**Chelsea Biala**

**CS31 Smallberg**

**Fall 2020**

1. The greatest obstacle I overcame was the general debugging process- I had a number of errors in my tallyVotes function at various stages that took me a little while to pinpoint. I also had a bit of difficulty parsing the digits from the state forecast strings into ints, but the write-up on the website helped a lot with that. Otherwise the process was relatively straightforward.
2. isSyntacticallyCorrect checks if the argument is a valid poll data string:

if the string is empty

return true

until no more characters left in string:

check for digit(s) at beginning

if not return false

check for valid state code

if not return false

check for party character

if not return false

repeats for next state forecast

returns true if iterates through whole argument string with no error

processCode counts votes in each individual state forecast:

if vote count is 0

return 0

checks if party argument is the one specified by main

converts vote count from characters to an integer

returns vote count in state forecast

(continued next page)

tallyVotes totals votes in poll data string for a particular party:

if the poll data string isn’t valid

return 1

if the party argument isn’t valid

return 2

repeatedly:

finds end of state forecast

runs processCode on that state forecast

if processCode returns 0

return 3

add value returned by processCode to temp variable

sets voteTally to final value of temp variable (all votes for that party added)

return 0

c. Empty string (“”, ‘d’, 999)

No votes (“CAR”, ‘d’, 999)

Wrong number of digits (“334CAr”, ‘d’, 999)

Invalid state code (“14CCd”, ‘d’, 999)

Lowercase state code (“14cad”, ‘d’, 999)

Invalid character (“14CAd12#mOr”, ‘d’, 999)

0 votes for party (“14CAr0Mod13ild”, ‘d’, 999)

2-digit 0 votes (“0014Cad”, ‘d’, 999)

1-digit vote counts (“5CAd5pAd1cor”, ‘d’, 999)

1-digit vote counts w/ 0 (“05CAd5pad1cor”, ‘d’, 999)

2-digit vote counts (“35CAd14par13cod”, ‘d’, 999)

Mixed # vote counts (“5CAd34pad3cor”, ‘d’, 999)

Non-alphabetical party (“5CAd34pad3cor”, ‘%’, 999)

(In all cases program should work as required as spec)