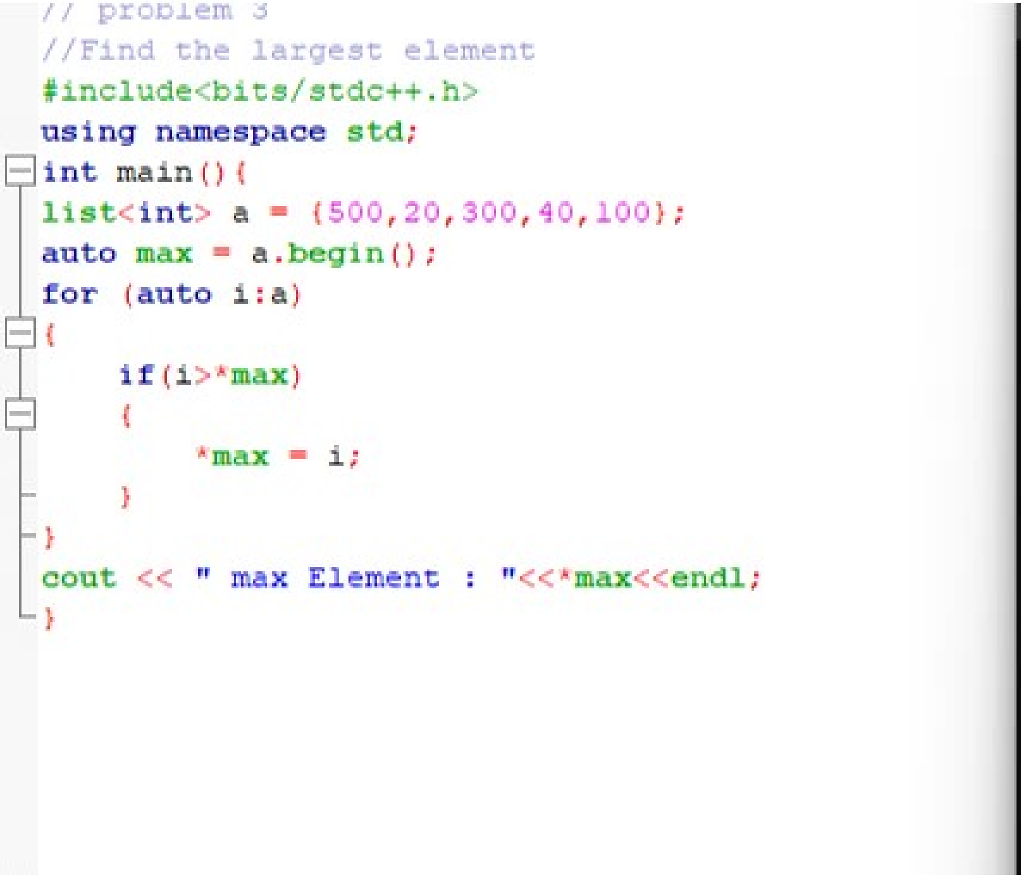
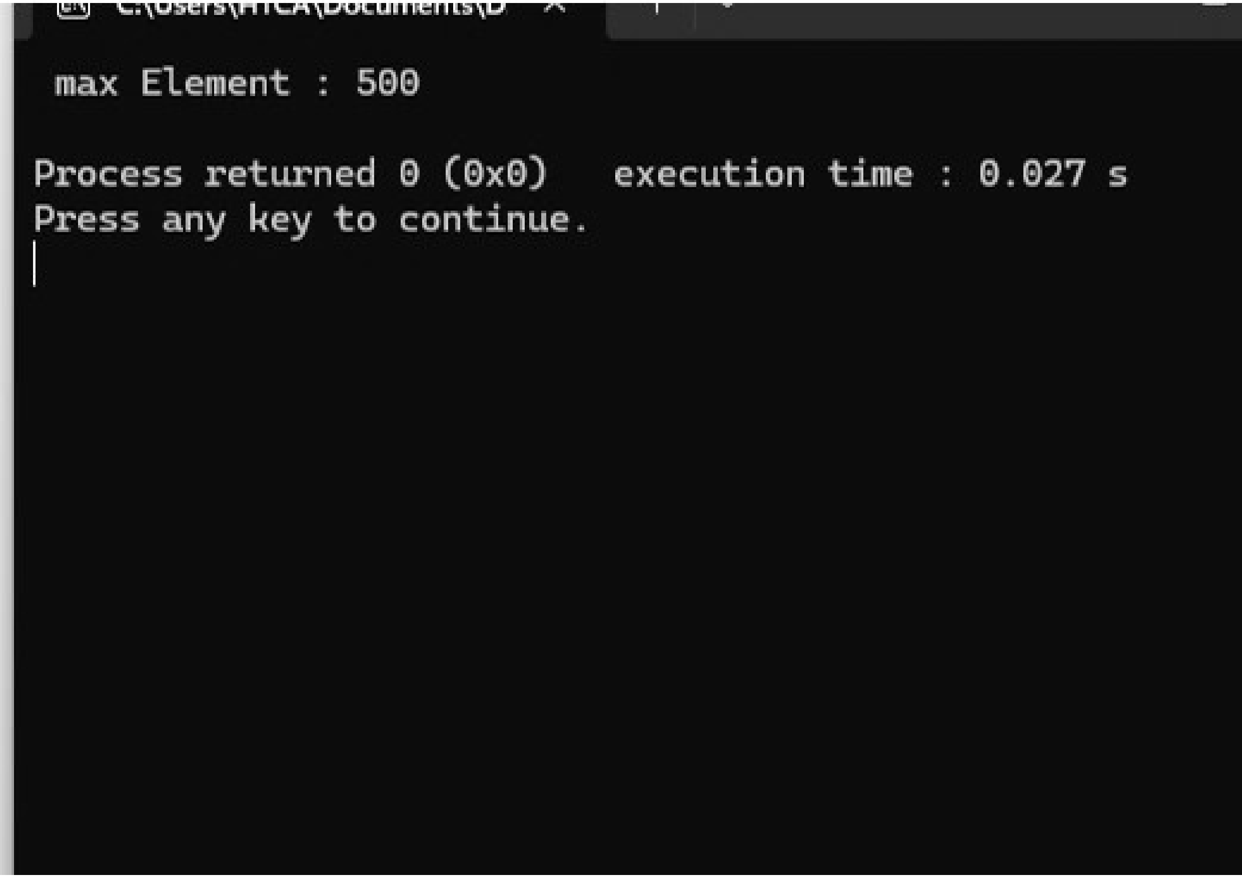
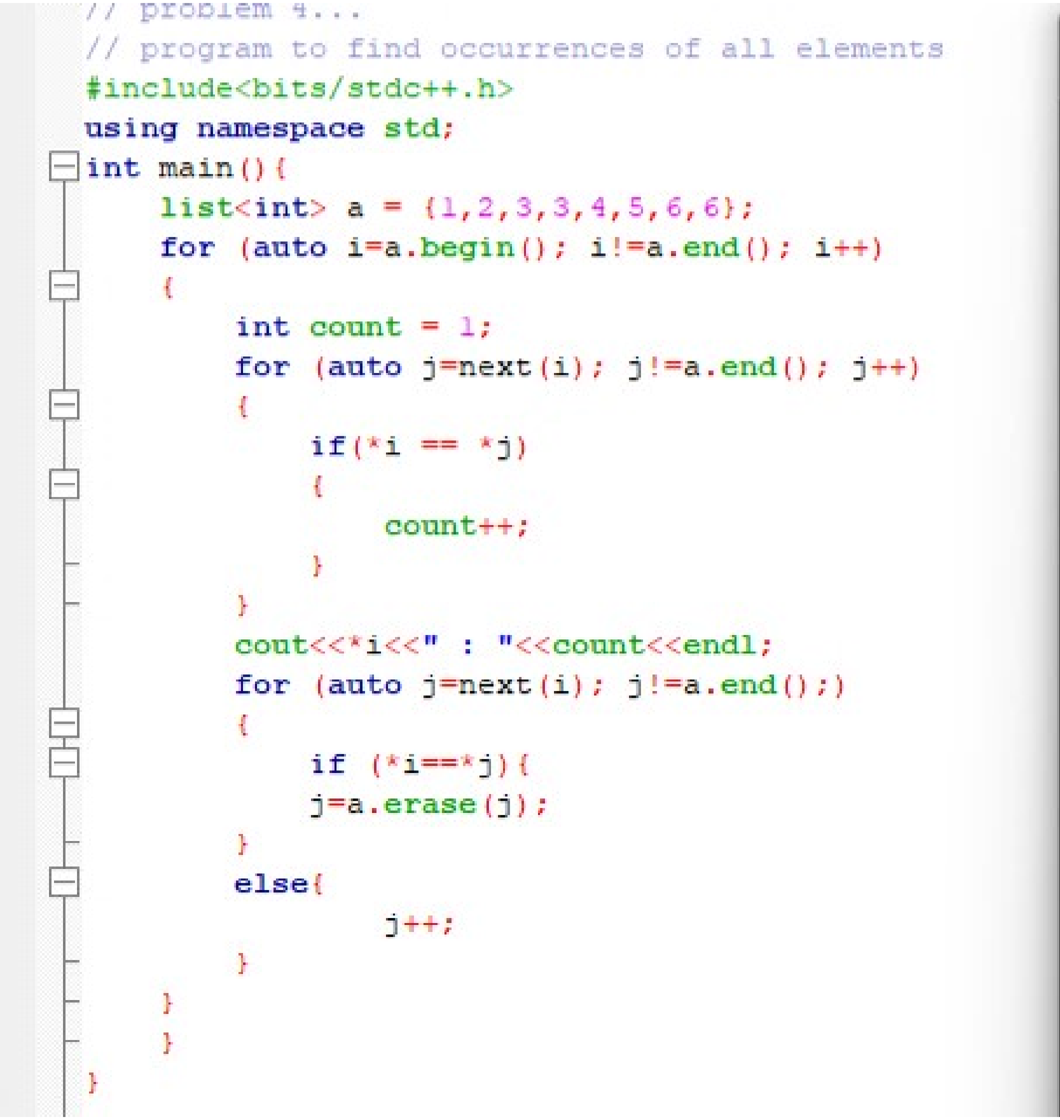


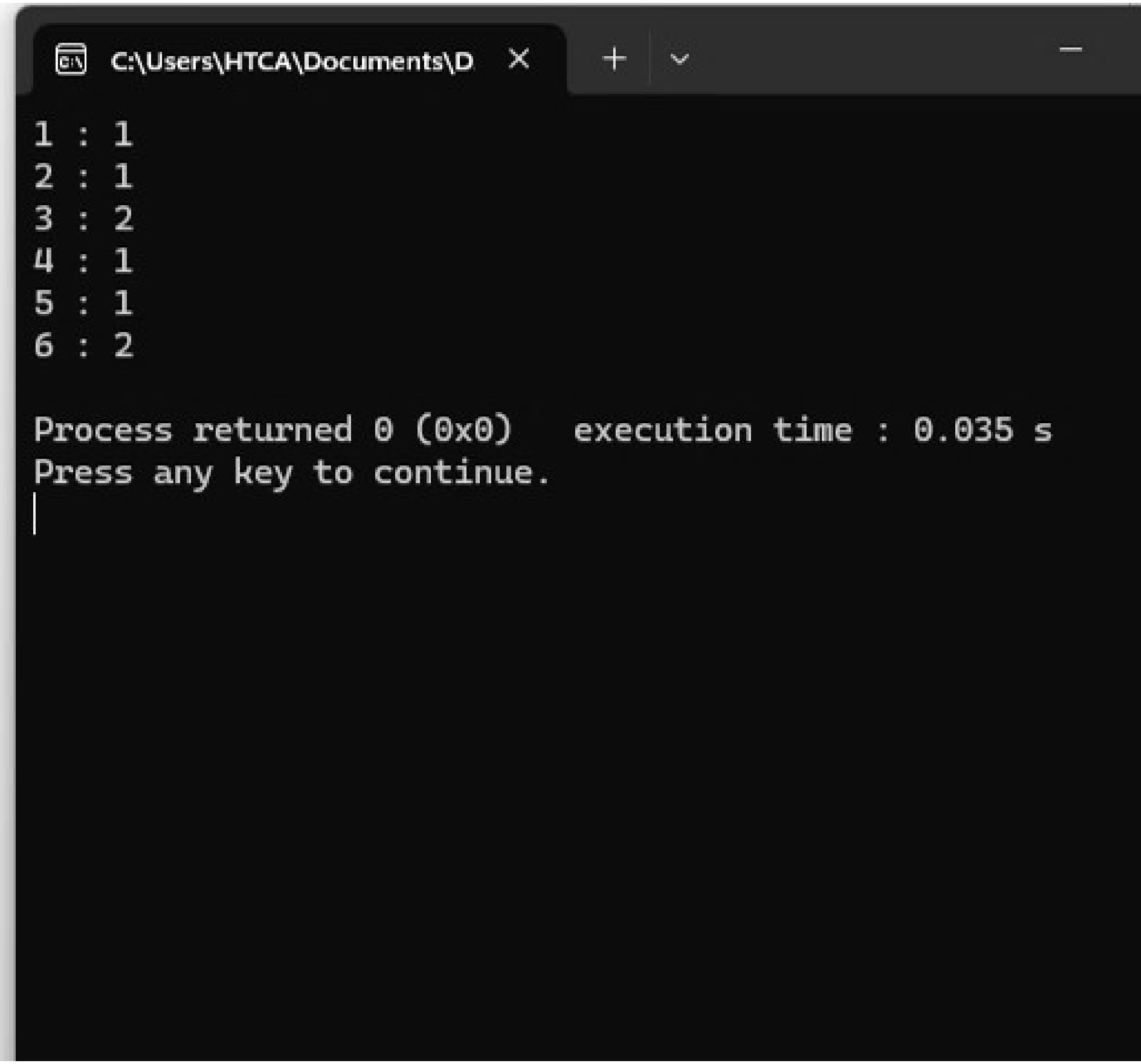
```
// problem 3
//Find the largest element
#include<bits/stdc++.h>
using namespace std;
int main(){
list<int> a = {500,20,300,40,100};
auto max = a.begin();
for (auto i:a)
{
    if(i>*max)
    {
        *max = i;
    }
}
cout << " max Element : "<<*max<<endl;
}
```



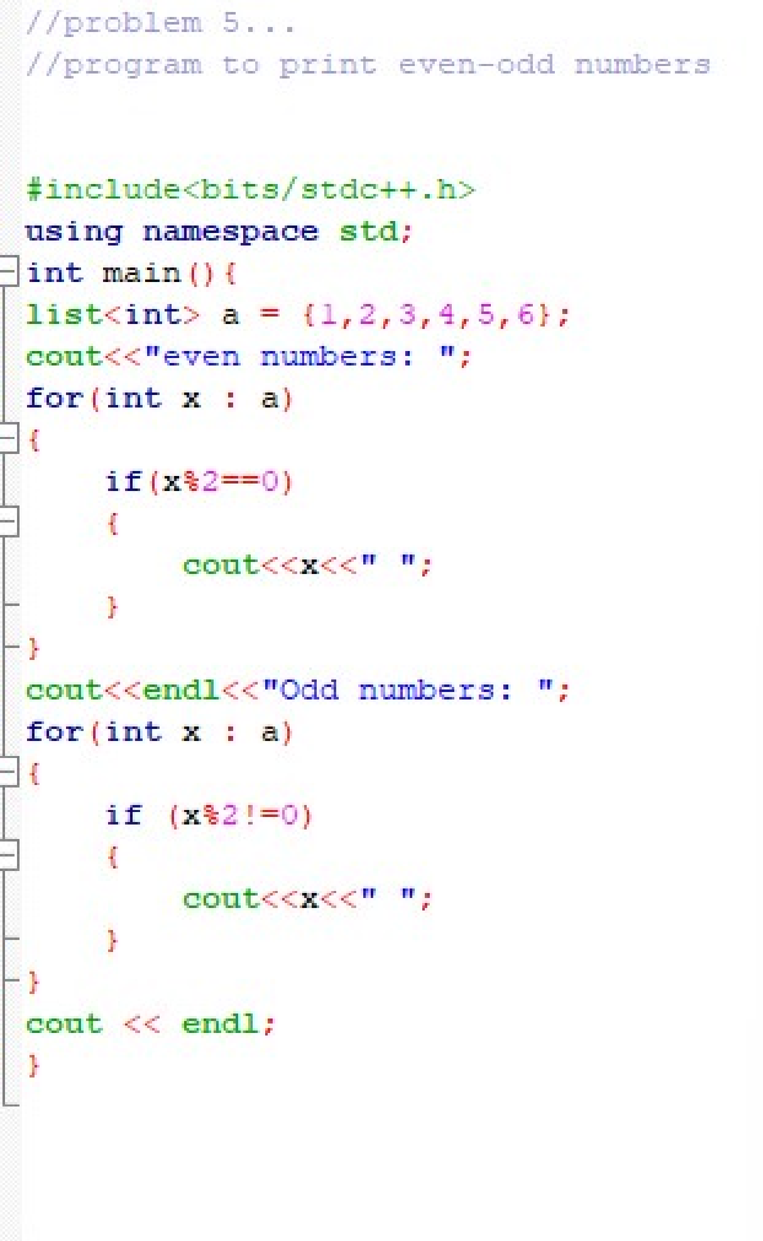
```
 max Element : 500

Process returned 0 (0x0)   execution time : 0.027 s
Press any key to continue.
```



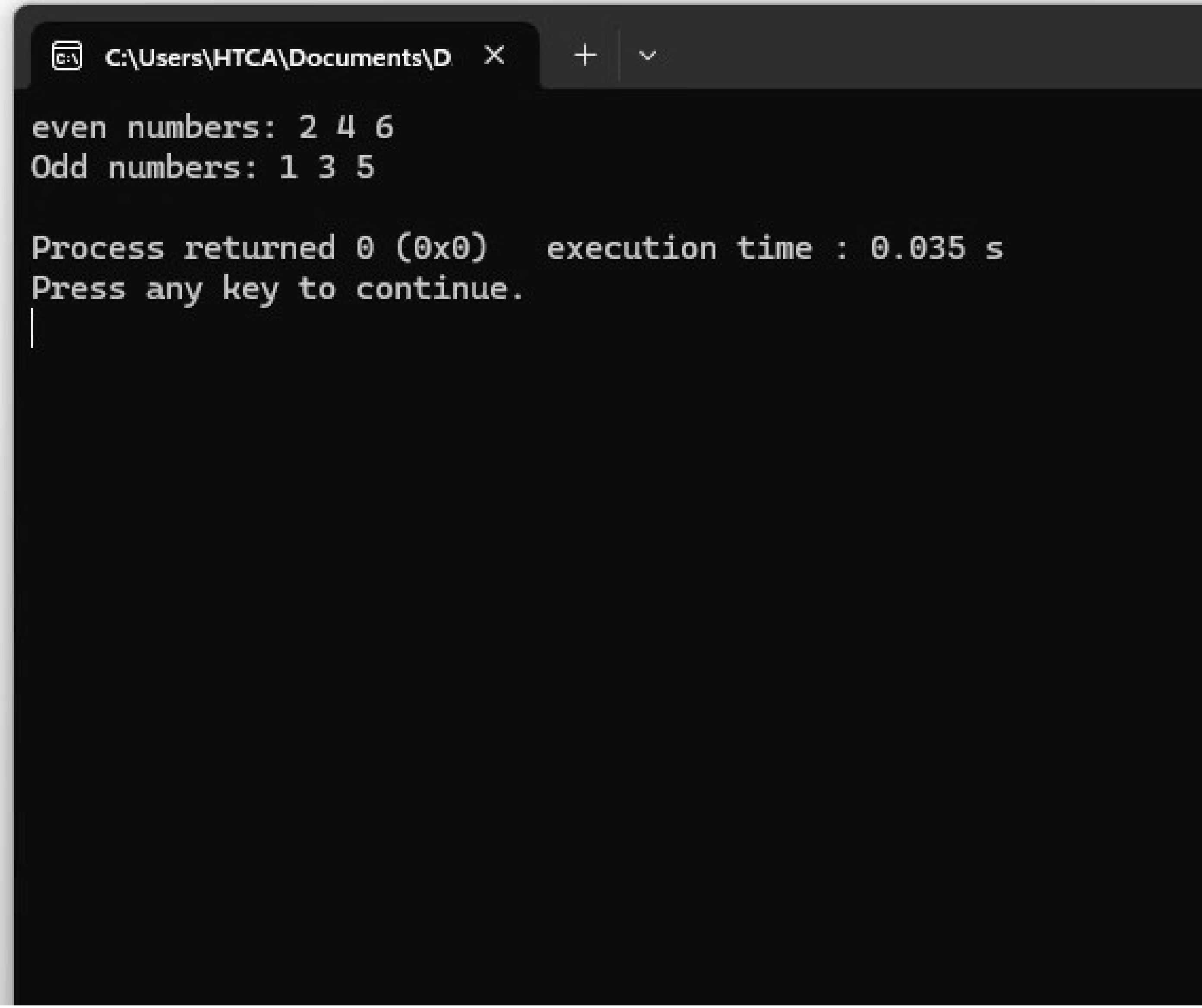
```cpp
// problem 4...
// program to find occurrences of all elements
#include<bits/stdc++.h>
using namespace std;
int main(){
    list<int> a = {1,2,3,3,4,5,6,6};
    for (auto i=a.begin(); i!=a.end(); i++)
    {
        int count = 1;
        for (auto j=next(i); j!=a.end(); j++)
        {
            if(*i == *j)
            {
                count++;
            }
        }
        cout<<*i<<" : "<<count<<endl;
        for (auto j=next(i); j!=a.end();)
        {
            if (*i==*j){
            j=a.erase(j);
        }
        else{
                j++;
        }
    }
    }
}
```



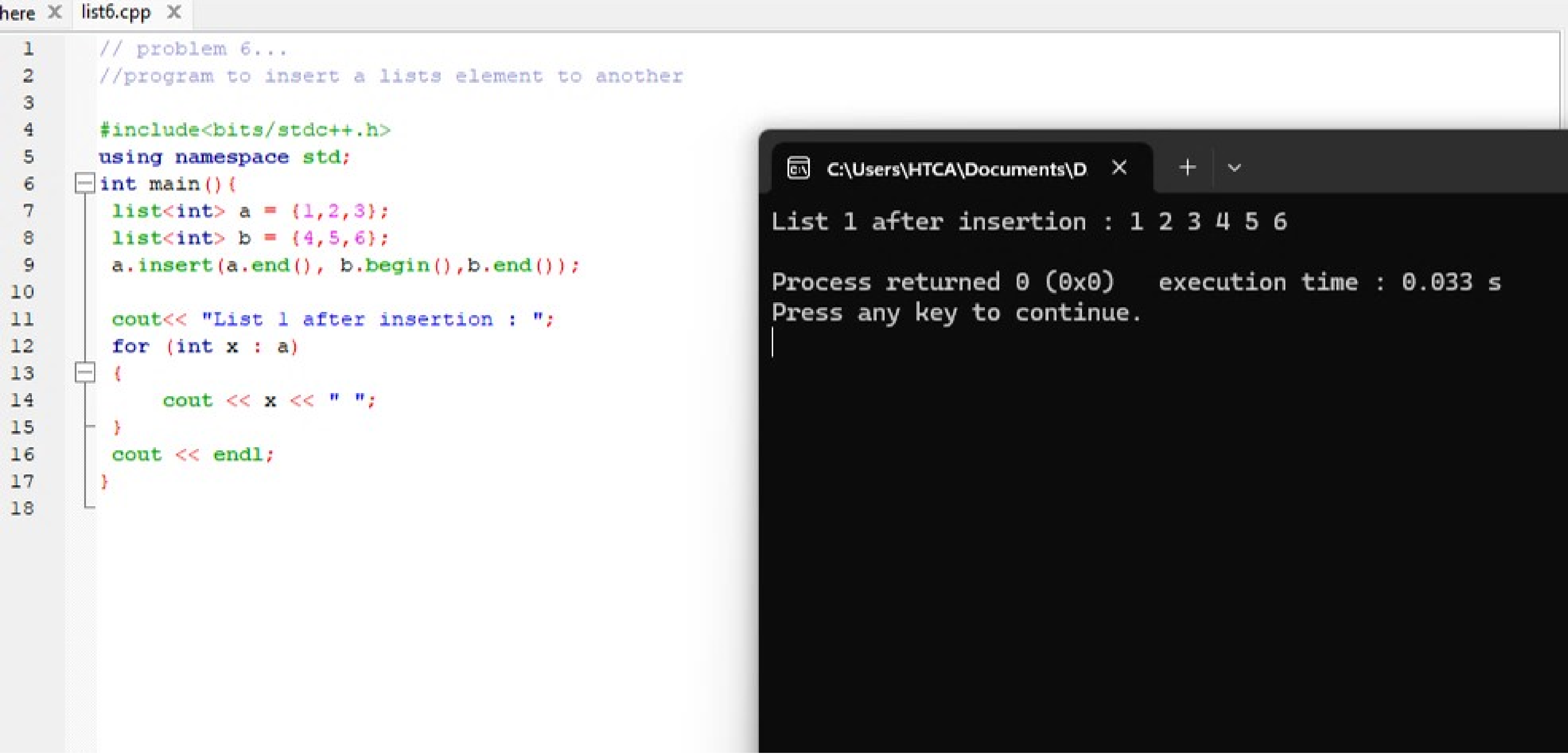
```
1 : 1
2 : 1
3 : 2
4 : 1
5 : 1
6 : 2

Process returned 0 (0x0)   execution time : 0.035 s
Press any key to continue.
```



```
//problem 5...
//program to print even-odd numbers


#include<bits/stdc++.h>
using namespace std;
int main(){
list<int> a = {1,2,3,4,5,6};
cout<<"even numbers: ";
for(int x : a)
{
    if(x%2==0)
    {
        cout<<x<<" ";
    }
}
cout<<endl<<"Odd numbers: ";
for(int x : a)
{
    if (x%2!=0)
    {
        cout<<x<<" ";
    }
}
cout << endl;
}
```

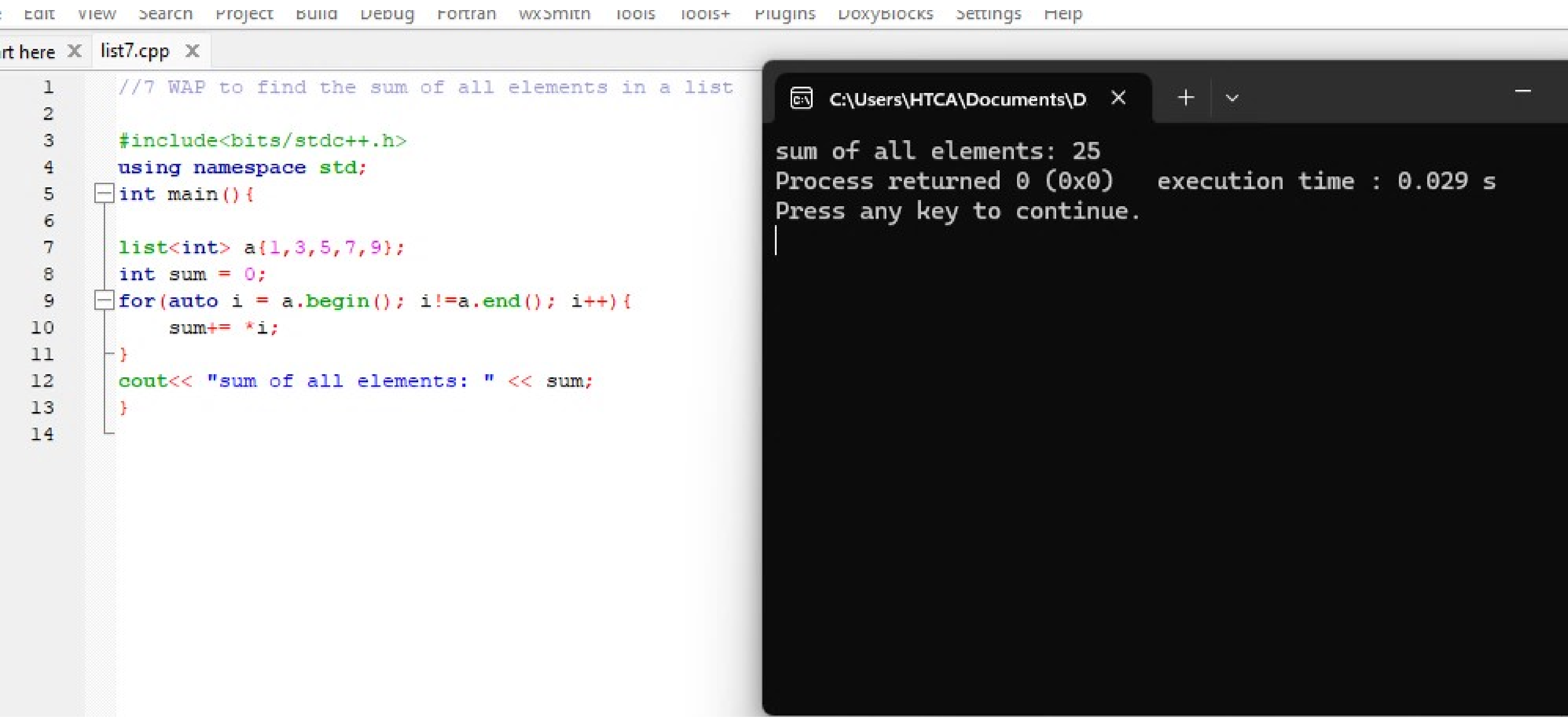


```
even numbers: 2 4 6
Odd numbers: 1 3 5

Process returned 0 (0x0)   execution time : 0.035 s
Press any key to continue.
```



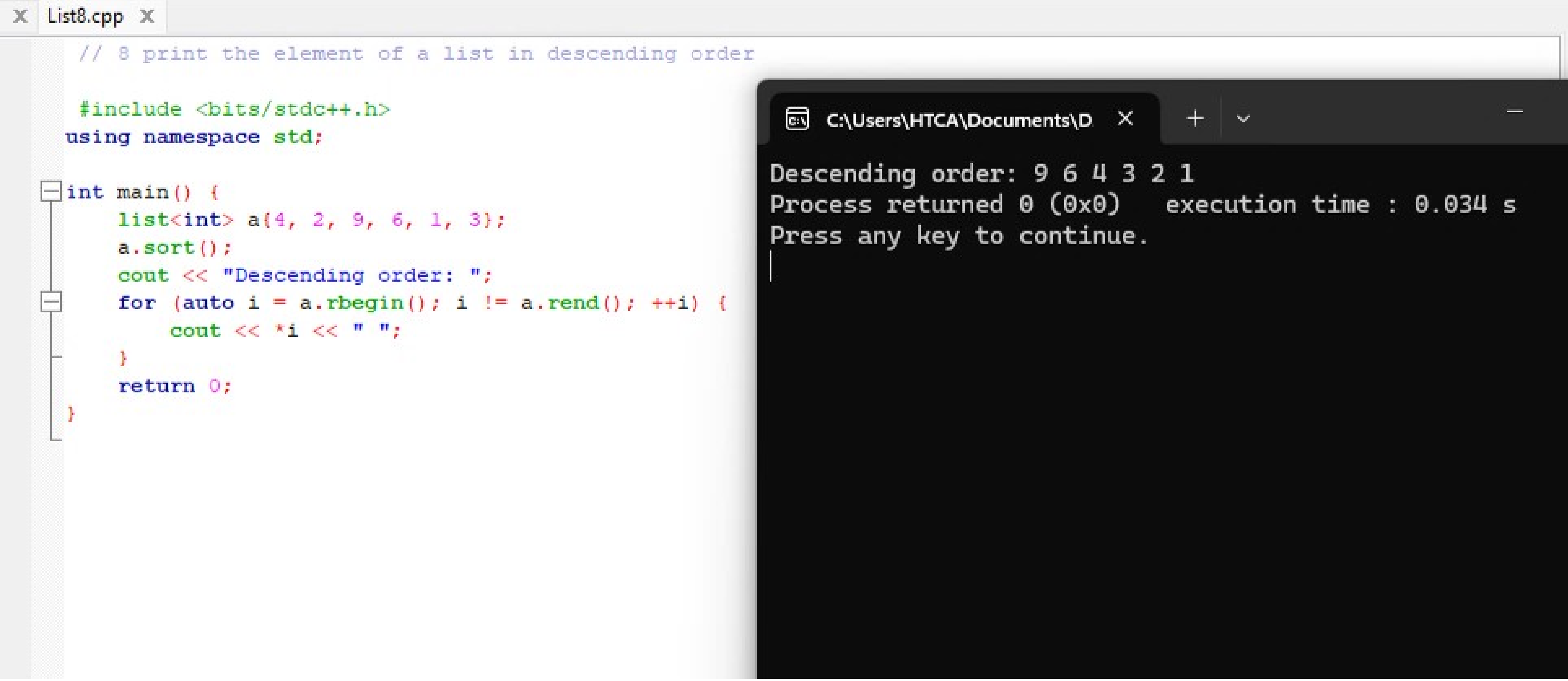
```cpp
// problem 6...
//program to insert a lists element to another

#include<bits/stdc++.h>
using namespace std;
int main(){
 list<int> a = {1,2,3};
 list<int> b = {4,5,6};
 a.insert(a.end(), b.begin(),b.end());

 cout<< "List 1 after insertion : ";
 for (int x : a)
 {
     cout << x << " ";
 }
 cout << endl;
}
```

```
List 1 after insertion : 1 2 3 4 5 6

Process returned 0 (0x0)   execution time : 0.033 s
Press any key to continue.
```


Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
rt here
list7.cpp
C:\Users\HTCA\Documents\D

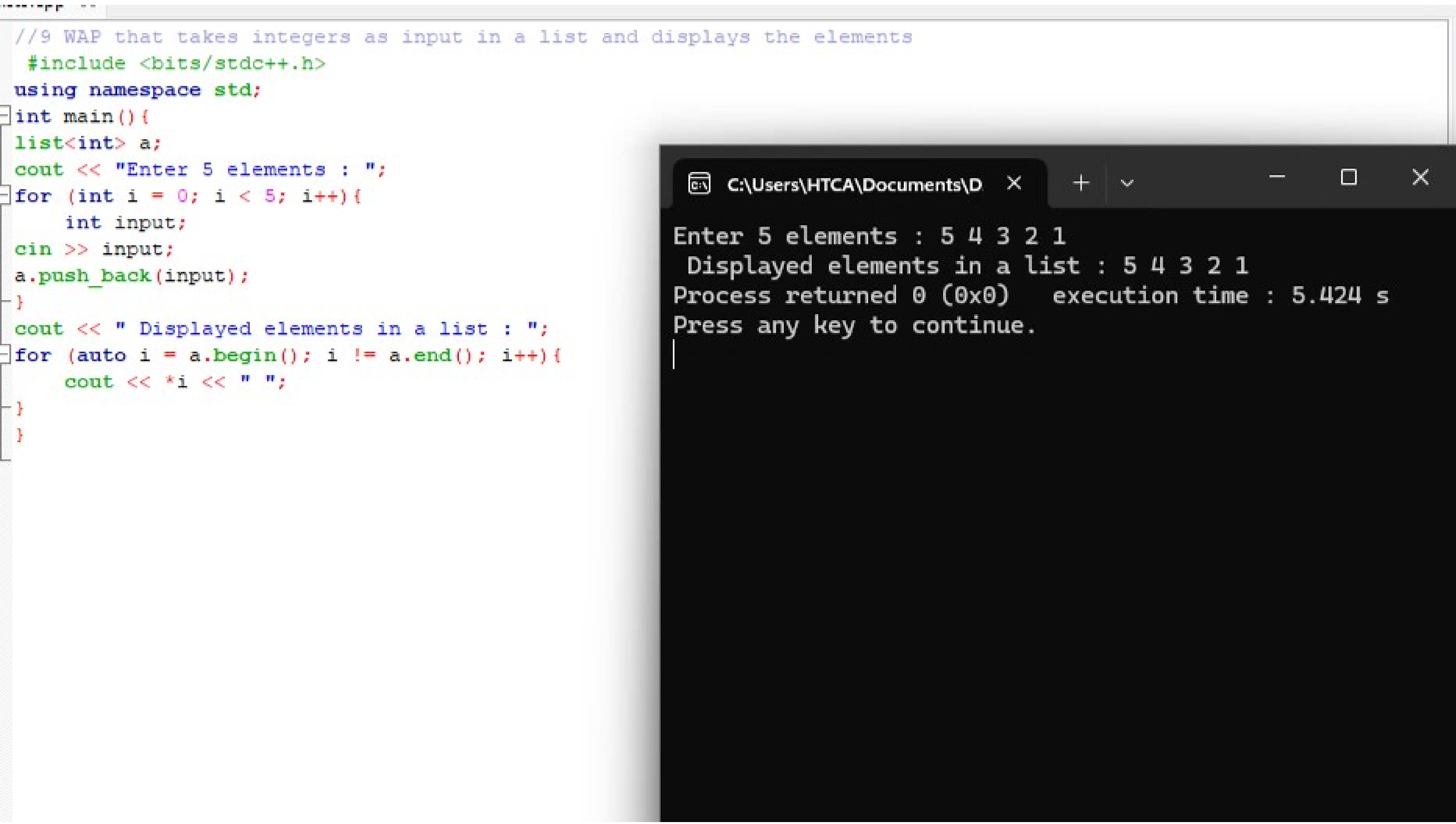
```cpp
//7 WAP to find the sum of all elements in a list

#include<bits/stdc++.h>
using namespace std;
int main(){

list<int> a{1,3,5,7,9};
int sum = 0;
for(auto i = a.begin(); i!=a.end(); i++){
    sum+= *i;
}
cout<< "sum of all elements: " << sum;
}
```

```
sum of all elements: 25
Process returned 0 (0x0)   execution time : 0.029 s
Press any key to continue.
```



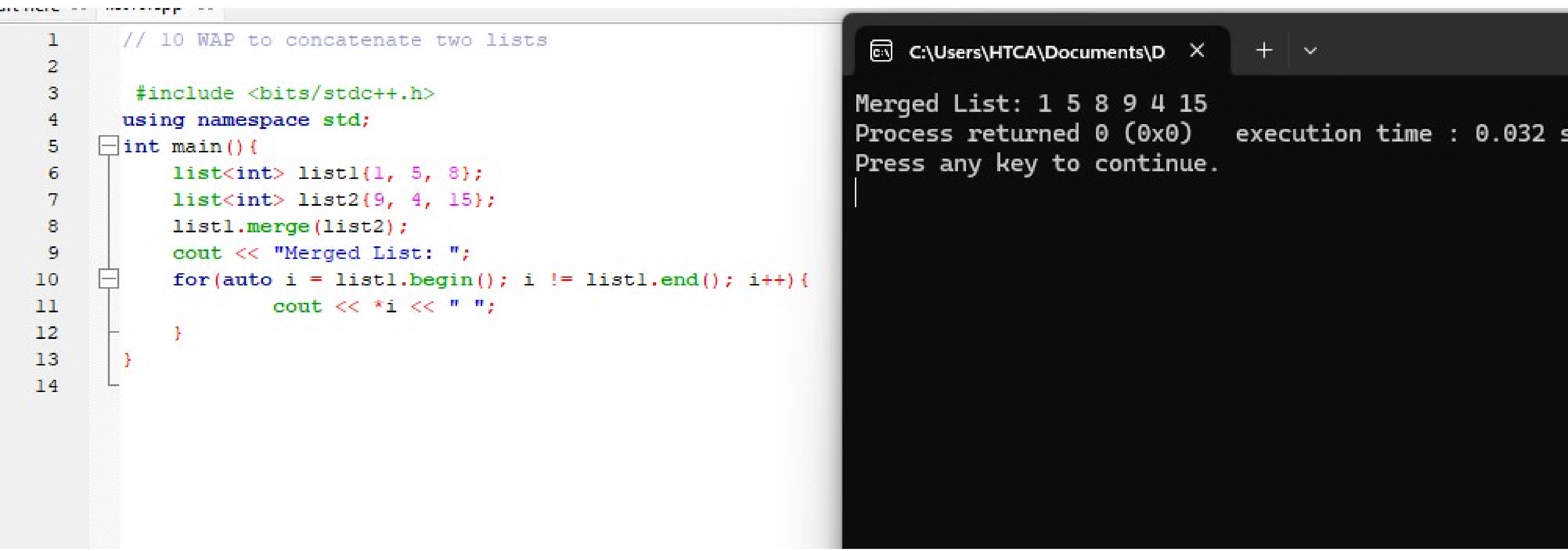
```
 // 8 print the element of a list in descending order

 #include <bits/stdc++.h>
using namespace std;

int main() {
    list<int> a{4, 2, 9, 6, 1, 3};
    a.sort();
    cout << "Descending order: ";
    for (auto i = a.rbegin(); i != a.rend(); ++i) {
        cout << *i << " ";
    }
    return 0;
}
```

```
Descending order: 9 6 4 3 2 1
Process returned 0 (0x0)   execution time : 0.034 s
Press any key to continue.
```



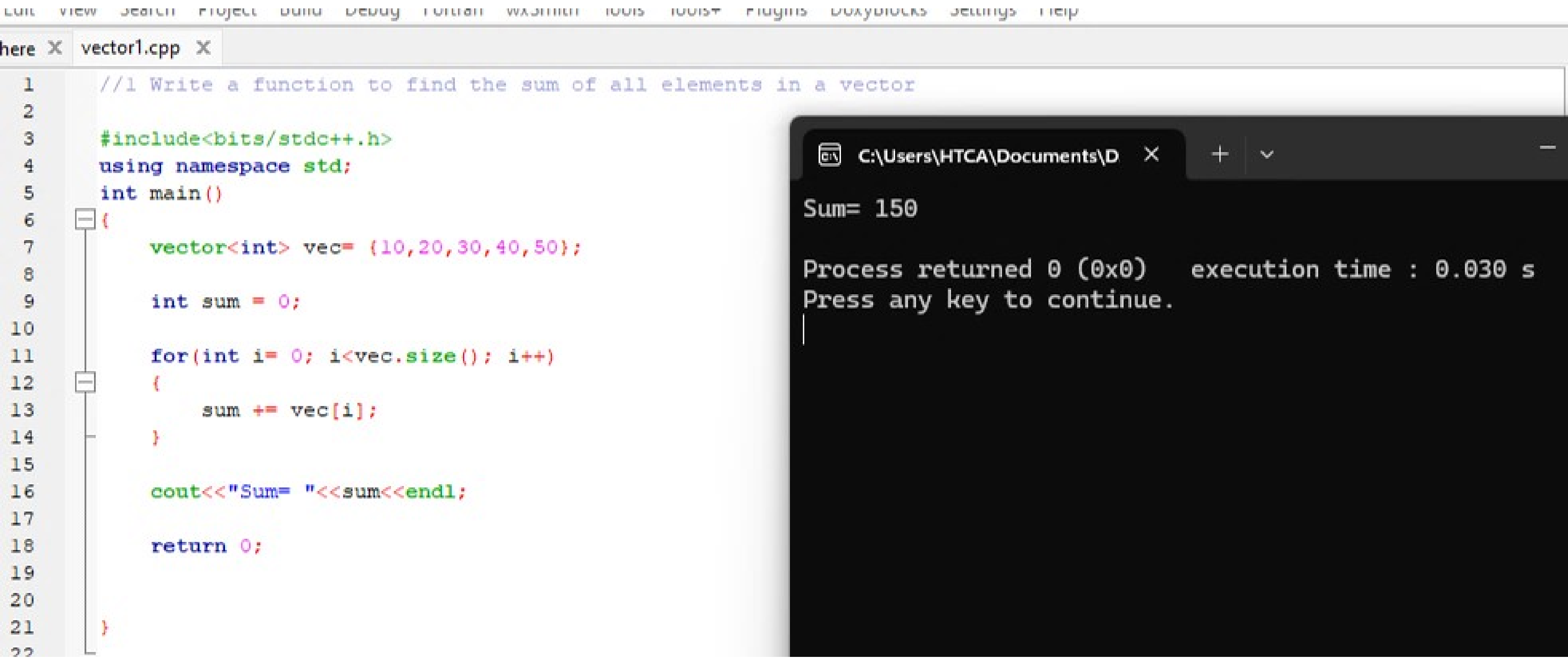
```
//9 WAP that takes integers as input in a list and displays the elements
 #include <bits/stdc++.h>
using namespace std;
int main(){
list<int> a;
cout << "Enter 5 elements : ";
for (int i = 0; i < 5; i++){
    int input;
cin >> input;
a.push_back(input);
}
cout << " Displayed elements in a list : ";
for (auto i = a.begin(); i != a.end(); i++){
    cout << *i << " ";
}
}
```

```
Enter 5 elements : 5 4 3 2 1
 Displayed elements in a list : 5 4 3 2 1
Process returned 0 (0x0)   execution time : 5.424 s
Press any key to continue.
```



```cpp
// 10 WAP to concatenate two lists

 #include <bits/stdc++.h>
using namespace std;
int main(){
    list<int> list1{1, 5, 8};
    list<int> list2{9, 4, 15};
    list1.merge(list2);
    cout << "Merged List: ";
    for(auto i = list1.begin(); i != list1.end(); i++){
            cout << *i << " ";
    }
}

```

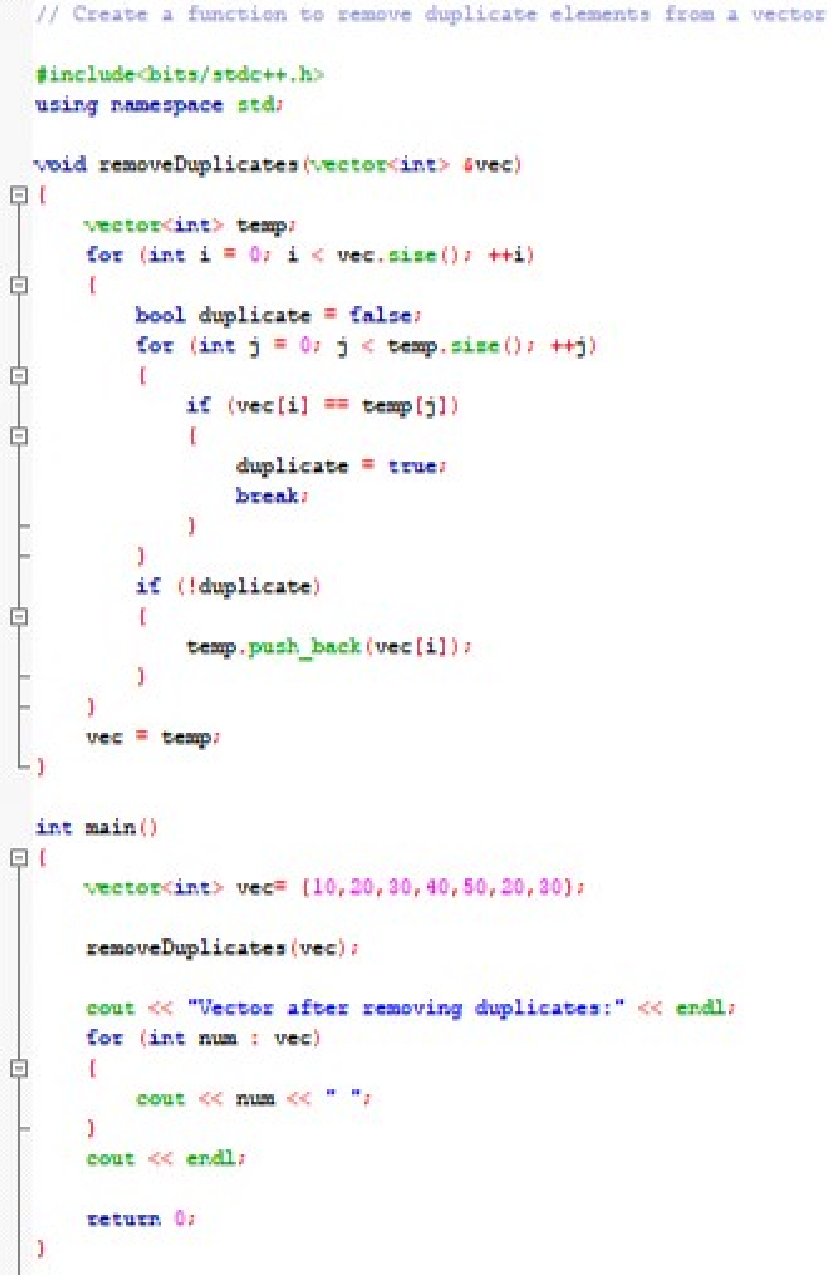
```
Merged List: 1 5 8 9 4 15
Process returned 0 (0x0)   execution time : 0.032 s
Press any key to continue.
```



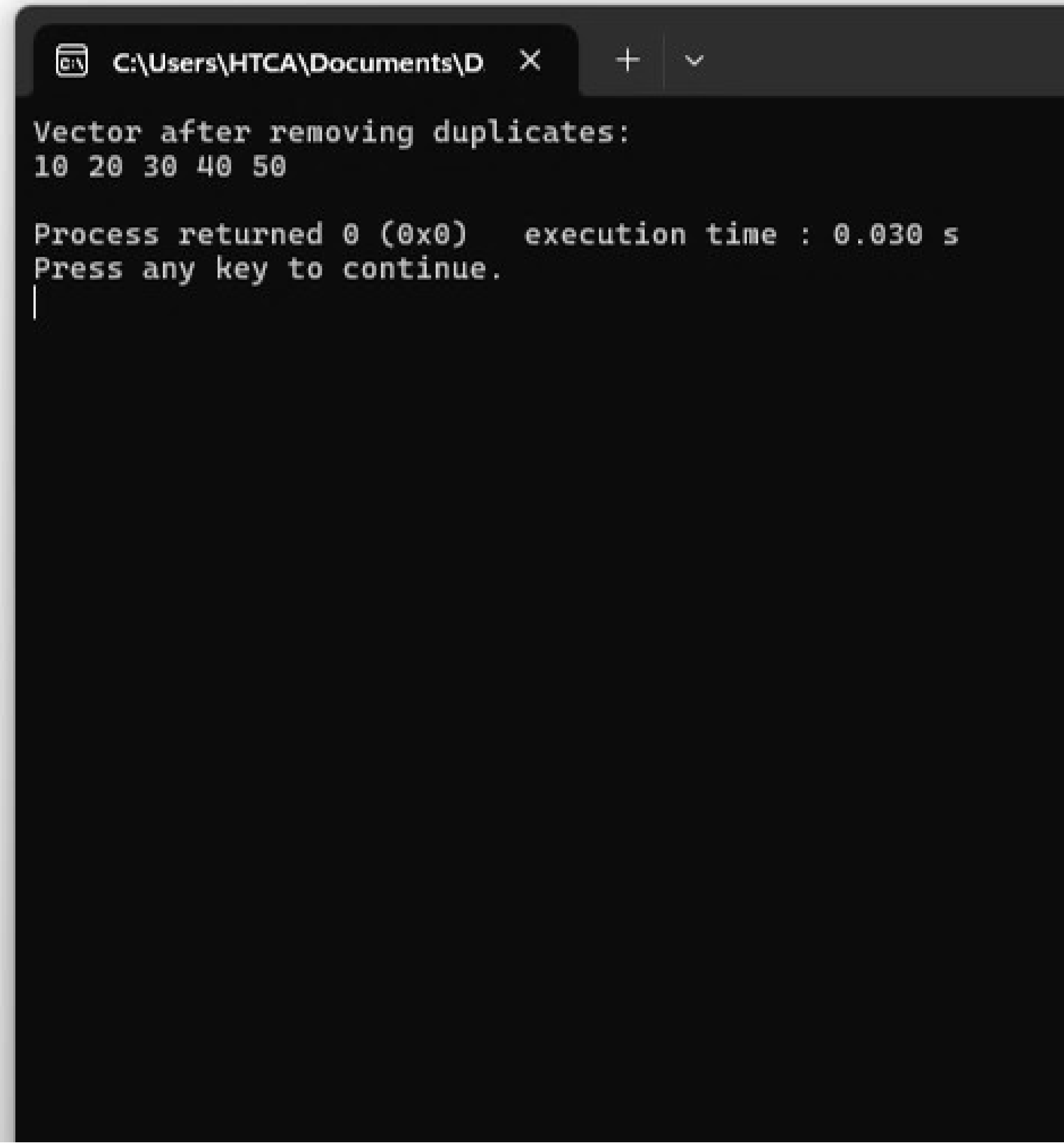
Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
here
vector1.cpp
C:\Users\HTCA\Documents\D

```cpp
//1 Write a function to find the sum of all elements in a vector

#include<bits/stdc++.h>
using namespace std;
int main()
{
    vector<int> vec= {10,20,30,40,50};

    int sum = 0;

    for(int i= 0; i<vec.size(); i++)
    {
        sum += vec[i];
    }

    cout<<"Sum= "<<sum<<endl;

    return 0;


}
```

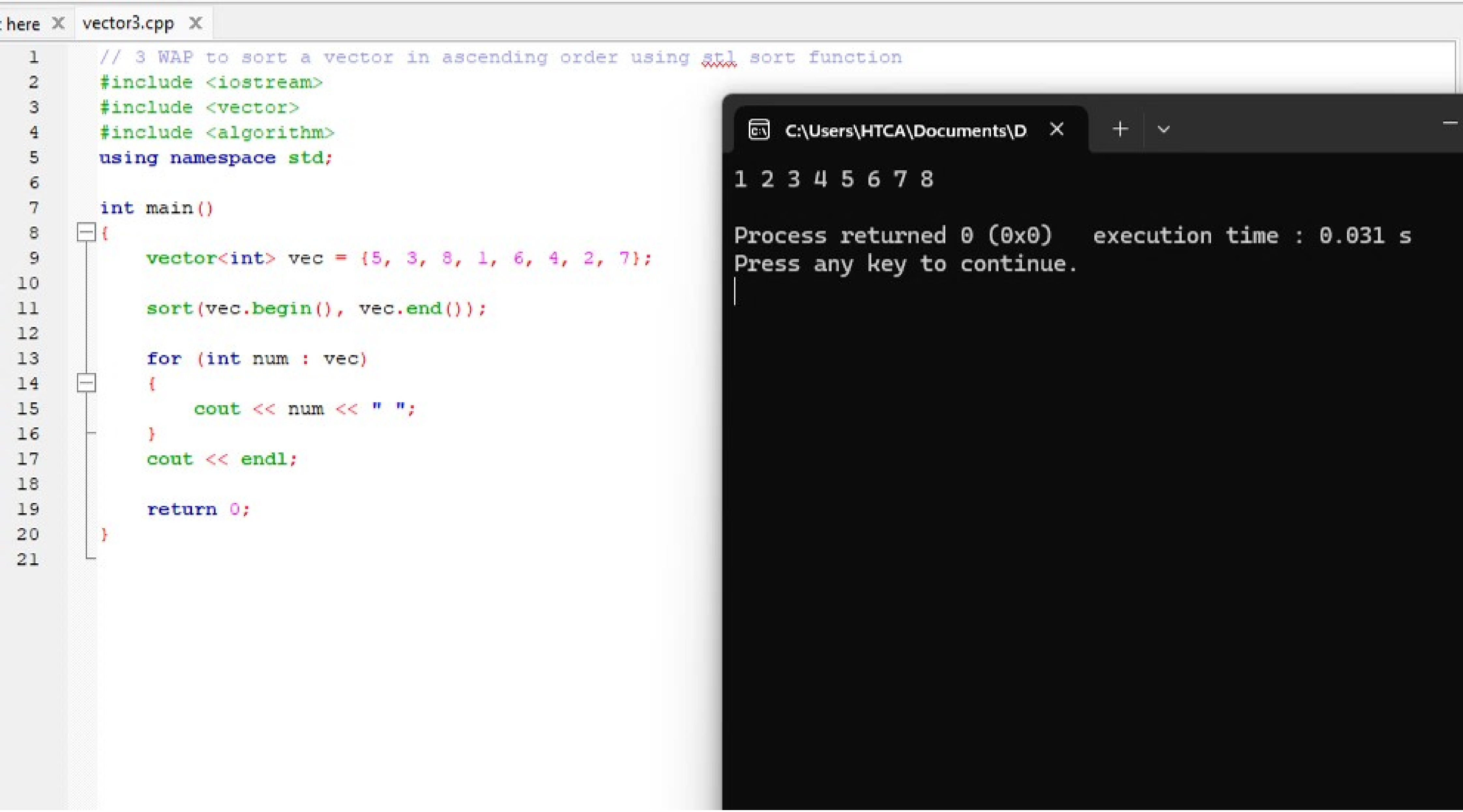
```
Sum= 150

Process returned 0 (0x0)   execution time : 0.030 s
Press any key to continue.
```



```cpp
// Create a function to remove duplicate elements from a vector

#include<bits/stdc++.h>
using namespace std;

void removeDuplicates(vector<int> &vec)
{
    vector<int> temp;
    for (int i = 0; i < vec.size(); ++i)
    {
        bool duplicate = false;
        for (int j = 0; j < temp.size(); ++j)
        {
            if (vec[i] == temp[j])
            {
                duplicate = true;
                break;
            }
        }
        if (!duplicate)
        {
            temp.push_back(vec[i]);
        }
    }
    vec = temp;
}

int main()
{
    vector<int> vec= {10,20,30,40,50,20,30};

    removeDuplicates(vec);

    cout << "Vector after removing duplicates:" << endl;
    for (int num : vec)
    {
        cout << num << " ";
    }
    cout << endl;

    return 0;
}
```

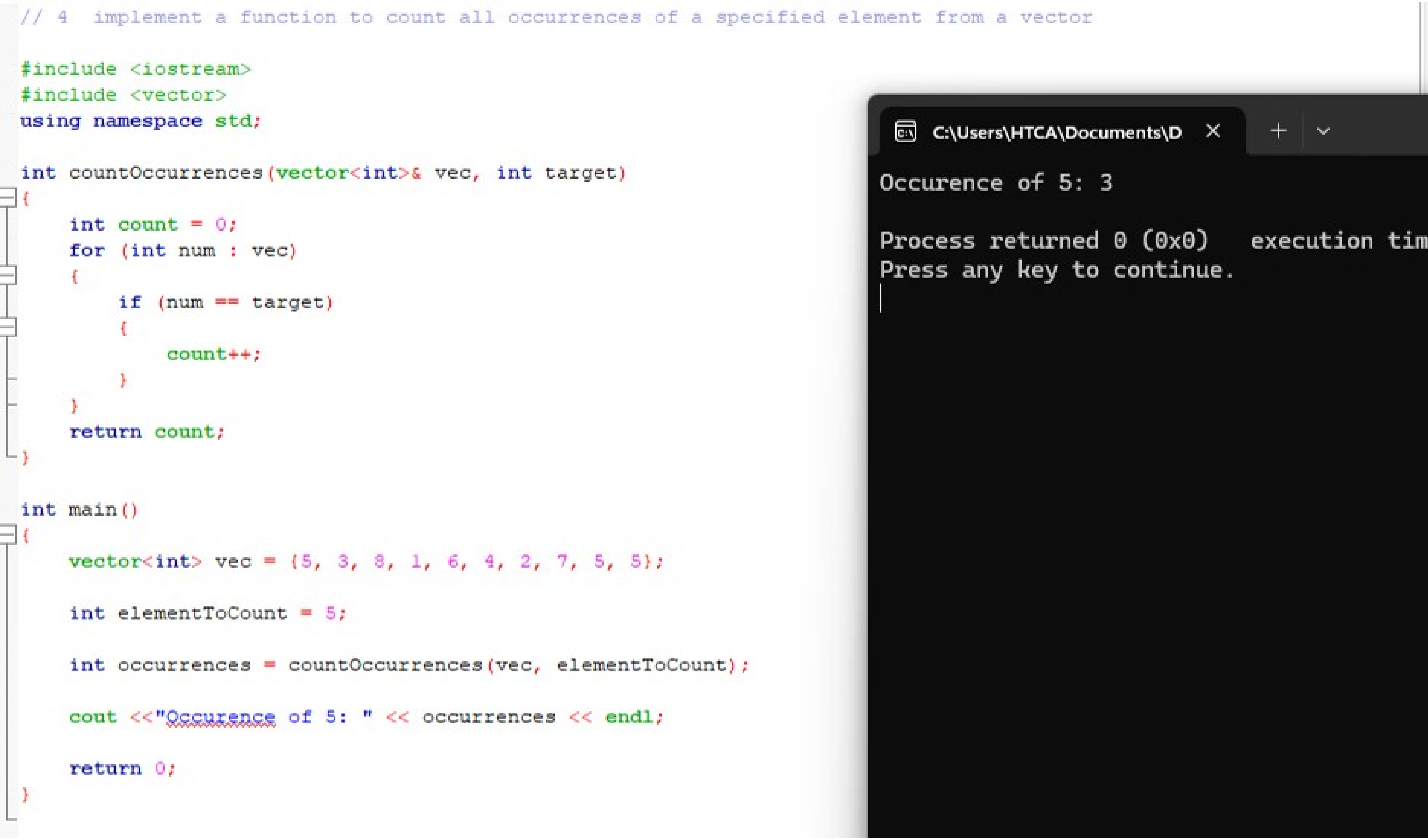


```
Vector after removing duplicates:
10 20 30 40 50

Process returned 0 (0x0)   execution time : 0.030 s
Press any key to continue.
```



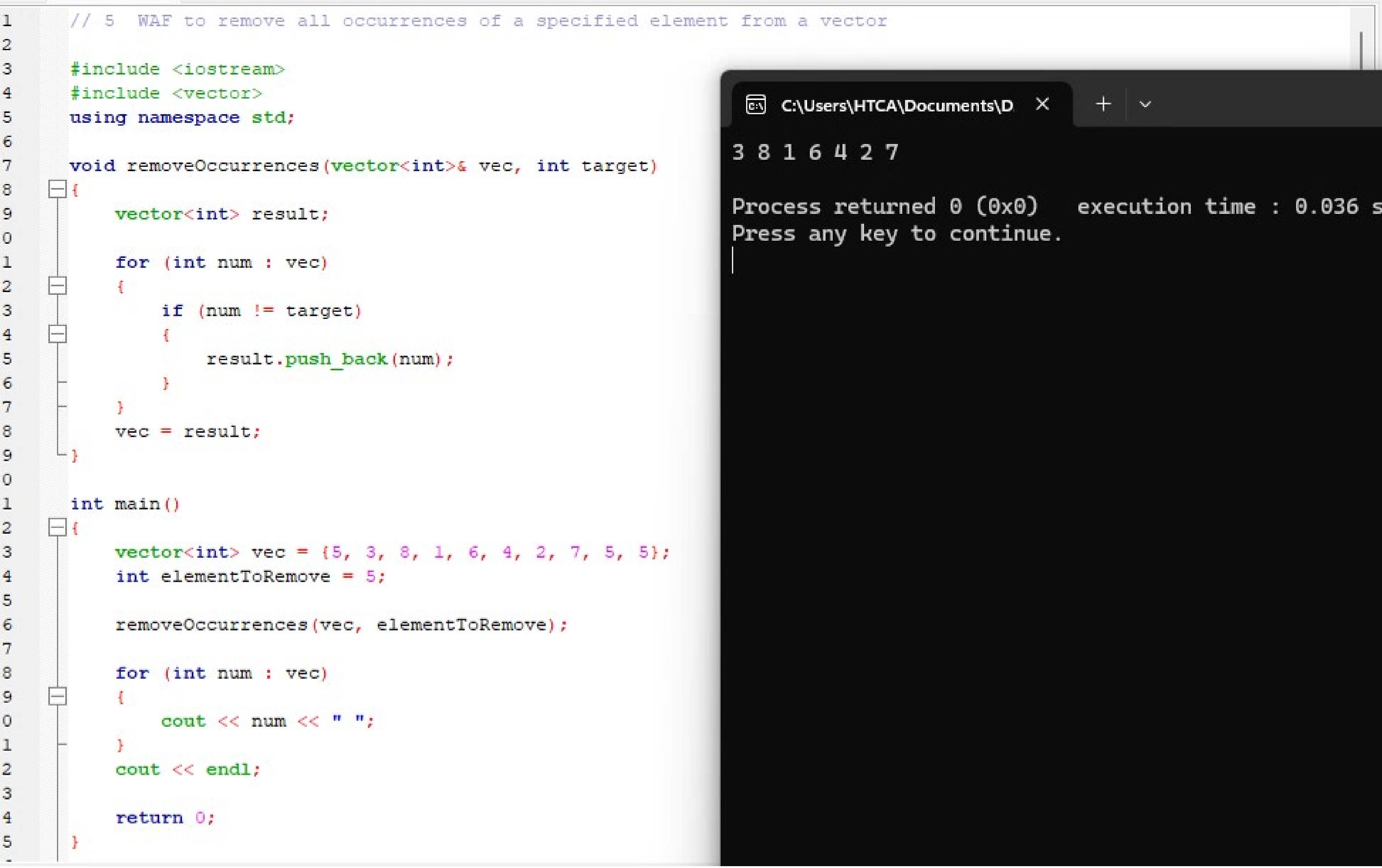
```cpp
// 3 WAP to sort a vector in ascending order using stl sort function
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main()
{
    vector<int> vec = {5, 3, 8, 1, 6, 4, 2, 7};

    sort(vec.begin(), vec.end());

    for (int num : vec)
    {
        cout << num << " ";
    }
    cout << endl;

    return 0;
}

```

```
1 2 3 4 5 6 7 8

Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```



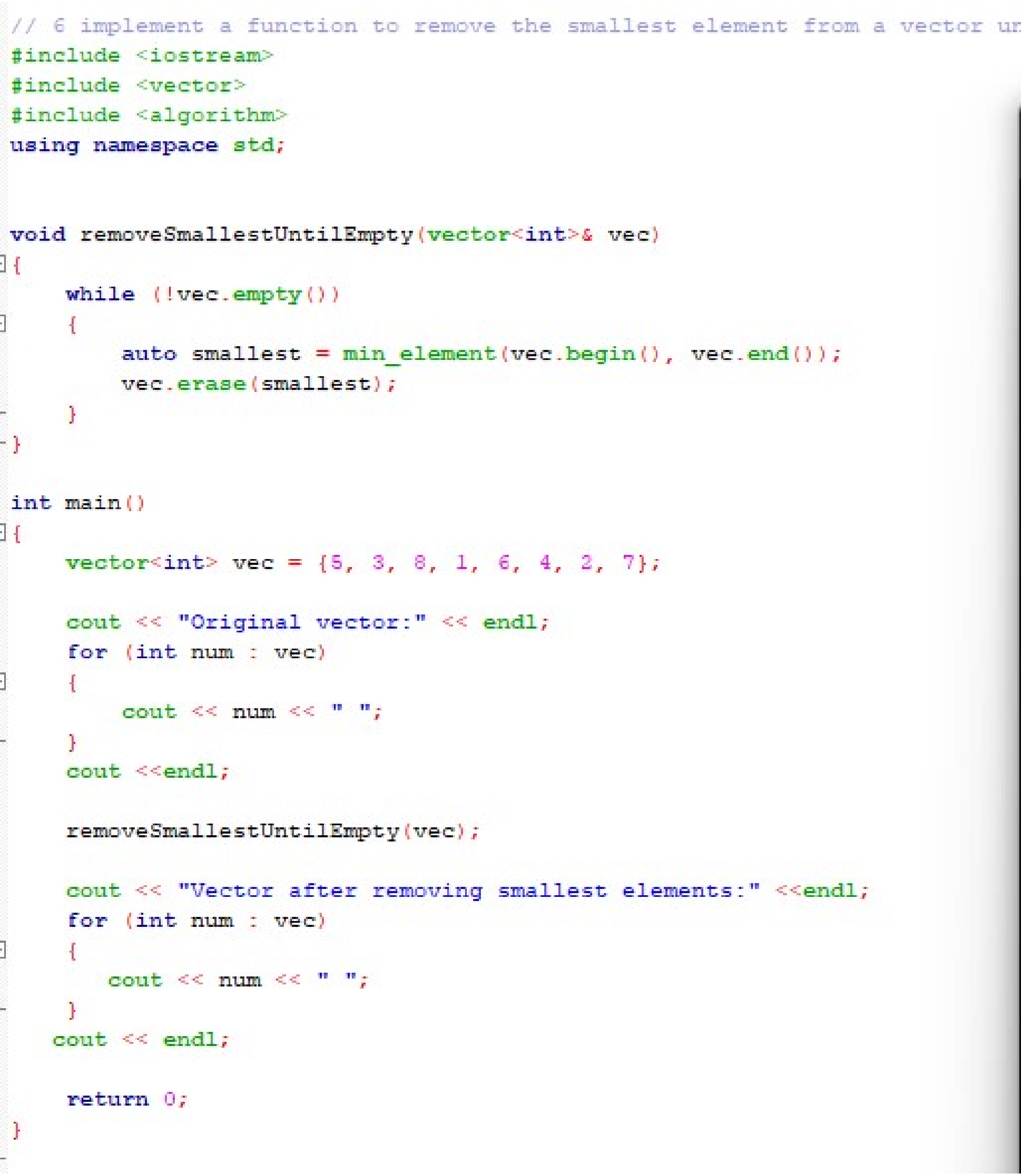
```cpp
// 4  implement a function to count all occurrences of a specified element from a vector

#include <iostream>
#include <vector>
using namespace std;

int countOccurrences(vector<int>& vec, int target)
{
    int count = 0;
    for (int num : vec)
    {
        if (num == target)
        {
            count++;
        }
    }
    return count;
}

int main()
{
    vector<int> vec = {5, 3, 8, 1, 6, 4, 2, 7, 5, 5};

    int elementToCount = 5;

    int occurrences = countOccurrences(vec, elementToCount);

    cout <<"Occurence of 5: " << occurrences << endl;

    return 0;
}
```

```
Occurence of 5: 3

Process returned 0 (0x0)   execution tim
Press any key to continue.
```

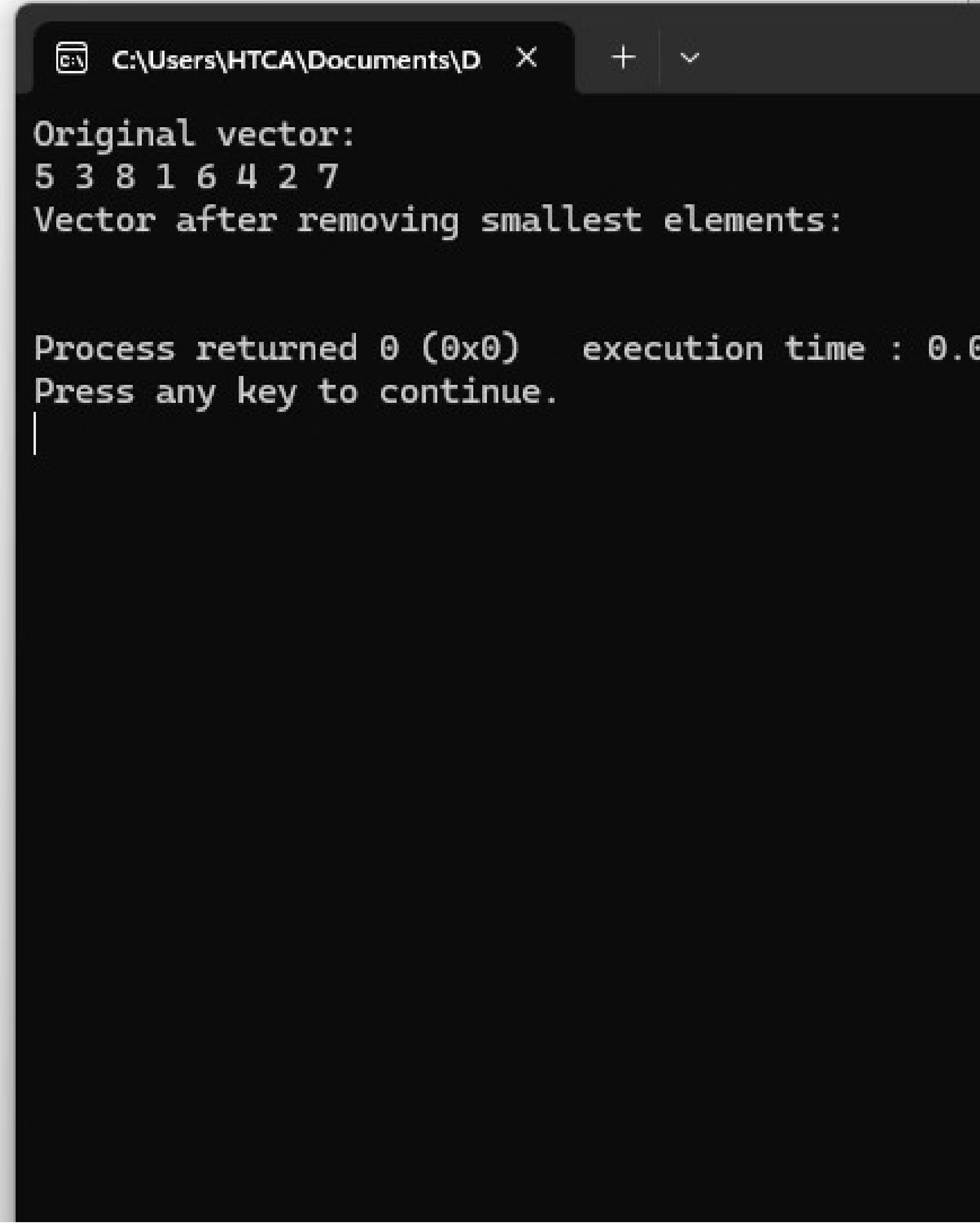


```cpp
// 5  WAF to remove all occurrences of a specified element from a vector

#include <iostream>
#include <vector>
using namespace std;

void removeOccurrences(vector<int>& vec, int target)
{
    vector<int> result;

    for (int num : vec)
    {
        if (num != target)
        {
            result.push_back(num);
        }
    }
    vec = result;
}

int main()
{
    vector<int> vec = {5, 3, 8, 1, 6, 4, 2, 7, 5, 5};
    int elementToRemove = 5;

    removeOccurrences(vec, elementToRemove);

    for (int num : vec)
    {
        cout << num << " ";
    }
    cout << endl;

    return 0;
}
```

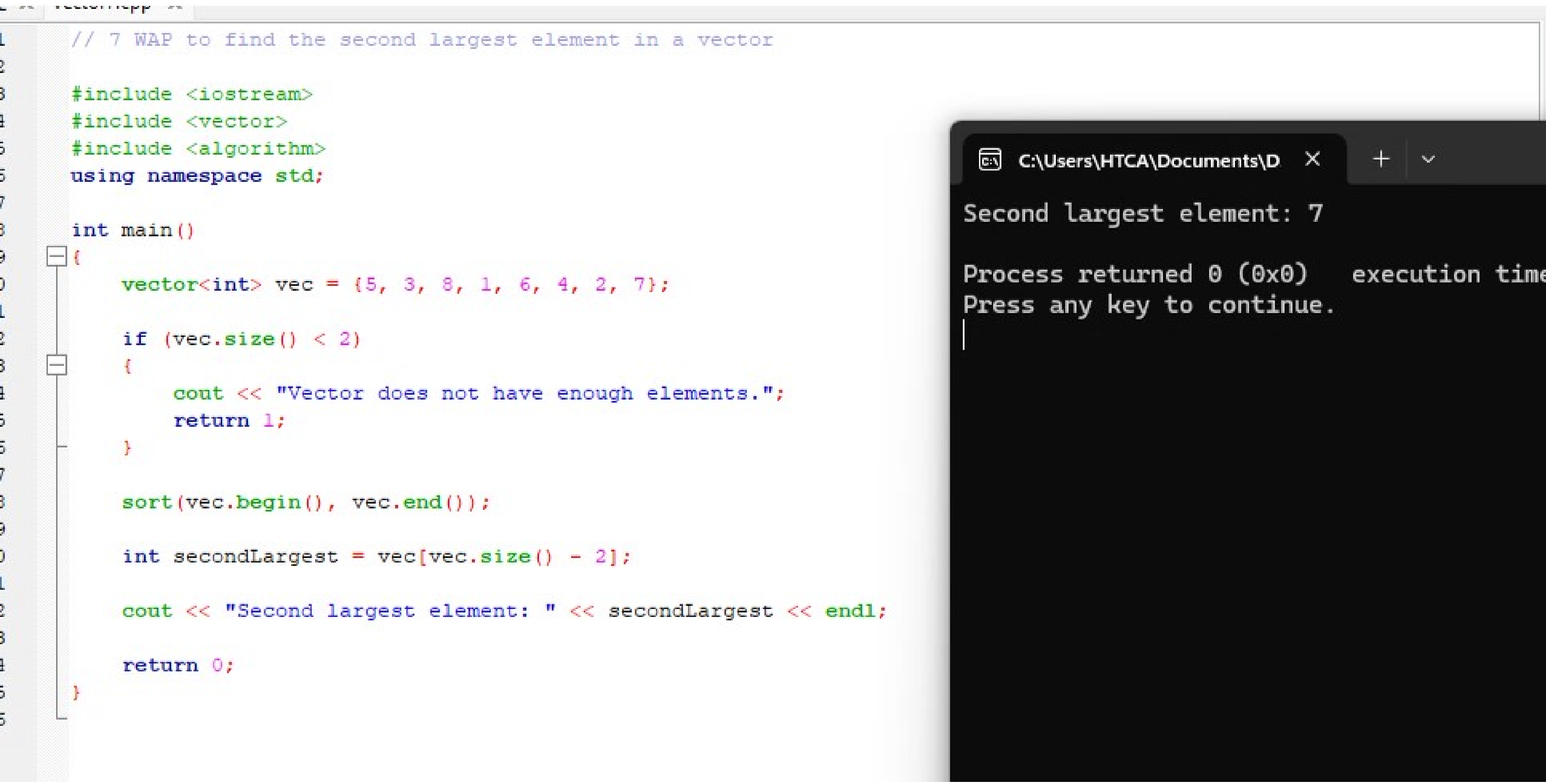
```
3 8 1 6 4 2 7

Process returned 0 (0x0)   execution time : 0.036 s
Press any key to continue.
```



```cpp
// 6 implement a function to remove the smallest element from a vector until its empty
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;


void removeSmallestUntilEmpty(vector<int>& vec)
{
    while (!vec.empty())
    {
        auto smallest = min_element(vec.begin(), vec.end());
        vec.erase(smallest);
    }
}

int main()
{
    vector<int> vec = {5, 3, 8, 1, 6, 4, 2, 7};

    cout << "Original vector:" << endl;
    for (int num : vec)
    {
        cout << num << " ";
    }
    cout <<endl;

    removeSmallestUntilEmpty(vec);

    cout << "Vector after removing smallest elements:" <<endl;
    for (int num : vec)
    {
       cout << num << " ";
    }
   cout << endl;

    return 0;
}
```

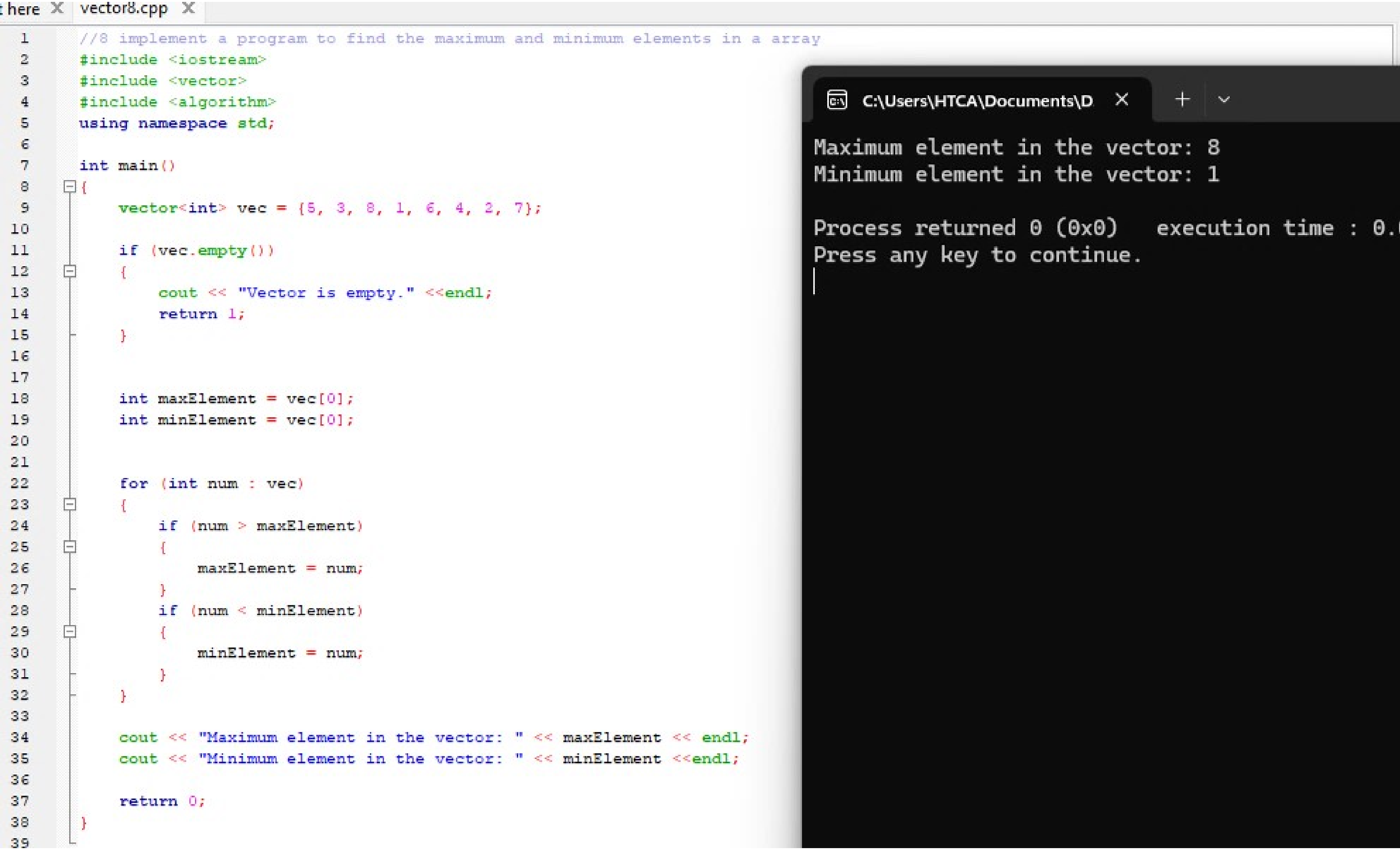


```
Original vector:
5 3 8 1 6 4 2 7
Vector after removing smallest elements:


Process returned 0 (0x0)   execution time : 0.0
Press any key to continue.
```



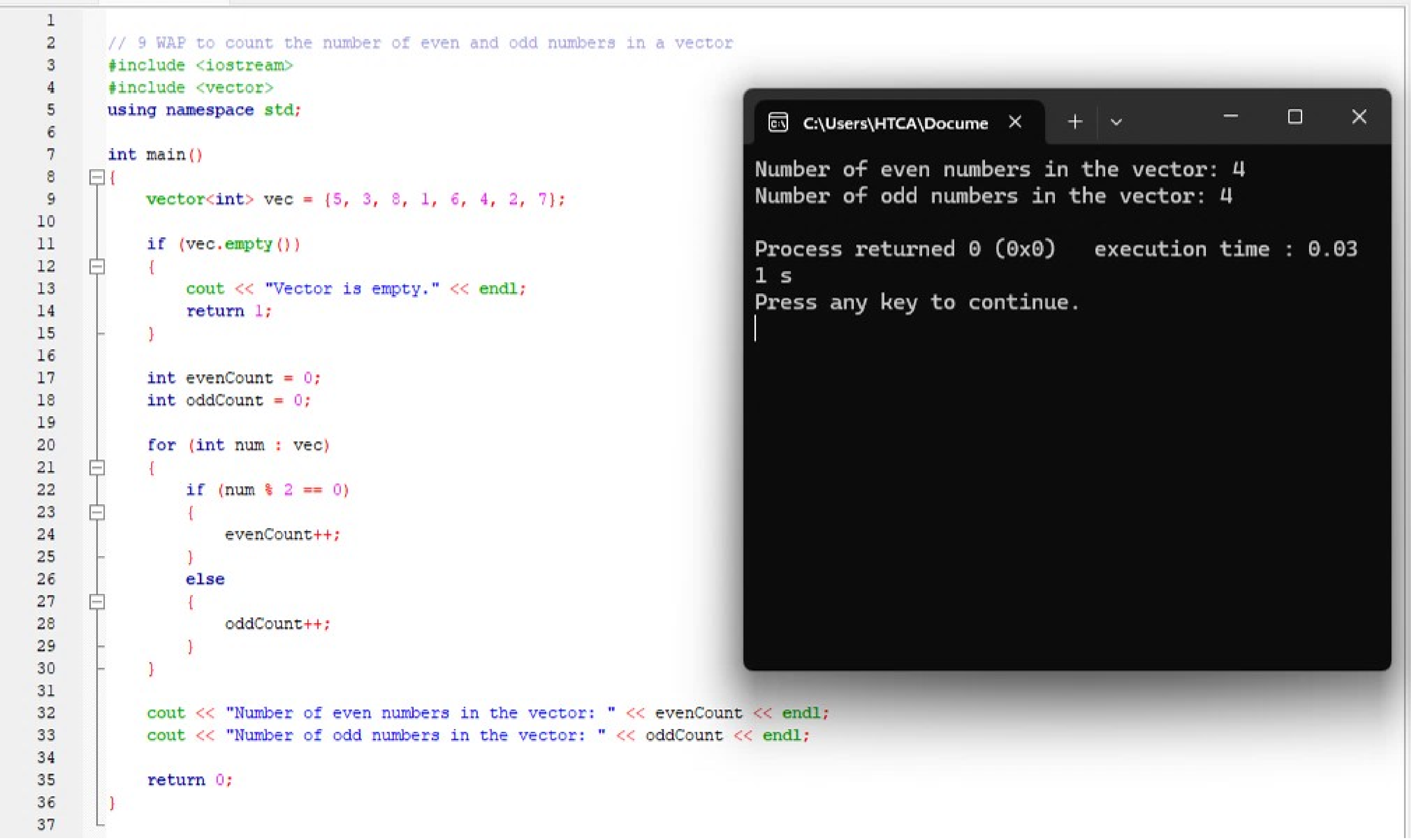
```cpp
// 7 WAP to find the second largest element in a vector

#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main()
{
    vector<int> vec = {5, 3, 8, 1, 6, 4, 2, 7};

    if (vec.size() < 2)
    {
        cout << "Vector does not have enough elements.";
        return 1;
    }

    sort(vec.begin(), vec.end());

    int secondLargest = vec[vec.size() - 2];

    cout << "Second largest element: " << secondLargest << endl;

    return 0;
}
```

```
Second largest element: 7

Process returned 0 (0x0)   execution time
Press any key to continue.
```



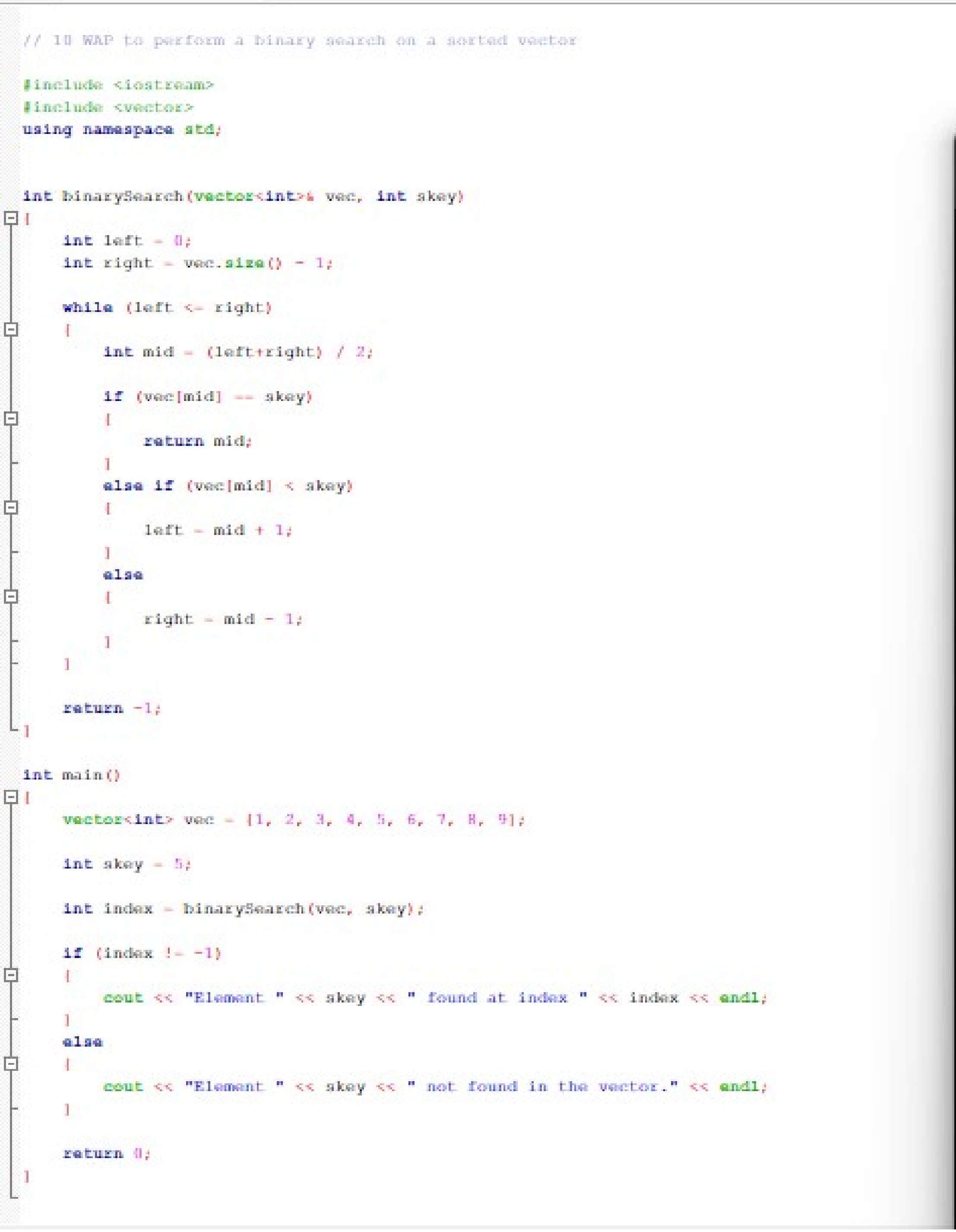
```cpp
//8 implement a program to find the maximum and minimum elements in a array
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main()
{
    vector<int> vec = {5, 3, 8, 1, 6, 4, 2, 7};

    if (vec.empty())
    {
        cout << "Vector is empty." <<endl;
        return 1;
    }


    int maxElement = vec[0];
    int minElement = vec[0];


    for (int num : vec)
    {
        if (num > maxElement)
        {
            maxElement = num;
        }
        if (num < minElement)
        {
            minElement = num;
        }
    }

    cout << "Maximum element in the vector: " << maxElement << endl;
    cout << "Minimum element in the vector: " << minElement <<endl;

    return 0;
}

```

```
Maximum element in the vector: 8
Minimum element in the vector: 1

Process returned 0 (0x0)   execution time : 0.
Press any key to continue.
```

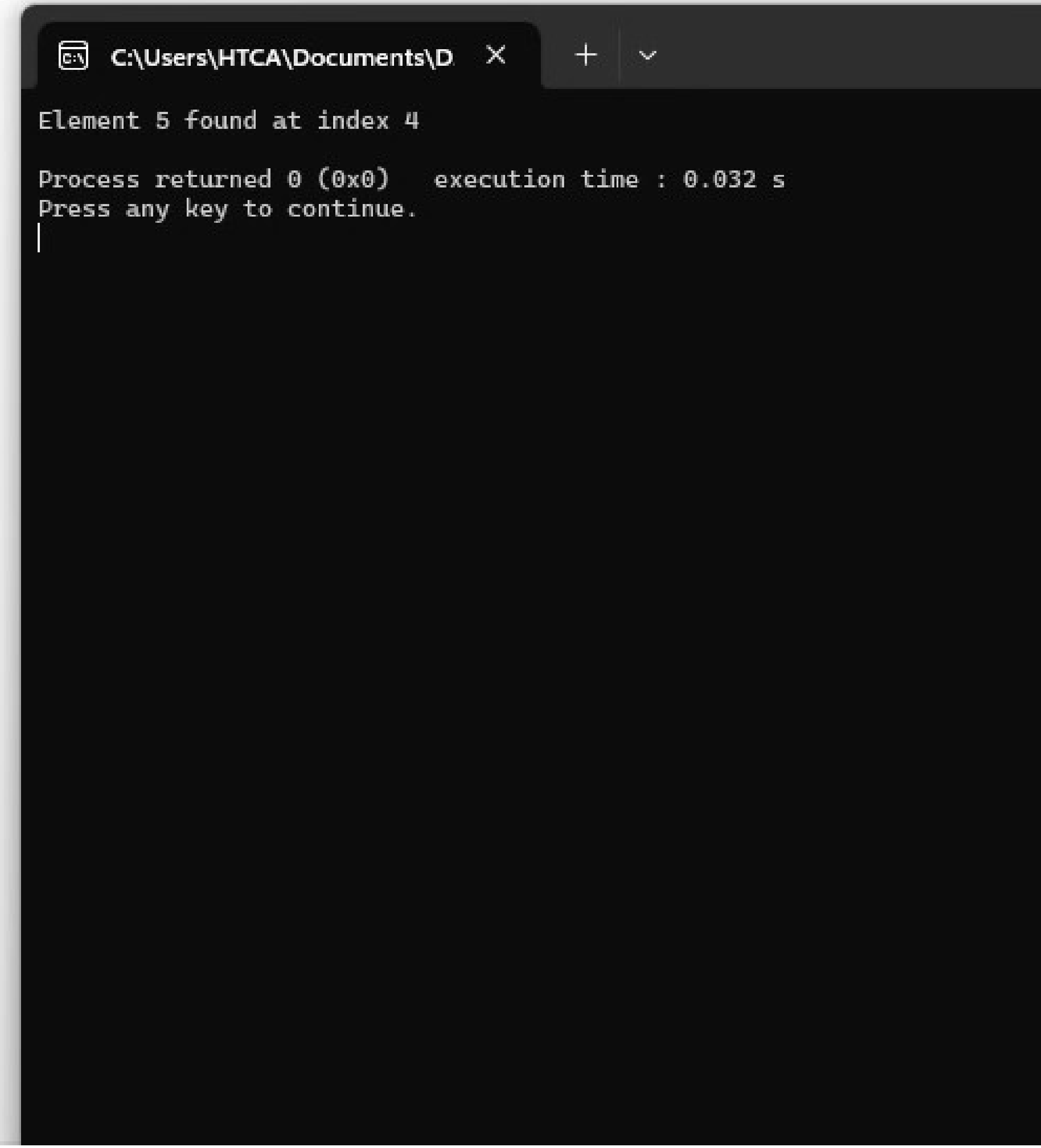


```cpp

// 9 WAP to count the number of even and odd numbers in a vector
#include <iostream>
#include <vector>
using namespace std;

int main()
{
    vector<int> vec = {5, 3, 8, 1, 6, 4, 2, 7};

    if (vec.empty())
    {
        cout << "Vector is empty." << endl;
        return 1;
    }

    int evenCount = 0;
    int oddCount = 0;

    for (int num : vec)
    {
        if (num % 2 == 0)
        {
            evenCount++;
        }
        else
        {
            oddCount++;
        }
    }

    cout << "Number of even numbers in the vector: " << evenCount << endl;
    cout << "Number of odd numbers in the vector: " << oddCount << endl;

    return 0;
}

```

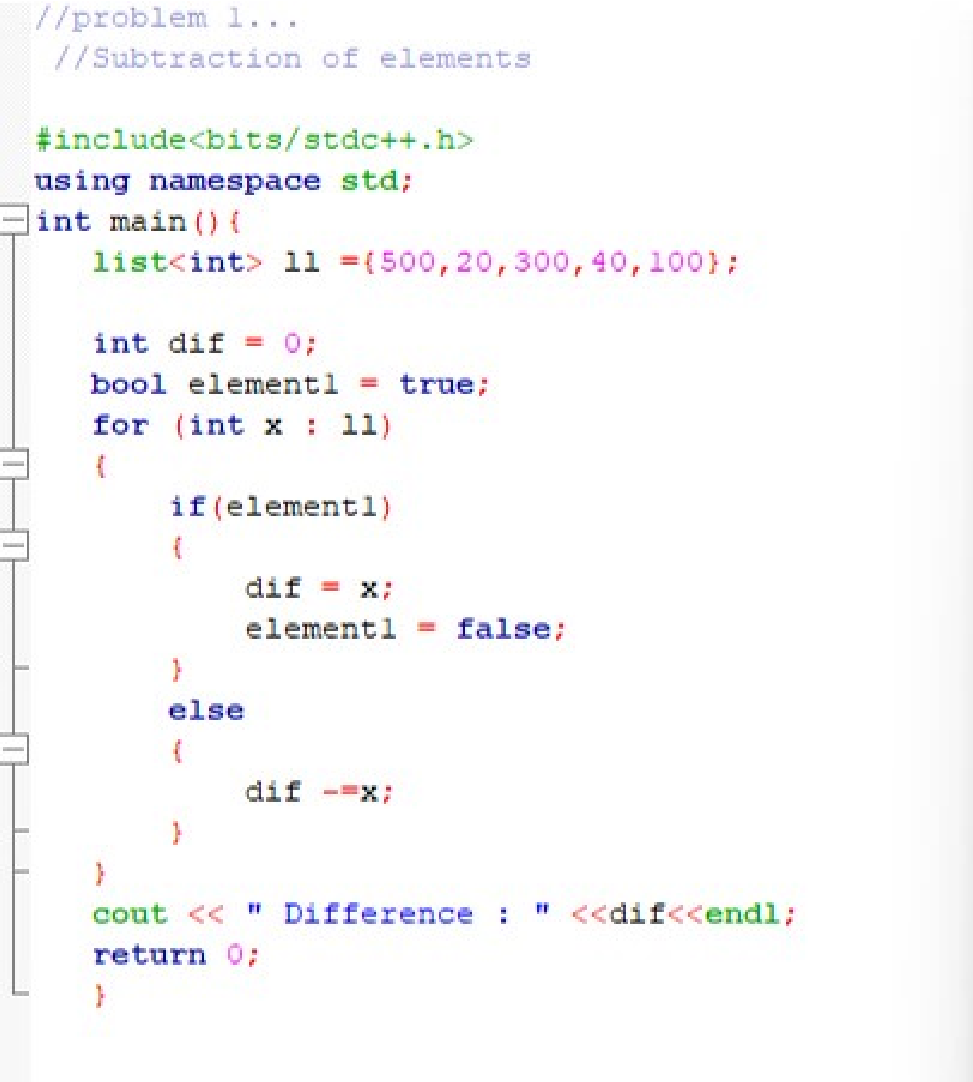
```
Number of even numbers in the vector: 4
Number of odd numbers in the vector: 4

Process returned 0 (0x0)   execution time : 0.03
1 s
Press any key to continue.
```



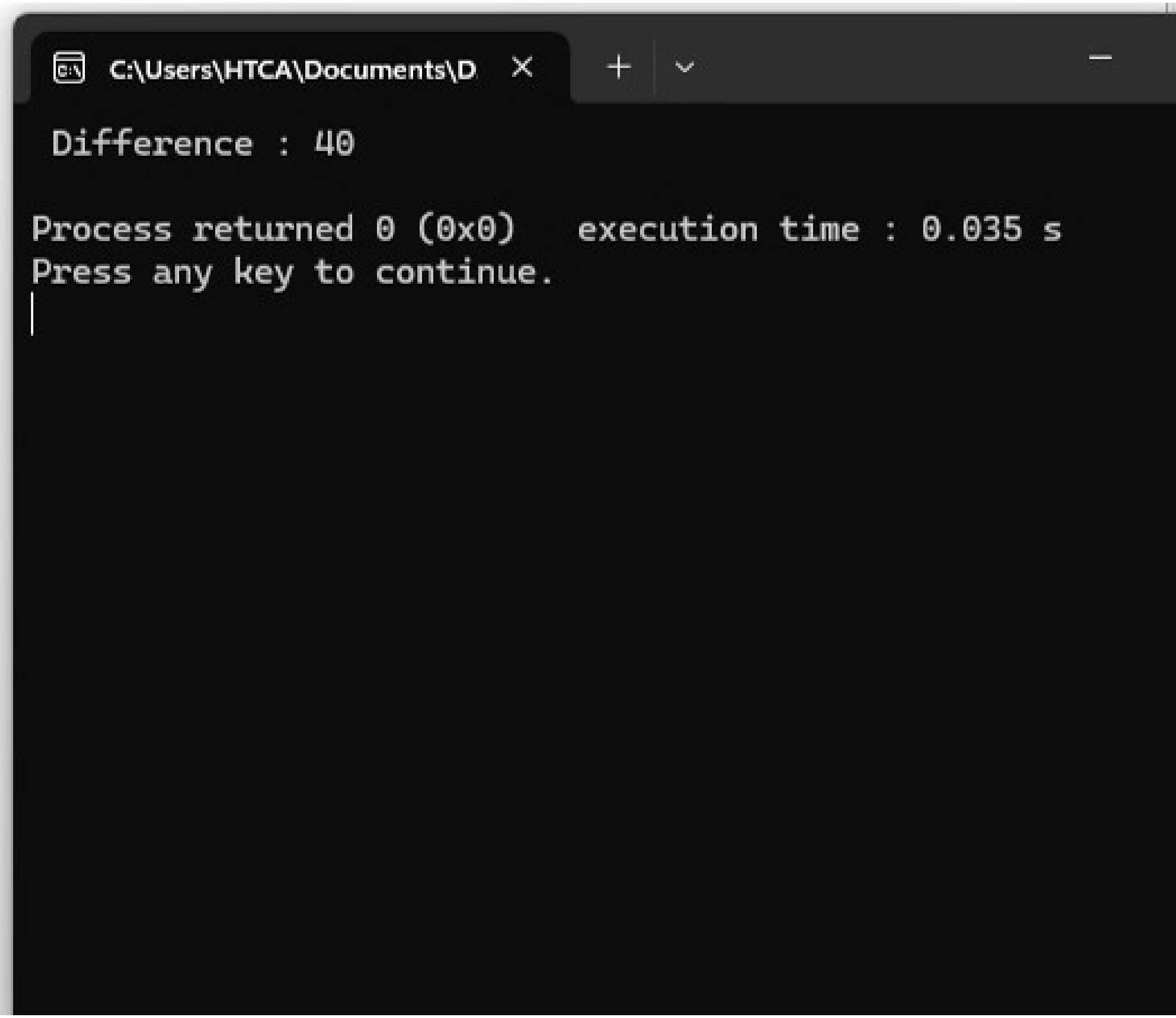
```cpp
// 10 WAP to perform a binary search on a sorted vector

#include <iostream>
#include <vector>
using namespace std;


int binarySearch(vector<int>& vec, int skey)
{
    int left = 0;
    int right = vec.size() - 1;

    while (left <= right)
    {
        int mid = (left+right) / 2;

        if (vec[mid] == skey)
        {
            return mid;
        }
        else if (vec[mid] < skey)
        {
            left = mid + 1;
        }
        else
        {
            right = mid - 1;
        }
    }

    return -1;
}

int main()
{
    vector<int> vec = {1, 2, 3, 4, 5, 6, 7, 8, 9};

    int skey = 5;

    int index = binarySearch(vec, skey);

    if (index != -1)
    {
        cout << "Element " << skey << " found at index " << index << endl;
    }
    else
    {
        cout << "Element " << skey << " not found in the vector." << endl;
    }

    return 0;
}
```



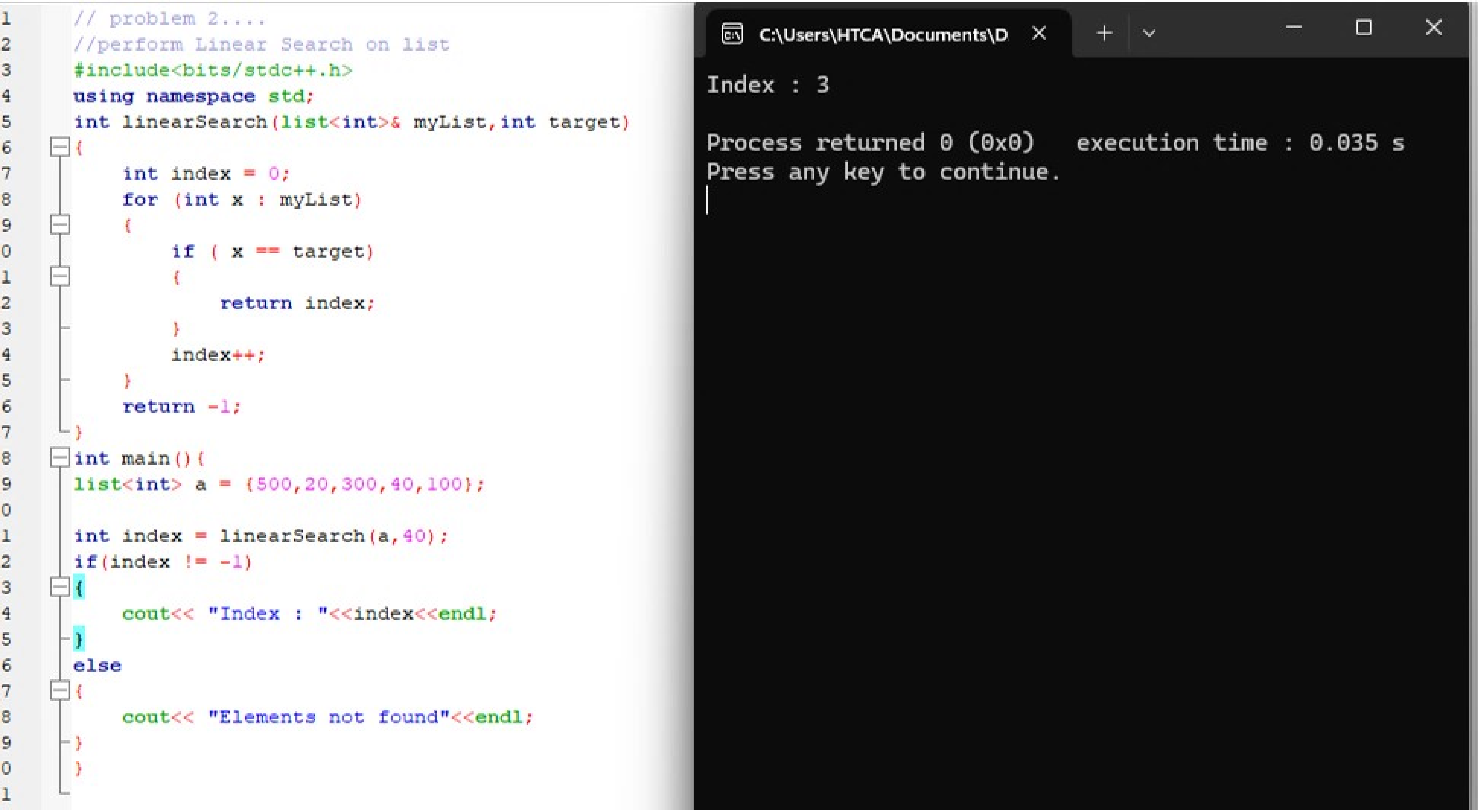
```
Element 5 found at index 4

Process returned 0 (0x0)   execution time : 0.032 s
Press any key to continue.
```



```cpp
//problem 1...
 //Subtraction of elements

#include<bits/stdc++.h>
using namespace std;
int main(){
   list<int> l1 ={500,20,300,40,100};

   int dif = 0;
   bool element1 = true;
   for (int x : l1)
   {
       if(element1)
       {
           dif = x;
           element1 = false;
       }
       else
       {
           dif -=x;
       }
   }
   cout << " Difference : " <<dif<<endl;
   return 0;
   }
```



```
 Difference : 40

Process returned 0 (0x0)   execution time : 0.035 s
Press any key to continue.
```



```cpp
// problem 2....
//perform Linear Search on list
#include<bits/stdc++.h>
using namespace std;
int linearSearch(list<int>& myList,int target)
{
    int index = 0;
    for (int x : myList)
    {
        if ( x == target)
        {
            return index;
        }
        index++;
    }
    return -1;
}
int main(){
list<int> a = {500,20,300,40,100};

int index = linearSearch(a,40);
if(index != -1)
{
    cout<< "Index : "<<index<<endl;
}
else
{
    cout<< "Elements not found"<<endl;
}
}
```

```
Index : 3

Process returned 0 (0x0)   execution time : 0.035 s
Press any key to continue.
```