```
/****
 *
 * This is a test program for StrLists, as defined in strlist.{h.C}, q.q.v.
 * The module test plan is as follows:
 *
 *  (1) Unit test the basic constructors: new, Put, and Push.
 *
 *  (2) Unit test the find operations on the results of the constructor tests.
 *
 *  (3) Unit test the basic selectors and destructors: Pull, Pop, RemoveNth
 *
 *  (4) Unit test the insert function.
 *
 *  (5) Stress test by constructing and destroying a large object.
 *
 *  (6) Unit test the Sort function and the destructor.
 *
 *  (7) Unit test the GetNth, Enum, and Print functions in the Dump service
 *      function, which is called from each of the preceding unit tests.
 *
 *  (8) Rerun the preceding unit-test sequence twice in succession.
 *
 */

#ifndef strlisttestIncluded
#define strlisttestIncluded

#include "strlist.h"


class StrListTest {

  friend class StrList;

  public:

    void DoIt();
    /*
     * Execute the test plan described above.
     */

    void DoItOnce();
    /*
     * Execute steps (1) through (5) of the test plan described above.
     */

    void TestBasicConstructors(StrList* sl);
    /*
     * Build a list with Put and Push.
     */

    void TestFinds(StrList* sl);
    /*
     * Test Finds before lists get clobbered.
     */

    void TestBasicSelectorsAndDestructor(StrList* sl);
    /*
     * Test Pull, Pop, and Remove.  GetNth test is in Dump.
     */

    void TestInserts(StrList* sl);
    /*
     * Test some inserts.
     */

    void StressTest();
    /*
     * Do some large-object stress testing.
     */

    void DoSort();
    /*
     * Test sorting.
     */

    void Dump(StrList* sl, char* prefix = "");
    /*
     * Dump results of a unit test.  Do so by dumping out the contents of a
     * list two differenct ways, to test both Enum and GetNth fully.  Note that
     * if a GetNth or RemoveNth was callled recenlty, we have to reset the enum
     * state in order to enum from the front.  The DumpNoEnumReset function
     * below does not do this resetting.  NOTE: DumpNoEnumReset still needs to
     * be written.
     */

    char* Strify(int i);
    /*
     * A little utility trinket to convert an in to a char*.
     */
};

#endif
```