Report

1. A brief description of notable obstacles

When executing the function rotateleft, my program keeps crashing at the line

a[pos+i] = a[pos+i+1];

Then I realize, for the for-loop, I set the condition as i + pos < n. This line would result in an undefined behavior, because there is no element at position n.

After change the condition to i + pos < n -1, the function would achieve the intended goals.

For the function Flip, there are two cases to consider due to the fact that n can be either an even or an odd number. In the for-loop, condition k < n/2 effectively takes care of both cases. If n = 7, elements at following positions will switch order (0,6) (1,5) (2,4). If n = 6, elements at following positions will switch order (0, 5) (1,4) (2,3).

1. Test data

AppendToAll

Normal case

string people[5] = { "hillary", "jeb", "rand", "ben", "john" };

int j = appendToAll(people, 5, "!!!");

0 case ( function should not make any change)

string people[5] = { "hillary", "jeb", "rand", "ben", "john" };

int j = appendToAll(people, 0, "!!!");

Lookup

Array contains target (return position)

string people[5] = { "hillary", "jeb", "rand", "ben", "john" };

int j = lookup(people, 5, "jeb");

Array does not contain target (return -1)

string people[5] = { "hillary", "jeb", "rand", "ben", "john" };

int j = lookup(people, 5, "jeB");

PositionOfMax

string cand[6] = { "bernie", "hillary", "donald", "marco", "carly", "ben" };

int k = positionOfMax(cand, 6); // returns 3, since marco is latest

RotateLeft

string running[5] = { "carly", "mike", "ted", "bernie", "jeb" };

int m = rotateLeft(running, 5, 1);

CountRuns

Normal case

string d[9] = {

"ben", "chris", "marco", "marco", "donald", "donald", "donald", "marco", "marco"

};

int p = countRuns(d, 9);

0 case (function should return 0)

string d[9] = {

"ben", "chris", "marco", "marco", "donald", "donald", "donald", "marco", "marco"

};

int p = countRuns(d, 0);

Flip

Odd number case

string running[6] = { "chris", "marco", " ", "ben", "donald", "john" };

int q = flip(running, 3);

Even number case

string running[6] = { "chris", "marco", " ", "ben", "donald", "john" };

int q = flip(running, 4);

Differ

string folks[6] = { "chris", "marco", "", "ben", "donald", "john" };

string group[5] = { "chris", "marco" , "","john", "carly" };

Normal case

int r = differ(folks, 6, group, 5); // returns 2

equal-up-to-the-point-where-one-or-both-runs-out case

int s = differ(folks, 2, group, 1); // returns 1

Subsequence

string names[10] = { "ted", "hillary", "rand", "bernie", "mike", "jeb" };

All-items-match case

string names1[10] = { "hillary", "rand", "bernie" };

int t = subsequence(names, 6, names1, 3); // returns 1

Some-items-match case

string names2[10] = { "ted", "bernie" };

int u = subsequence(names, 5, names2, 2); // returns -1

No-item-match case

string names2[10] = { "t", "ber" };

int u = subsequence(names, 5, names2, 2); // returns -1

LookupAny

string names[10] = { "ted", "hillary", "rand", "bernie", "mike", "jeb" };

string set1[10] = { "carly", "mike", "bernie", "hillary" };

int v = lookupAny(names, 6, set1, 4); // returns 1 (a1 has "hillary" there)

string set2[10] = { "ben", "donald" };

int w = lookupAny(names, 6, set2, 2); // returns -1 (a1 has none)

Split

Splitter is not an item in the array

string cand[6] = { "bernie", "hillary", "donald", "marco", "carly", "ben" };

int x = split(cand, 6, "chris"); // returns 3

Splitter is an item in the array

string cand2[4] = { "donald", "hillary", "jeb", "ben" };

int y = split(cand2, 4, "donald"); // returns 1