STL

Vector:

Array এবং vector এর মধ্যে main পার্থক্য Array তে size first এ বলে দেওয়া লাগে এবং এটি sized কিন্তু vector এ size বলা লাগে না ইচ্ছা মতো নেওয়া যায়।

- vector <int> vec1;
- vec1.push_back(element) // কোন element add করবে vector এ
- vec1.pop_back(); // last element delete করে দিবে
- vec1.size(); // vector এর size বলে দিবে।
- V[i] // vector Handel.
- vec1.insert(iterator, how many times, element to insert)

↳ vector<int>:: iterator iter = vec1.begin();

vec1.insert(iter, 56);

এখানে iter first element এ point করছে value এর আগে বসবে। iter+3 করলে এটি 4 এ point করবে এবং value এর আগে বসবে।

data = {1, 2, 3}

- data.insert(data.begin()+2, x);
- data.insert(data.end()-1, 8)
- data.insert(data.begin(), 2, x) // 2টা x add হবে।
- data[i] // access.
- data.front() // প্রথম data
- data.back() // শেষের data.
- convert vector to ~~data~~ array.

```
vector<int> vec = {3, 2, 1, 4};
int *arry = vec.data();
```

*Now arry is new array from vector.

- update:

```
data[position] = value
```

- data1.swap(data2) // দুটি ভিন্ন vector এর মধ্যে data swap হবে
- data.pop_back() // শেষের element delete.
- data.clear() // all vector will be clean.
- data.erase(data.begin()+2, data.begin()+3) // নির্দিষ্ট value delete করবে

```
# vector input.
for(i=0; i<n; i++)
{
    cin>>a;
    v.push_back(a);
}
```

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

Map :

⇒ Map use করা হয় key value pair store করতে।

Hriday → 70

Tanvin → 80

```
map<string, int> marksMap;
```

```
marksMap["Harry"] = 98;        //take input.
```

```
map<string, int> :: iterator iter;
for(iter = marksMap.begin(); iter != marksMap.end(); iter++)
{
    cout << (*iter).first << " " << (*iter).second << endl;
}
```

Index/key

```
marksMap.insert({{"Hriday", 90}, {"Tanvin", 80}});   //take input.
```

```
marksMap.size();   // size
```

```
marksMap["Hriday"] ──→ result will be 90.
  └ marksMap.at["Hriday"]
```

```
for (auto it : marksMap)        //এখানে key এর value গুলো আগে auto sort হয় ]
{
    cout << it.first << " " << it.second << endl;
}
```

marksMap.erase(key) // or specify key to delete করতে চাই

Map এর মধ্যে first এর টা অর্থাৎ key [mp[1]=10 declare করার পর আবার যদি mp[1]=20 delcare করি তাহলে 10 কে update করে 20 থাকবে]

একটা map কে অন্য map এ copy করা যায়।

mp.swap(mp1) // অর্থাৎ mp এর value mp1 এবং mp1 এর গুলো mp এর মধ্যে চলে যাবে।

mp.count(1) //এখানে 1 return করবে যদি map এ এই keyword থাকে।

Map input:

```
for(i=1, i<5, i++)
{
    int a,b;
    cin >> a >> b;
    mp.insert(pair<int, int>(a,b))
}
```

List:

- insertion & deletion first করতে List use করা হয় কিন্তু array তে first access হয়।

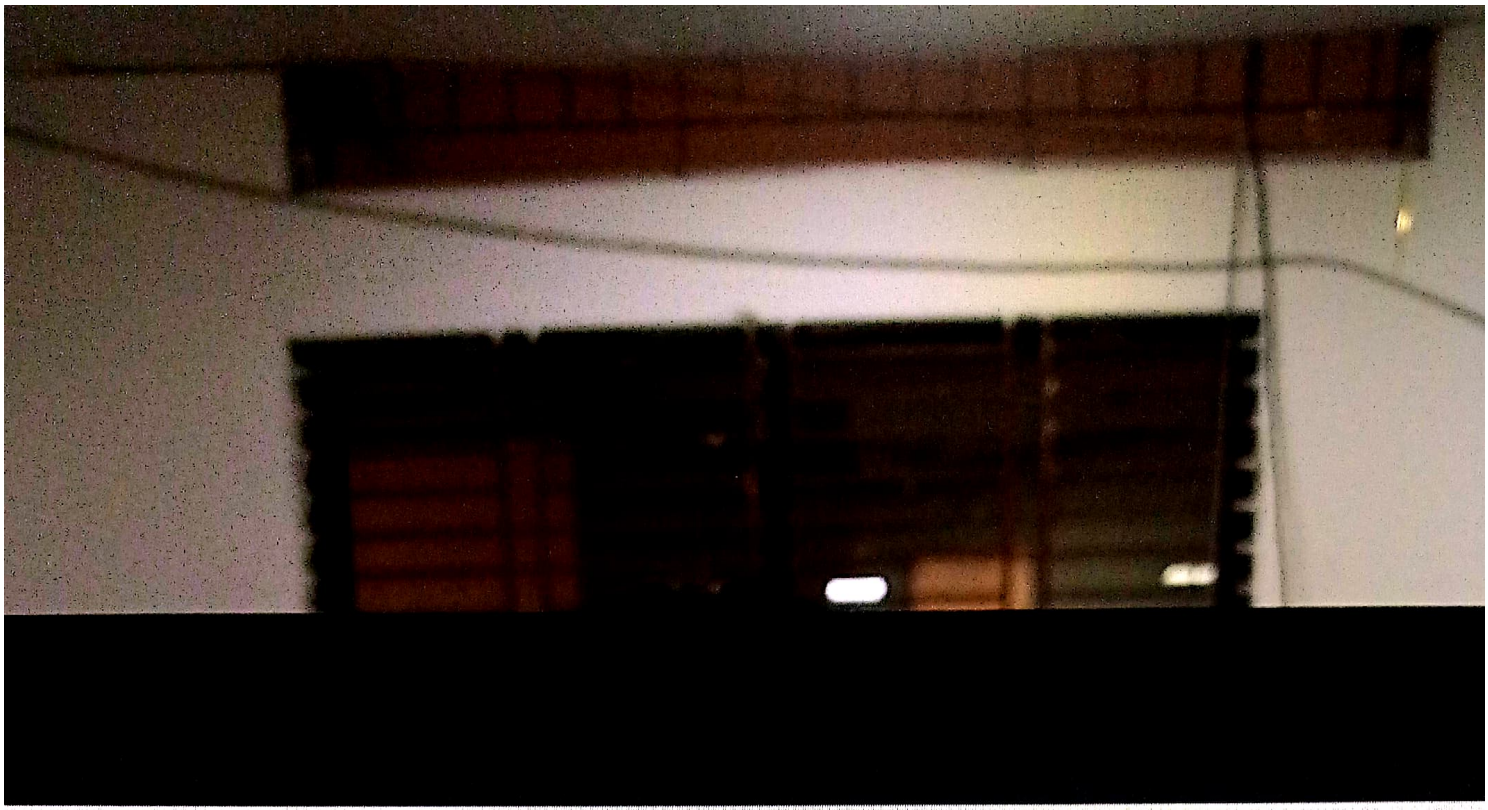
  5 6 7 8

- Array তে contigious memory allocate হয় এতে সমস্যা হলো 6 delete করতে হলে 7 এবং 8 কে shif করতে হবে।

কিন্তু list এ এই কোন নেই 6 delete করতে হলে 5 এর connection 7 এর সাথে করে দিলেই হবে।

- list<int> list1;

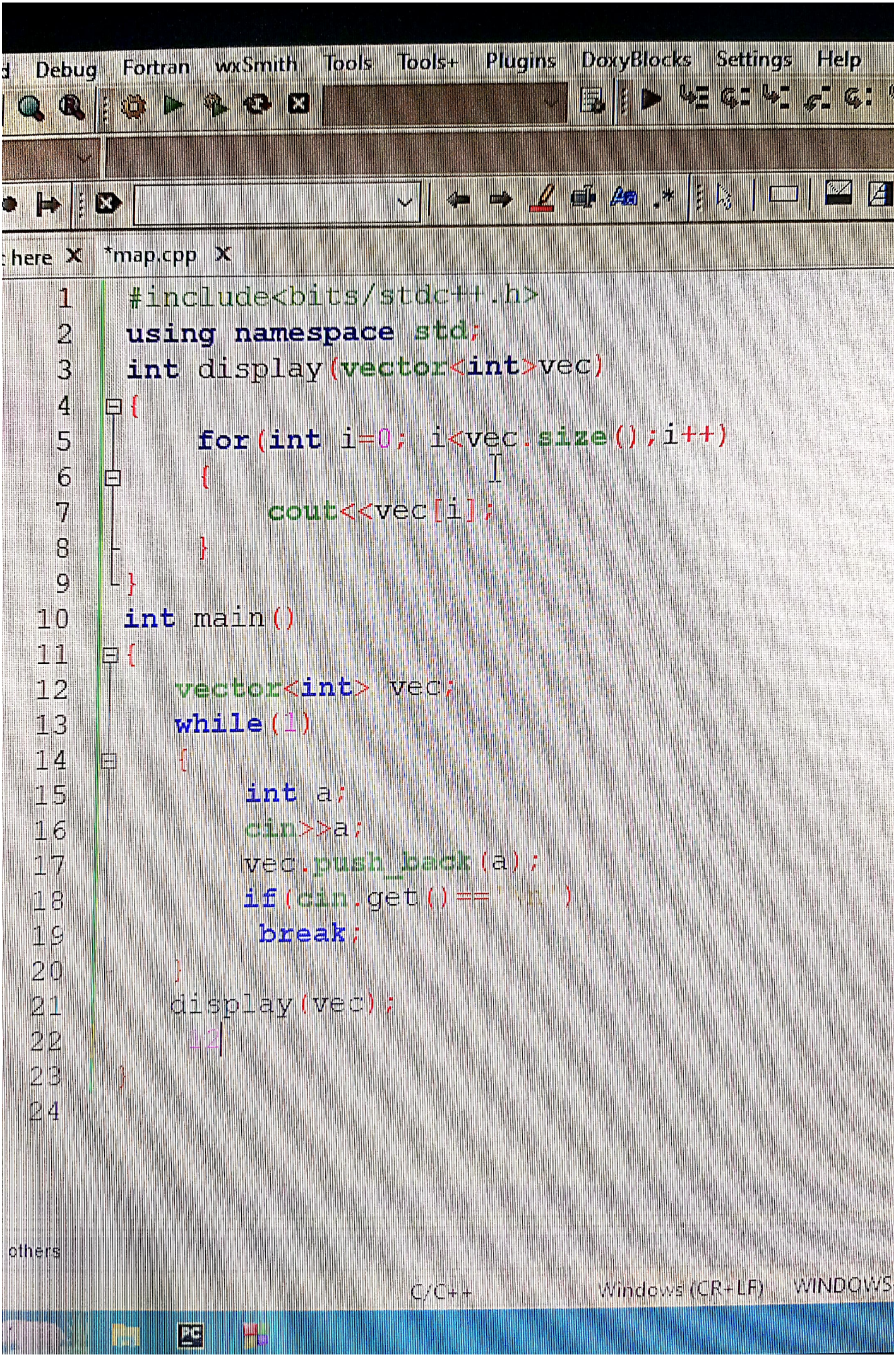
- list1.push_back(5);



uild Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

art here × map.cpp ×

```cpp
#include<bits/stdc++.h>
using namespace std;
int main()
{
    map<int, int> mp;
    for(int i=0;i<3;i++)
    {
        int a,b;
        cin>>a>>b;
        mp.insert(pair<int,int>(a,b));
    }
    for(auto it:mp)
    {
        cout<<it.first<<" "<<it.second<<endl;
    }
}
```


Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
here
*map.cpp
#include<bits/stdc++.h>
using namespace std;
int display(vector<int>vec)
{
for(int i=0; i<vec.size();i++)
{
cout<<vec[i];
}
}
int main()
{
vector<int> vec;
while(1)
{
int a;
cin>>a;
vec.push_back(a);
if(cin.get()=='\n')
break;
}
display(vec);
}
others
C/C++
Windows (CR+LF)
WINDOWS

map.cpp - Notepad

File Edit View

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    map<int,int>m;              // it can be string also
    for(int i=0;i<5;i++)
    {
        int x,y;
        cin>>x>>y;
        m[x]=y;
    }
    for(auto it=m.begin();it!=m.end();it++)   //iterator
    {
        cout<<it->first<<it->second;
    }

}
```

Ln 1, Col 1

map2.cpp - Notepad

File Edit View

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    map<string,int>m;              // it can be string also
    for(int i=0;i<5;i++)
    {
        string str;
        cin>>str;
        int y;
        cin>>y;

        m[str]=y;
    }
    for(auto it=m.begin();it!=m.end();it++)   //iterator
    {
        cout<<it->first<<it->second;
    }

}
```

Ln 1, Col 1

multimap.cpp - Notepad

File Edit View

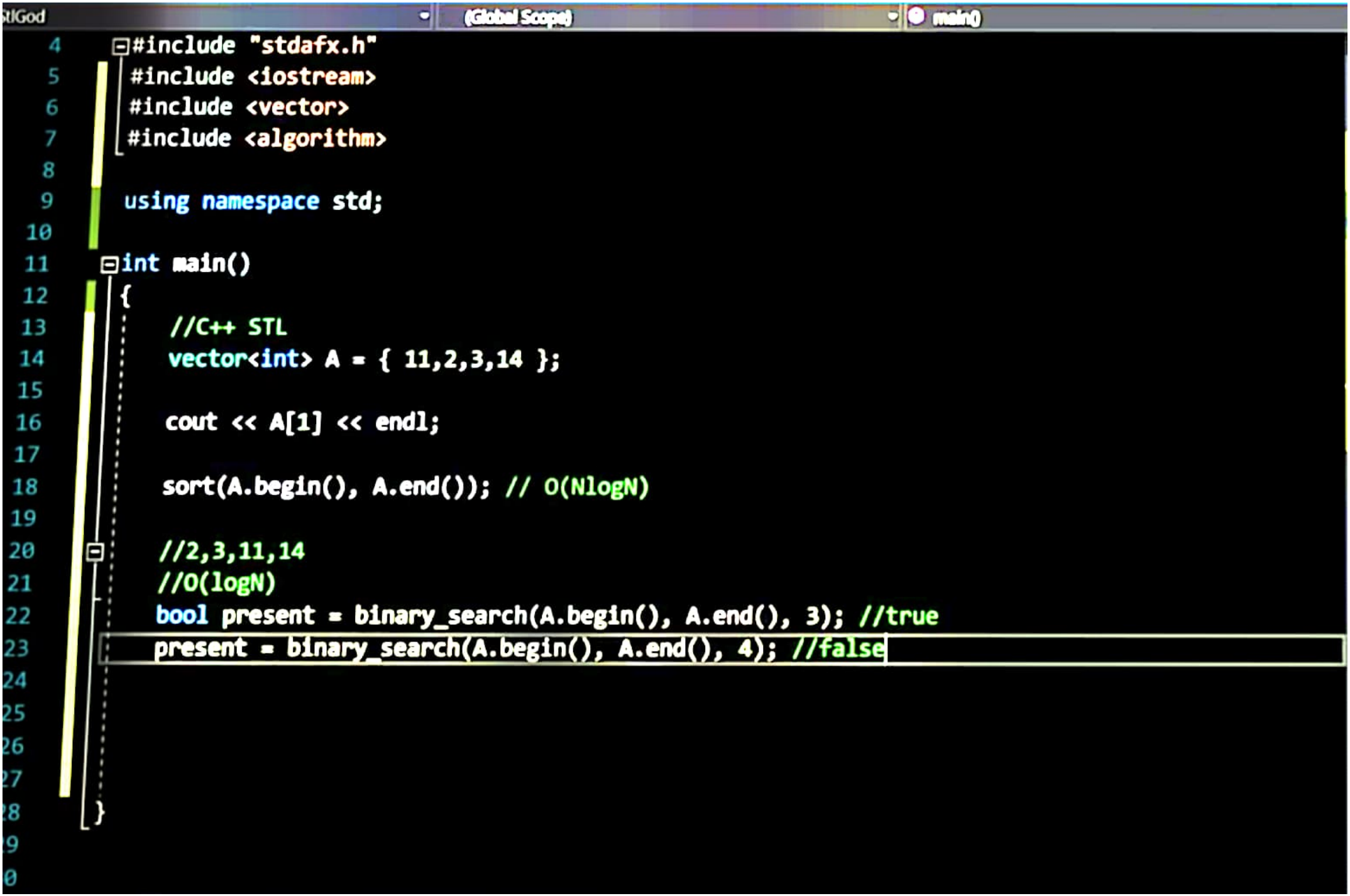
```
// Map ar multimap er modde partokko map e duplicate value cut kore dey multimap e dey na
#include<bits/stdc++.h>
using namespace std;
int main()
{
    multimap<int,int>m;                    // it can be string also
     for(int i=0;i<5;i++)
     {
         int x,y;
         cin>>x>>y;
         m.insert({x,y});
     }
     for(auto it=m.begin();it!=m.end();it++)   //iterator
     {
         cout<<it->first<<it->second;
     }

}
```

v3.cpp - Notepad

File Edit View

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

int display(vector<int> vec1)
{
    for(int i=0;i<vec1.size();i++)
    {
        cout<<vec1[i]<<" ";
    }
    cout<<endl;
}

int main()
{
    vector<int> vec1;
    vec1 = {1,2,3,4};
    display(vec1);
    vec1.push_back(5);
    display(vec1);
    vec1.pop_back();
    display(vec1);
    vector<int> vec2=vec1;
    display(vec2);

    vec1.insert(vec1.begin()+2,5,10);  // data insert
    display(vec1);

    vec2[2]=99; //update
    display(vec2);

    vec1.erase(vec1.begin()+2,vec1.begin()+6);
    display(vec1);

    int *arry = vec1.data();  //Convert vector into array
}
```

Ln 1, Col 1


StlGod
(Global Scope)
main()
#include "stdafx.h"
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
int main()
{
//C++ STL
vector<int> A = { 11,2,3,14 };
cout << A[1] << endl;
sort(A.begin(), A.end()); // O(NlogN)
//2,3,11,14
//O(logN)
bool present = binary_search(A.begin(), A.end(), 3); //true
present = binary_search(A.begin(), A.end(), 4); //false
}

```
cout << A[1] << endl;

sort(A.begin(), A.end()); // O(NlogN)

//2,3,11,14
//O(logN)
bool present = binary_search(A.begin(), A.end(), 3); //true
present = binary_search(A.begin(), A.end(), 4); //false

A.push_back(100);
present = binary_search(A.begin(), A.end(), 100); //true

//2,3,11,14,100
A.push_back(100);
A.push_back(100);
A.push_back(100);
A.push_back(100);
//2,3,11,14,100, 100, 100, 100, 123
A.push_back(123);

vector<int>::iterator it = lower_bound(A.begin(), A.end(), 100);
```

```
cout << A[1] << endl;

sort(A.begin(), A.end()); // O(NlogN)

//2,3,11,14
//O(logN)
bool present = binary_search(A.begin(), A.end(), 3); //true
present = binary_search(A.begin(), A.end(), 4); //false

A.push_back(100);
present = binary_search(A.begin(), A.end(), 100); //true

//2,3,11,14,100
A.push_back(100);
A.push_back(100);
A.push_back(100);
A.push_back(100);
//2,3,11,14,100, 100, 100, 100, 123
A.push_back(123);

vector<int>::iterator it = lower_bound(A.begin(), A.end(), 100); // >=
vector<int>::iterator it2 = upper_bound(A.begin(), A.end(), 100); // >
```

```
cout << A[1] << endl;

sort(A.begin(), A.end()); // O(NlogN)

//2,3,11,14
//O(logN)
bool present = binary_search(A.begin(), A.end(), 3); //true
present = binary_search(A.begin(), A.end(), 4); //false

A.push_back(100);
present = binary_search(A.begin(), A.end(), 100); //true

//2,3,11,14,100
A.push_back(100);
A.push_back(100);
A.push_back(100);
A.push_back(100);
//2,3,11,14,100, 100, 100, 100, 123
A.push_back(123);

vector<int>::iterator it = lower_bound(A.begin(), A.end(), 100); // >=
vector<int>::iterator it2 = upper_bound(A.begin(), A.end(), 100); // >

cout << *it << " " << *it2 << endl;
cout << it2 - it << endl; //
```