

## 0\_cpp\_stl.cpp

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  // Pairs
5  pair<int , int> p = {1, 2};
6  cout<<p.first<<" "<<p.second;
7
8  pair<int, pair<int, int>> p = {1, {2, 3}};
9  cout<<p.first<<" "<<p.second.first<<" "<<p.second.second;
10
11  pair<int int> p[] = {{1, 2}, {3, 4}, {5, 6}};
12  cout<<p[1].second;
13
14  // Vectors
15  vector<int> v;
16
17  v.push_back(1);
18  v.emplace_back(2);
19
20  vector<pair<int, int>> v;
21  v.push_back({1, 2});
22  v.emplace_back(1, 2);
23
24  v.push_back({1, 2});
25  v.pop_back();
26
27  vector<int> v(5);
28  vector<int> v(5, 0);
29  vector<int> v(5, -1);
30
31  vector<int> v(v1);
32
33
34  vector<int>::iterator it = v.begin();
35
36  it++
37  cout<<*(it);
38
39  it = it + 2;
40  cout<<*(it);
41
42  vector<int>::iterator it = v.begin();
43  vector<int>::iterator it = v.end();
44  vector<int>::iterator it = v.rend();
45  vector<int>::iterator it = v.rbegin();
46
47  cout<<v[0];
48  cout<<v.at(0);
49  cout<<v.front();
50  cout<<v.back();
51  cout<<v.size();
```

```

52
53     for(vector<int>::iterator it = v.begin(); it != v.end(); it++){
54         cout<<*(it)<<" ";
55     }
56
57     for (auto it=v.begin(); it!=v.end(); it++) {
58         cout<<*(it)<<" ";
59     }
60
61     for(auto it : v){
62         cout<<it<<" ";
63     }
64
65     v.erase(v.begin() + 1);
66     v.erase(v.begin() + 1, v.begin() + 3); // [start, end)
67
68     vector<int> v(2, 100);
69     v.insert(v.begin(), 300)
70     v.insert(v.begin() + 1, 2, 200)
71     v.insert(v.begin(), v2.begin(), v2.end());
72
73     v1.swap(v2);
74     v.clear();
75     cout<<v.empty();
76
77
78 // List (doubly linked list)
79     list<int> ls;
80
81     ls.push_back(1);
82     ls.emplace_back(2);
83
84     ls.push_front(1);
85     ls.emplace_front(2);
86
87     ls.pop_front(1);
88     ls.pop_back(2);
89
90 // Deque
91     deque<int> dq;
92     dq.push_back(1);
93     dq.emplace_back(2);
94     dq.push_front(1);
95     dq.emplace_front(2);
96
97     dq.pop_back();
98     dq.pop_front();
99
100     dq.front();
101     dq.back();
102
103 // Stack
104     stack<int> st;
105     st.push(1);

```

```
106     st.emplace(2);
107     st.pop();
108
109     st.top();
110
111     st.size();
112     st.empty();
113
114 // Queue
115     queue<int> q;
116     q.push(1);
117     q.emplace(2);
118     q.pop();
119
120     q.front();
121     q.back();
122
123     q.front() += 4;
124
125     q.size();
126     q.empty();
127
128
129 // Priority Queue
130     // Max Heap
131     priority_queue<int> pq;
132     pq.push(1);
133     pq.emplace(2);
134
135     pq.top();
136
137     pq.pop();
138
139     // Min Heap
140     priority_queue<int, vector<int>, greater<int>> pq;
141
142 // Set
143     set<int> s;
144     s.insert(1);
145     s.emplace(2);
146
147     auto it = s.find(1);
148     s.erase(it);
149     s.erase(2);
150
151     auto it = s.lower_bound(1);
152     auto it = s.upper_bound(1);
153
154
155 // Multiset
156     multiset<int> ms;
157     ms.insert(1);
158     ms.insert(1);
159     ms.insert(1);
```

```

160
161     ms.erase(1)
162
163     int cnt = ms.count(1);
164
165     ms.erase(ms.find(1))
166
167     ms.erase(ms.find(1), ms.find(1)+2);
168
169 // unordered set
170     unordered_set<int> us;
171     us.insert(1);
172     us.insert(1);
173     us.insert(2);
174
175 // Map
176     map<int, int> m;
177     map<int, pair<int, int>> m;
178     map<pair<int, int>, int> m;
179
180     m[1] = 3;
181     map.emplace({1, 3})
182     map.insert({1, 3});
183
184     for(auto it: m){
185         cout<<it.first<<" "<<it.second<<endl;
186     }
187
188     cout<<m[1];
189
190     auto it = m.find(1);
191     cout<<*(it).second;
192
193 // Multimap
194 // Sort
195     sort(a, a+n);
196     sort(a.begin(), a.end());
197
198     sort(a+2, a+4);
199     sort(a, a+n, greater<int>);
200
201     pair<int, int> a[] = {{1, 2}, {3, 4}, {5, 6}};
202     sort(a, a+n, comp);
203
204     bool comp(pair<int, int> &a, pair<int, int> &b){
205         if(a.first < b.first){
206             return true;
207         }else if(a.first == b.first){
208             return a.second > b.second;
209         }
210         return false;
211     }
212
213 // Basics

```

```

214     int num  = 7;
215     int cnt  = __builtin_popcount(num); // no of set bits
216
217     long long num = 1234567843232;
218     long long cnt = __builtin_popcountll(num);
219
220     // Permutations
221     string s = "abc";
222     string s = "123";
223     string s = "321";
224     sort(s.begin(), s.end());
225
226     do {
227         cout<<s<<<endl;
228     } while(next_permutation(s.begin(), s.end()));
229
230     int maxele = *max_element(a, a+n);
231     int minele = *min_element(a, a+n);
232
233

```