```
/*
 *  Simple example of a CUnit unit test.
 *
 *  This program (crudely) demonstrates a very simple "black box"
 *  test of the standard library functions fprintf() and fread().
 *  It uses suite initialization and cleanup functions to open
 *  and close a common temporary file used by the test functions.
 *  The test functions then write to and read from the temporary
 *  file in the course of testing the library functions.
 *
 *  The 2 test functions are added to a single CUnit suite, and
 *  then run using the CUnit Basic interface.  The output of the
 *  program (on CUnit version 2.0-2) is:
 *
 *           CUnit : A Unit testing framework for C.
 *           http://cunit.sourceforge.net/
 *
 *       Suite: Suite_1
 *         Test: test of fprintf() ... passed
 *         Test: test of fread() ... passed
 *
 *       --Run Summary: Type      Total     Ran  Passed  Failed
 *                      suites        1       1     n/a       0
 *                      tests         2       2       2       0
 *                      asserts       5       5       5       0
 */

#include <stdio.h>
#include <string.h>
#include "CUnit/Basic.h"

/* Pointer to the file used by the tests. */
static FILE* temp_file = NULL;

/* The suite initialization function.
 * Opens the temporary file used by the tests.
 * Returns zero on success, non-zero otherwise.
 */
int init_suite1(void)
{
   if (NULL == (temp_file = fopen("temp.txt", "w+"))) {
      return -1;
   }
   else {
      return 0;
   }
}

/* The suite cleanup function.
 * Closes the temporary file used by the tests.
 * Returns zero on success, non-zero otherwise.
 */
int clean_suite1(void)
{
   if (0 != fclose(temp_file)) {
      return -1;
   }
   else {
      temp_file = NULL;
      return 0;
   }
}

/* Simple test of fprintf().
 * Writes test data to the temporary file and checks
 * whether the expected number of bytes were written.
 */
void testFPRINTF(void)
{
   int i1 = 10;

   if (NULL != temp_file) {
      CU_ASSERT(0 == fprintf(temp_file, ""));
      CU_ASSERT(2 == fprintf(temp_file, "Q\n"));
      CU_ASSERT(7 == fprintf(temp_file, "i1 = %d", i1));
   }
}

/* Simple test of fread().
 * Reads the data previously written by testFPRINTF()
 * and checks whether the expected characters are present.
 * Must be run after testFPRINTF().
 */
void testFREAD(void)
{
   unsigned char buffer[20];

   if (NULL != temp_file) {
      rewind(temp_file);
      CU_ASSERT(9 == fread(buffer, sizeof(unsigned char), 20, temp_file));
      CU_ASSERT(0 == strncmp(buffer, "Q\ni1 = 10", 9));
   }
}

/* The main() function for setting up and running the tests.
 * Returns a CUE_SUCCESS on successful running, another
 * CUnit error code on failure.
 */
int main()
{
   CU_pSuite pSuite = NULL;

   /* initialize the CUnit test registry */
   if (CUE_SUCCESS != CU_initialize_registry())
      return CU_get_error();

   /* add a suite to the registry */
   pSuite = CU_add_suite("Suite_1", init_suite1, clean_suite1);
   if (NULL == pSuite) {
      CU_cleanup_registry();
      return CU_get_error();
   }

   /* add the tests to the suite */
   /* NOTE - ORDER IS IMPORTANT - MUST TEST fread() AFTER fprintf() */
   if ((NULL == CU_add_test(pSuite, "test of fprintf()", testFPRINTF)) ||
       (NULL == CU_add_test(pSuite, "test of fread()", testFREAD)))
   {
      CU_cleanup_registry();
      return CU_get_error();
   }

   /* Run all tests using the CUnit Basic interface */
   CU_basic_set_mode(CU_BRM_VERBOSE);
   CU_basic_run_tests();
   CU_cleanup_registry();
   return CU_get_error();
}
```