```
// $Id: insert_forward_list.cpp,v 1.18 2016-08-12 15:27:12-07 - - $

// Example of insert_ascending to a forward_list.
// This runs in O(n) time.
// Obviously there are better algorithms.

#include <forward_list>
#include <iostream>
#include <iterator>
#include <string>
#include <vector>
using namespace std;

// Insert ascending order but no duplicates.
// Can't use any equality operator, only less.
template <typename T, class Less = less<T>>
void insert_ascending (forward_list<T>& list, const T& item) {
   Less less;
   auto curr = list.begin();
   auto prev = list.end();
   while (curr != list.end()) {
      if (not (less (*curr, item))) break;
      prev = curr;
      ++curr;
   }
   if (prev == list.end()) {
      list.push_front (item);
   }else if (curr == list.end() or less (item, *curr)) {
      list.insert_after (prev, item);
   }
}

int main() {
   forward_list<string> list;
   istream_iterator<string> cin_itor (cin);
   istream_iterator<string> end_file;
   ostream_iterator<string> cout_itor (cout, "\n");
   vector<string> data (cin_itor, end_file);
   cout << endl << "Unsorted data:" << endl;
   copy (data.begin(), data.end(), cout_itor);
   for (auto& word: data) insert_ascending (list, word);
   cout << endl << "Sorted data:" << endl;
   copy (list.begin(), list.end(), cout_itor);
}

/*
//TEST// echo hello world foo bar baz qux This is some test data. \
//TEST// | insert_forward_list >insert_forward_list.out 2>&1
//TEST// mkpspdf insert_forward_list.ps insert_forward_list.cpp* \
//TEST//         insert_forward_list.out
*/
```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting insert_forward_list.cpp
insert_forward_list.cpp: $Id: insert_forward_list.cpp,v 1.18 2016-08-12 15:27:12-07 - - $
g++ -g -O0 -std=gnu++14 -rdynamic -Wall -Wextra -Wold-style-cast insert_forward_list.cpp -o insert_forward_list -lglut -lGLU -lGL -lX11 -lrt -lm
cpplint.py.perl insert_forward_list.cpp
Done processing insert_forward_list.cpp
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished insert_forward_list.cpp
```

```

Unsorted data:
hello
world
foo
bar
baz
qux
This
is
some
test
data.

Sorted data:
This
bar
baz
data.
foo
hello
is
qux
some
test
world
```