1. The first obstacle is in function 3. When I have the maximum string, how can I find its position. I end up by calling the lookup function inside the positionOfMax function. The second is in subsequence function: how to pinpoint the starting point of comparison. I design a loop keep comparing each element in a1 with a2[0], the first element in a2. If they are identical, start another loop which compares the rest of a2. The final big problem is the divide function. I first come up with a solution is to create another array. Using a loop a an if statement to compare the element in a and the divider, storing it in the new array if it is less that the divider. Then using another loop to store the rest of the elements that are not in the new array. However, it is said in the project specification that it can be done without creating a new array. I came up with another method: bubble sorting. First sort the element inside the array from small to larger, then find the threshold element that is larger than the divider, returning its position.
2. appendToAll():

string people[5] = { "donald", "joe", "mike", "lindsey", "kamala" }; (people, 5, “!!!”)

lookup():

string people[5] = { "donald", "joe", "mike", "lindsey", "kamala" }; (people, 5, “joe”)

positionOfMax():

string people[5] = { "donald", "joe", "mike", "lindsey", "kamala" }; (people, 5)

rotateLeft():

string people[5] = { "donald", "joe", "mike", "lindsey", "kamala" }; (people, 5, 1)

countRuns():

string d[9] = {"susan", "donald", "mike", "mike", "joe", "joe", "joe", "mike", "mike"}; (d, 9)

flip():

string people[5] = { "donald", "joe", "mike", "lindsey", "kamala" }; (people, 3)

differ()

string folks[6] = { "donald", "mike", "", "susan", "sara", "jamie" };

string group[5] = { "donald", "mike", "jamie", "", "susan" };

(folks, 6, group, 5) (folks, 2, group, 1)

Subsequence()

string names[10] = { "kamala", "mark", "sara", "martha", "donald", "lindsey" };

string names1[10] = { "mark", "sara", "martha" };

(names, 6, names1, 3) (names, 5, names1, 2)

lookupAny()

string names[10] = { "kamala", "mark", "sara", "martha", "donald", "lindsey" };

string set1[10] = { "jamie", "donald", "martha", "mark" };

(names, 6, set1, 4) (names, 6, set2, 2)

Divide()

string candidate[6] = { "jamie", "lindsey", "mark", "susan", "joe", "donald" };

(candidate, 6, "kamala")

string candidate2[4] = { "mark", "sara", "lindsey", "mike" };

(candidate2, 4, "mike")