Preethika Kiruveedula

204912895

Project 4 Report

1. Some notable obstacles I overcame while completing this project are: figuring out the logic behind each function and making sure to consider each exception to the tasks. It was extremely simple to overlook the small aspects of each condition. For example, I had trouble accounting for simple things such as what to do when a variable would become negative. A main issue for me was figuring out how to format each function in an efficient way that would account for all bad inputs. The function I struggled with the most was rotateLeft, but after visualizing it using paper notecards I could write the function.
2. Test cases:

string a[10] = { "brian", "steven", "timothy", "sean" ,"sean", "linda", "sarah","sarah", "megan", "jade" };

assert(countRuns(a, 10) == 8);

assert(appendToAll(a, 4, "?") == 4 && a[0] == "brian?" && a[3] == "sean?");

assert(appendToAll(a, -3, "?") == -1);

assert(flip(a, 10) == 10 && a[0] == "jade" & a[9] == "brian?");

string b[7] = { "ally", "blake", "leyna", "sarah", "zahra", "jacob", "shannon" };

string c[4] = { "blake", "leyna", "joseph", "shannon" };

string s[3] = { "sarah", "zahra", "jacob" };

assert(lookup(b, 7, "ally") == 0);

assert(lookup(b, 7, "jacob") == 5);

assert(lookup(c, 3, "will") == -1);

assert(lookup(c, -2, "leyna") == -1);

assert(lookupAny(b, 7, c, 4) == 1);

assert(lookupAny(a, 10, c, 4) == -1);

assert(lookupAny(a, -1, c, 4) == -1);

assert(subsequence(b, 7, s, 3) == 3);

assert(subsequence(b, 7, c, 4) == -1);

assert(subsequence(b, -2, c, 4) == -1);

string z[4] = { "nithika", "harshita", "likhita", "preethika" };

string r[2] = { "nithika", "harshita" };

string t[3] = { "dawn", "summer", "autumn" };

assert(differ(z, 4, r, 2) == 2);

assert(differ(z, 0, r, 2) == 0);

assert(differ(z, -1, r, 2) == -1);

assert(positionOfMax(z, 4) == 3);

assert(positionOfMax(r, -2) == -1);

assert(countRuns(z, 4) == 4);

assert(rotateLeft(z, 4, 1) == 1 && z[1] == "likhita" && z[3] == "harshita");

assert(rotateLeft(s, 3, 1) == 1 && s[0] == "sarah" && s[2] == "zahra");

assert(rotateLeft(s, 3, -2) == -1);

assert(rotateLeft(s, -2, 1) == -1);

assert(split(t, 3, "jolly") == 1);

assert(split(r, 2, "apple") == 0);

assert(split(r, -1, "open") == -1);

assert(split(r, 2, "zebra") == 2);

assert(split(r, 2, "king") == 1);