**2a) Obstacles**:

A huge obstacle I had with project three was getting started. I read the project specs the day they were released, but had no idea where to begin. This led to procrastination and lots of stress. Another issue I had was developing the countSeats function. I had trouble developing a solution to to correctly loop through each state chunk.

**2b) Pseudocode:**

IsValidPollString

* Check for empty input
* Adds a comma to end of string to make condition of chunking data the same
* Check for comma, if comma, check if state poll data valid
* Enter function dataChunk
  + First check for valid state code
  + Ensure party codes are surrounded by digits
  + Last char should be a party char
  + Return true if all pass
* Set state poll data empty again, loop through input to find next state’s poll data and repeat until ensure valid string
* Return true once gone through entire input string

countSeats

* Ensure again valid input
* Make sure valid party char is passed in
* Make the party char uppercase and the input uppercase to compare same casing
* Check condition for comma, or it being the last char in the string
* Find comma enter loop checking for number of votes
* Find a digit, converts it from ascii to ints
* Adding votes to overall sum if matches party
* Reset state input and continue down line until find next comma or last char

**2c) List of test data:**

1. (“”, d, seats)
   * input empty string to make sure still returns valid string with no votes
2. (“ne3d, ct4r”, d, seats)
   * ensure quits program because of space in input
3. (“ne3d5n2d,NY9p2d”, d, seats)
   * make sure still compute all ‘d’ votes even though not consecutive in string
4. (“CT5D,NY9R16D1I,VT,ne3r00D”, d, seats)
   * to ensure correctly pass isValidPollString function
5. (“ZT5D,NY9R16D1I,VT,ne3r00D”, d, seats)
   * ensure fails isValidPollString because not a valid state
6. seats = -999;

(countSeats("CT5D,NY9R16D1I,VT,ne3r00D", 'd', seats) == 0 && seats == 21);

* Ensure that correctly setting seats to the right amount of votes and correctly returns 0 after successful run

1. seats = -999; // so we can detect whether countSeats changes seats

assert(countSeats("CT5D,NY9R16D1I,VT,ne3r00D", '%', seats) == 2 && seats == -999);

* Ensures that because it was invalid input for party char, countSeats needs to return 2 and also, seats needs to be untouched and still equal -999

1. (“ne”, d, seats)
   * Just a state code is still valid, make sure no votes allocated to party
2. (“ne3b4nct2d3, d, seats)

* Make sure it fails isValidPollString because does not end with a party

1. (“IA4D,ms9d”, d, seats)

* Make sure casing doesn’t matter and still add 9 and 4 votes to party D