Name: Morgan Lim

UID: 105 168 668

a: One notable obstacle I overcame was interpreting the specifications the way that the professor wanted it be interpreted. It took a lot of rereading of the specifications, browsing the discussion forum, and asking my peers to get a clear idea of what was expected of the project. Another obstacle I encountered was making sure that loops did not “walk off” the boundary of the arrays. Since it has been a while since I worked with loops, there were a few functions that initially tried to access nonexistent indices of the array.

b: Test cases:

string folks[8] = {"samwell", "jon", "margaery", "daenerys", "tyrion", "sansa", "howard123", "jon"};

string people[5] = { "samwell", "jon", "margaery", "daenerys", "tyrion" };

string names[5] = { "a", "b", "c", "d", "e" };

string namesB[5] = {"e", "d", "c", "b", "a"};

string list[7] = { "a100", "bottle", "face34", "geek56", "geek56","sun3day", "z3ro\_s3bum"};

string a[6] = { "alpha", "beta", "gamma", "gamma", "beta", "delta" };

//valid test cases

assert(hasDuplicates(a, 3) == false);

assert(hasDuplicates(a, 6) == true);

assert(hasDuplicates(folks, 5) == false);

assert(hasDuplicates(folks, 8) == true);

assert(hasDuplicates(list, 7) == true);

assert(hasDuplicates(list, 4) == false);

//invalid test cases; size <= 0

assert(hasDuplicates(folks, -1) == false);

assert(hasDuplicates(list, -5) == false);

assert(hasDuplicates(list, 0) == false);

//valid test cases

assert(countAllDigits(folks, 8) == 3);

assert(countAllDigits(list, 7) == 12);

assert(countAllDigits(a, 6) == 0);

assert(countAllDigits(list, 4) == 7);

//invalid test cases; size <= 0

assert(countAllDigits(folks, -108) == -1);

assert(countAllDigits(list, 0) == -1);

//valid test cases

assert(isInDecreasingOrder(people, 5) == false);

assert(isInDecreasingOrder(names, 5) == true);

assert(isInDecreasingOrder(people, 0) == true);

assert(isInDecreasingOrder(names, 1) == true);

assert(isInDecreasingOrder(namesB, 5) == false);

//invalid test cases; size < 0

assert(isInDecreasingOrder(names, -1) == false);

assert(isInDecreasingOrder(namesB, -5) == false);

//valid test cases

assert(replaceAllCharacters(people, 5, 'z', 'a') == 0);

assert(replaceAllCharacters(people, 5, 'a', '\_') == 4);

assert(replaceAllCharacters(list, 4, 'e', '!') == 4);

//invalid test cases; size <= 0

assert(replaceAllCharacters(people, -5, 'z', 'a') == -1);

assert(replaceAllCharacters(list, 0, 'e', '!') == -1);

//valid test cases

assert(shiftRight(people, 5, 3, "foo") == 3);

assert(shiftRight(names, 5, 4, "\_") == 4);

//invalid test cases; amount or size < 0

assert(shiftRight(people, -5, 3, "foo") == -1);

assert(shiftRight(names, 5, -4, "\_") == -1);