**Practical : 3**

**Aim: Implement a function of sequential search and count the steps executed by function on various inputs for best case and worst case. Also write complexity in each case and draw a comparative chart.**

**Program**:

#include <stdio.h>

#include<stdlib.h>

int count;

int sequential\_search(int arr[], int n ,int ele)

{

int pos=0;

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

{

count++;

if(arr[i]==ele)

{

count++;

int pos=i;

count++;

return pos;

}

}

return -1;

}

int main()

{

int ele,n=10,position;

count=0;

int a[n];

// int a[]={27,10,24,56,57,12,32,87,99,89};

printf("Array Element:");

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

{

a[i] = rand()%100;

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

}

printf("\nEnter any element to search : ");

scanf("%d",&ele);

position = sequential\_search(a,n,ele);

if(position==-1){

printf("\nElement not found");

}

else {

printf("Element is at %d",position+1);

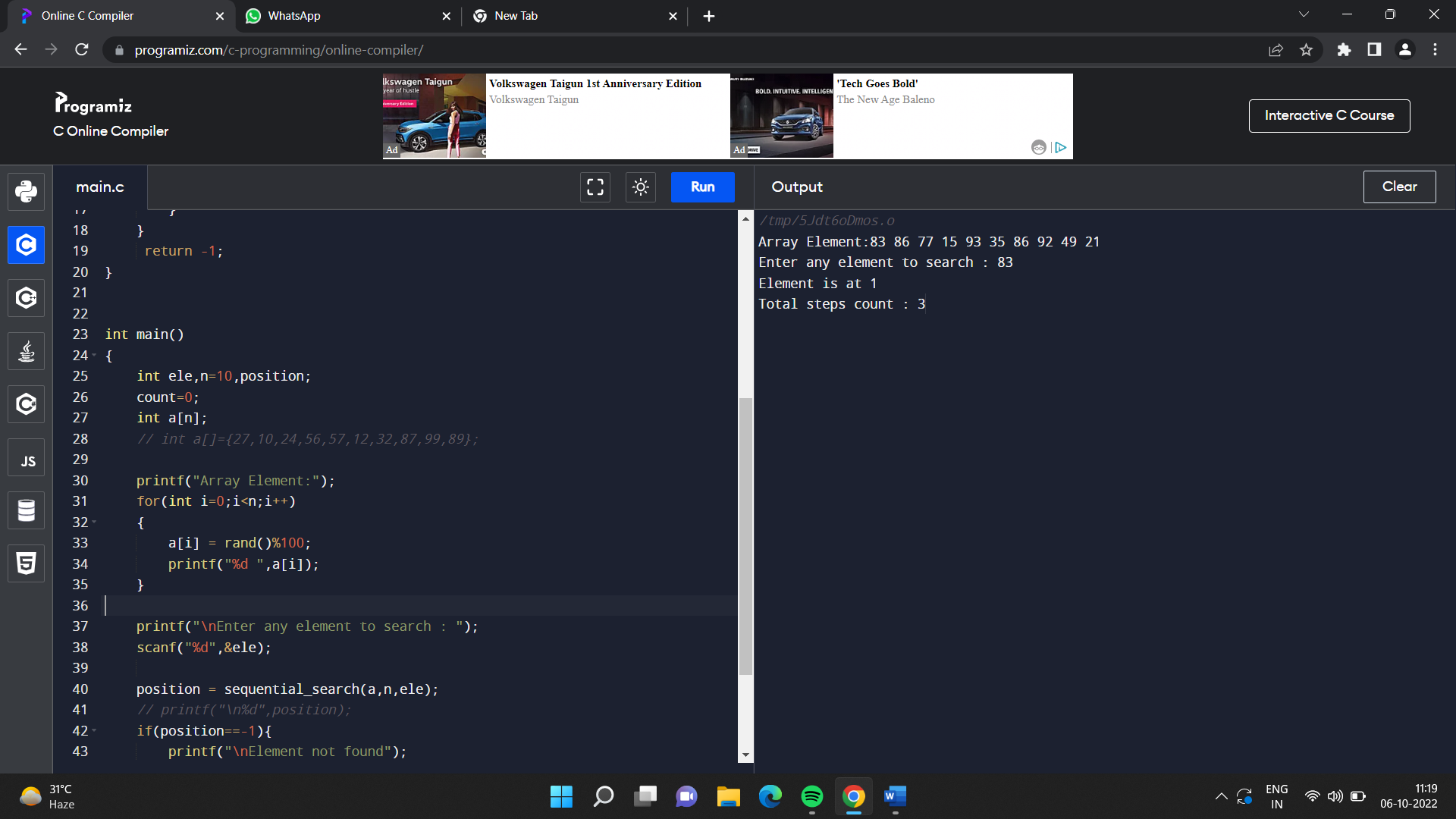
}

printf("\nTotal steps count : %d",count);

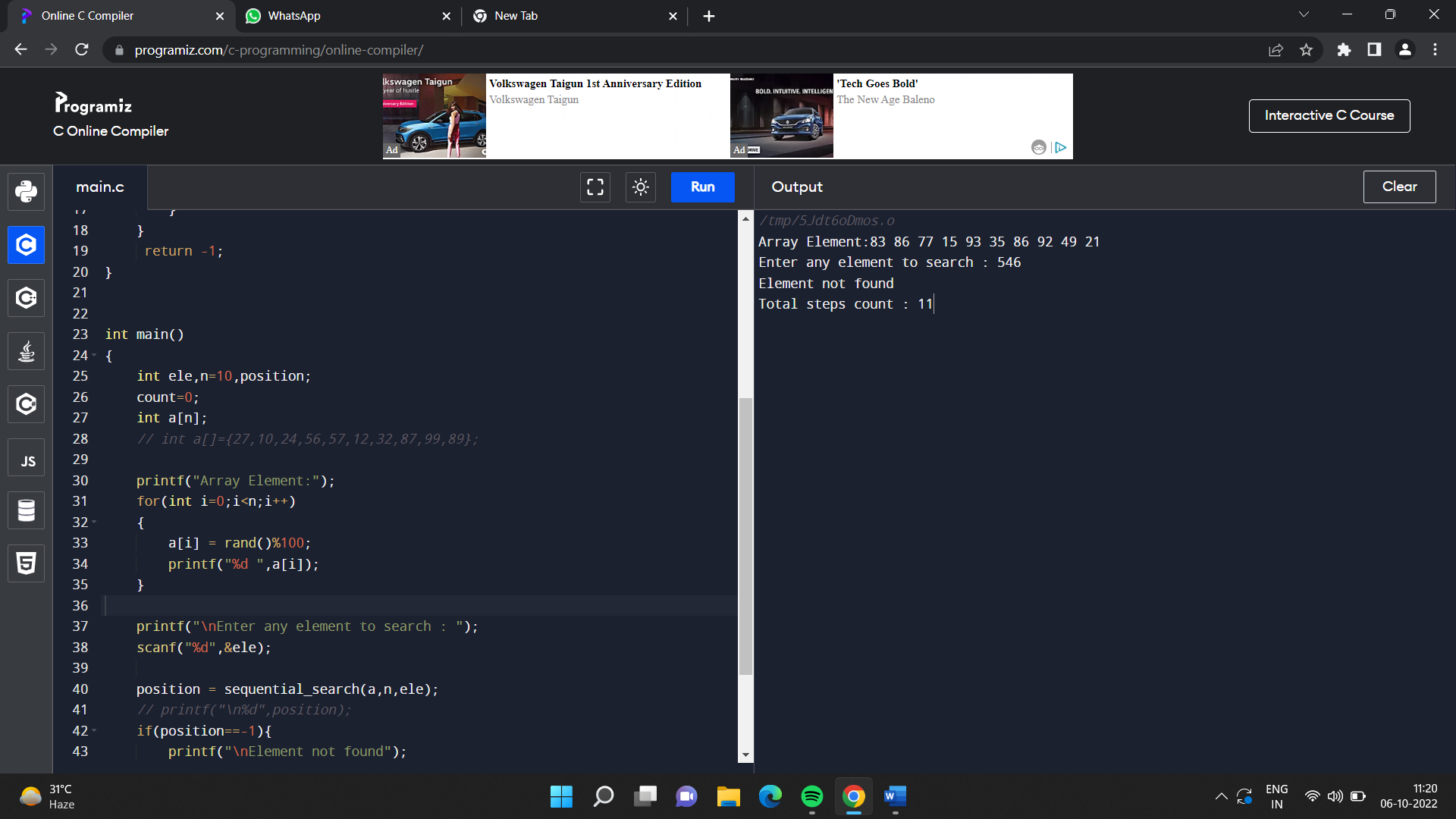
}

* **Output:**

Best Case:



Worst Case:



|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **N** | **Steps count** | |
| **Best Case** | **Worst Case** |
| 1 | 5 | 3 | 7 |
| 2 | 10 | 3 | 11 |
| 3 | 20 | 3 | 22 |
| 4 | 50 | 3 | 52 |
| 5 | 100 | 3 | 102 |

**Time Complexity:**

Best Case - O(1)

Worst Case- O(n)

**Data Analysis:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Understanding of a Practical (3) | Implementation of a Practical (5) | Documentation (2) | Total (10) | Sign |
|  |  |  |  |  |