Jiachen Liu

CS 31 Report

a.

some of the obstacles includes:

These function I wrote are really interconnected, for example, moveToEnd and moveToBeginning are used in various functions.

One of the functions, moveToEnd is excutable but does not do entirely what I wanted and I went along with it. Then I implemented this function in a lot of my other functions, and the result did not go well. So next time I need to make sure one of the function execute completely well and then write other functions based on the original function.

The other obstacles include that the assert statement in my code did not function well. It runs really good when you use a for loop to show all the string element. (the debugger jumps directly to a breakpoint in other files, it seems to be written in assembly or such. I’m not sure since I haven’t really learned it yet). I spend a lot of time looking at it but it turns out that the program missed a “=” in an if statement.

b.

For the test cases, I mainly used “assert” in this tests:

I mainly used the string example given in the CS project description.

Besides the tests I’m going to list below, I test the n is negative for every function and it works fine.

My main function basically account for most test cases I can think of, so I’m going to just post it here:

string h[7] = { "peter", "lois", "meg", "chris", "", "meg", "stewie" };

assert(tally(h, 7, "meg") == 2); // test whether it functions normally.

assert(tally(h, 7, "") == 1); // to see if it recognize empty string.

assert(tally(h, 7, "quagmire") == 0); // test whether it functions well if the string is not in the array.

assert(tally(h, -10, "quagmire") == -1); // test whether it outputs -1 when the n is negnative.

assert(tally(h, 4, "")==0); // it's in the string but it not within the n range.

assert(tally(h, 0, "meg") == 0); //test what does it do when n is zero.

assert(findLast(h, 7, "meg") == 5); // a successful case for findLast function.

assert(findLast(h, 2, "meg") == -1); // what if the function can't find the designated string?

assert(findLast(h, -2, "meg") == -1); //to see the function returns -1 if n is negative.

int bg;

int en;

assert(findFirstSequence(h, 7, "chris", bg, en) && bg == 3 && en == 3); // test whether it works normally.

string test[10]={"peter" , "lois", "meg", "chris", "", "meg", "stewie", "lois" , "lois"};

assert(findFirstSequence(test, 12, "lois", bg, en) && bg == 1 && en == 1); //test whether it only takes the earileset set.

assert(findFirstSequence(test, -12, "lois", bg, en)==false); // test what happens when

assert(findFirstSequence(test, -12, "nobody", bg, en)==false); // test what if the name is not in the string array.

string g[4] = { "peter", "lois", "chris", "stewie" };

assert(positionOfMin(g, 4) == 2); // check the position of a string.

assert(positionOfMin(g, -1) == -1); // what if n is negative.

assert(positionOfMin(g, 0) == -1); //what if the arrary have no elements.

assert(disagree(h, 4, g, 4) == 2); //test if it works normally.

assert(disagree(h, 4, g, 1) == 1); //test what if one of them is really short.

string family[5] = { "peter", "lois", "meg", "chris", "stewie" };

string people[6] = { "peter", "lois", "quagmire", "cleveland" };

assert(disagree(family, 5, people, 4)==2); //test if it works normally.

assert(disagree(family, 2, people, 1)==1);//test what if one of them is really short.

assert(subsequence(h, 7, g, 4)); //this tests good

assert(moveToEnd(g, 4, 1) == 1 && g[1] == "chris" && g[3] == "lois"); // check if it works normally.

string f[4] = { "chris", "stewie", "meg", "lois" };

assert(moveToBeginning(f, 4, 2) == 2 && f[0] == "meg" && f[2] == "stewie"); // this works, as one can see in my report, I tested a lot of this cases

string e[5] = { "chris", "chris", "chris", "meg", "meg" };

assert(removeDups(e, 5) == 2 && e[1] == "meg");// test if this function works.

string x[4] = { "brian", "lois", "lois", "quagmire" };

string y[4] = { "chris", "lois", "meg", "stewie" };

string z[10];

assert(mingle(x, 4, y, 4, z, 10) == 8 && z[5] == "meg"); //mingle is kind of weird in my program.

assert(divide(h, 7, "meg") == 3); // test if this function works.

assert(divide(x, 4, "lois")==1); // what if there are multiple lois.

cout << "All tests succeeded" << endl << endl;

besides mingle might not work so well, when I do the following assert, it doesn’t work so well either.

assert(removeDups(x, 5) == 3 && x[2] == "quagmire");

Some thing that is not complete yet:

Still the mingle one works now, but it work funny now and then, and I’m pretty sure there are cases that it does not fit (my xcode when excute the mingle function acts really weird. I think it might be that I access the elements that are out of the range).