Project 3 Report

I encountered a few challenges during this project, most notably ones relating to the difficulty of reading my own code. My program contained numerous nested loops and conditional statements and when I had to debug a particular section, I found it quite difficult to locate the sources of any errors. My first try at this code involved using arrays, but the Linux server flagged an error because it did not the array length to be an uninitialized variable. My second try used substrings, but the program inexplicably multiplied the units digit of any number by 2, something I found very challenging to solve.

This program takes in a poll data string and a party letter and calculates the number of electoral votes a candidate for that party receives in total.

The function isValidUppercaseStateCode() checks if the state code in a forecast is valid. There are 51 acceptable state codes, which are all listed in uppercase letters. The parameter is a string and the function returns a boolean value. The pseudocode for this method is:

bool isValidUppercaseStateCode(string stateCode) {

list acceptable state codes

if a match is found,

return true

else return false

}

The function isSyntacticallyCorrect() checks if the sequence of forecasts is a valid poll data string. A forecast consists of either 1 or 2 digits, followed by state code, followed by a party code. Any number of forecasts can be joined together to form a poll data string. The letters can be either uppercase or lowercase, but no colons, spaces, or special characters are allowed. This function calls on the isValidUppercaseStateCode() method. The parameter is a string and the function returns a boolean value. The pseudocode for this method is:

bool isSyntacticallyCorrect(string pollData) {

if string is empty,

return true

repeat:

if char isn’t a letter or number,

return false

if string doesn’t begin with a valid number or doesn’t end with a letter,

return false

if any numbers have over 2 digits,

return false

for each forecast:

if state code is invalid,

return false

for each forecast:

if there aren’t exactly 3 letters,

return false

conditions met, return true

}

The function tallyVotes() calculates the number of electoral votes a candidate from a party will receive. An integer value is returned and there are 3 parameters: a string, a character, and an integer. The third parameter is passed by reference and when its value is changed by this method, the original variable also gets modified. This function calls on the isSyntacticallyCorrect() method to check if the poll data string is valid. The pseudocode for this method is:

int tallyVotes(string pollData, char party, int& voteTally) {

if string has wrong syntax,

return 1

if party isn’t a letter,

return 2

if a forecast has 0 votes,

return 3

if string is empty,

return 0

find number of forecasts

create votes and party arrays

for each forecast:

add to party array

add to votes array

increment position in both arrays

repeat:

if element in party array matches with specific party,

add number of votes to total

return 0

}

I used multiple test cases to check what results a variety of inputs would produce.

1. “”: to see how it responds to an empty string
2. 00cad: to make sure it returns 3 with double zero
3. 0Cad: to make sure it returns 3 with single zero
4. 10cAD0txR: to make sure it returns 3 with a zero in a later forecast
5. 55CAd10txd, with party d: to make sure it adds the 2 numbers
6. 23nyg7njl, with party r: because none of the forecasts has the r party
7. 1csd2txr: with invalid state code
8. Cad23txr: with invalid poll data string
9. 02car34txd87ctl45mee2FLd: for a long poll data string with many forecasts
10. 123qad76txd: for an invalid poll data string and an invalid state code