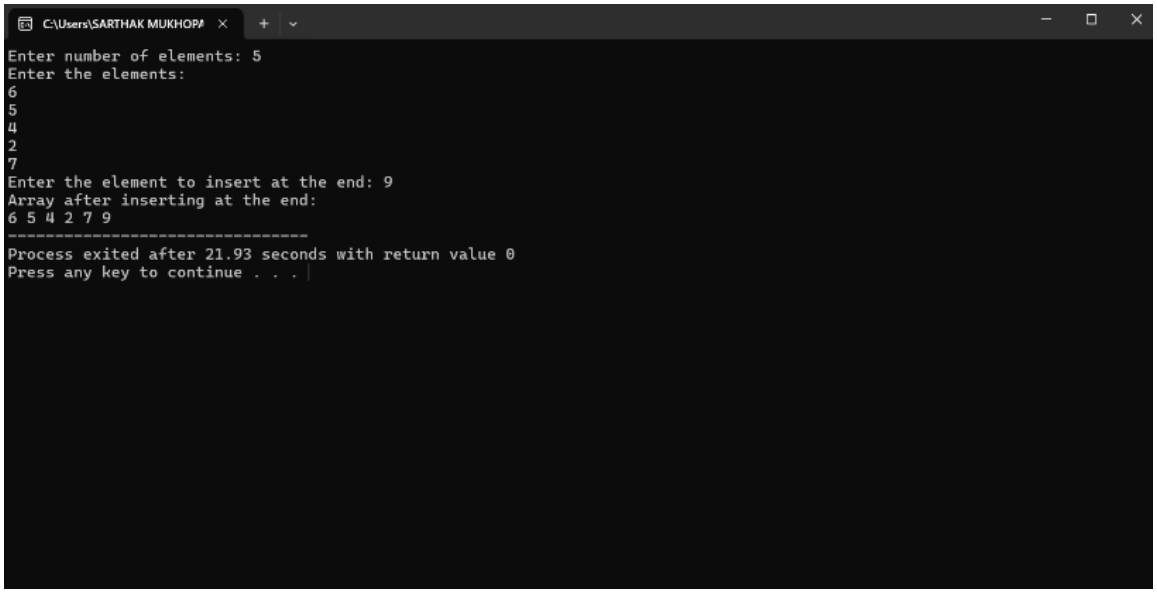
Q1) Insert an element at end of array

```c
#include <stdio.h>

int main() {
    int arr[100];
    int n, i, newElement;

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

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

    printf("Enter the element to insert at the end: ");
    scanf("%d", &newElement);

    arr[n] = newElement;
    n++;

    printf("Array after inserting at the end:\n");
    for(i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

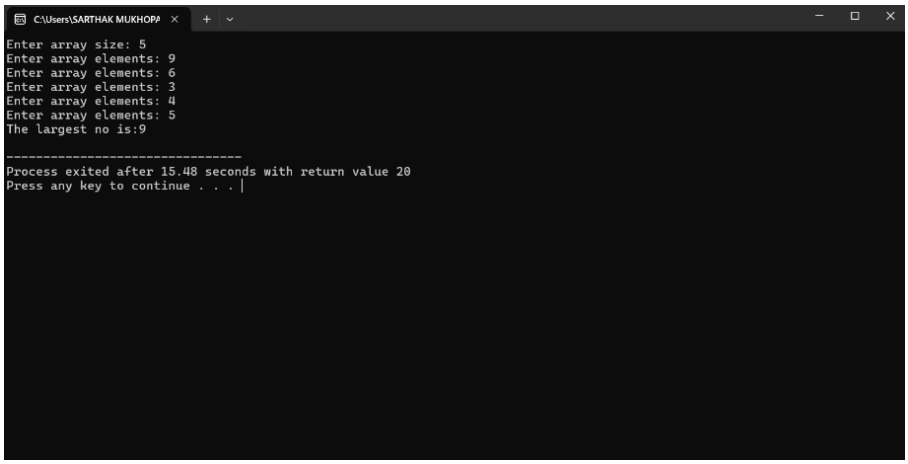
OUTPUT



Q2) Find largest element in array

```c
#include<stdio.h>
void main(){
    int i,n,largest;
    printf("Enter array size: ");
    scanf("%d",&n);
    int arr[n];
    for(i=0;i<n;i++){
        printf("Enter array elements: ");
        scanf("%d",&arr[i]);
    }
    largest=arr[0];
    for(i=1;i<n;i++){
        if(arr[i]>largest){
            largest=arr[i];
        }
    }
    printf("The largest no is:%d\n",largest);
}
```

OUTPUT

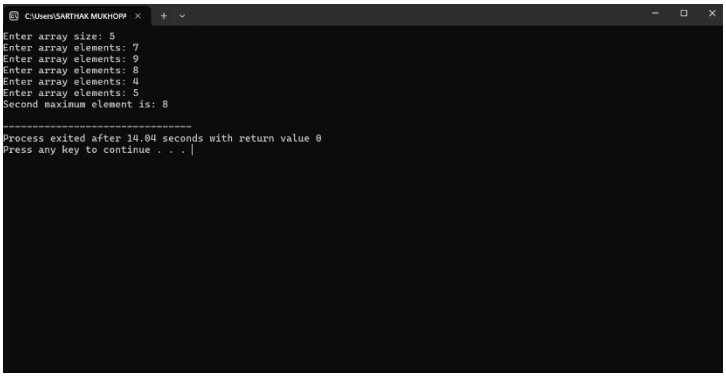


Q3) Find second largest element

```c
    scanf("%d",&arr[i]);
}
if (arr[0] > arr[1]) {
    max = arr[0];
    secmax = arr[1];
} else {
    max = arr[1];
    secmax = arr[0];
}
for(i = 2; i < n; i++) {
    if(arr[i] > max) {
        secmax = max;
        max = arr[i];
    } else if(arr[i] > secmax && arr[i] != max) {
        secmax = arr[i];
    }
}
printf("Second maximum element is: %d\n", secmax);
return 0;
}
```

```c
//input an array of n integer element and find the second highest among them
#include<stdio.h>
#include<limits.h>
int main(){
    int i,j,n,max,secmax;
    printf("Enter array size: ");
    scanf("%d",&n);
    int arr[n];
    for(i=0;i<n;i++){
        printf("Enter array elements: ");
        scanf("%d",&arr[i]);
    }
    if (arr[0] > arr[1]) {
        max = arr[0];
        secmax = arr[1];
    } else {
        max = arr[1];
        secmax = arr[0];
    }
    for(i = 2; i < n; i++) {
        if(arr[i] > max) {
            secmax = max;
            max = arr[i];
```

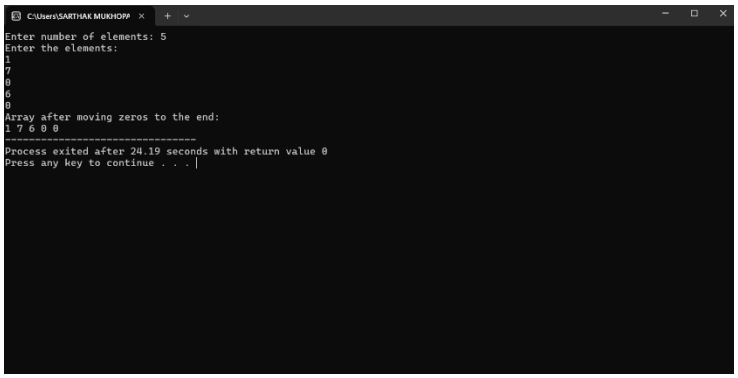
OUTPUT



Q4)Write a program to move all zeros to end

```c
#include <stdio.h>
int main() {
    int arr[100], n, i, j = 0;
    int result[100];

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

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

    for(i = 0; i < n; i++) {
        if(arr[i] != 0) {
            result[j] = arr[i];
            j++;
        }
    }

    while(j < n) {
        result[j] = 0;
        j++;
    }

    printf("Array after moving zeros to the end:\n");
    for(i = 0; i < n; i++) {
        printf("%d ", result[i]);
    }

    return 0;
}
```

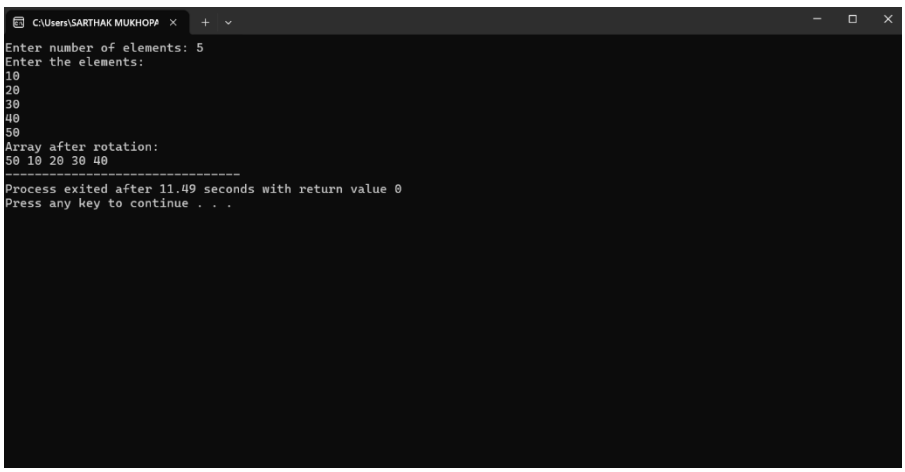
OUTPUT



Q5)Rotate array by one

```c
#include <stdio.h>
int main() {
    int arr[100], n, i, last;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter the elements:\n");
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    last = arr[n - 1];
    for(i = n - 1; i > 0; i--) {
        arr[i] = arr[i - 1];
    }
    arr[0] = last;
    printf("Array after rotation:\n");
    for(i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

OUTPUT



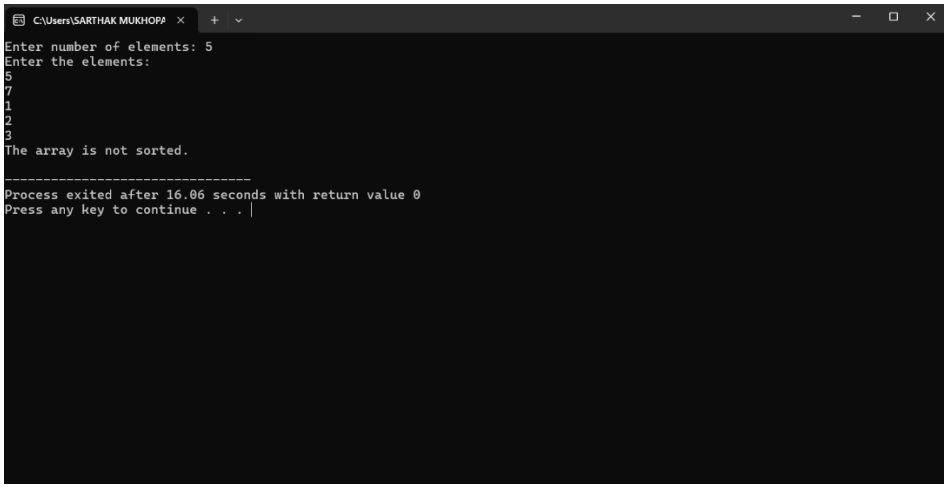
Q6)Write a program to check if array is sorted

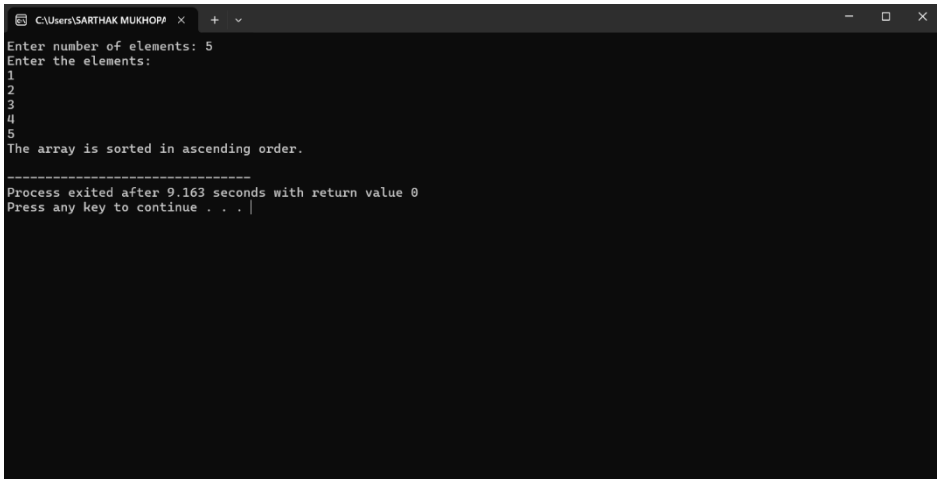
```c
#include <stdio.h>
int main() {
    int arr[100], n, i, isSorted = 1;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter the elements:\n");
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    // Check if array is sorted in ascending order
    for(i = 0; i < n - 1; i++) {
        if(arr[i] > arr[i + 1]) {
            isSorted = 0;
            break;
        }
    }
    if(isSorted) {
        printf("The array is sorted in ascending order.\n");
    } else {
        printf("The array is not sorted.\n");
    }
    return 0;
}
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 19 Sel: 0 Lines: 24 Length: 600 Insert Done parsing in 0.015 seconds

OUTPUT

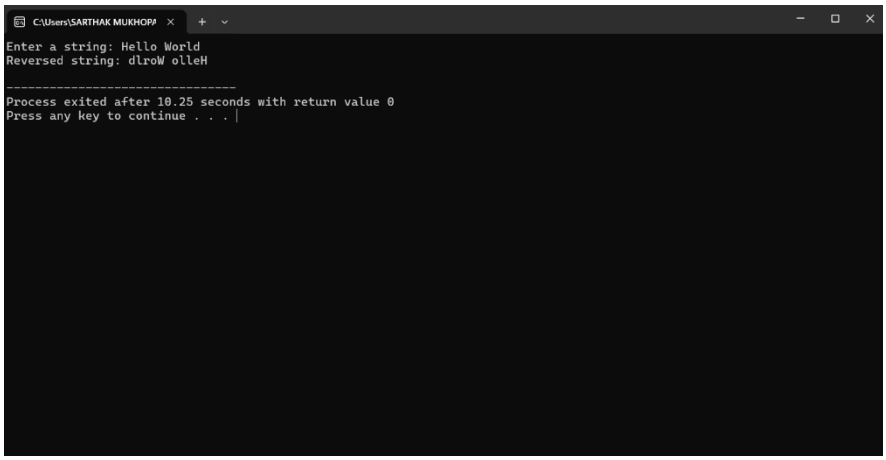




Q7)Write a program to reverse a string

```c
#include <stdio.h>
#include <string.h>
int main() {
    char str[100];
    int i, len;
    printf("Enter a string: ");
    gets(str);

    len = strlen(str);

    for(i = 0; i < len / 2; i++) {
        char temp = str[i];
        str[i] = str[len - i - 1];
        str[len - i - 1] = temp;
    }
    printf("Reversed string: %s\n", str);
    return 0;
}
```

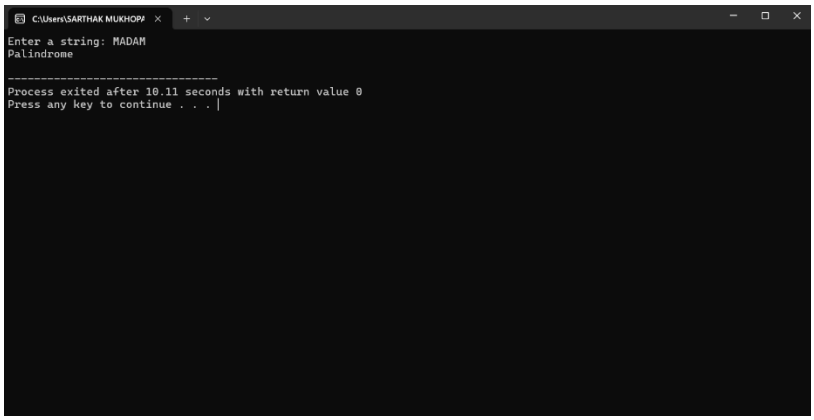
OUTPUT

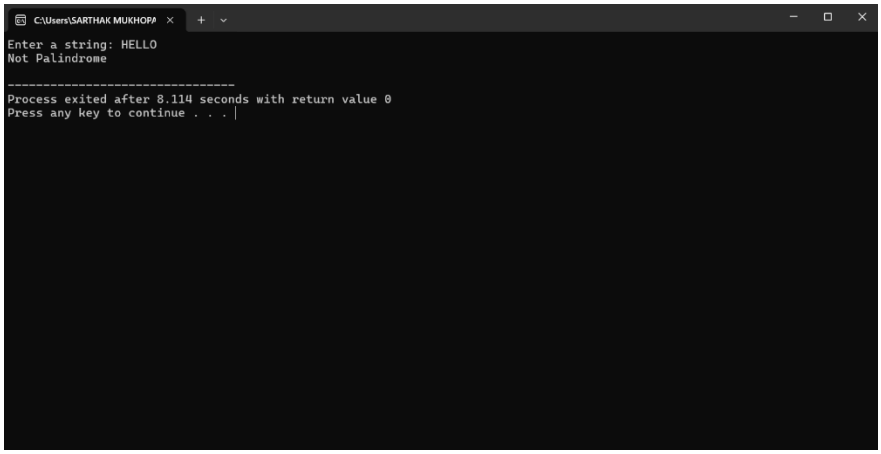


Q8) Write a program to check if a string is palindrome

```c
#include <stdio.h>
#include <string.h>
int main() {
    char str[100];
    printf("Enter a string: ");
    scanf("%s", str);

    int len = strlen(str), i, flag = 1;
    for(i = 0; i < len / 2; i++) {
        if(str[i] != str[len - i - 1]) {
            flag = 0;
            break;
        }
    }
    printf("%s\n", flag ? "Palindrome" : "Not Palindrome");
    return 0;
}
```

OUTPUT

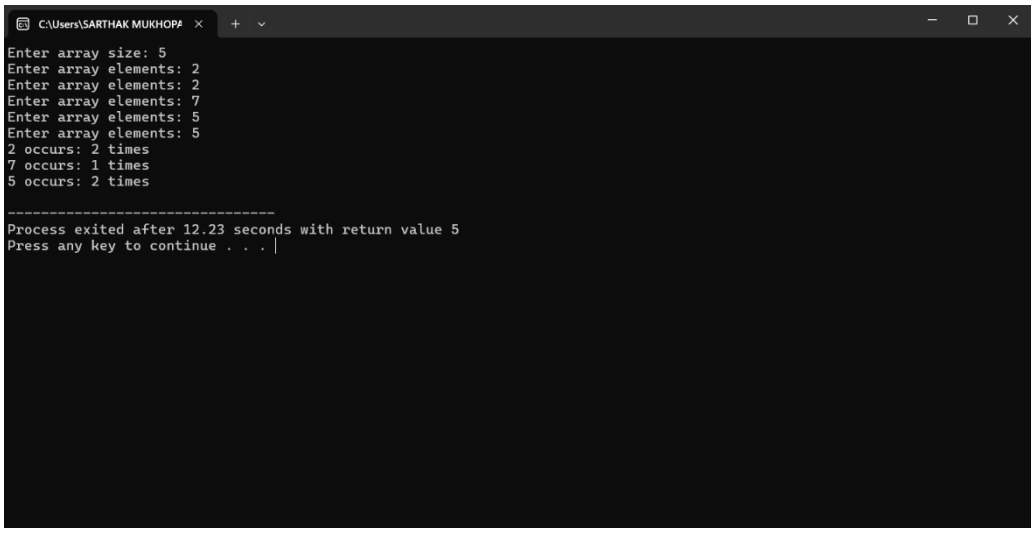




Q9)Write a program to count frequency of array elements

```c
#include<stdio.h>
void main(){
    int i,n,count,j;
    printf("Enter array size: ");
    scanf("%d",&n);
    int arr1[n];
    int arr2[n];
    for(i=0;i<n;i++){
        printf("Enter array elements: ");
        scanf("%d",&arr1[i]);
        arr2[i]=-1;
    }
    for(i=0;i<n;i++){
        count=1;
        for(j=i+1;j<n;j++){
            if(arr1[i]==arr1[j]){
                count++;
                arr2[j]=0;
            }
        }
        if(arr2[i]!=0){
            arr2[i]=count;
        }
    }
    for(i=0;i<n;i++){
        if(arr2[i]!=0)
            printf("%d occurs: %d times\n",arr1[i],arr2[i]);
    }
}
```

OUTPUT



Q10)Reverse of an array

```c
#include<stdio.h>
int main(){
    int i,n,j,temp;
    printf("Enter the size: ");
    scanf("%d",&n);
    int a[n];
    for(i=0;i<n;i++){
        printf("Enter the elements: ");
        scanf("%d",&a[i]);
    }
    for(i=0,j=n-1;i<n/2;i++,j--){
        temp=a[i];
        a[i]=a[j];
        a[j]=temp;
    }
    for(i=0;i<n;i++){
        printf("%d",a[i]);
    }

}
```

Compiler Resources Compile Log Debug Find Results

Line: 15 Col: 6 Sel: 0 Lines: 20 Length: 321 Insert Done parsing in 0.016 seconds

OUTPUT