

STL

C++ Standard Template Library

STL有什么

- 算法 (Algorithms)
- 容器 (Containers)
- 函数 (Functors)
- 迭代器 (Iterators)

- 简单容器
 - pair
- 列表容器
 - **vector**
 - list (slist)
 - deque
 - **queue** (priority_queue)
 - **stack**
- 联系容器
 - **set** (**multiset**, unordered_set)
 - **map** (multimap, unordered_map)
- 其他容器
 - **string**
 - bitset
 - valarray

stack

先进后出

```
stack <int> st;  
st.push(x);      // 入栈  
st.pop();        // 出栈  
st.top();        // 查看栈顶元素  
st.empty();      // 是否为空  
st.size();       // 元素个数
```

queue

先进先出

```
queue <int> q;  
q.push(x);           // 入队  
q.pop();             // 出队  
q.front();           // 查看队列首元素  
q.empty();           // 是否为空  
q.size();            // 元素个数
```

set

自动排序去重!

```
1  #include <bits/stdc++.h>
2  #define MAXN 1001
3  using namespace std;
4
5  int main() {
6      int x = 1;
7
8      set <int> s;
9      s.insert(x); // 插入元素
10     s.erase(x);  // 删除元素
11     s.count(x);   // 判断元素是否在set里 (bool! )
12
13     set <int> :: iterator it = s.find(x); C++98
14     cout << *it << endl;
15
16     auto it14 = s.find(x); C++14
17     cout << *it14 << endl;
18
19     if(s.find(x) != s.end()) cout << *(s.find(x)) << endl;
20
21     return 0;
22 }
```

set

```
1  #include <bits/stdc++.h>
2  #define MAXN 1001
3  using namespace std;
4
5  int main() {
6      set<int> s;
7      s.insert(10);
8      s.insert(30);
9      s.insert(20);
10
11     for (set<int>::iterator it = s.begin(); it != s.end(); ++it) {
12         cout << *it << " ";
13     }
14     cout << endl;
15
16     return 0;
17 }
```

C++98的遍历

set

```
1  #include <bits/stdc++.h>
2  #define MAXN 1001
3  using namespace std;
4
5  int main() {
6      set<int> s;
7      s.insert(10);
8      s.insert(30);
9      s.insert(20);
10
11      for (auto x : s) {
12          cout << x << " ";
13      }
14      cout << endl;
15
16      return 0;
17 }
```

C++14的遍历

multiset

```
1  #include <bits/stdc++.h>
2  #define MAXN 1001
3  using namespace std;
4
5  int main() {
6      multiset<int> s;
7      s.insert(1);
8      s.insert(1);
9      s.insert(4);
10     s.insert(5);
11     s.insert(1);
12     s.insert(4);
13
14     for (auto x : s) {
15         cout << x << " ";
16     }
17     cout << endl;
18
19     return 0;
20 }
```

排序但不去重！！

map

```
1  #include <bits/stdc++.h>
2  #define MAXN 1001
3  using namespace std;
4
5  int main() {
6      int x = 1;
7
8      map <string, int> mp;
9      mp["apple"] = 3;
10     mp["banana"]++;
11
12     cout << mp["apple"] << endl;
13     cout << mp["banana"] << endl;
14     cout << mp["orange"] << endl;
15
16     mp.erase("banana");
17
18     return 0;
19 }
```

map <key, val> mp;

按照key排序

map

```
1  #include <bits/stdc++.h>
2  #define MAXN 1001
3  using namespace std;
4
5  int main() {
6      int x = 1;
7
8      map <string, int> mp;
9      mp["apple"] = 3;
10     mp["orange"] = 0;
11     mp["banana"]++;
12
13     map <string, int> :: iterator it = mp.find("banana");
14     if (it != mp.end())
15         cout << "Found banana, value = " << mp["banana"] << endl;
16     else cout << "Not found" << endl;
17
18     map <string, int> :: iterator it1 = mp.find("orange");
19     if (it1 == mp.end()) cout << "No orange!" << endl;
20     else cout << "Has orange!" << endl;
21
22     for(map <string, int> :: iterator i = mp.begin(); i != mp.end(); i++) {
23         cout << i->first << " " << i->second << endl;
24     }
25     return 0;
26 }
```

```
1  Found banana, value = 1
2  Has orange!
3  apple 3
4  banana 1
5  orange 0
```

小练习

15:45-16:45