Edward Chu

504772148

CS 31 Report: Assignment 3:

Notable obstacles that I overcame:

1. It took me a while to figure out how to make sure that I got all of the digits from the poll data inputted. To do this, I used a while loop: while the current digit is a number, the digit is added to a string.
2. It also took me a while to figure out how to turn the string mentioned above into an integer. To do this, I found a function called stoi() that turns a string into an integer.

Description of program:

bool hasCorrectSyntax(string pollData):

1. Check if string is empty string, return true if true
2. Check is string is a string with length 1, return false if true
3. Turn any letters in the input, poll data, into uppercase letters
4. Repeatedly:
   1. Check whether the state code is valid, using the function provided by Dr. Smallberg, return false if invalid state code
   2. Check whether end of string is reached after the state code, return false if it’s reached (No poll data after state code)
   3. Scroll past all of the digits, check if party is a letter, return false if party is not a letter
   4. Scroll past the party name to reach next section of poll data
5. Return true when end of the whole poll data is reached

int countVotes(string pollData, char party, int& voteCount):

1. Turn any letters in the input, poll data, into uppercase letters
2. Check if poll data has correct syntax, return 1 if false
3. Check if any states have 0 predicted electoral votes, return 2 if true
4. Check if the party name is a letter, return 3 if false
5. Turn the party name into an uppercase letter and initialize voteCount to 0
6. Repeatedly:
   1. Scroll past the state code
   2. Store all of the numbers after the state code into a temporary string
   3. Add the integer value of the string to the vote count if the party name after the string is the party required by the input
7. End of function

Test Cases:

hasCorrectSyntax:

assert(hasCorrectSyntax("TX38RCA55D")); //Normal correct

assert(hasCorrectSyntax("VA3RCA55DAL55LUT677777777777D")); //Normal correct

assert(hasCorrectSyntax("TX38RCA00000000D")); //State with 0 votes, should be true

assert(hasCorrectSyntax("")); //Empty string, should be true

assert(!hasCorrectSyntax("MX38RCA55D")); //Wrong state code at start, should be false

assert(!hasCorrectSyntax("TX38RCA55")); //Missing party name at end

assert(!hasCorrectSyntax("AZRCA55D")); //Missing vote number in middle

assert(!hasCorrectSyntax("TX38RA55D")); //Missing party name in middle

assert(!hasCorrectSyntax("TX38%CA55D")); //Symbols for party names, should be false

assert(!hasCorrectSyntax("TX38RCD55D")); //Wrong state code in middle

assert(!hasCorrectSyntax("tx38rCA55DMs6rnY29dUT06LL")); //Extra letter on the end

assert(!hasCorrectSyntax("INVALIDSTRING")); //String with no numbers

assert(!hasCorrectSyntax("AZD")); //String with no numbers

assert(!hasCorrectSyntax("AZ")); //String with no numbers

assert(!hasCorrectSyntax("A")); //One letter string

assert(!hasCorrectSyntax("AZ38DC27D")); //Missing party name in middle

assert(!hasCorrectSyntax("AZ38RDC27")); //Missing party name at end

assert(!hasCorrectSyntax("AZ38R DC27R")); //Space in string

countVotes:

Votes is initialized to -999 before each test

assert(countVotes("TX38RCA55DMs6rnY29dUT06L", 'R', votes) == 0 && votes == 44); //Normal

assert(countVotes("TX38RCA55DMs6rnY29dUT06L", 'L', votes) == 0 && votes == 6); //Normal

assert(countVotes("TX38RCA55DMs6rnY29dUT06L", 'd', votes) == 0 && votes == 84); //Lowercase party

assert(countVotes("", 'R', votes) == 0 && votes == 0); //Empty string

assert(countVotes("TX38RCA55D", '%', votes) == 3 && votes == -999); //Party name not letter

assert(countVotes("INVALIDSTRING", 'D', votes) == 1 && votes == -999); //Invalid pollData assert(countVotes("TX38%CA55D", 'D', votes) == 1 && votes == -999); //Invalid pollData with non-letter

assert(countVotes("TX38RCA55DMs6rnY00dUT09L", 'R', votes) == 2 && votes == -999); //State with 0 votes

assert(countVotes("TX000000000RCA55DMs6rnY10dUT09L", 'R', votes) == 2 && votes == -999); //State with 0 votes at start