0_cpp_stl.cpp

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

```
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
         for(auto it : v){
61
              cout<<it<< ";</pre>
 62
         }
 63
 64
65
         v.erase(v.begin() + 1);
66
         v.erase(v.begin() + 1, v.begin() + 3); // [start, end)
67
         vector<int> \mathbf{v}(2, 100);
 68
 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();</pre>
 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
         ls.pop front(1);
87
88
         ls.pop_back(2);
89
90
     // Deque
         deque<int> dq;
91
92
         dq.push_back(1);
         dq.emplace_back(2);
93
94
         dq.push_front(1);
95
         dq.emplace_front(2);
96
97
         dq.pop_back();
98
         dq.pop_front();
99
         dq.front();
100
101
         dq.back();
102
103
     // Stack
104
         stack<int> st;
105
         st.push(1);
```

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

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

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