a.The greatest obstacle that I overcame would definitely be figuring out a method to attempt the project. I was confused about how to separate the poll data string but eventually, I realized that I could separate it by commas and make a temporary string for the substring in between each comma. After this, however, the last substring wasn’t accounted for because there were no commas at the end of the poll data string. I decided to string on a comma to poll data string to account for this. Then, I was also stuck on how to check the temporary string to see if there was a state, either one or two digits, and then a party. I decided to do multiple if statements to check these and this is in a for loop so it would account for each index. I was also kind of stuck on how to convert a string into an int in the countSeats function so I just subtracted ‘0’ from the character number in the pollstring. The biggest obstacle would be my isValidPartyResult function would skip an index running through the for loop once and I figured out I had to decrement j at the end of the for loop.

b.

define boolean isValidUppercaseStateCode function with a parameter of string stateCode

define each state code

return the boolean value

define boolean isValidParty function with a parameter of character Party

capitilize the character Party

check if Party is a letter

return the boolean value

define boolean isValidState function with a parameter of string state

capitilize each character in string state

for each letter on string state

string on each character to string upperState

use function isValidUppercaseStateCode on string upperState

return the boolean value

define boolean isValidPartyResult function with a parameter of string PartyResult

go through each index of PartyResult until there’s no indexes left, starting with i being 0:

if the string begins with a state

assign j to 2

go through each index starting at 2

repeatedly until a boolean is reached

check if j is less than the string size

check if there’s a digit

check if there’s another digit or party

if there’s a digit

check if there’s a party

if any of the code above fails

return false

or else

return true

define boolean isValidPollString function with a parameter of string pollData

check if it’s an empty string

return true

add a comma to the end of string

repeatedly for each index of the string

check if there’s a space

return false

or else check where the comma is located

string everything from the first comma to the most recent comma

use function isValidPartyResult to evaluate each string

return boolean value

define integer countSeats function with parameters string pollData, char party, and int& seatCount

if party isn’t a valid party

return 2

if pollstring isn’t a valid pollstring

return 1

or else

repeatedly

check for the party in the pollstring

check if there’s a number before the party

add the number to seats

return 0

C.

assert(isValidPollString("MD")); //checks if one state is true

assert(isValidPollString("")); //checks if an empty string is true

assert(!isValidPollString("CA3 D45e24t")); //checks if there's a space it returns false

assert(!isValidPollString("CT5D,Ny9Rf")); //a state with no digits before it

assert(!isValidPollString("Ny9R33")); // a number with no party after it

assert(!isValidPollString("CT3"));

assert(isValidPollString("CT5D,NY9R16D1I,VT,ne3r00D")); //checks if multiple states and results can be taken in

assert(!isValidPollString("ZT5D,NY9R16D1I,VT,ne3r00D")); // checks if multiple states and reults can be taken in

assert(!isValidPollString("CT5D,NY9R16D1I,VT2,ne3r00D")); //digit with no state with long poll string

assert(!isValidPollString("ZT5D,NY9R16D1I,VT,ne3r00D"));

int seats;

seats = -999;

assert(countSeats("CT5D,NY9R16D1I,VT,ne3r00D", 'd', seats) == 0 && seats == 21); // so we can detect whether countSeats sets seats

seats = -999;

assert(countSeats("CT5D,NY9R16D1I,VT,ne3r00D", '%', seats) == 2 && seats == -999); //checks if it returns 1 for invalid party

seats = -999;

assert(countSeats("Md42e5c2d,ca31e63e3g,ny1d", 'e', seats) == 0 && seats == 136); // so we can detect whether countSeats sets seats

seats = -999;

assert(countSeats("ZT5D,NY9R16D1I,VT,ne3r00D", 'e', seats) == 1 && seats == -999); //checks if invalid strings returns 2

assert(countSeats("Md42e5c2d,ca31e63e3g,ny1d", 'd', seats) == 0 && seats == 3); //checks if seats are set to 3 //checks if it evaluates the d in “md”