1. The first obstacle was how to set up the main structure. This one is easy though. I decided to set up the main function following by three supplemental functions. All the prompting for input an outputting for result processes will be finished in the main function. The second obstacle is how to check the string is in correct format. What I do is to break the string into several groups, each of which has 4 or 5 elements, since only 4 or 5 characters can be converted to a valid forecast. As for groups for 4 elements, the first element must be a digit, representing the number of votes; the second and third will be a valid state code and the fourth will be the party code, which are all characters. As for groups with 5 elements, the only difference is that it has 2-digits votes. The third obstacle is probably the most annoying one: how to test whether two-character state code is valid or not. There are 4 \* 51 = 204 possibilities. I set up an array of strings which store all these possibilities and build up a loop to test them one by one. It is kind of awkward, but it works well. The final obstacle is a common one: how to find a particular character in a string consecutively. I use the string.find(‘text’, starting point) function and a loop to keep updating the starting point until no longer find the particular character.

Main()

Prompt user enter a poll data

Prompt user enter a party code

Get return value from tallyvote function

If value is 1

Data poll string is not correct

Else if value is 2

Party code is not a letter

Else if value is 3

There exist 0 vote

Else

Output result

isSyntacticallcorrect()

If poll data string is empty, return true

If size is less than 4, return false

Repeatedly

Increase k by 1 If poll data at k is digit

Else Return false

If poll data at k is digit Check if match the valid state code

Else check if match the valid state code

Increase k by 1 if poll data at k is digit

isStateCodeCorrect()

get a substring of length 2 from k

return false if one character is not character

repeatedly check if it matches with the state code array

tallyVote()

return 1 if isSyntsticallyCorrect returns false

return 2 if party code is not a character

return 3 if string at 0 is 0

return3 if finding 0 that at 1

repeatedly

return 3 if there is 0 vote

else updating the starting point

set voteTally to 0

skip the rest of function if the string is empty

make the party code available in both upper- and lower-case format

find the party code between 0 and 4 or 5

repeatedly

find the party code one that starts at x

tallying the vote

updating x

repeatedly

find the party code two that starts at y

tallying vote

updating y

1. One valid forecast (34cad, d)

Party code with upper letter (34cad, D)

Several valid forecasts (20cad15AkD5nyr, d)

Invalid forecast (format) (1ccadcs31cad, d)

Invalid forecast (state code) (15bbD, d)

Forecasts that contain at least 1 zero vote (12cad0txr, d)

Empty forecast ( , d)

Empty party code (13cad, )

Empty forecast and party code ( , ) (return 1)

Invalid forecast and invalid party code (1ccad06nyr, @)

Valid forecast, but invalid party code (16car20nyd, @)

0 vote in total (13car21nyd8txg, f)