STL Vector in C++

vector is a dynamic array provided by the C++ Standard Template Library (STL). It can resize automatically when elements are added or removed, and it provides fast random access.

---

1. Include Header

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

---

2. Creating a Vector

```
vector<int> v1;             // empty vector of ints
vector<int> v2(5);          // vector of 5 integers initialized to 0
vector<int> v3(5, 10);      // vector of 5 integers initialized to 10
vector<int> v4 = {1, 2, 3}; // initializer list
```

---

3. Common Functions with Examples

3.1. Add Elements

```
vector<int> v;
v.push_back(4); // Adds 4 at the end
```

Amortized Time Complexity: O(1)

3.1.2. Access Elements

```
cout << v[0] << endl;      // Direct access (no bounds check)
cout << v.at(0) << endl;   // Safe access (throws exception if out of bounds)
cout << v.front() << endl; // Access first element
cout << v.back() << endl;  // Access last element
```

3.1.3. Size and Capacity

```
cout << v.size() << endl; // Number of elements
cout << v.capacity();     // Allocated storage
```

3.1.4. Shrink to Fit

```
v.shrink_to_fit(); // Removes extra spaces
```

3.1.5. Iterate Over Vector

```
for (int x : v)
    cout << x << " "; // range-based loop
for (int i = 0; i < v.size(); i++)
    cout << v[i] << " "; // index loop
```

3.1.6. Insert and Erase

```
v.insert(v.begin() + 1, 99);           // Insert 99 at index 1
v.erase(v.begin() + 1);                // Remove element at index 1
v.erase(v.begin() + 1, v.begin() + 3); // Remove element at from index 1 to index 2
```

3.1.7. Remove Last Element

```
v.pop_back(); // Remove last element
```

3.1.8. Clear All Elements

```
v.clear(); // vector becomes empty
```

3.1.9. Check Empty

```
if (v.empty())
        cout << "Empty";
```

---

4. Vector Iterator

```
vector<int>::iterator it;
```

```
for (it = v.begin(); it != v.end(); it++)
{
    cout << *it << " ";
}
```

---

5. Sorting a Vector

```
sort(v.begin(), v.end());
```

---

6. Reverse a Vector

```
reverse(v.begin(), v.end());
```

---

7. Find Element

```
auto it = find(v.begin(), v.end(), 3);
if (it != v.end())
    cout << "Found at index: " << (it - v.begin());
```

---

8. Get Unique Values

```
sort(v.begin(), v.end());
auto last = unique(v.begin(), v.end());
v.erase(last, v.end());
```

---

9. Get min and max element

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    vector<int> v = {1, 4, 5, 1, 2, 3, 5, 6, 4, 1, 2, 5, 3};

    auto it = min_element(v.begin(), v.end());
    auto it2 = max_element(v.begin(), v.end());
    cout << *it << " " << *it2;

    return 0;
}
```

10. Get count

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    vector<int> v = {1, 4, 5, 1, 2, 3, 5, 6, 4, 1, 2, 5, 3};

    cout << count(v.begin(), v.end(), 1);

    return 0;
}
```

11. Fill all elements

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    vector<int> v = {1, 4, 5, 1, 2, 3, 5, 6, 4, 1, 2, 5, 3};
    fill(v.begin(), v.end(), 0);
    for (auto &x : v)
        cout << x << " ";
    return 0;
}
```

12. Rotate elements

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    vector<int> v = {1, 2, 3, 4, 5};

    rotate(v.begin(), v.begin() + 3, v.end()); // left rotation
    rotate(v.begin(), v.end() - 3, v.end()); // right rotation

    for (int &x : v)
        cout << x << " ";
    return 0;
}
// O(N)
```

13. Get Sum

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    vector<int> v = {1, 2, 3, 4, 5};
    int s = accumulate(v.begin(), v.end(), 0);
    cout << s << endl;
    return 0;
}
```

14. Practice Problems

1. **Remove Duplicates from a List**
   Given a list of integers, remove all duplicates and print the unique elements in sorted order.

```
#include <bits/stdc++.h>

using namespace std;

int main()

{

    vector<int> v = {1, 4, 5, 1, 2, 3, 5, 6, 4, 1, 2, 5, 3};

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

    v.erase(unique(v.begin(), v.end()), v.end());

    for (auto &x : v)

    {

        cout << x << endl;

    }

    return 0;

}
```

2. **Frequency Count**
   Count how many times each unique number appears in a vector.

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    vector<int> v = {1,4,5,1,2,3,5,6,4,1,2,5,3};
    vector<int> freq(7, 0);
    for(auto &x: v) freq[x]++;
    sort(v.begin(), v.end());
    v.erase(unique(v.begin(), v.end()), v.end());
    for(auto &x: v)
    {
        cout<<x<<" "<<freq[x]<<endl;
    }
    return 0;
}
```

3. **Sort by Frequency**
   Given a vector of integers, sort the elements by their frequency (most frequent first).

```
#include <bits/stdc++.h>
using namespace std;
vector<int> freq(7, 0);
bool comp(int &a, int &b)
{
    if (freq[a] == freq[b])
    {
        return a < b;
    }
    else
        return freq[a] > freq[b];
}
int main()
{
    vector<int> v = {1, 4, 5, 1, 2, 3, 5, 6, 4, 1, 2, 5, 3};

    for (auto &x : v)
        freq[x]++;
    sort(v.begin(), v.end(), comp);
    for (auto &x : v)
        cout << x << " " << freq[x] << endl;
    return 0;
}
```

4. **Erase All Occurrences**
 Remove all occurrences of a given value from a vector.

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    vector<int> v = {1, 4, 5, 1, 2, 3, 5, 6, 4, 1, 2, 5, 3};
    while (find(v.begin(), v.end(), 1) != v.end())
    {
        v.erase(find(v.begin(), v.end(), 1));
    }
    for (int &x : v)
        cout << x << " ";
    return 0;
}
// O(N^2)
```

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    vector<int> v = {1, 4, 5, 1, 2, 3, 5, 6, 4, 1, 2, 5, 3};

    v.erase(remove(v.begin(), v.end(), 1), v.end());
    for (int &x : v)
        cout << x << " ";
    return 0;
}
// O(N)
```

5. **Sort vector of objects**
   Given a vector of objects, sort the elements.

```
#include <bits/stdc++.h>
using namespace std;
class Students
{
public:
    int roll;
    int number;
    Student(int r, int n)
    {
        roll = r;
        number = n;
    }
};
bool comp(Student &a, Student &b)
{
    if (a.number == b.number)
    {
        return a.roll < b.roll;
    }
    else
        return a.number < b.number;
}
int main()
{
    vector<Student> v = {Student(1, 99), Student(5, 80), Student(3, 80)};
    sort(v.begin(), v.end(), comp);
    for (auto &x : v)
    {
        cout << x.roll << " " << x.number << endl;
    }
    return 0;
}
```

15. More Practice Problems

1. https://codeforces.com/group/c3FDI9EUi9/contest/262795/problem/A
2. https://codeforces.com/group/c3FDI9EUi9/contest/262795/problem/D
3. https://codeforces.com/group/c3FDI9EUi9/contest/262795/problem/I

4. https://codeforces.com/group/c3FDI9EUi9/contest/262795/problem/L
5. https://codeforces.com/group/c3FDI9EUi9/contest/262795/problem/J