**Submission of Your Work**

You need to prepare and submit ONE SINGLE MS Word document to Canvas (in your lab section) as LastName\_FirstName\_Labxy.doc. It must contain:

* Your NAME only on page 1
* For ***each*** question:
  + Specify the question number.
  + Source code. Copy/Paste your final source code. You must include standard “comment header” even if code is provided. *Do Not* paste a snippet of your source code, it must be copy/pasted.
  + Initial test plan. After reading the question requirements, but **before** beginning any coding, create the test case table, below, completed through column Expected Output. Include in your report.
  + Final test plan. Write your program then complete the **test table** with actual output results and include in your report *AFTER* your source code.
  + Output results. Paste in a snippet of output showing results for **every listed test case in your final test plan**, labeled with test case #

Test Table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Expected Output | Actual Output | Test Pass / Fail |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |

* Add / delete rows from Test Table as necessary
* Modify column widths as necessary
* Test both valid and invalid input
* Test for every output expected
* If failure is an expected output and it happens then that test Passes
* Any test that fails means the program must be fixed so that it passes the test
  + Failing tests need a new test row, ie 1a, 1b, etc, showing corrections from original

**Program 1.**

Write a function to calculate the square root of a float number with the following interface:

double squareRoot( double x)

{

// assert that x is not negative

...

}

The function should return an approximation to using series approximation.

Let x0 = x/2. Then x­n+1 = (xn + x/xn)/2. Keep computing terms until the difference between xn and xn+1 is less than 0.0001. In addition, if the input variable x is negative, your function should stop the execution via the assert( ) function, which is a built-in C++ function. You need to add a function call to assert( ) at the beginning of the above function to guarantee the precondition of this function is correct.

In your main function, design it to request the user to enter as many numbers as they want (continue to enter values? “y” or “n”). Test cases must include values 3,0, and -3, as well as others you think are appropriate.

The specifics of the assert message shown depends on the specific implementation in the compiler, but it should include: the <*bool exp*> whose assertion failed, the name of the source file, and the line number where it happened. A usual expression format is:  
  
Assertion failed: *<bool exp>* file *filename*, line *line number*

#include <cassert>

void print\_number(int myInt) {

assert (myInt > 5);

The program will abort if myInt has a value of 5 or less because myInt > 5 evaluates to false.

**Program 2.**

Write a main( ) program that requests the user to enter a file name, then open the file with that name.

1. If the file does not exist, display an error message.
2. If the file exists but contains NO data, display an error message and use an assertion to stop the program.
3. If the file exists and contains data then your program calls a function that has the following prototype:

void readIntFile(ifstream &x, int intArray [ ], const int size, int &length);

This function reads all the integers in the file and stores them into array intArray and returns the total number of integers stored in that file in variable *length*. You must design a while loop with a test for end-of-file (eof) condition.

The variable *size* is the declared size of the array. The size of the array for this program is 20 (declare as a global constant). Remember, you are required to perform a check to ensure you do not overflow the array.

Main() will conclude by calling function printFileValues to print the values that are stored in the array. The prototype for the function is

void printFileValues(const int intArray [ ], const int &length);

You must run tests to verify the correctness of your program: empty input file, too many values for array, invalid data, as well as any others you think are appropriate.

NOTE: These checks will be REQUIRED on all future programs that use input files

**Program 3.**

Write a recursive function that *returns* a reversed string. Be sure to use all base cases as test cases.

**Program 4.**

No test table necessary. List the three legs of the software development stool.