**ADA LAB-2**

* **BINARY SEARCH**
  + - * **WRITTEN PROGRAM**

//recursive binary search

#include<iostream>

using namespace std;

int binarySearch(int arr[], int p, int r, int num)

{

if (p <= r)

{

int mid = (p + r)/2;

if (arr[mid] == num)

return mid ;

if (arr[mid] > num)

return binarySearch(arr, p, mid-1, num);

if (arr[mid] > num)

return binarySearch(arr, mid+1, r, num);

}

return -1;

}

int main(void)

{

int arr[100],size,x,i;

cout<<"enter size of the array: ";

cin>>size;

cout<<"enter elements in the array: "<<endl;

for(i=0;i<=size-1;i++)

{

cout<<"arr["<<i<<"]: ";

cin>>arr[i];

}

cout<<"enter the element to be searched for: "<<endl;

cin>>x;

int index = binarySearch (arr, 0, size-1, x);

if(index == -1)

cout<<" element is not present";

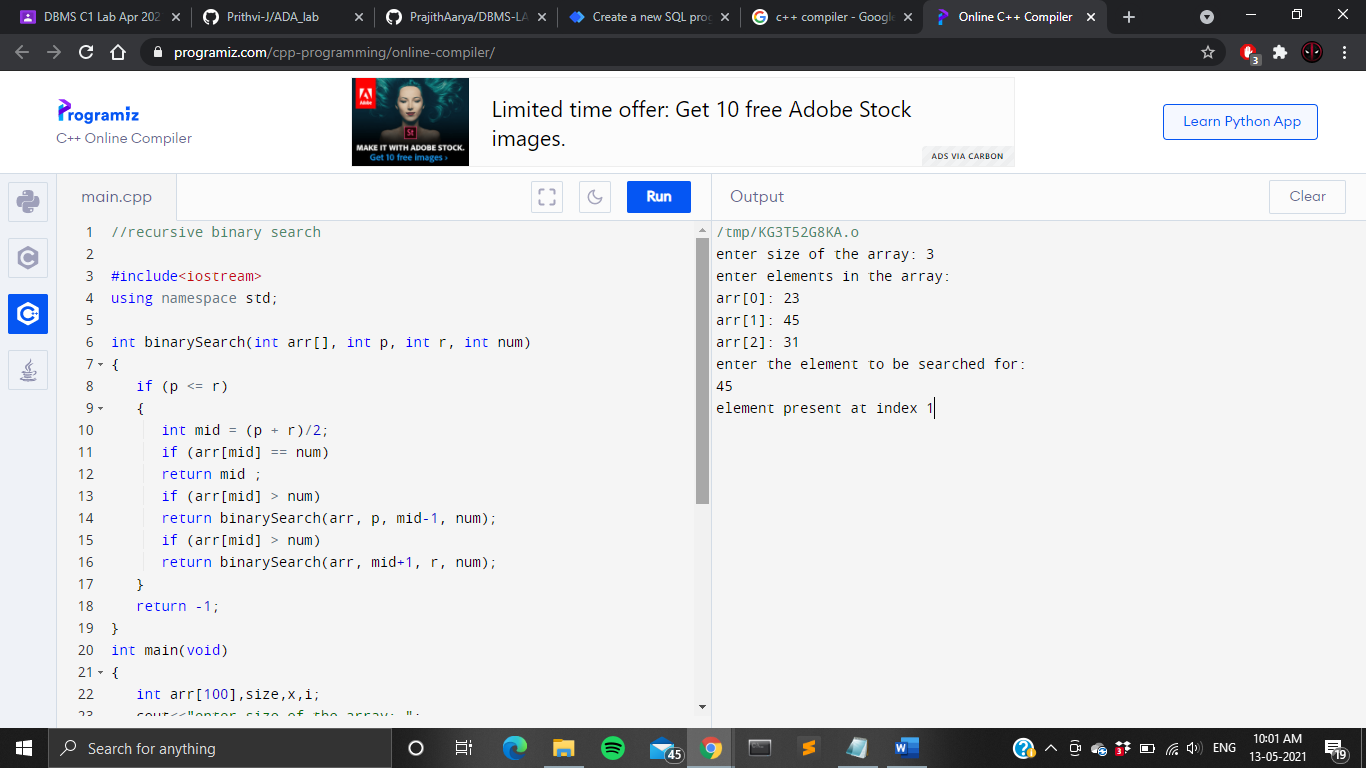
else

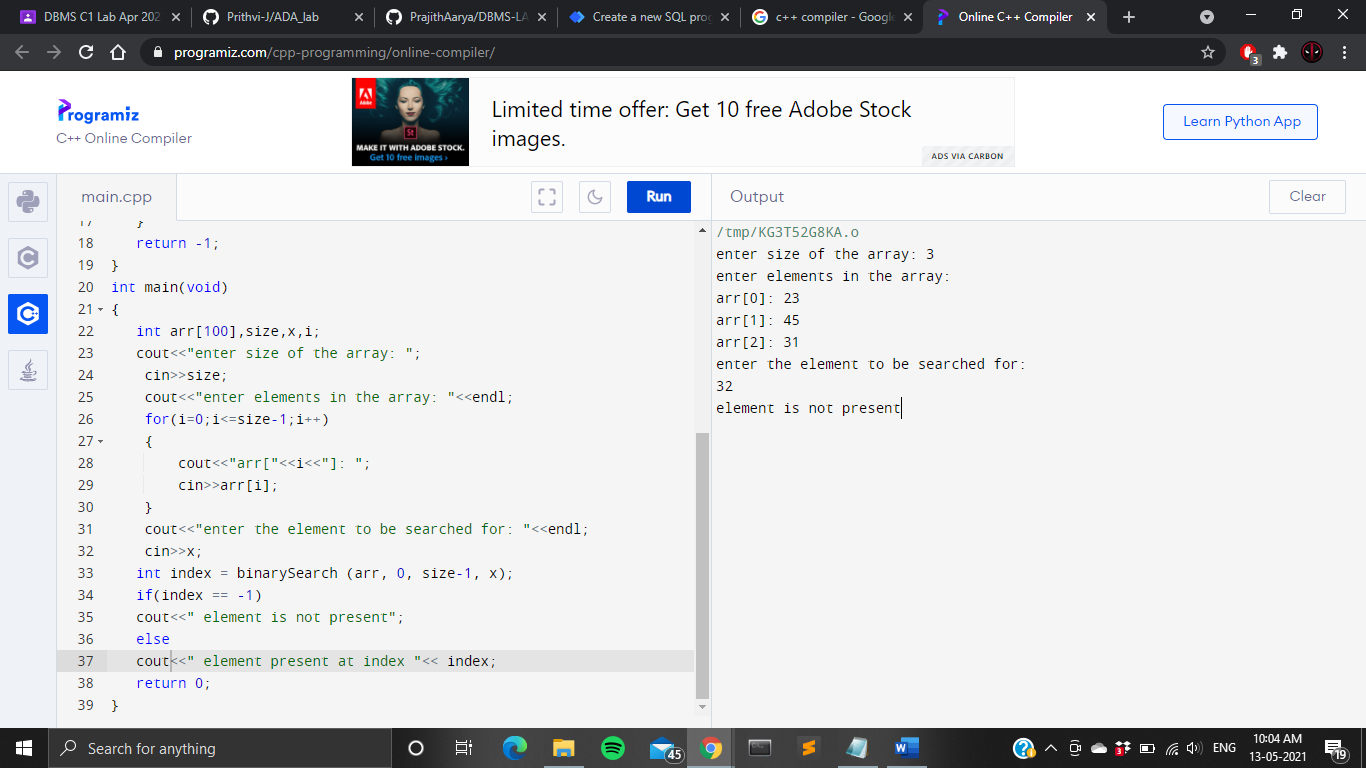
cout<< " element present at index "<< index;

return 0;

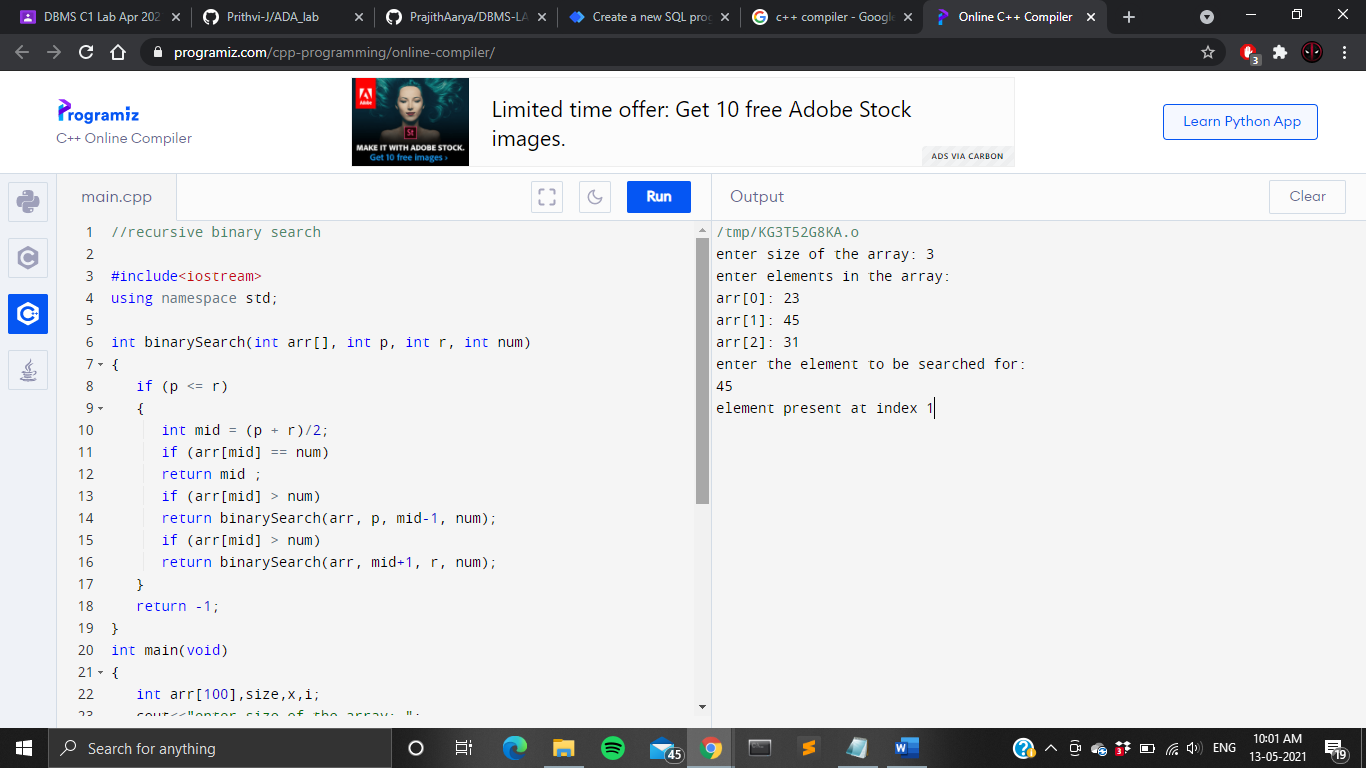
}

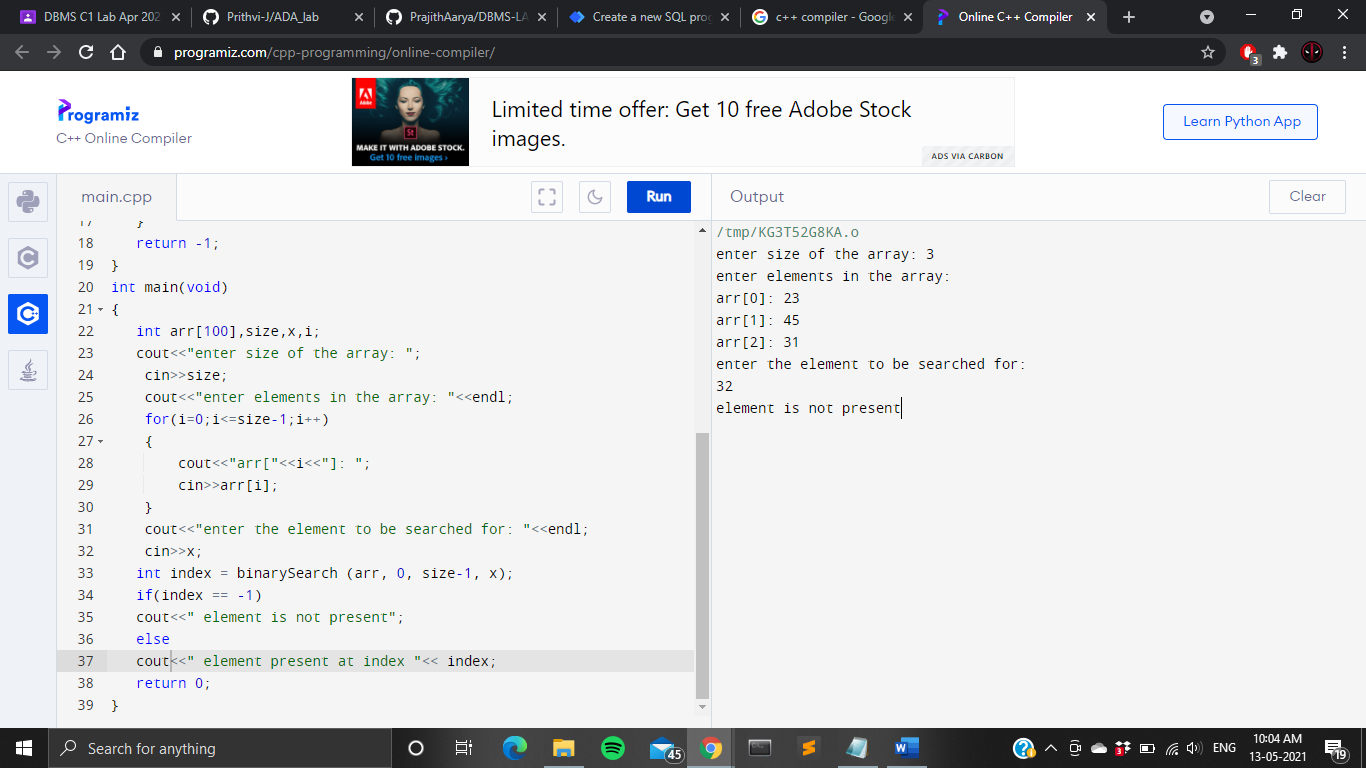
* + - * **PROGRAM**





* + - * **OUTPUT**





* **LINEAR SEARCH**
  + - * **WRITTEN PROGRAM**

//recursive linear search

#include <iostream>

using namespace std;

int linSearch(int arr[], int l,int r, int x)

{

if (r < l)

return -1;

if (arr[l] == x)

return l;

if (arr[r] == x)

return r;

return linSearch(arr, l + 1,r - 1, x);

}

int main()

{

int arr[100] , i, x, size;

cout<<"enter size of the array: ";

cin>>size;

cout<<"enter elements in the array: "<<endl;

for(i=0;i<=size-1;i++)

{

cout<<"arr["<<i<<"]: ";

cin>>arr[i];

}

cout<<"enter the element to be searched for: "<<endl;

cin>>x;

int index = linSearch(arr, 0, size - 1, x);

if (index != -1)

cout << "Element " << x<< " is present at index "<< index;

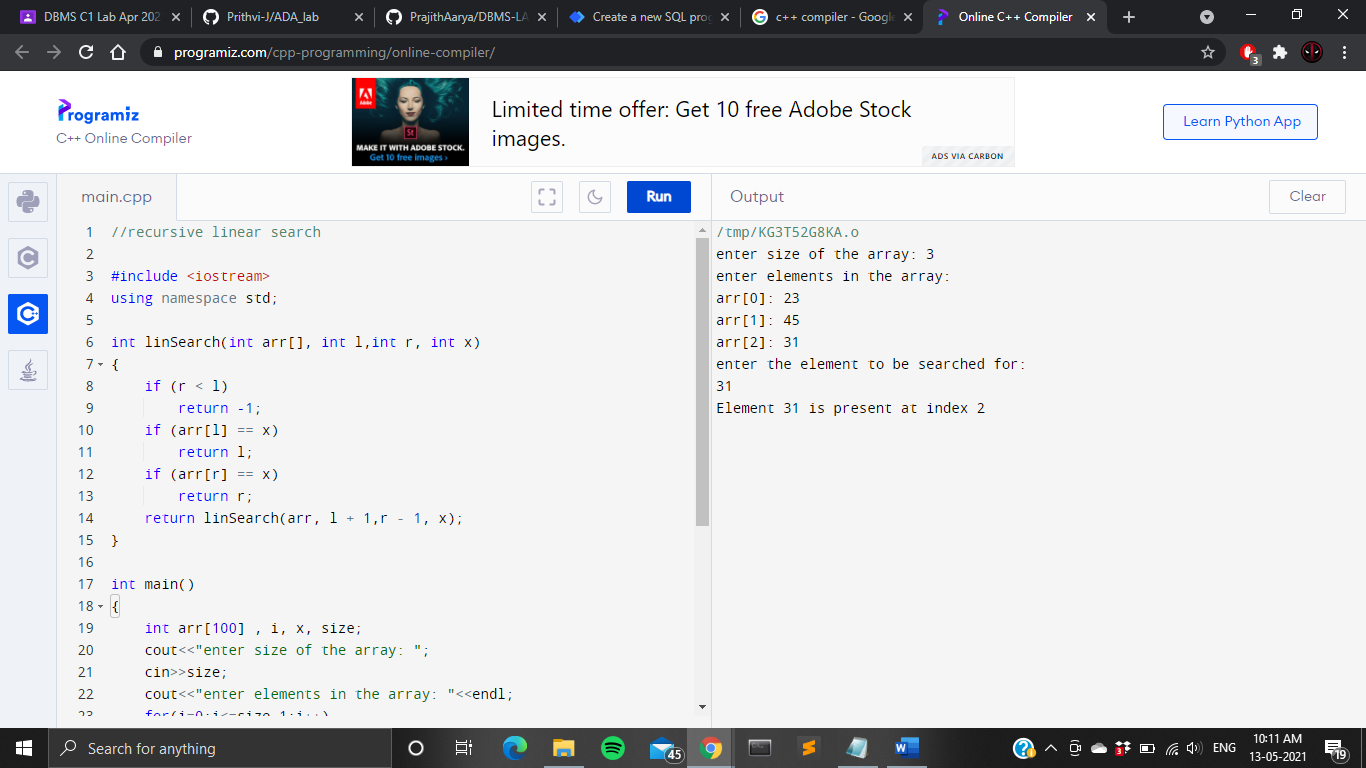
else

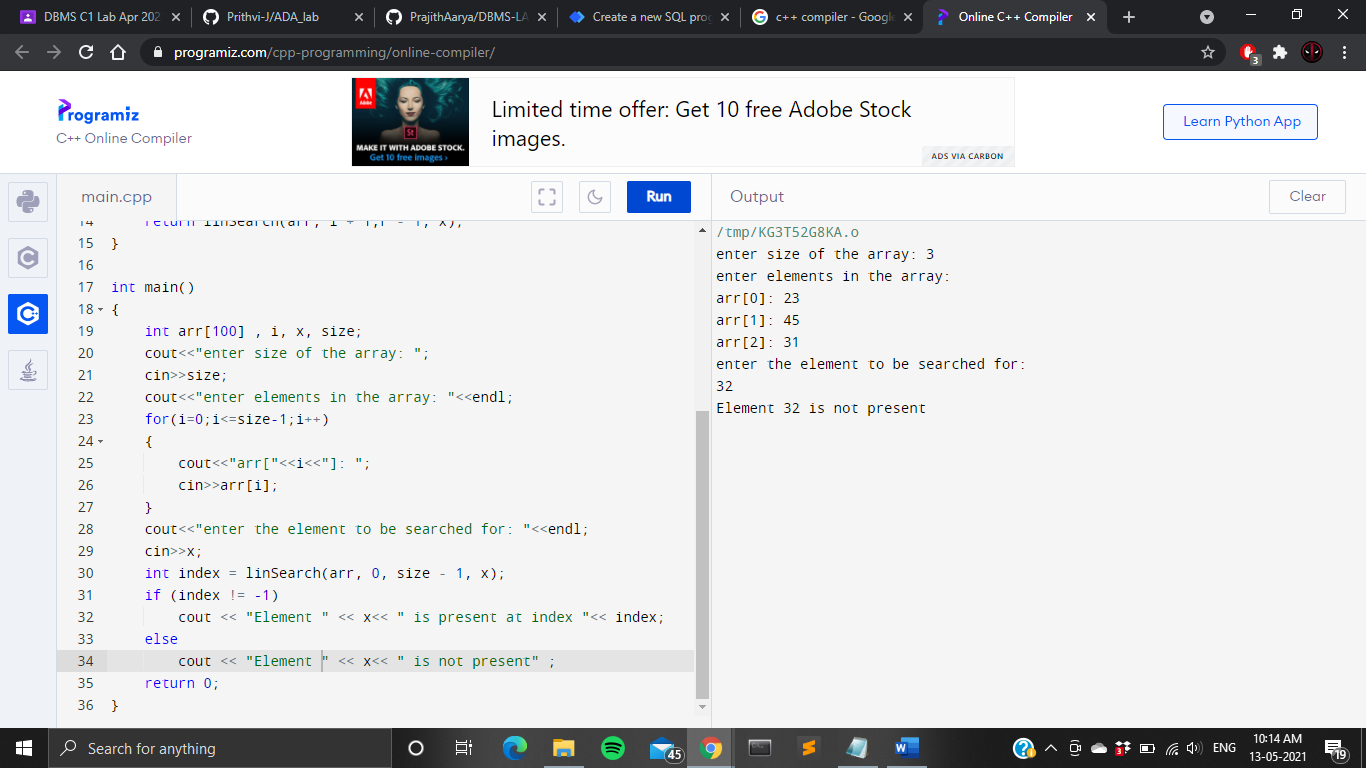
cout << "Element" << x<< " is not present" ;

return 0;

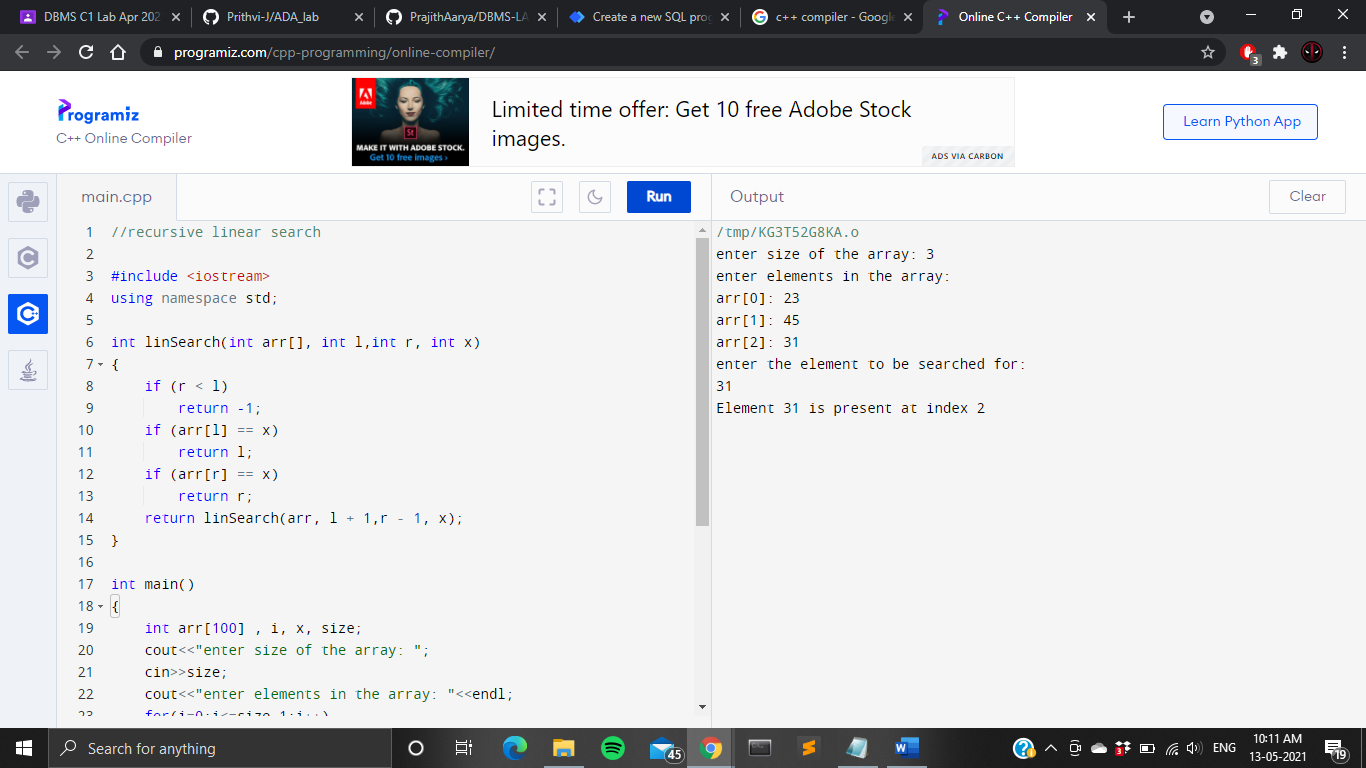
}

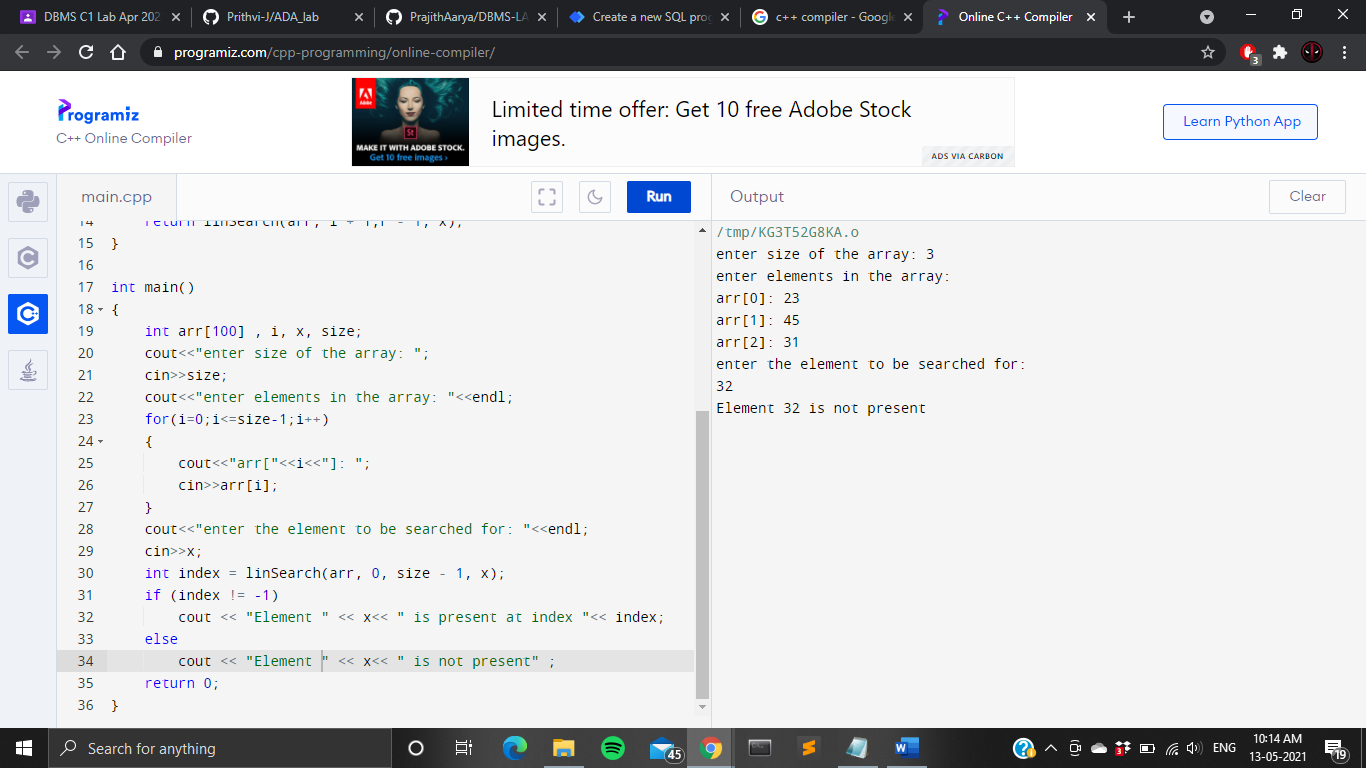
* + - * **PROGRAM**





* + - * **OUTPUT**





* **FIBONACCI SERIES**
  + **WRITTEN PROGRAM**

#include <iostream>

using namespace std;

int fib(int x)

{

if((x==1)||(x==0))

{

return(x);

}

else

{

return(fib(x-1)+fib(x-2));

}

}

int main()

{

int x,i=0;

cout<<"Enter the number of terms of series : ";

cin>>x;

cout<<"\nFibonnaci Series : ";

while(i<x)

{

cout<<" "<<fib(i);

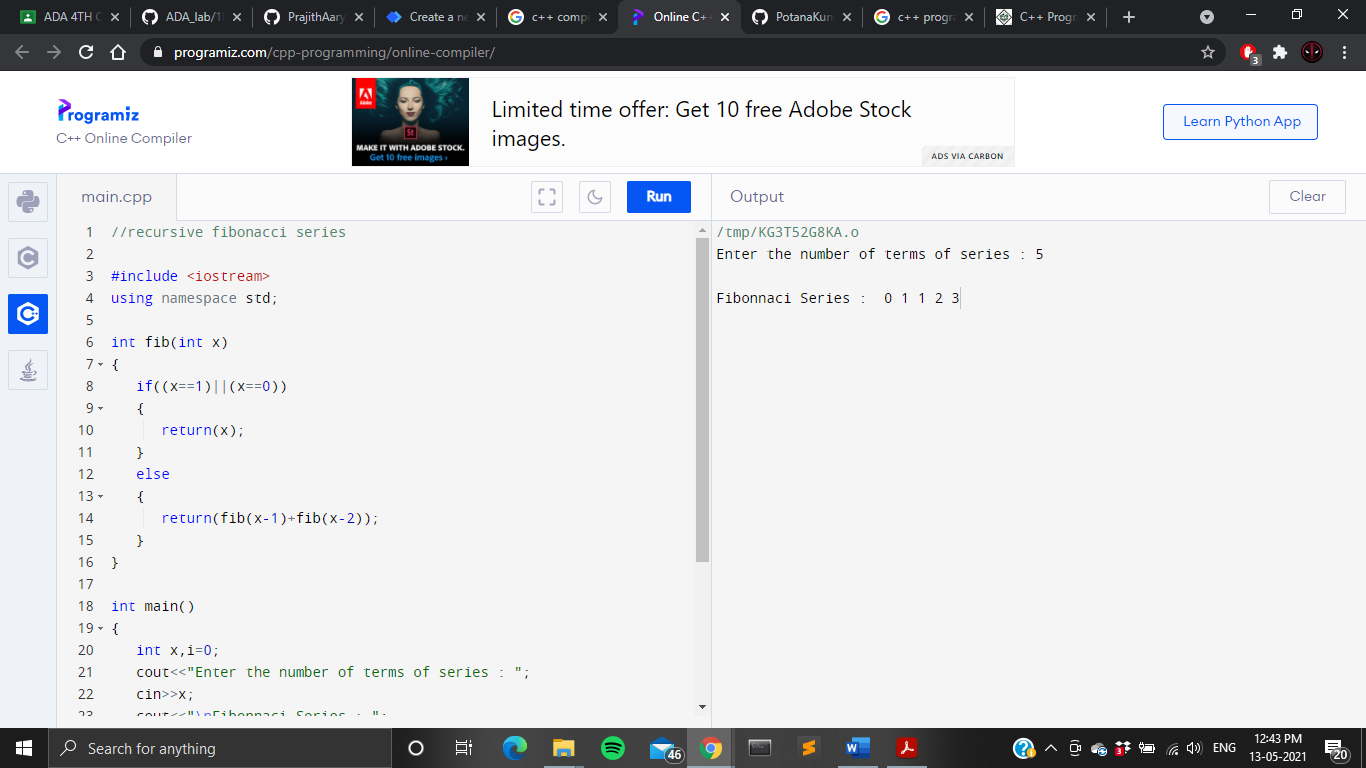
i++;

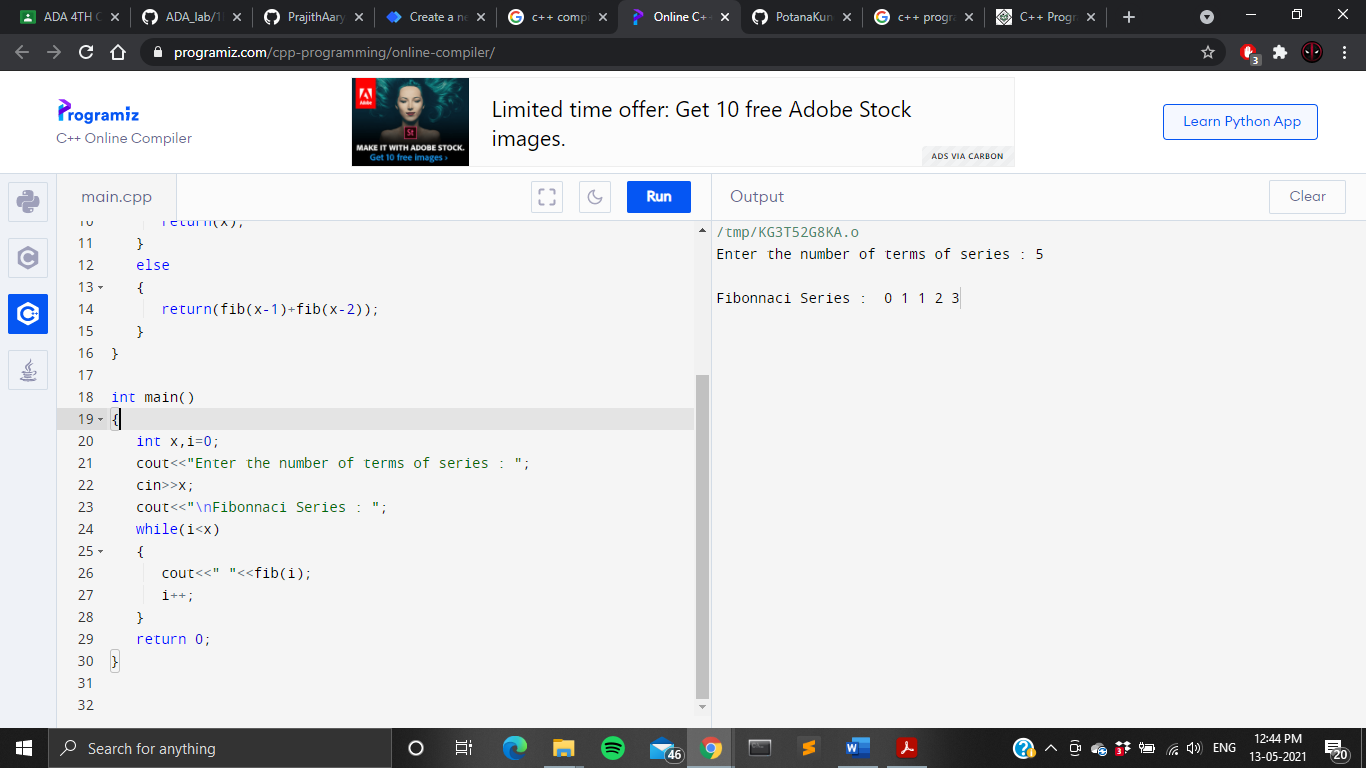
}

return 0;

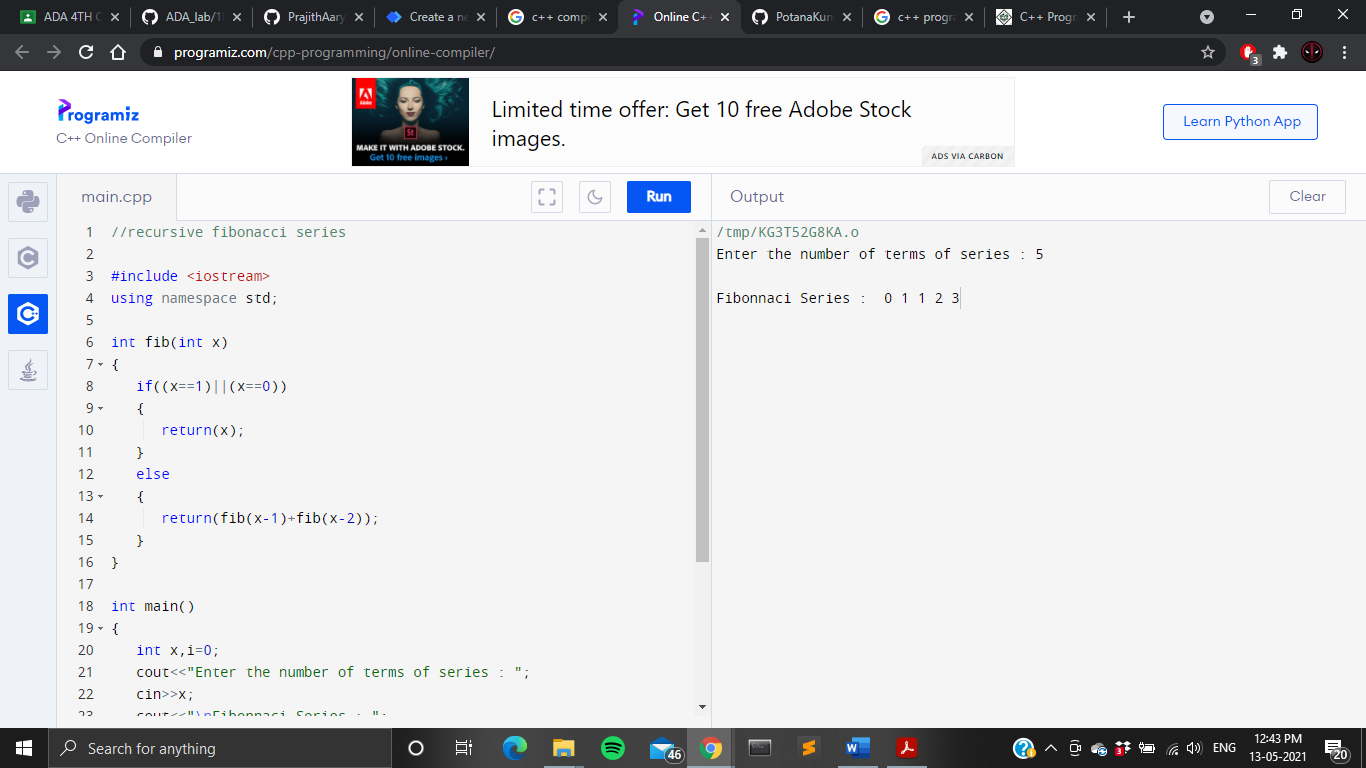
}

* + **PROGRAM**





* + **OUTPUT**



* **RECURSIVE BUBBLE SORT** 
  + **WRITTEN PROGRAM**

#include <iostream>

using namespace std;

void bubbleSort(int arr[], int n)

{

int temp;

if (n == 1)

return;

for (int i=0; i<n-1; i++)

if (arr[i] > arr[i+1])

{

temp=arr[i];

arr[i]=arr[i+1];

arr[i+1]=temp;

}

bubbleSort(arr, n-1);

}

int main()

{

int n ,arr[100];

cout<<"Enter no.of elements : ";

cin>>n;

cout<<"Enter the elements in the array : "<<endl;

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

cin>>arr[i];

cout<<endl;

bubbleSort(arr, n);

cout<<"Sorted array : "<<endl;

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

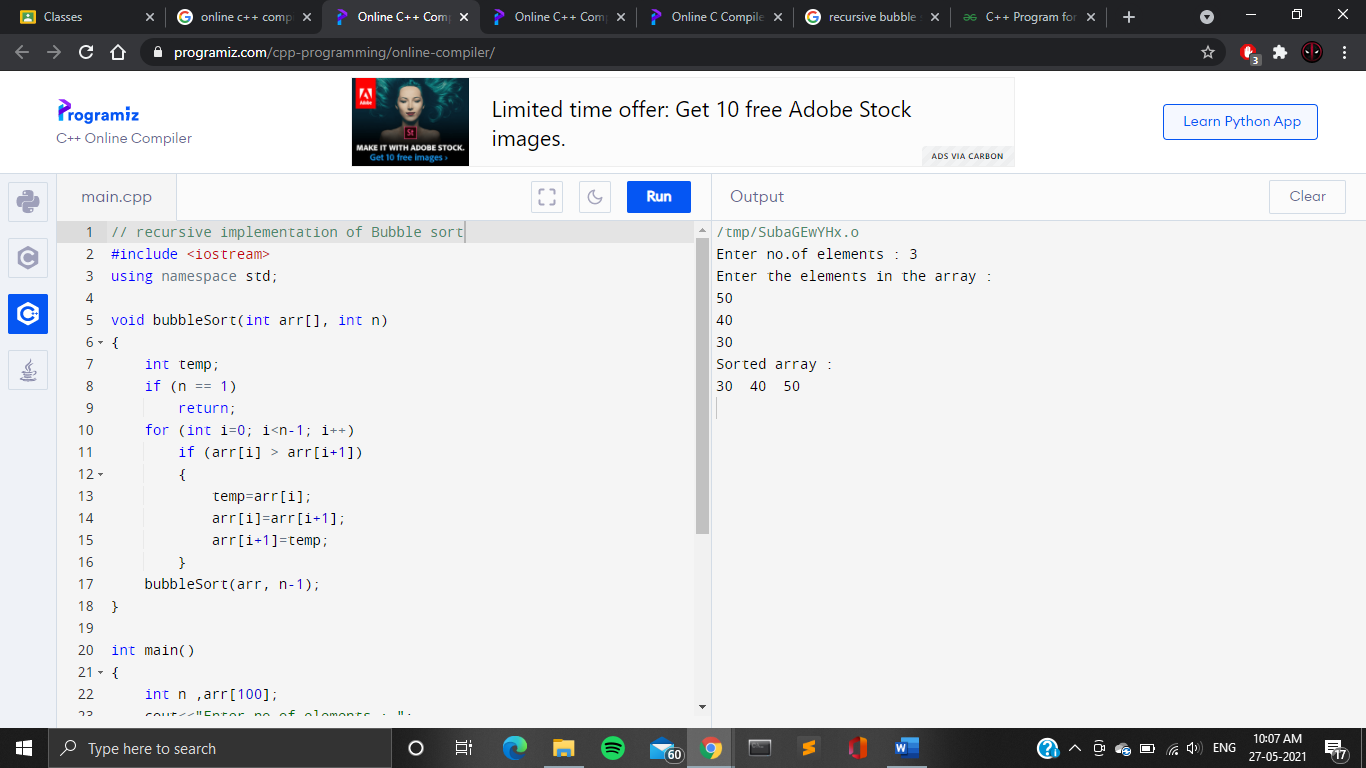
cout<<arr[i]<<"\t";

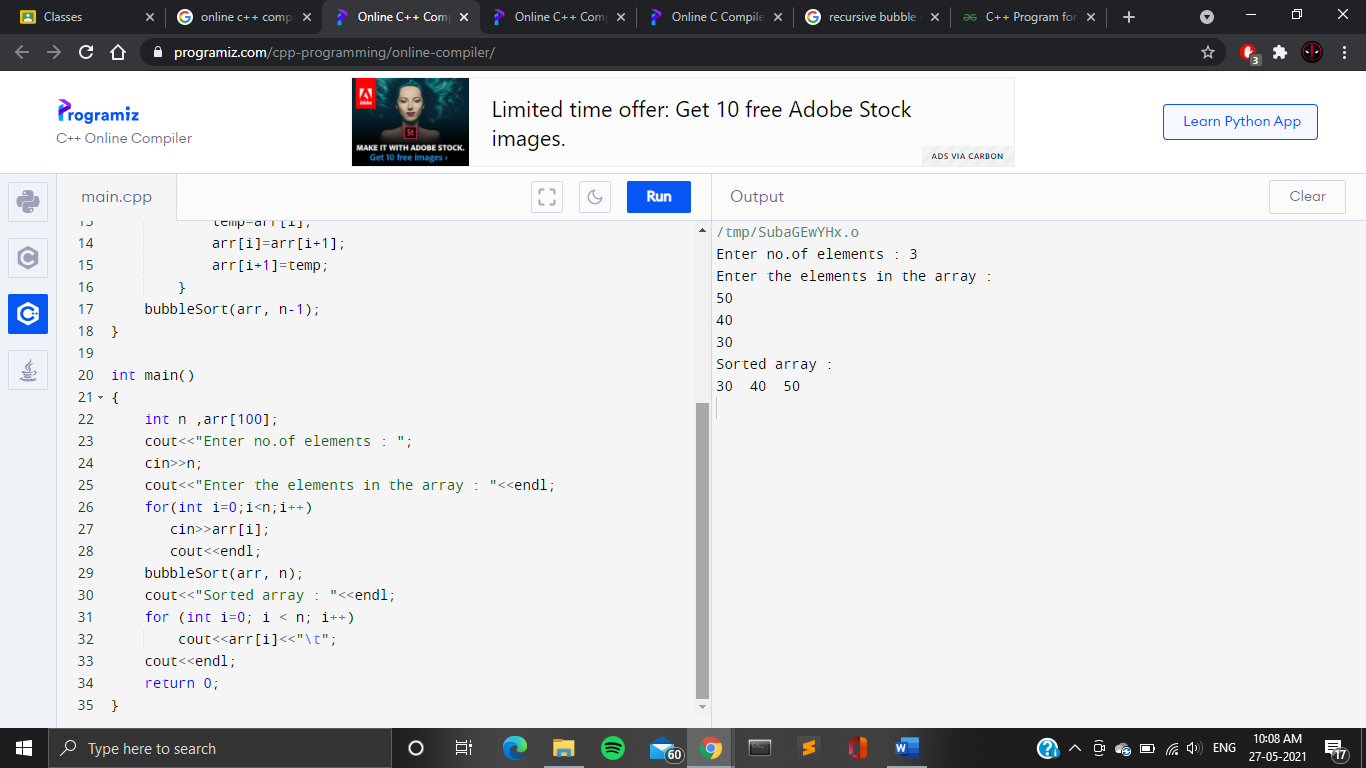
cout<<endl;

return 0;

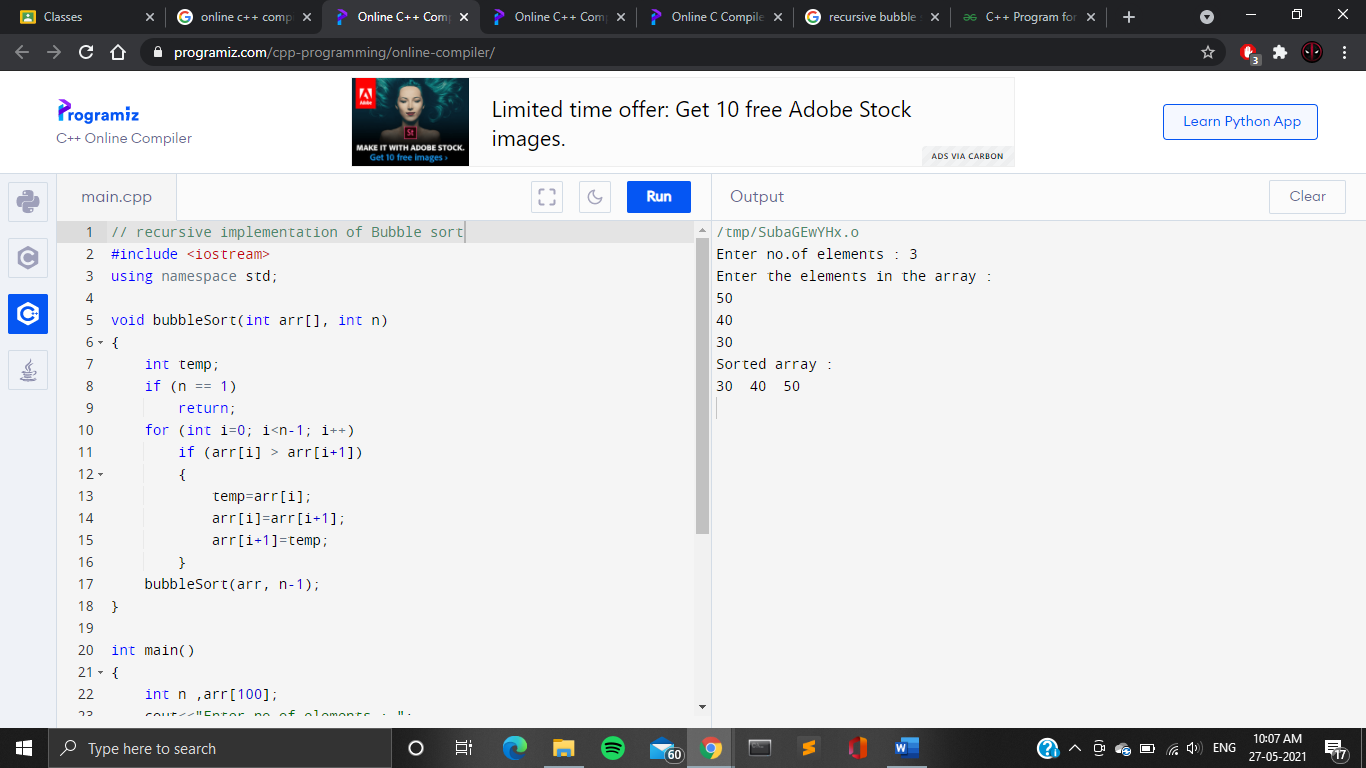
}

* + **PROGRAM**





* + **OUTPUT**



* **RECURSIVE SELECTION SORT**
  + **WRITTEN PROGRAM**

#include <iostream>

using namespace std;

void insertionSort(int arr[], int n)

{

if (n <= 1)

return;

insertionSort( arr, n-1 );

int last = arr[n-1];

int j = n-2;

while (j >= 0 && arr[j] > last)

{

arr[j+1] = arr[j];

j--;

}

arr[j+1] = last;

}

int main()

{

int n ,arr[100];

cout<<"Enter no.of elements : ";

cin>>n;

cout<<"Enter the elements in the array : "<<endl;

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

cin>>arr[i];

cout<<endl;

insertionSort(arr, n);

cout<<"Sorted array : "<<endl;

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

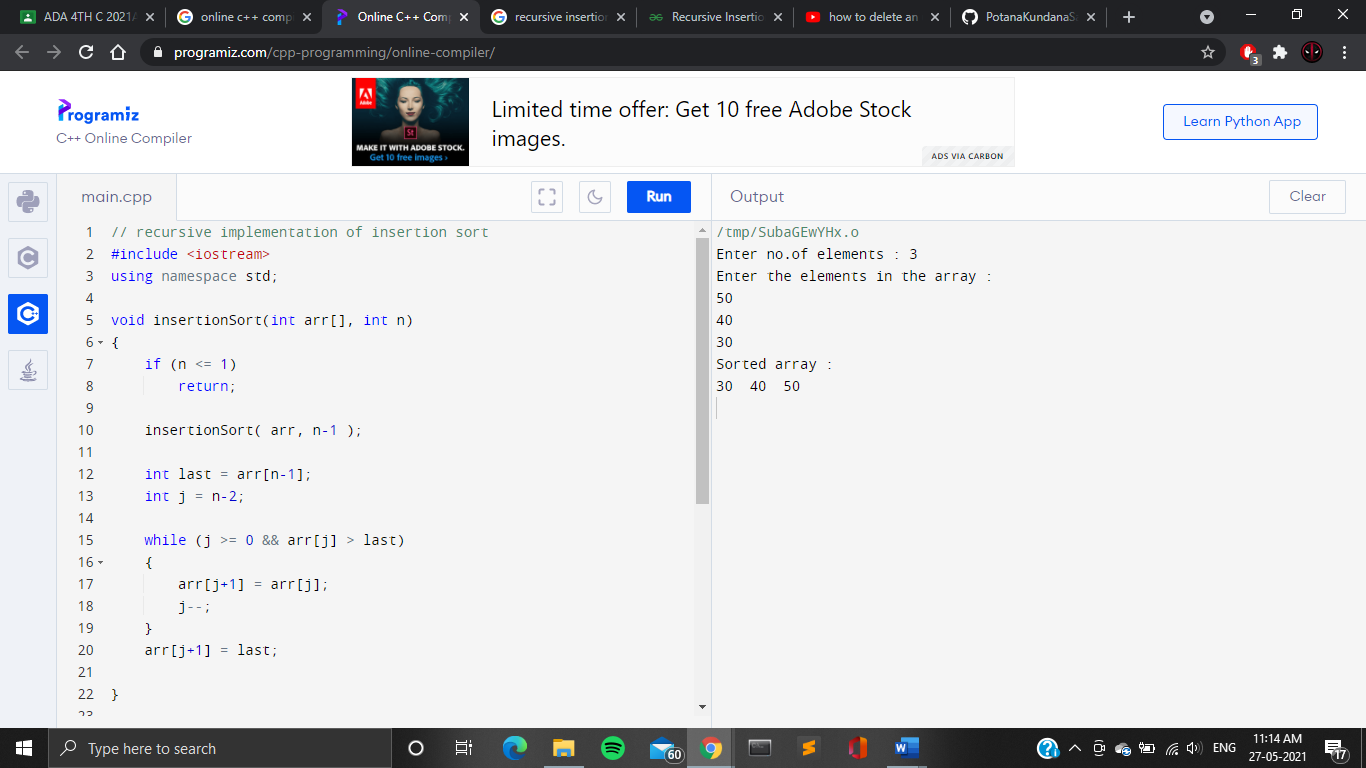
cout<<arr[i]<<"\t";

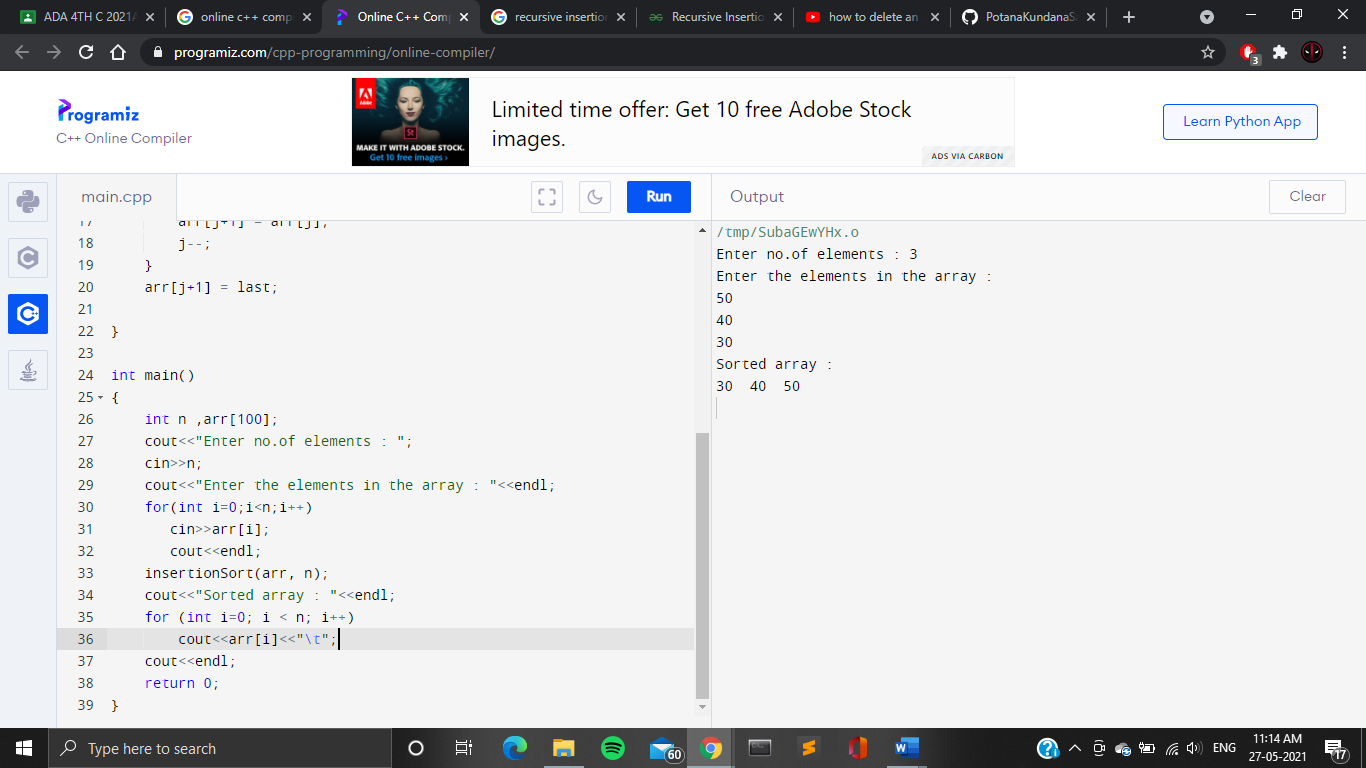
cout<<endl;

return 0;

}

* + **PROGRAM**





* + **OUTPUT**

