

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

为什么要讲STL

- 因为是必考的
- 只要你在写C++，你就离不开STL
- 老师老师，我手写不行吗？
- 又慢又难写，为什么不用**现成的最佳模板**呢

万能头文件

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

pair为什么不讲

- struct 结构体

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  struct node {
5      int first, second;
6  };
7
8  int main() {
9      pair <int, int> p;
10     p.first = 1;
11     p.second = 2;
12     cout << p.first << " " << p.second << endl;
13
14     node a;
15     a.first = 1;
16     a.second = 2;
17     cout << a.first << " " << a.second << endl;
18
19     return 0;
20 }
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  struct node {
5      string str;
6      int first, second, third;
7  };
8
9  int main() {
10     node a;
11     a.str = "CUHKSZ";
12     a.first = 1;
13     a.second = 2;
14     a.third = 3;
15     cout << a.str << endl;
16     cout << a.first << " " << a.second << endl;
17     cout << a.third << endl;
18     return 0;
19 }
```

sort

```
1  #include <bits/stdc++.h>
2  #define MAXN 10001
3  using namespace std;
4
5  int s[MAXN];
6
7  int main() {
8      int n;
9      cin >> n;
10     for(int x = 1; x <= n; x++) cin >> s[x];
11     sort(s+1, s+n+1);
12     for(int x = 1; x <= n; x++) cout << s[x] << " ";
13     cout << endl;
14     return 0;
15 }
```

对数组 s 进行从小到大排序

sort with cmp

```
int s[MAXN];
```

```
bool cmp (int x, int y) {  
    return x < y;  
}
```

```
sort(s+1, s+n+1, cmp);
```

从小到大

```
int s[MAXN];
```

```
bool cmp (int x, int y) {  
    return x > y;  
}
```

```
sort(s+1, s+n+1, cmp);
```

从大到小

sort with cmp

```
1  #include <bits/stdc++.h>
2  #define MAXN 10001
3  using namespace std;
4
5  struct node {
6      int score;
7      string name;
8  } s[MAXN];
9
10 bool cmp (node x, node y) {
11     return x.score > y.score;
12 }
13
14 int main() {
15     int n;
16     cin >> n;
17     for(int x = 1; x <= n; x++) cin >> s[x].name >> s[x].score;
18     sort(s+1, s+n+1, cmp);
19     for(int x = 1; x <= n; x++) cout << s[x].name << " ";
20     cout << endl;
21     return 0;
22 }
```

按 score 排序

```
1  #include <bits/stdc++.h>
2  #define MAXN 10001
3  using namespace std;
4
5  struct node {
6      int math, english;
7      string name;
8  } s[MAXN];
9
10 bool cmp (node x, node y) {
11     if(x.math != y.math) return x.math > y.math;
12     else return x.english > y.english;
13 }
```

先按 math 排序
再按 english

```
1  #include <bits/stdc++.h>
2  #define MAXN 10001
3  using namespace std;
4
5  string s[MAXN];
6
7  bool cmp(string x, string y) {
8      return x < y;
9  }
```

按字典序

binary_search

```
1  #include <bits/stdc++.h>
2  #define MAXN 10001
3  using namespace std;
4
5  int s[MAXN];
6
7  int main() {
8      int n, a;
9      cin >> n >> a;
10     for(int x = 1; x <= n; x++) cin >> s[x];
11     sort(s+1, s+n+1);
12     cout << binary_search(s+1, s+n+1, a);
13     cout << endl;
14     return 0;
15 }
```

对有序数组 s 进行
二分搜索查找 a
返回bool

vector

- 动态数组 `vector <类型> 名称;`
- 支持对数组的所有函数，而且用起来比较方便
- 还有额外的函数

```
vector <int> v1;  
vector <string> vs;  
vector <node> v2;  
vector <int> g[MAXN];
```

```
vector <int> v(n);  
vector <int> v(n, 5);  
vector <int> v = {1, 1, 4};  
vector <vector <int>> v2;
```

vector的基础操作

```
1  #include <bits/stdc++.h>
2  #define MAXN 10001
3  using namespace std;
4
5  int main() {
6      int a = 1, b = 2;
7      vector <int> v;
8      v.push_back(a); // 在末尾插入元素 a
9      v.push_back(b); // 在末尾插入元素 b
10     v.pop_back(); // 删除末尾元素
11     int sz = v.size(); // 返回vector的大小 (长度)
12     if (v.empty()) cout << "Empty!"; // 判断是否为空
13     v.clear(); // 清空
14
15     // 遍历
16     for(int i = 0; i < v.size(); i++) cout << v[i] << " ";
17     cout << endl;
18
19     return 0;
20 }
```

sort和binary_search (vector)

```
1  #include <bits/stdc++.h>
2  #define MAXN 10001
3  using namespace std;
4
5  bool cmp(int x, int y) {
6      return x > y;
7  }
8
9  int main() {
10     vector<int> v1, v2;
11     int n, a;
12     cin >> n >> a;
13     for(int i = 1; i <= n; i++) {
14         int inp;
15         cin >> inp;
16         v1.push_back(inp);
17         v2.push_back(inp);
18     }
19     sort(v1.begin(), v1.end());
20     sort(v2.begin(), v2.end(), cmp);
21     for(int i = 0; i < v1.size(); i++) cout << v1[i] << " ";
22     cout << endl;
23     for(int i = 0; i < v2.size(); i++) cout << v2[i] << " ";
24     cout << endl;
25     cout << binary_search(v1.begin(), v1.end(), a) << endl;
26     return 0;
27 }
```

// 对下标是 1 到 n 的数组排序

sort(s+1, s+n+1, cmp);

// 对 vector 排序

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

vector的进阶操作

```
v.resize(10); // 变成长度10, 新增元素为0
```

```
v.resize(10, -1); // 新增元素赋值为-1
```

```
v.insert(v.begin()+2, 100); // 在第3个位置插入100
```

```
v.erase(v.begin()+2); // 删除第3个元素
```

```
v.erase(v.begin(), v.begin()+3); // 删除前3个
```

```
vector<int> a = v; // copy (赋值)
```

```
swap(v, a); // 交换两个vector内容
```

unique

```
1  #include <bits/stdc++.h>
2  #define MAXN 10001
3  using namespace std;
4
5  int s[MAXN];
6
7  int main() {
8      int n;
9      cin >> n;
10     for(int x = 1; x <= n; x++) cin >> s[x];
11     sort(s+1, s+n+1);
12     n = unique(s+1, s+n+1) - (s+1);
13     for(int x = 1; x <= n; x++) cout << s[x] << " ";
14     cout << endl;
15     return 0;
16 }
```

对数组 s 的连续相同元素
进行去重
(与sort结合食用更加)

unique (vector)

对 vector 进行排序后去重

```
1  #include <bits/stdc++.h>
2  #define MAXN 10001
3  using namespace std;
4
5  int main() {
6      int n;
7      cin >> n;
8      vector <int> v(n);
9      for(int i = 0; i < v.size(); i++) cin >> v[i];
10     sort(v.begin(), v.end());
11     v.erase(unique(v.begin(), v.end()), v.end());
12     for(int i = 0; i < v.size(); i++) cout << v[i] << " ";
13     cout << endl;
14     return 0;
15 }
```

next_permutation

不停的找下一个排列

```
1  #include <bits/stdc++.h>
2  #define MAXN 10001
3  using namespace std;
4
5  int s[MAXN];
6
7  int main() {
8      int n;
9      cin >> n;
10     for(int x = 1; x <= n; x++) cin >> s[x];
11     do {
12         for(int x = 1; x <= n; x++) cout << s[x] << " ";
13         cout << endl;
14     } while (next_permutation(s+1, s+n+1));
15     return 0;
16 }
```

```
1  #include <bits/stdc++.h>
2  #define MAXN 10001
3  using namespace std;
4
5  int main() {
6      int n;
7      cin >> n;
8      vector <int> v(n);
9      for(int x = 0; x < n; x++) cin >> v[x];
10     do {
11         for(int x = 0; x < n; x++) cout << v[x] << " ";
12         cout << endl;
13     } while (next_permutation(v.begin(), v.end()));
14     return 0;
15 }
```

string的基础操作

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main() {
5      string s;           // 定义空字符串
6      s = "hello";        // 给字符串赋值
7      cin >> s;           // 读入一个单词（遇空格、换行结束）
8      getline(cin, s);    // 读入整行（含空格）
9      cout << s << endl; // 输出字符串
10
11     int l = s.length(); // 字符串的长度
12
13     // 遍历字符串
14     for(int i = 0; i < n; i++) {
15         cout << s[i] << " ";
16     }
17     cout << endl;
18     return 0;
19 }
```

string的进阶操作

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main() {
5      string a, b;
6      a = "abcd";
7      b = "efg";
8      string str = a + b; // 拼接
9      str += a;           // 拼接
10     bool tf = (a < b); // 字典序比大小
```

```
12     // 查找子串第一次出现的位置, 返回下标或npos
13     if(str.find("cde") != string::npos) {
14         cout << str.find("cde") << endl;
15     }
16
17     // 提取子串 (从下标 3 开始提取长度为 2 的子串)
18     cout << str.substr(3, 2) << endl;
19
20     // 替换子串 (从下标 3 开始将长度为 2 的子串替换成 xzy)
21     str.replace(3, 2, "xzy");
22
23     // 删除子串 (从下标 0 开始删除长度为 2 的子串)
24     str.erase(0, 2);
25
26     // 插入子串 (从下标 2 开始插入 ijk)
27     str.insert(2, "ijk");
28
29     cout << str << endl;
30     return 0;
31 }
```

```

50 }
51
52     return 0;
53
54
55     conf << endl; \\ e 2 4
56     for(int i = 0; i < 3; i++) conf << z[i] << " ";
57     level26(z, z+3);
58     int z[3] = {4, 2, e};
59
60
61
62     conf << endl; \\ 3 5 7
63     for(int i = 0; i < lambda26(); i++) conf << lambda[i] << " ";
64     level26(lambda.begin(), lambda.end());
65     vector<int> lambda = {7, 5, 3};
66
67
68
69     conf << (g == p) << endl; \\ false (7)
70     level26(g.begin(), g.end());
71     string p = g;
72     string g = "gpcqdcps";
73     \\ 快速排序字符串
74
75
76
77     conf << str << endl; \\ 357
78     level26(str.begin(), str.end());
79     string str = "753";
80     int main() {
81
82
83     namespace zfq:
84     #include <bits/stdc++.h>

```

转回，着急

别着急，有反转

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main() {
5      string str = "123";
6      reverse(str.begin(), str.end());
7      cout << str << endl; // 321
8
9      // 快速判断回文字符串
10     string a = "abcdedcba";
11     string b = a;
12     reverse(a.begin(), a.end());
13     cout << (a == b) << endl; // true (1)
14
15     vector <int> v = {1, 2, 3};
16     reverse(v.begin(), v.end());
17     for(int i = 0; i < v.size(); i++) cout << v[i] << " ";
18     cout << endl; // 3 2 1
19
20     int s[3] = {4, 5, 6};
21     reverse(s, s+3);
22     for(int i = 0; i < 3; i++) cout << s[i] << " ";
23     cout << endl; // 6 5 4
24
25     return 0;
26 }
```

类型转换 (大于等于C++11)

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main() {
5      string str = "123";
6      int num = stoi(str);
7      long long lnum = stoll(str);
8
9      string flstr = "10.123";
10     float flnum = stof(flstr);
11     double dbnum = stod(flstr);
12
13     int a = 114514;
14     string b = to_string(a);
15     return 0;
16 }
```

Dev-CPP用不了
不是很建议在赛场上用

训练赛

10:30-12:00