**Question 1 part 1:** Unit tests have been implemented using CUnit. The updated code is included in this .zip. It runs the unit tests I had on the old largest() and then runs unit tests on the fixedLargest() to show it fixed.

**Question 1 part 2:** I got a check for this, so I left it out.

**Question 2:** The mock object has been fixed and called mock\_max.c and is included in the .zip with the header I had

**Question 3:** I got checks for all of these, so I left them out.

**Question 4:** An example of a non-persistant XSS attack:

1.url is made that specifies a script as part of the search term

2.distribute link somehow(probably spam mail)

3.user clicks link, and the script runs

5.the attacker sets the user’s cookie in their browser as their own

6.now the attacker is doing stuff as the user

How to prevent: Don’t click on spam mail. I mean seriously? Also, escape strings so you don’t execute it. Or maybe validate and sanitize the strings, maybe clearing cookies or linking cookies to IP addresses. Or just disabling scripts.

**Question 5:** An example would be the floating point version of Pi, something like 3.14159. Compare that to math.pi from the system library. 3.14159 is not the same as 3.1415926535 because the second number is technically larger.

How can this mistake be avoided? Compare floats with a tolerance. So, compare them based on the shortest number, not all numbers(effectively ignoring the last 5 numbers of math.pi in this example). While this isn’t accurate per se, it is valid with the tolerance