[j++]=a[i];
        }
    }
    printf("\nArranged the elements of the array such that all even numbers are followed by all odd 
numbers:\n");
    for(i=0; i<n; i++){
        a[i]=b[i];
        printf("%d ",a[i]);
    }
 }
```

Output:-



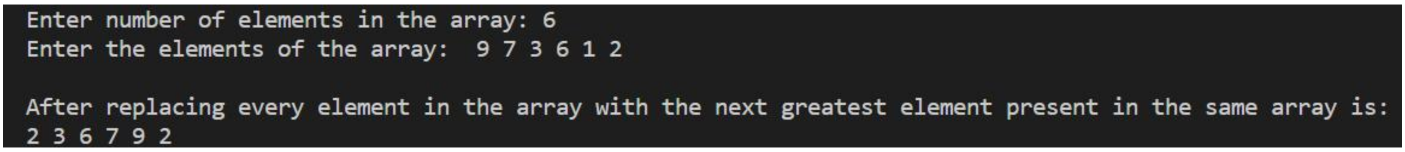
```
Enter number elements of the  array: 6
Enter the elements in array: 4 8 1 2 7 3

Arranged the elements of the array such that all even numbers are followed by all odd numbers:
2 4 8 1 3 7
```

Q3. Write a program to replace every element in the array with the next greatest element present in the same array.

Code:-

```
#include <stdio.h>

int main(){
    //input
    int n;
    printf("Enter number of elements in the array: ");
    scanf("%d", &n);
    int a[n];
    int i;
    printf("Enter the elements of the array: ");
    for (i = 0; i < n; i++){
        scanf("%d", &a[i]);
    }
    //sorting
    for (i = 0; i < n; i++){
        for (int j = i + 1; j < n; j++){
            if (a[j] < a[i]){
                int temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
    // replacing
    for (i = 0; i < n; i++){
        if (i < n && (i + 1) == n){
            a[i] = a[0];
        }
        else{
            a[i] = a[i + 1];
        }
    }
    // printing
    printf("\nAfter replacing every element in the array with the next greatest element present in the
same array is:\n");
    for (i = 0; i < n; i++){
        printf("%d ", a[i]);
    }
    return 0;
}
```

Output:-



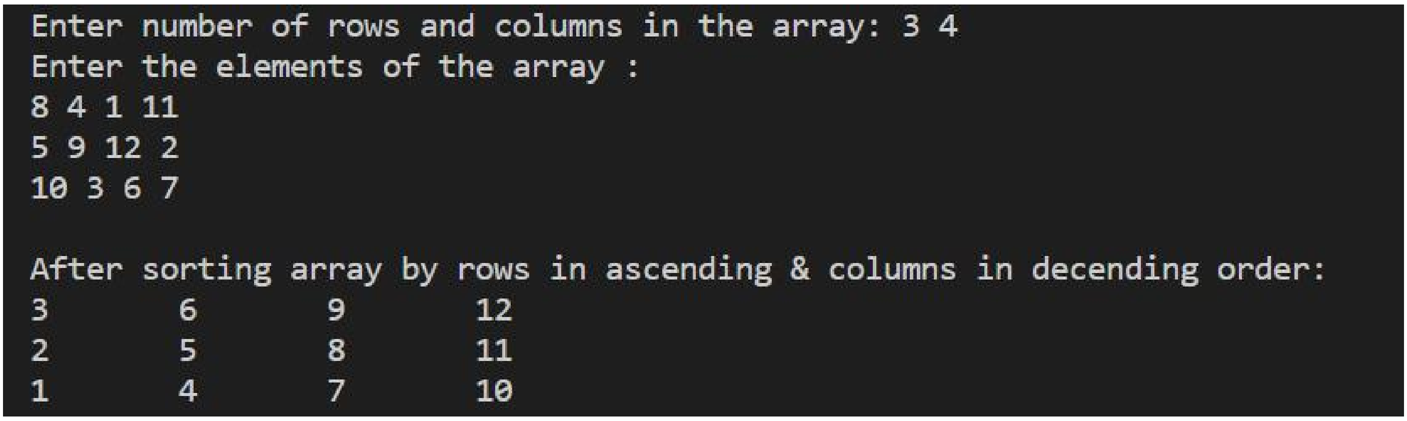
```
Enter number of elements in the array: 6
Enter the elements of the array:  9 7 3 6 1 2

After replacing every element in the array with the next greatest element present in the same array is:
2 3 6 7 9 2
```

Q4. WAP to replace every array element by multiplication of previous and next of an n element.

Code:-

```
#include<stdio.h>

int main(){
    //input
    int n;
    printf("Enter the number of elements on the array: ");
    scanf("%d",&n);
    //array
    int a[n];
    int b[n];
    printf("Enter the elements of the array: ");
    int i;
    for ( i = 0; i < n; i++){
        scanf("%d",&a[i]);
    }
    b[0] = a[1];
    //multiplication
    for ( i = 0; i < n; i++){
        if (i>0 && i<n){
            b[i] = a[i-1] * a[i+1];
        }
        if (i<n && (i+1) == n){
            b [i] = a[i-1];
        }

    }
    //printing
    printf("\nResult after replacing every array elements by multiplication of previous and next of an
array element:\n");
    for ( i = 0; i < n; i++){
        printf("%d\t",b[i]);
   }

    return 0;
}
```

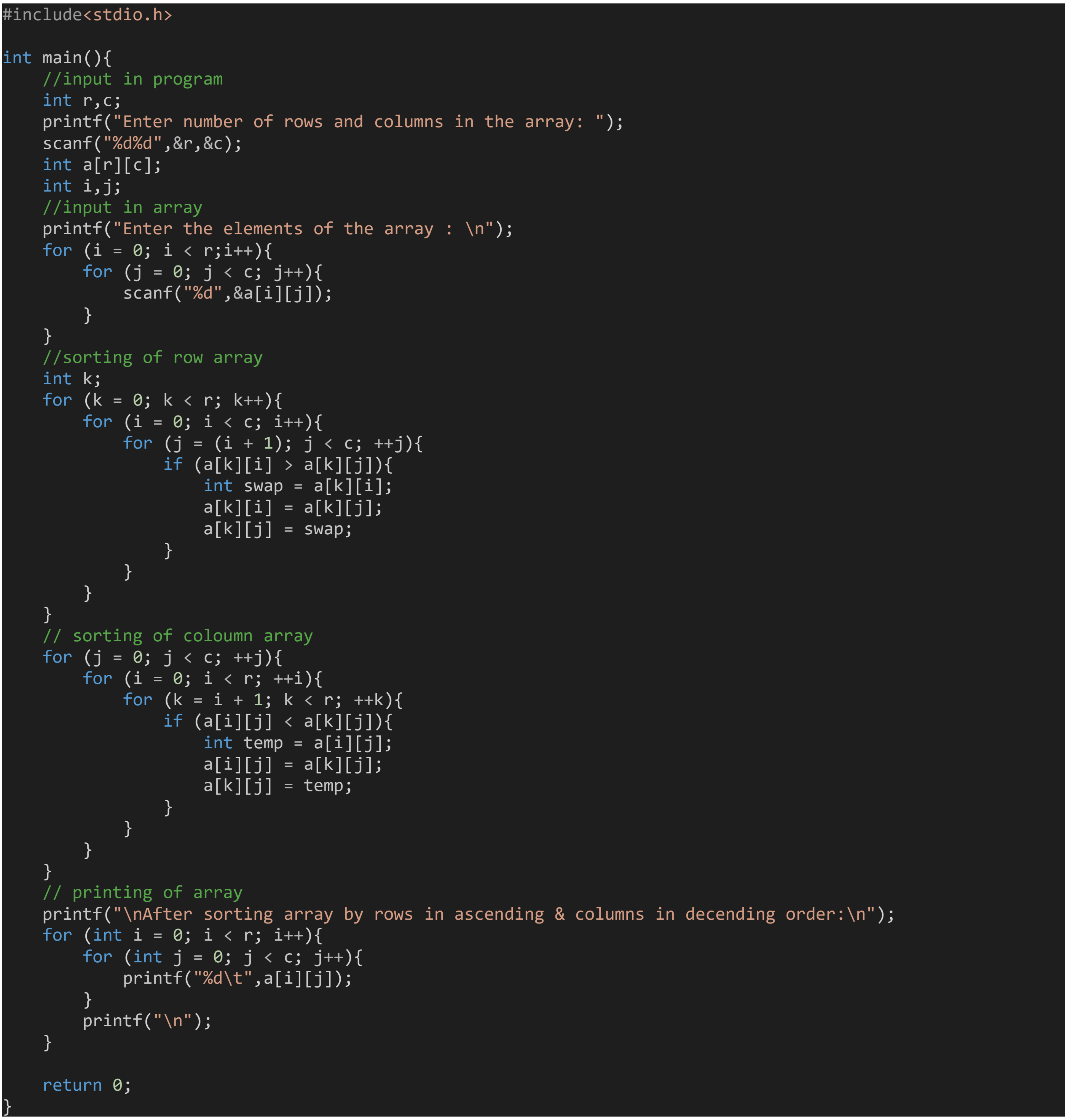
Output:-



```
Enter number of rows and columns in the array: 3 4
Enter the elements of the array :
8 4 1 11
5 9 12 2
10 3 6 7

After sorting array by rows in ascending & columns in decending order:
3       6       9       12
2       5       8       11
1       4       7       10
```

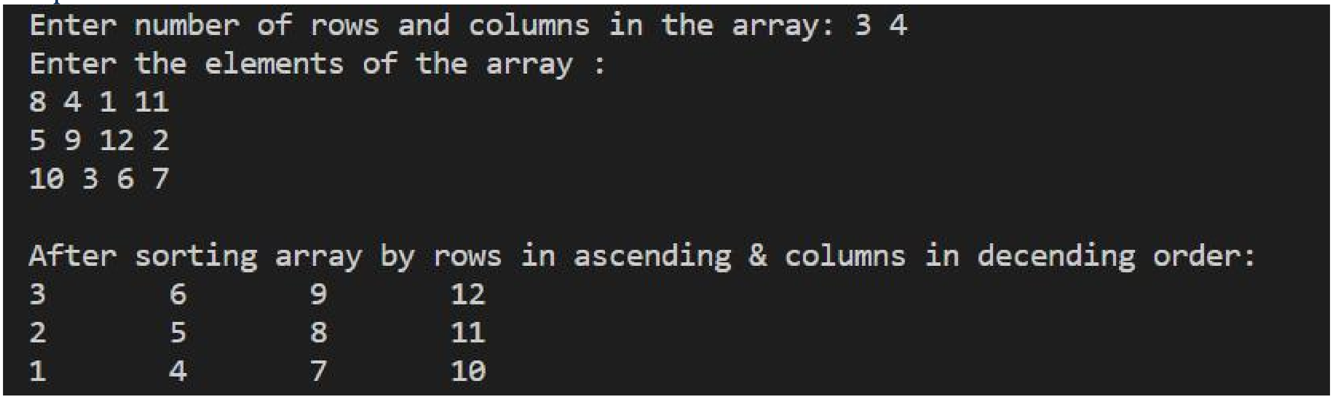
Q5. WAP to sort rows of a matrix having m rows and n columns in ascending & columns in descending order.

Code:-



```
#include<stdio.h>

int main(){
    //input in program
    int r,c;
    printf("Enter number of rows and columns in the array: ");
    scanf("%d%d",&r,&c);
    int a[r][c];
    int i,j;
    //input in array
    printf("Enter the elements of the array : \n");
    for (i = 0; i < r;i++){
        for (j = 0; j < c; j++){
            scanf("%d",&a[i][j]);
        }
    }
    //sorting of row array
    int k;
    for (k = 0; k < r; k++){
        for (i = 0; i < c; i++){
            for (j = (i + 1); j < c; ++j){
                if (a[k][i] > a[k][j]){
                    int swap = a[k][i];
                    a[k][i] = a[k][j];
                    a[k][j] = swap;
                }
            }
        }
    }
    // sorting of coloumn array
    for (j = 0; j < c; ++j){
        for (i = 0; i < r; ++i){
            for (k = i + 1; k < r; ++k){
                if (a[i][j] < a[k][j]){
                    int temp = a[i][j];
                    a[i][j] = a[k][j];
                    a[k][j] = temp;
                }
            }
        }
    }
    // printing of array
    printf("\nAfter sorting array by rows in ascending & columns in decending order:\n");
    for (int i = 0; i < r; i++){
        for (int j = 0; j < c; j++){
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

Output:-



```
Enter number of rows and columns in the array: 3 4
Enter the elements of the array :
8 4 1 11
5 9 12 2
10 3 6 7

After sorting array by rows in ascending & columns in decending order:
3       6       9       12
2       5       8       11
1       4       7       10
```