Tyler Stovsky

705512370

1. A notable obstacle I encountered and overcame was when I was trying to figure out if a number of positive or negative cases was a leading zero, a non leading zero, or a regular digit. Since a non leading zero is valid but a leading zero isn’t, I realized I needed to check if the next character was a digit or not. However, while doing this I encountered many out of bounds errors that occurred if the zero I was checking was at the end of a string. To fix this, I checked to make sure the next character’s index was less than the length of the string before I checked the next character. This way, the function would return false before it checked the next character if it was the end of a string.
2. In isValidResultString, first I check for an empty string, then I incrementally evaluate the string character by character, checking each condition. First I make sure the string starts with ‘R’, then if the next character is a nonzero digit, then if the next is a ‘+’ or ‘-‘, then if the next character is a digit and what kind of digit (leading/nonleading zero, regular digit), then if the next is a ‘+’ or ‘-‘ and if its the opposite as the first one, then again for another digit and what kind. I then loop this process for each batch.

In positiveTests, I make sure the string is valid and then add up all the numbers directly after a ‘+’ in the string. For negativeTests, I do the same thing but for numbers after a ‘-‘. For totalTests, I do the same thing but for numbers after a ‘R’.

For batches, I count all the times ‘R’ appears in the string.

For all my functions, I make great use of the extractNumber function provided to us. It has allowed me to pull an integer from a string, while also moving the index to the end of the number in the string for me.

c. Empty string (“”)

String with just spaces (“ “)

No tests (“R”)

First character is not ‘R’ (“r”)

First character is not ‘R’ (“r6-3+3”)

Begins with an integer (4)

Too short, leading zero for total tests (“R0”)

Too short, non leading zero for total tests (“R4”)

Too short, non ‘+’ or ‘-‘ symbol (“R8\_”)

Too short, ‘+’ at the end (“R500+”)

Too short, ‘-‘ at the end (“R42-“)

Too short, zero for first case (“R5-0”)

Too short, first case ending in zero (“R20+10”)

Too short, first case leading zero (“R35-01”)

Too short, second non ‘+’ or ‘-‘ symbol (“R20-6h”)

Too short, second symbol same as first (“R6+3+”)

Too short, second symbol opposite (“R90-45+”)

Last case leading zero (“R21-14+07”)

Valid string with a zero (“R5+0-5”)

Last case zero valid string (“R5-5+0”)

Last case zero invalid string (“R15+14-0”)

Positive and negative don’t equal total tests (“R50-25+24”)

Extra characters (“R60-20+40 hello”)

Zero total cases (“R0+0-0”)

No positive tests (“R9+-9”)

No negative tests (“R9+9-“)

Leading zeros in positive and negative tests (“R8-004+000004”)

Leading zero in total tests (“R007+3-4”)

Second symbol same as the first (“R20+10+10”)

Multi digit int values valid string (“R1000-200+800”)

Valid string multiple batches (“R2-1+1R63+40-23”)

First batch is valid, second is not (“R400-50+350R0-0+0”)

Second batch too short (“R10+5-5R”)