Eric Yang

304263623

1. One of the most difficult problems for me throughout this project was remembering that the number of interesting elements in the array that gets entered into the function isn’t the position of the last element of the array. I forgot to subtract 1 from each of the “for loops” and ended up with correct answers only half the time. I changed the loop condition to “k != n” in order to solve this problem

Another challenge was starting with the rotateLeft function. I first tried to use multiple strings to rearrange the array but could not figure out how to move the string at “pos” to the end. I decided to just try using a second array and assigning the first strings from the original array. Then I assigned the second half and lastly attached the string at “pos.” I also managed to mess this up because I completely forgot to copy the second array back to the original. I figured this out after I finished every function and decided to check each array output. I went back to change this mistake for the functions that used a second array.

1. string cast[6] = { "sansa", "robb", "daenerys", "tyrion", "jon", "arya" };

string cast2[6] = { "sansa", "robb", "daenerys", "tom", "jon", "arya" };

string cast3[2] = { "tyrion", "jon",};

Append to all (appendToAll(cast, 6, “!!!”))

Append to only first half (appendToAll(cast, 3, “!!!”))

Append bad argument (appendToAll(cast, -1, “!!!”))

Look up existing string (lookup(cast, 6, “tyrion”))

Look up non-existing string (lookup(cast, 6, “tom”))

Look up string not in range (lookup(cast, 2, “tyrion”))

Look up bad argument (lookup(cast, -1, “tyrion”))

Position of max (positionOfMax(cast, 6))

Position of max bad argument (positionOfMax(cast, -1))

Rotate left middle string (rotateLeft(cast, 6, 2))

Rotate left last string (rotateLeft(cast, 6, 5))

Rotate left bad argument (rotateLeft(cast, -1, 2))

Rotate right middle string (rotateRight(cast, 6, 2))

Rotate right first string (rotateRight(cast, 6, 0))

Rotate right bad argument (rotateRight(cast, -1, 2))

Flip all string (flip(cast, 6))

Flip first 4 strings (flip(cast, 4))

Flip bad argument (flip(cast, -1))

Differ all strings (differ(cast, 6, cast2, 6))

Differ n2 smaller (differ(cast, 6, cast2, 3))

Differ n1 smaller (differ(cast, 3, cast2, 6))

Differ bad argument (differ(cast, -1, cast2, -1))

Subsequence exists (subsequence(cast, 6, cast3, 2))

Subsequence does not exist (subsequence(cast2, 6, cast3, 2))

Subsequence bad argument (subsequence(cast, -1, cast3, 2))

Look up any first exists (lookupAny(cast, 6, cast3, 2))

Look up any does not exist (lookupAny(cast2, 4, cast3, 2))

Look up any bad argument (lookupAny(cast2, -1, cast3, 2))

Split middle (split(cast, 6, “jorah”))

Split end ((split(cast, 6, “zzz”))

Split beginning (split(cast, 6, “aaa”))

Split bad argument (split(cast, -1, “jorah”))