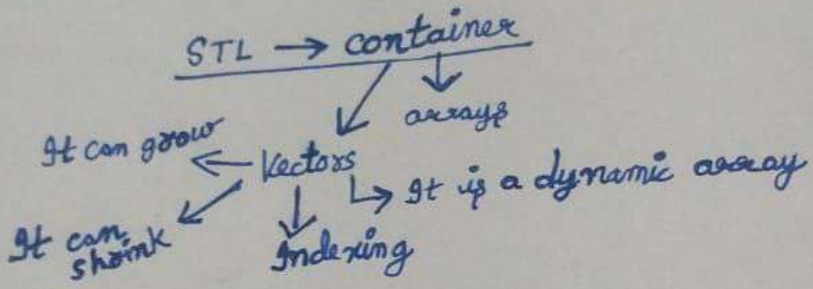
C++ STL



Header File :-

- #include <vector> - Header required to use vectors in your program.
- Alternative : If using #include <bits/stdc++.h>, then vector is already and doesn't need to be defined separately.

Vector Declaration :-

- vector <type> v
- Syntax : vector <type> variable_name

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

int main() {
    vector<int> v;
    cout << v.size() << "\n";

    v.push_back(1); //Adds element 1 at the end.
    v.push_back(2);
    v.push_back(3);

    v.insert(v.begin() + 10, 4); // Trying to insert at index 10
    // Inserting at index 10 is invalid if the size of v < 10.


    v.pop_back();  // Removes the last element.

    //two methods for printing elements of vector

    // 1st method
    for (int i = 0; i < v.size(); i++) {
        cout << v[i] << " ";
    }

    //2nd method
    for(int num:v){
        cout << num << " ";
    }

    v.erase(v.begin() + 1);
    // Erases the element at index 1.
    // v.begin() + 1 points to the second element (0-based indexing).

    auto it = find(v.begin(), v.end(), 2);
    cout << *it << "\n"; //output : 2

    // Searches for the element 2 in the vector v from v.begin() to v.end().
    // If found, it returns an iterator pointing to the element, otherwise it returns v.end().
```

```cpp
sort(v.begin(), v.end()); // Sorts the vector in ascending order.
// The sort function rearranges the elements in the vector from the beginning to the end.

auto it = binary_search(v.begin(), v.end(), 4);
cout << it << "\n"; // Output: 0 or 1 (false or true)
// Checks if the element 4 is present in the sorted vector v.

int idx = lower_bound(v.begin(), v.end(), 2) - v.begin();
cout << idx << "\n"; // Output: 0 or index of the first occurrence of 2
// lower_bound returns an iterator pointing to the first element that is not less than 2
// If 2 is present, it returns the index of the first occurrence of 2

int idx2  = upper_bound(v.begin(), v.end(), 2) - v.begin();
cout << idx2 << "\n"; // Output: index of the first element greater than 2
// upper_bound returns an iterator pointing to the first element that is greater than 2
// If 2 is present, it returns the index of the first element greater

cout <<upper_bound(v.begin(), v.end(), 2) - lower_bound(v.begin(), v.end(), 2) << "\n";
// This line calculates the count of occurrences of the element 2 in the vector v.

cout << accumulate(v.begin(), v.end(), 0) << "\n";
// This line calculates the sum of all elements in the vector v.

cout << *max_element(v.begin(), v.end()) << "\n";
// This line finds the maximum element in the vector v.

vector<int> arr(10,1);
// This line creates a vector of size 10, initialized with the value 1.

vector<vector<int>> vec(3, vector<int>(3, 0));
// This line creates a 2D vector (3x3) initialized with the value 0.
for(int i = 0; i < vec.size(); i++) {
    for(int j = 0; j < vec[i].size(); j++) {
        cout << vec[i][j] << " ";
    }
    cout << "\n";
}
```