A) the first obstacle I encountered was in my shiftRight function, I was walking off the array and to fix that I cycled through the iteration backwards and added the following if statement **if** (n-1-amount-i== 0)**break**;. I also had issues with a test case when amount = 0 and so I incorporated that into the first if statement. The rotateLeft function was by far the hardest to figure out, trying to understand and implement the Modulo Operation allowed me to reduce the number of rotations when amount was a multiple of n. I also created a tempArray where the variable length was undefined and this was caught in the g++ compiler, and I didn’t have time to fix it :(

B)The test cases I used where mainly the ones that were provided as well as edge cases that tried to break the code;

assert(locateMinimum(folks, 8) == 3); //test case provided

assert(locateMinimum(folks, 5) == 3); //test edge case

assert(locateMinimum(a, 3) == 0); //test edge case

assert(countCommonElements(folks, 8, data, 5) == 1); //test provided

assert(countCommonElements(folks, 3, data, 3) == 0); //test edge case

assert(countVowels(data, 5) == 8); //test case provided

assert(countVowels(folks, 8) == 21); //test case to see if function works

assert(hasThreeOfX(data, 5, "mama") == **true**); // test case provided

assert(hasThreeOfX(folks, 8, "jon") == **false**);

assert(hasThreeOfX(test, 5, "ju") == **true**);

assert(majorityElement(data, 5) == "mama");

assert(majorityElement(folks, 8) == "");

assert(majorityElement(test, 5) == "ju");

assert(shiftRight(data, 5, 2,"foo") == **true**);

assert(shiftRight(data, -5, 10, "foobar") == **false**);

assert(shiftRight(data, 5, 10, "bar") == **true**);

assert(rotateLeft(folks, 8, 1) == **true**);

assert(rotateLeft(folks, 8, 2) == **true**);

assert(rotateLeft(test, 5, 2) == **true**);

assert(rotateLeft(folks, 2, 12) == **false**);

assert(shiftRight(codeboard , 1, 0, "foo" ) == **false**);

assert(replaceAll(folks, 8, 'A', 'Z') == 0);