Stewart Dulaney

Report for Project 4: Array of Sunshine

a. While writing the code for Project 4, I encountered many obstacles but the two most notable obstacles I overcame involved writing the code for the "int flip(string a[], int n);" and "int differ(const string a1[], int n1, const string a2[], int n2);" functions.

The first noteworthy issue I overcame while writing the "flip" function was writing a for statement with correct intialization and stay-in-loop condition. Because the function reverses the order of the first n elements of the array, not up to the nth position in the array, I had to discover by trial and error that I needed a to count backwards and that the correct for statement is "for(int k = (n-1); k >= 0; k--)". The next problem I had was copying the elements in the array to another array in reverse order. I tried using expressions inside of assignment statements, but I eventually discovered that I needed another variable to use for the position in my second array. So I intialized a variable "int x = 0", and increased it by one after the for loop assigned a variable to that position. Because the for loop was counting backwards, this effectively copied the original array in reverse order to the other array. The only step remaining was to copy the new order of elements back to the original array.

The second notable problem I overcame involved writing the code for the "differ" function. My issue was accounting for the case when the arrays are equal up to one point where one of both run out. At first I thought I needed to if statements to divide the code, "if(n1 > n2)" and "if(n2 > n1)", so that the program would only test until the smaller array ran out. However, I could see that the code was getting to be too repetitive and needed to be consolidated. I used a for loop with a nested if-else statement, which returned the position when there was an inequality, and returned a different variable n if the elements were equal. I kept getting an error message that the function wasn't producing the right output. By displaying the output of the function on the screen, I discovered that the else statement was the problem. I needed to place the "return n;" statement outside of the for loop, so that is was executed only when all the elements in the arrays were the same, not just one. The function then produced the correct output.

b. To test external functions:

```
assert(appendToAll(folks, 6, "!!!") == 6);
assert(folks[0] == "george!!!" && folks[5] == "buster!!!");
assert(lookup(names, 5, "mary") == 2);
assert(lookup(names, 5, "ellen") == 3);
assert(positionOfMax(letters, 6) == 5);
assert(positionOfMax(persons, 6) == 3);
assert(rotateLeft(humans, 5, 0) == 0);
assert(humans[1] == "bart" && humans[4] == "homer");
assert(rotateRight(simpsons, 5, 2) == 2);
assert(simpsons[0] == "bart" && simpsons[1] == "homer" && simpsons[2] == "marge");
assert(flip(beings, 4) == 4);
assert(beings[0] == "haley" && beings[1] == "" && beings[2] == "phil" && beings[3] == "claire");
assert(differ(group1, 6, group2, 4) == 2);
assert(differ(group1, 2, group2, 1) == 1);
```

Reasons for each test:

1. Verifies output of appendToAll
2. Verifies changes to string made by appendToAll
3. Verifies output of lookup
4. Verifies output of lookup
5. Verifies output of positionOfMax
6. Verifies output of positionOfMax
7. Verifies output of rotateLeft
8. Verifies changes to string made by rotateLeft
9. Verifies output of rotateRight
10. Verifies changes to string made by rotateRight
11. Verifies output of flip
12. Verifies changes to string made by flip
13. Verifies output of differ
14. Verifies output of differ

Note: I was not able to implement the functions "subsequence", "lookupany", and "split", so my program will not handle test cases for those functions correctly.