1. The biggest problem I faced during this project is figuring out how to separate a poll data string into individual state forecasts. Since an individual forecast statement can have different lengths based on how many digit is used to represent the number of votes, it is tricky to identify which characters within the forecast represent the number of votes and where the forecast statement ends. I solved this by checking the 4th character of each forecast to determine the whether the statement is 4 characters long or 5 characters long. Knowing how long the forecast statement is, I can also figure out where the statement ends.
2. hasCorrectSyntax:

if pollData is empty,

return true

if pollData is less than 4 characters long,

return false

repeatedly:

if first 3 characters do not consist of state code and a digit,

return false

identify the 4th character to determine the length of the state forecast

make sure the 5th character is a letter if the forecast is 5 characters long

if do not have enough characters remaining in the string to make a forecast,

return false

read the next forecast in the string

once the loop finishes without returning false, return true

countVotes:

if pollData has bad syntax,

return 1

if party is not a letter,

return 3

repeatedly:

read the 4th character in the string to determine length of the forecast

records the length of current forecast

read the # of votes

if no votes,

return 2

add # of votes to running total

read the next forecast in the string

set voteCount to equal to the running total of votes

return 0

1. List of test cases:

|  |  |  |
| --- | --- | --- |
| string pollData | char party | reason |
| CA55DTX38R Ms6rnY29dUT06L | R | Make sure a correctly formatted data string works |
| CA55DTX38R Ms6rnY29dUT06L | D | Make sure a correctly formatted data string works, this time with a different party |
| CA55DTX38R Ms6rnY29dUT06L | @ | party must be a letter |
| CA5@DTX38R Ms6rnY29dUT06L | D | The number of votes must consist of digits |
| CA55DTX38R Ms6rnY29dUT06L | d | Make sure party is not case-sensitive |
| CA55DTX38R Ms6rnY29dUT06L | r | Make sure party is not case-sensitive, this time with a different party |
| C@55DTX38R Ms6rnY29dUT06L | D | The state codes must be valid state codes consist of letters |
| aasdfa | D | Make sure hasCorrectSyntax can identify data that doesn’t fit the requirement of state forecast |
|  | R | Make sure empty string passes hasCorrectSyntax |
| CA55D TX38R Ms6r nY29d UT06L | R | Make sure data with spacing doesn’t pass hasCorrectSyntax |