

## C++-队列-支持优先级的队列

题目描述：实现一个支持优先级的队列，高优先级先出队列；同优先级时先进先出。 如果两个输入数据和优先级都相同，则后一个数据不入队列被丢弃。 队列存储的数据内容是一个整数。

输入描述：一组待存入队列的数据（包含内容和优先级）

输出描述：队列的数据内容（优先级信息输出时不再体现）

补充说明：不用考虑输入数据不合法的情况，测试数据不超过 100 个

### 示例

#### 示例1

输入：(10,1),(20,1),(30,2),(40,3)

输出：40,30,10,20

说明：输入样例中，向队列写入了4个数据，每个数据由数据内容和优先级组成。

输入和输出内容都不含空格。

数据40的优先级最高，所以最先输出，其次是30；10和20优先级相同，所以按输入顺序输出。

#### 示例2

输入：(10,1),(10,1),(30,2),(40,3)

输出：40,30,10

说明：输入样例中，向队列写入了4个数据，每个数据由数据内容和优先级组成。

输入和输出内容都不含空格。

数据40的优先级最高，所以最先输出，其次是30；两个10和10构成重复数据，被丢弃一个。

```

1  #include <iostream>
2  using namespace std;
3
4  struct qnode {
5      int value;
6      int level;
7      qnode* next;
8      qnode() :value(NULL), level(NULL), next(nullptr) {}
9      bool operator==(const qnode& qn)const{
10         return (value==qn.value&&level==qn.level);
11     }
12 };
13
14 class pqueue {
15 private:
16     qnode* head;
17 public:
18     pqueue() :head(new qnode) {}
19     void push(const int& value, const int& level) {
20         if(have(value,level)){
21             return;
22         }
23         auto p = head;
24         while (p->next) {
25             if (p->next->level < level)
26                 break;
27             p = p->next;
28         }
29         auto q = p->next;
30         p->next = new qnode;
31         p->next->value = value;
32         p->next->level = level;
33         p->next->next = q;
34     }
35 }

```

```

36 | bool have(const int& value, const int& level)const{
37 |
38 |     bool flag = false;
39 |     auto p = head->next;
40 |     while (p) {
41 |         if(p->value==value&&p->level==level){
42 |             flag = true;
43 |             break;
44 |         }
45 |         p = p->next;
46 |     }
47 |
48 |     return flag;
49 |
50 | }
51 |
52 | void print()const {
53 |     auto p = head->next;
54 |     while (p) {
55 |         cout << p->value ;
56 |         if(p->next)
57 |             cout<<',';
58 |         p = p->next;
59 |     }
60 | }
61 | };
62 |

```

```

62
63 int main() {
64
65     string s;
66     string sub;
67     int value, level;
68     bool flag = true;
69     cin >> s;
70
71     pqueue pq;
72
73     for (int i = 0; i < s.size(); i++) {
74
75         const char& c = s[i];
76         if (isdigit(c)) {
77             sub.append(1, c);
78         }
79         else if (!sub.empty()) {
80             if (flag) {
81                 flag = false;
82                 value = stoi(sub);
83                 sub.clear();
84             }
85             else {
86                 flag = true;
87                 level = stoi(sub);
88
89                 pq.push(value, level);
90
91                 sub.clear();
92             }
93         }
94     }
95
96     pq.print();
97
98
99 }
100 // 64 位输出请用 printf("%lld")

```