

```
1: // $Id: testxless.cpp,v 1.21 2014-04-24 17:42:23-07 - - $
2:
3: // Example of how to use xless object on a pair of strings.
4:
5: #include <iomanip>
6: #include <iostream>
7: #include <string>
8:
9: using namespace std;
10:
11: template <typename Type>
12: struct xless {
13:     bool operator() (const Type &left, const Type &right) const {
14:         return left < right;
15:     }
16: };
17:
18: template <typename Type, class Less=xless<Type>>
19: void compare (const Type &left, const Type &right) {
20:     Less less;
21:     cout << boolalpha;
22:     bool is_less = less (left, right);
23:     bool is_equal = not less (left, right) and not less (right, left);
24:     bool is_greater = not is_less and not is_equal;
25:     cout << left << " < " << right << " = " << is_less << endl;
26:     cout << left << " == " << right << " = " << is_equal << endl;
27:     cout << left << " > " << right << " = " << is_greater << endl;
28:     cout << endl;
29: }
30:
31: int main () {
32:     compare ("hello", "world");
33:     compare ("qwert", "qwert");
34:     compare ("zxcvb", "asdfg");
35:     return 0;
36: }
37:
38: //TEST// testxless >testxless.out 2>&1
39: //TEST// mkpspdf testxless.ps testxless.cpp* testxless.out
40:
```

[illegible]

```
1: hello < world = false
2: hello == world = false
3: hello > world = true
4:
5: qwert < qwert = false
6: qwert == qwert = true
7: qwert > qwert = false
8:
9: zxcvb < asdfg = false
10: zxcvb == asdfg = false
11: zxcvb > asdfg = true
12:
```