Harsh Chobisa

Prof. David Smallberg

CS 31

October 28, 2018

Project 3 Report

The biggest struggle that I had was understanding what was to be done. When I first read the spec, it went completely over my head. In order to solve this problem, I read the spec line by line and diagrammed the problem. After I did this, I had a better idea of what was to be done in the project and was able to start.

The next problem I had was structuring by code for hasProperSyntax. In the end, I ended up creating a pseudocode model of what was to be checked by the program, and using the pseudocode, I had a better model of the program, thus allowing me to start and complete the writing of the program.

The program is composed of three functions. The first is isValidUppercaseStateCode. This function takes in a string and outputs true if it a valid state code string and false if it is not. This function was provided in the spec.

The next function is hasProperSyntax. hasProperSyntax takes in a string and returns true if the string is a valid poll data string and returns false otherwise. Pseudocode:

Append a comma to the end of the string

Make the entire string uppercase

Cycle through the string until reaching the end

Check if the first two letters are valid state code

While you do not hit a comma

Check if the next char is a digit

Scroll past the digit

Check if the next char is a letter

Scroll past the comma

Return true

The final function tallySeats takes in a poll data string, a party character, and seatTally. It sets seatTally to the projected number of seats the party indicated by the character will win as predicted by poll data string. Pseudocode:

If it is not a poll data string, return 1

If party is not a letter, return 2

Make the poll data string and party uppercase

Cycle through the string

If a character equals party

Check for digits behind party and add them to a seatTally

Return 0

Test cases:

These check that hasProperSyntax works properly. The tests start with a blank string, and increase in complexity, from a state code, to a state code and aprty results, to multiple party results, to a full poll data string.

assert(hasProperSyntax(""));

assert(hasProperSyntax("NY"));

assert(hasProperSyntax("NY1D"));

assert(hasProperSyntax("NY1D33R"));

assert(hasProperSyntax("NY9R17D1I,Vt,NJ3d5r4D,KS4R"));

These check for input errors by the user.

//check for input errors

assert((tallySeats(" ", 'D', seatTally) == 1));

assert((tallySeats("CAA99D", 'D', seatTally) == 1));

assert((tallySeats("CAA99DD", 'D', seatTally) == 1));

assert((tallySeats("CA99D", '2', seatTally) == 2));

These check that tallySeats works properly. The tests start with a blank string, and increase in complexity, from a state code, to a state code and aprty results, to multiple party results, to a full poll data string.

assert((tallySeats("", 'D', seatTally) == 0) && seatTally==0);

assert((tallySeats("NY", 'D', seatTally) == 0) && seatTally==0);

assert((tallySeats("NY1d", 'D', seatTally) == 0) && seatTally==1);

assert((tallySeats("CA99D", 'D', seatTally) == 0) && seatTally==99);

assert((tallySeats("NY1D33R", 'R', seatTally) == 0) && seatTally==33);

assert((tallySeats("NY12R1D33R", 'R', seatTally) == 0) && seatTally==45);

assert((tallySeats("NY9R17D1I,Vt,NJ3d5r4D,KS4R", 'D', seatTally) == 0) && seatTally==24);

assert((tallySeats("NY9R17D1I,Vt,NJ3d5r4D,KS4R,NY9R17D1I,NJ3d5r4D,KS4R", 'D', seatTally) == 0) && seatTally==48);

A test provided by professor Smallberg to ensure consistency of style:

assert(hasProperSyntax("MA9D,KS4R") && hasProperSyntax("KS4R,MA9D"));

assert(hasProperSyntax("MA9D,,KS4R") == hasProperSyntax("KS4R,,MA9D"));