**Test**

Q1. Program to access 2D Array Elements Using Pointer

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int m, n, k = 0;**

**cout << "Enter the size of the row: " << endl;**

**cin>>m;**

**cout << "Enter the size of the coloumn: " << endl;**

**cin>>n;**

**int\* arr = new int[m \* n];**

**for (int i = 0; i < m; i++) {**

**for (int j = 0; j < n; j++) {**

**\*(arr + i \* n + j) = ++k;**

**}**

**}**

**for (int i = 0; i < m; i++) {**

**for (int j = 0; j < n; j++) {**

**cout << \*(arr + i \* n + j)<< " ";**

**}**

**cout << endl;**

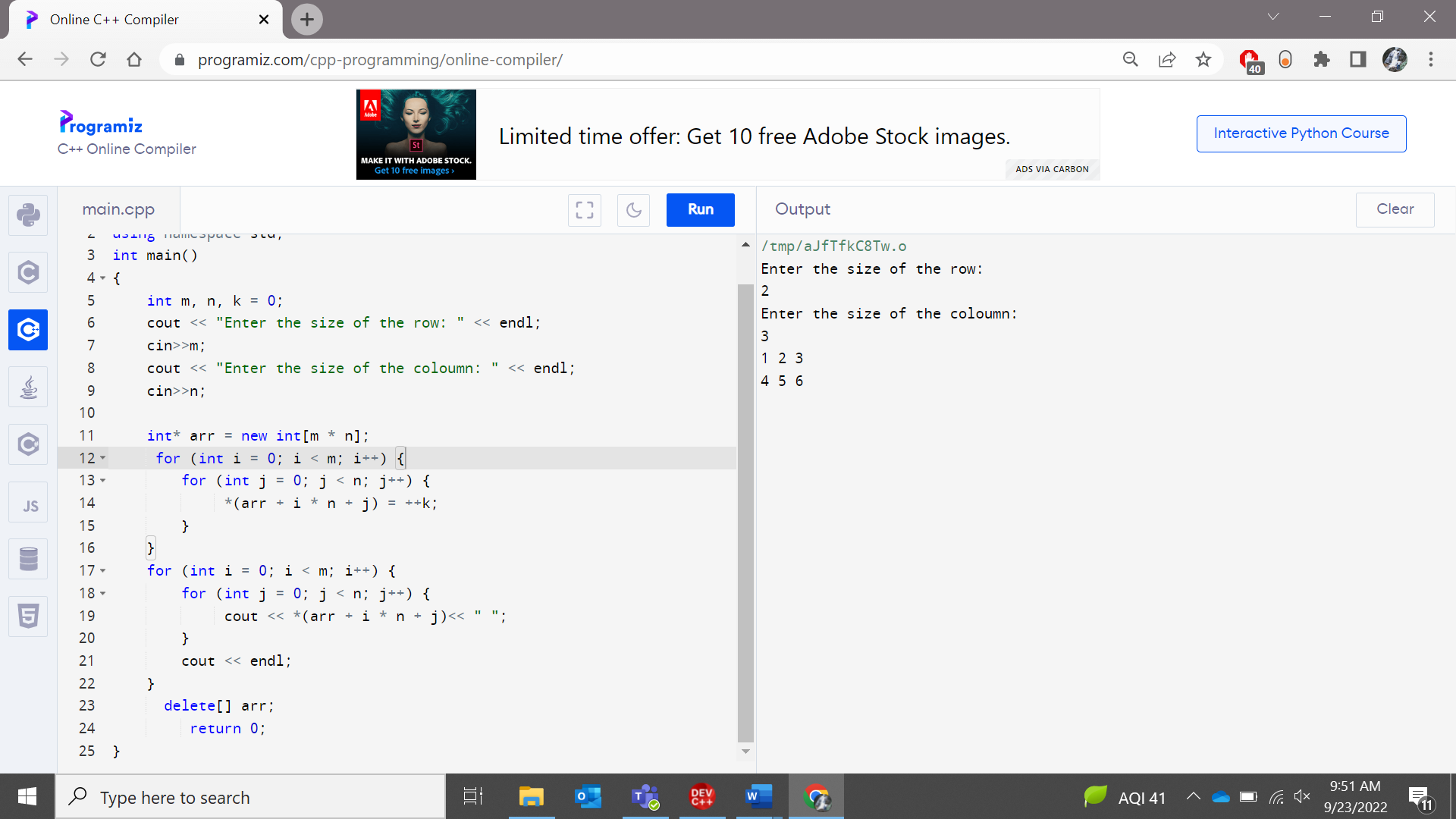
**}**

**delete[] arr;**

**return 0;**

**}**

**Output:**



Q2. Given an integer array num, return true if any value appears at least twice in the array, and return false if every element is distinct.

Input: num = [-1,2,3,4,1] Output: false

Input: num = [1,5,3,1] Output: true

**Code:**

**Output:**

#include <iostream>

using namespace std;

bool check(int num[], int n, int k)

{

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

int j = i + 1;

int range = k;

while (range > 0 and j < n) {

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

return true;

j++;

range--;

}

}

return false;

}

int main()

{

int n;

cout << "Enter the size of the array: " << endl;

cin>>n;

int num[n];

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

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

{

cin>>num[k];

}

if (check(num, n, 1)==true)

{

cout << "true" << endl;

}

else

cout << "false" << endl;

}

Graphical user interface, text, application

Description automatically generated

Q3. create a pointer array of size 5 Input:[10,20,30,40,50] Print the following outputs using pointer

1. First element of array

2. do an arithmetic operation to array element to get value of 15

3. print the second and third element using 2 different ways.

Code:

Output:

#include <iostream>

using namespace std;

int main()

{

int \*p;

int (\*ptr)[5];

int arr[5];

cout << "Enter the elements of the array " << endl;

for(int i=0;i<5;i++){

cin>>arr[i];

}

cout << "The first element is " <<arr[0] <<endl;

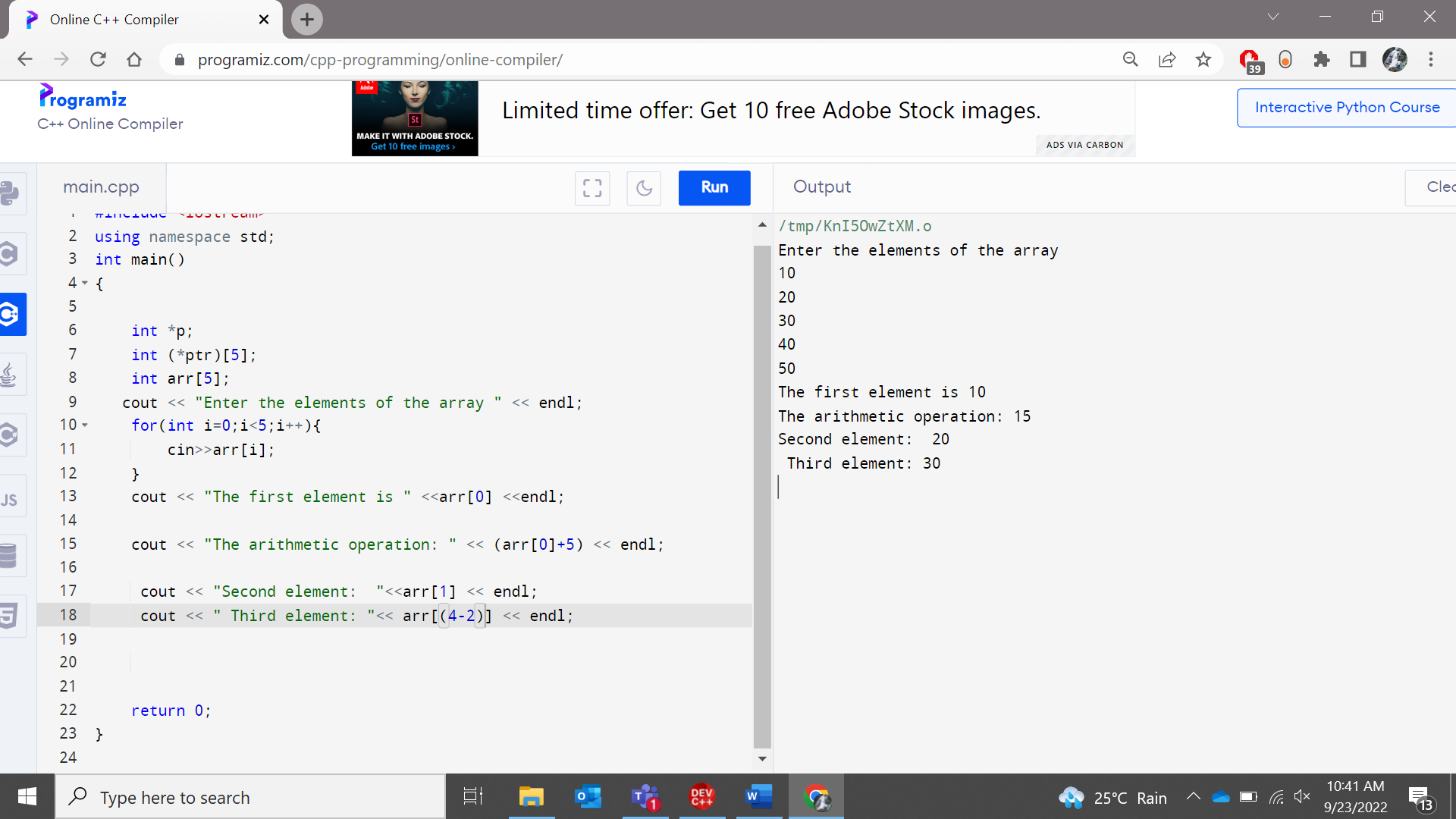
cout << "The arithmetic operation: " << (arr[0]+5) << endl;

cout << "Second element: "<<arr[1] << endl;

cout << " Third element: "<< arr[(4-2)] << endl;

return 0;

}



Q4. Inputs: int x; int y = 10; const int \* p = &y;

Output: 1. Assign pointer value to x

2. Modify the pointer value to 100.

Code:

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int x;**

**cout << "Enter the value of x: " << endl;**

**cin>>x;**

**int y = 100;**

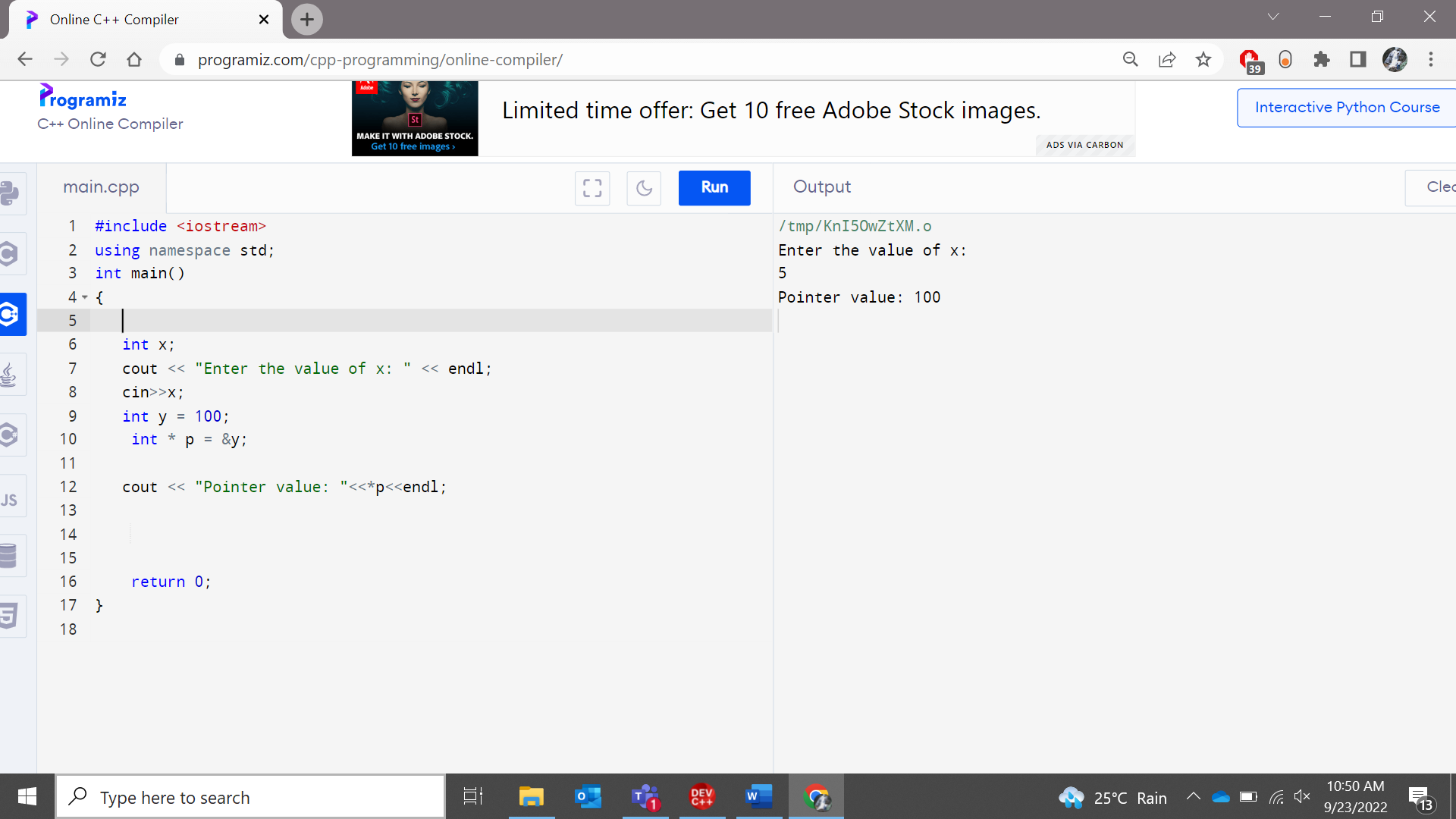
**int \* p = &y;**

**cout << "Pointer value: "<<\*p<<endl;**

**return 0;**

**}**

**Output:**



Q5 Declare two variables var1 and var2 and also aconstant pointer ‘ptr’ and made to point var1. output:

1. Make ptr value as user input.

2. Make ptr is to point var2 and print the value ptr is pointing to.

Code:

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int var1 = 0, var2 = 0;**

**cout << "Enter the value of var1: "<< endl;**

**cin>> var1;**

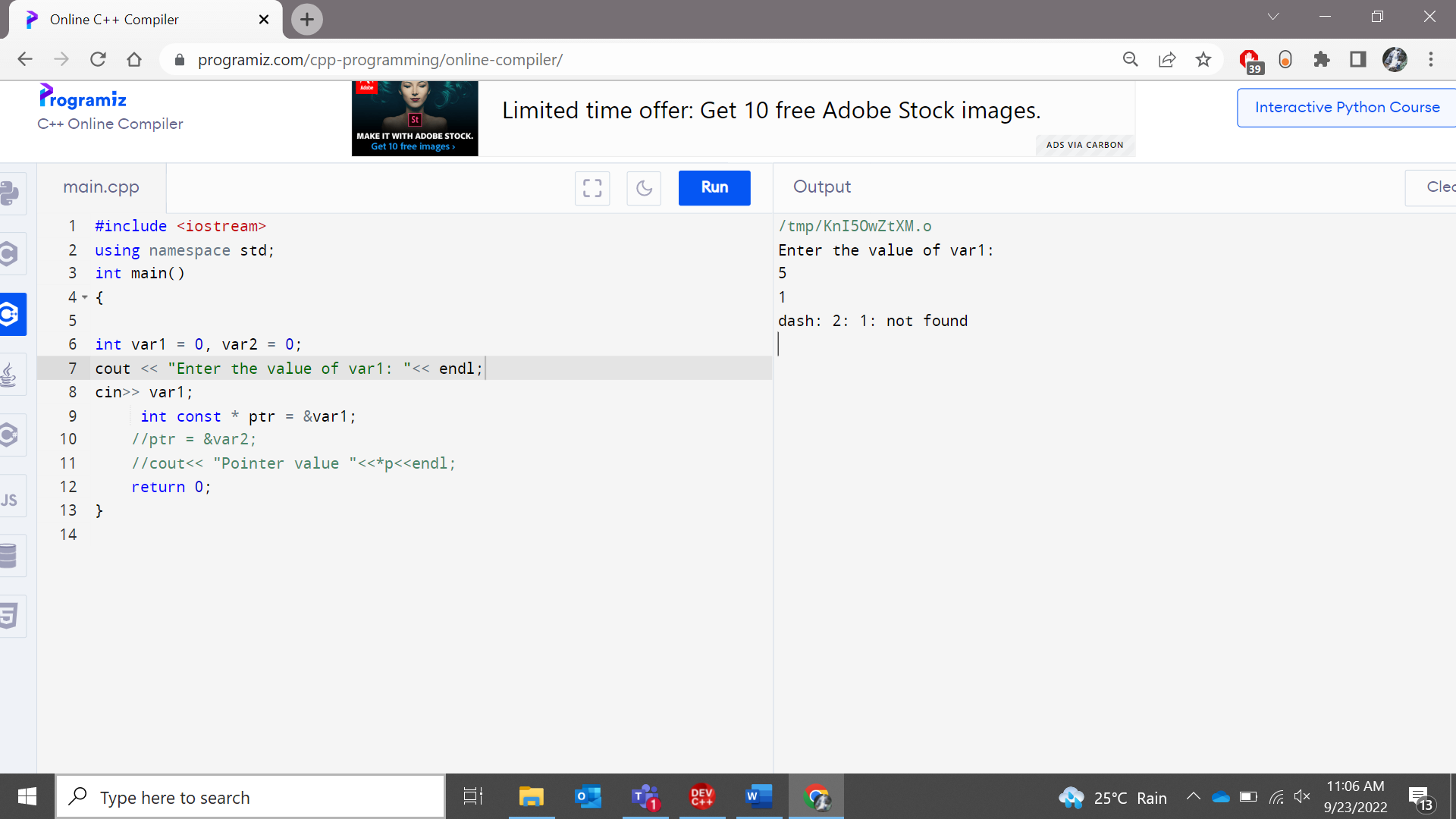
**int const \* ptr = &var1;**

**//ptr = &var2;**

**//cout<< "Pointer value "<<\*p<<endl;**

**return 0;**

**}**



**Explanation:**

**Const pointer variable cannot point to a variable. It is only readable.**

**Hence throws an exception.**

Q6. A thief trying to escape from a jail. He has to cross N walls each with varying heights (every height is greater than 0). He climbs X feet every time. But, due to the slippery nature of those walls, every time he slips back by Y feet. Now the task is to calculate the total number of jumps required to cross all walls and escape from the jail.

Input : heights[] = {11, 11}

X = 10;

Y = 1;

Output : 4

Input : heights[] = {11, 10, 10, 9}

X = 10;

Y = 1;

Output : 5

**Code :**