MAIN OBSTACLES

1. Figuring out how to build the first function incrementally.
2. Figuring out how to set the conditions in the loops, especially the first while loop in “bool”.
3. Finding a way to count the number of position in the whole string of pollData.
4. Finding a way to calculate and add up the number in each forecast by studying the “technique for processing strings”.
5. Finding the exact position to set voteCount a value and let it unchanged when a return 1,2,3.

BRIEF DESCRIPTION

1. bool hasCorrectSyntax(string pollData)

when it is the case of no forecast

when there is not a single whole forecast

for the case if there is one more alphabet in the end of the string

make sure that the state code is correct

transform all state code into upper letters

separately make sure state codes are valid

make sure there are vote counts after state code

make sure it is a digit after state code

find the vote count numbers

check the validity of vote count numbers and existence of party code

check the validity of party code

return the result of bool function

1. int countVotes(string pollData, char party, int& voteCount)

make sure it is a valid string input

make sure it is a valid party input

count the number of votes for a specific party

set the value into voteCount and return the correct value for the int function.

TEST CASES

Count = 123

1. assert(countVotes("ca12321d", 'd', count) == 1 && count == 123);

the case when number of digit in the forecast is invalid

failed at first because of I though the number can be arbitrary.

1. assert(countVotes("Ca12D", 'd', count) == 0 && count == 12);

the case when there is one forecast with different cases of letter in state code and party code.

1. assert(countVotes("qq12d", 'd', count) == 1 && count == 123);

the case when there is an invalid state code.

1. assert(countVotes("ca12.", 'd', count) == 3 && count == 123);

the case when there is an invalid party code

1. assert(countVotes("ca12dal31Cak12D", 'd', count) == 0 && count == 24);

the case of several forecast

1. assert(countVotes("ca32dAl12cAK00d", 'd', count) == 2 && count == 123);

the case with one zero vote