```cpp
/* The following code example is taken from the book
 * "The C++ Standard Library - A Tutorial and Reference"
 * by Nicolai M. Josuttis, Addison-Wesley, 1999
 *
 * (C) Copyright Nicolai M. Josuttis 1999.
 * Permission to copy, use, modify, sell and distribute this software
 * is granted provided this copyright notice appears in all copies.
 * This software is provided "as is" without express or implied
 * warranty, and with no claim as to its suitability for any purpose.
 */
#include "algostuff.hpp"
using namespace std;

bool lessLength (const string& s1, const string& s2)
{
    return s1.length() < s2.length();
}

int main()
{
    vector<string> coll1;
    vector<string> coll2;

    // fill both collections with the same elements
    coll1.push_back ("1xxx");
    coll1.push_back ("2x");
    coll1.push_back ("3x");
    coll1.push_back ("4x");
    coll1.push_back ("5xx");
    coll1.push_back ("6xxxx");
    coll1.push_back ("7xx");
    coll1.push_back ("8xxx");
    coll1.push_back ("9xx");
    coll1.push_back ("10xxx");
    coll1.push_back ("11");
    coll1.push_back ("12");
    coll1.push_back ("13");
    coll1.push_back ("14xx");
    coll1.push_back ("15");
    coll1.push_back ("16");
    coll1.push_back ("17");
    coll2 = coll1;

    PRINT_ELEMENTS(coll1,"on entry:\n ");

    // sort (according to the length of the strings)
    sort (coll1.begin(), coll1.end(),           // range
          lessLength);                          // criterion
    stable_sort (coll2.begin(), coll2.end(),    // range
                 lessLength);                   // criterion

    PRINT_ELEMENTS(coll1,"\nwith sort():\n ");
    PRINT_ELEMENTS(coll2,"\nwith stable_sort():\n ");
}
```