

```
1: // $Id: predobjsort.cpp,v 1.5 2015-02-05 19:09:40-08 - - $
2:
3: #include <algorithm>
4: #include <iostream>
5: #include <string>
6: #include <vector>
7:
8: using namespace std;
9:
10: template <class item_t>
11: struct incrcmp {
12:     bool operator () (const item_t &left, const item_t &right) {
13:         return left < right;
14:     }
15: };
16:
17: template <class item_t>
18: struct decrcmp {
19:     bool operator () (const item_t &left, const item_t &right) {
20:         return left > right;
21:     }
22: };
23:
24: template <class item_t>
25: struct lengthcmp {
26:     bool operator () (const item_t &left, const item_t &right) {
27:         return left.size() < right.size()
28:             or (not (right.size() < left.size()) and left < right);
29:     }
30: };
31:
32: template <class itor_t>
33: void print (string label, const itor_t &begin, const itor_t &end) {
34:     cout << endl << label << " order:" << endl;
35:     for (itor_t itor = begin; itor != end; ++itor) {
36:         cout << " " << *itor;
37:     }
38:     cout << endl;
39: }
40:
41: int main (int argc, char **argv) {
42:     char **end = argv + argc;
43:     vector<string> vec;
44:     for (char **itor = argv; itor < end; ++itor) vec.push_back (*itor);
45:     print ("initial", vec.begin(), vec.end());
46:
47:     sort (vec.begin(), vec.end());
48:     print ("default sort", vec.begin(), vec.end());
49:
50:     sort (vec.begin(), vec.end(), incrcmp<string>());
51:     print ("incrcmp<string>", vec.begin(), vec.end());
52:
53:     sort (vec.begin(), vec.end(), decrcmp<string>());
54:     print ("decrmp<string>", vec.begin(), vec.end());
55:
56:     sort (vec.begin(), vec.end(), lengthcmp<string>());
57:     print ("lengthcmp<string>", vec.begin(), vec.end());
58:
```

```
59:     return 0;
60: }
61:
62: //TEST// predobjsort hello world foo bar baz quux >predobjsort.out 2>&1
63: //TEST// mkpspdf predobjsort.ps predobjsort.cpp* predobjsort.out
64:
```

[illegible]

```
1:
2: initial order:
3:  predobjsort hello world foo bar baz quux
4:
5: default sort order:
6:  bar baz foo hello predobjsort quux world
7:
8: incrcmp<string> order:
9:  bar baz foo hello predobjsort quux world
10:
11: decrcmp<string> order:
12:  world quux predobjsort hello foo baz bar
13:
14: lengthcmp<string> order:
15:  bar baz foo quux hello world predobjsort
```