Sadhana Vadrevu

CS 31

Professor Smallberg

Project 3 Report

**Obstacles:**

One of the obstacles I had to overcome in this project were figuring out how to organize the hasProperSyntax function. At first, I was overwhelmed by how many different cases could cause the function to return false. However, after reading the FAQs and seeing how to build up the function incrementally, organizing it became much easier. Once I was able to solve this issue, the next big problem I had was actually implementing the checks. I got stuck on two specific problems: how to check if the party code is only one letter and how to make sure the party code is only preceded by one or two digits. Once we learned about the isalpha and isdigit functions, checking the party code became much more straightforward. I realized that I needed to check that the character at the party code position needed to return true for isalpha and the character after the code needed to return true for isdigit. If either of these tests returned false, the pollData was not valid. Checking that the party code was preceded by at most two digits took longer for me to figure out, but after drawing out what I wanted the code to do, I realized I needed another variable to count the number of digits before the party code. If this variable was greater than 2 when the party code was reached, then the pollData was not valid. Drawing out what I wanted the code to do helped me figure out the solutions to all these issues.

**Program Description:**

Overview: My program has two major parts: one that checks whether a string is valid a valid pollData string and another that processes the pollData to count the number of seats a specified party is predicted to win.

*hasProperSyntax*: checks if a string is valid as poll data

*if pollData starts or ends with a comma,*

*pollData is not valid*

*while pollData is valid:*

*while characters in pollData are not commas:*

*add pollData characters to stateForecast*

*if stateForecast is empty,*

*continue with loop*

*else if stateForecast has one character,*

*pollData is not valid*

*else if stateForecast begins with a valid state code,*

*if stateForecast has no party results,*

*pollData remains valid*

*else if stateForecast has three characters,*

*pollData is not valid*

*break out of loop*

*AND else if state code is not followed by a digit or the last character of the state forecast is not a letter,*

*pollData is not valid*

*break out of loop*

*otherwise, if stateForecast has a valid state code and a valid overall format,*

*check each character of stateForecast:*

*if character is a digit,*

*add to count of number of digits before party code*

*if character is a letter,*

*if letter character is not followed by a digit*

*pollData is not valid*

*break out of loop*

*otherwise, if party code is followed by a digit,*

*if number of digits preceding the party code is less than or equal to two,*

*set count of digits to zero*

*continue with loop*

*otherwise, if number of digits is greater than 2,*

*pollData is not valid*

*break*

*otherwise, if stateForecast does not start with a valid state code,*

*pollData is not valid*

*otherwise, if state forecas*

*pollData is not valid*

*(return to beginning of while loop)*

*return whether pollData is valid or not*

*tallySeats:* counts number of seats a specified party is predicted to win

*if pollData does not have proper syntax,*

*return 1*

*else if specified party code is not a letter,*

*return 2*

*otherwise,*

*if character in poll data matches party code*

*add up digits preceding the code*

*set seatTally to this sum*

*return 0*

**Test Cases:**

pollData string is invalid, tallySeats returns 1, and seatTally is unchanged:

(“ThisIsNotPollData”, ‘d’, seats) *//nonsense string*

(“FL8D,MA6R17D5I,pa,hi3r07D,TX69R,”, ‘d’, seats) *//extra comma at end*

(“,FL8D,MA6R17D5I,pa,hi3r07D,TX69R”, ‘d’, seats) *//comma at beginning*

(“ FL8D,MA6R17D5I,pa,hi3r07D,TX69R”, ‘d’, seats) *// space at beginning*

(“FL8D,MA6R17D5I,pa,hi3r07D,TX69R ”, ‘d’, seats) *// space at end*

(“FL8D,MA6R17D5I,pa,hi3 r07D,TX69R”, ’d’, seats) *//space in middle*

(“ “, ‘d’, seats) *// string is just a space*

(“FL8D,MA6[17D5I,pa,hi3r07D,TX69R”, ‘d’, seats) *//non-letter/number/comma character*

(“FL8D,Mp6R17D5I,pa,hi3r07D,TX69R”, ‘d’, seats) *//invalid state code (random letters)*

(“FL8D,MA6R17D5I,pa,h3r07D,TX69R”, ’d’, seats) *//invalid state code (only one letter)*

(“8D,6R17D5I,3r07D,69R”, ‘d’, seats) *//missing all state codes*

(“FL8D,MA6R17D5I,pa,hi3r07D,69R”, ‘d’, seats) *// missing a state code*

(“K”, ‘d’, seats) *//not a state forecast*

(“FL8D,MA6R17D5I,pa,hi3r07D,TX699R”, ‘d’, seats) *//invalid party result (too many digits)* (“FL8,MA6R17D5I,pa,hi3r07D,TX69R”, ‘d’, seats) *//invalid party result (no party code)* (“FL8D,MA6R17D5I,pa,hi3r07D,TX69Rt”, ‘d’, seats) *//invalid party result (more than one party code)*

(“FL8D,MAR17D5I,pa,hi3r07D,TX69R”, ‘d’, seats) *//invalid party result (no numbers)*

(“FL8D,MA6R17D5I,pa,hi3r07DTX69R”, ‘d’, seats) *//missing a comma between two state forecasts*

party character is invalid, tallySeats returns 2, and seatTally is unchanged:

(“FL8D,MA6R17D5I,pa,hi3r07D,TX69R”, ‘$’, seats) *// is a special character*

(“FL8D,MA6R17D5I,pa,hi3r07D,TX69R”, ‘1’, seats) *//is a number*

(“FL8D,MA6R17D5I,pa,hi3r07D,TX69R”, ‘ ‘, seats) *//is a space*

pollData and party are both valid, tallySeats returns 0, and seats is set to the correct value:

(“FL8D,MA6R17D5I,pa,hi3r07D,TX69R”, ‘d’, seats) – seats set to 32

(“FL8D,MA6R17D5I,pa,hi3r07D,TX69R”, ‘D’, seats) – seats set to 32 *// valid regardless of* party *capitalization*

(“FL8D,MA6R17D5I,pa,hi3r07D,TX69R”, ‘r’, seats) – seats set to 78 *// valid for other* partys

(“FL0D,MA6R00D5I,pa,hi3r0D,TX69R”, ‘d’, seats) – seats set to 0 // *if party result is 0*

(“FL8D,MA6R17D5I,pa,hi3r07D,TX69R”, ‘g’, seats) – seats set to 0 *//if* party *character not present in string*

(“FL8D,MA6R5I,pa,hi3r,TX69R”, ‘d’, seats) – seats set to 8 *// if only one party result for* party

(“FL8D,MA6R17D5I8d,pa,hi3r07D,TX69R”, ‘d’, seats) – seats set to 40 *// if more than one party result for* party

(“FL8D”, ‘d’, seats) – seats set to 8 *// if only one state forecast*

(“FL,MA,pa,hi,TX”, ‘d’, seats) – seats set to 0 // *if no party results*

(“FL”, ‘d’, seats) – seats set to 0 *// if no party results*

(“”, ‘d’, seats) – seats set to 0 *// if empty string*

(“fl8D,ma6R17D5I,pa,hi3r07D,tx69R”, ‘d’, seats) – seats set to 22 *//state codes not uppercase*

(“Fl8D,ma6R17D5I,pa,Hi3r07D,tX69R”, ‘r’, seats) – seats set to 18 *// state codes not uppercase*