1. A brief description of notable obstacles you overcame.

At first my bool function kept failing to compile and because it had multiple mistakes. I had to start from scratch multiple times. In the process of making myIsValidResultString function I realized that, given a string s, the last position would be s.size() -1 instead of s.size(). Furthermore, I had to change a lot of code I had inside for loops because it resulted in false when the input was a valid string.

1. A description of the design of your program.

When constructing my isValidResultString function I mainly thought about how a valid string would look. I realized that there is a certain sequence a string must obey in order to be valid. This sequence is “R(batch test results)+(positive results)-(negative results)” or “R(batch test results)-(negative results)+(positive results)”. I realized that there is 1 R, -, + sign per batch and that each symbol is separated by numbers which must obey:

batch test results = batch positive results + batch negative results

before checking another batch which must always start with an R. My design was based on ensuring that these two rules were obeyed along with some other minor rules. Using a for loop I extended this idea to accommodate multiple batches which allowed me to solve the main problem of determining whether an input is valid or not. From there it was much easier to construct the rest of the functions.

1. A list of the test data that could be used to thoroughly test your program, along with the reason for each test. You don't have to include the results of the tests, but you must note which test cases your program does not handle correctly. (This could happen if you didn't have time to write a complete solution, or if you ran out of time while still debugging a supposedly complete solution.)

The following rules must be obeyed by a correct program:

1. Input String must start with a batch which starts with an R
2. Input String cannot have any letter other than R
3. After an R there must be a number that represents the total number of results for that batch:

* This number must be nonzero
* This number must not have any leading zeroes

1. There must be a + or – sign after the total number of results of a batch
2. There must be a number that represents the positive/negative results for a batch after a + or – sign

* This number must not have any leading zeroes

1. There must be a – sign after the number of positive results for a batch and a + – sign after the number of negative results for a batch
2. (batch test results = batch positive results + batch negative) results must hold true for every batch in the string and (total test results = total positive results + total negative results) must hold true for the whole string
3. A batch must only have one +, -, and R sign, no more or no less
4. A string can have multiple batches as long as each batch is written without any whitespace or characters in between them

Below are some test cases that test these rules

**Test Case 1:**

Reason: no extra characters are allowed

Input 1: R1+0-1 abcdefg

Input 2: R1abcdefg+0-1

Input 3: "R3-1+2 "

Input 4: R3-1+2R2-1+1 02392

Expected output: false

**Test Case 2:**

Reason: no letter other than R is allowed

Input: r1+1-0

Expected output: false

**Test Case 3:**

Reason: a number of positive tests is required

Input: R1+-1

Expected output: false

**Test Case 4:**

Reason: a number of negative tests is required

Input: R1+1-

Expected output: false

**Test Case 5:**

Reason: must have a number of cases to report

Input 1: R0+0-0

Input 2: R0-0+0

Input 3: R10-5+5R

Expected output: false

**Test Case 6:**

Reason: leading zeros not allowed

Input 1: R05+00003-0002

Input 2: R5+00003-2

Input 3: R5+5-00

Input 4: R05+5-0

Expected output: false

**Test Case 7:**

Reason: a + and a - are required for a single batch

Input 1: R5-5

Input 2: R5+5

Expected output: false

**Test Case 8:**

Reason: can only have one – and one + for a single batch, cannot have less or more

Input 1: R5-3-2

Input 2: R5+3+2

Input 3: R5

Input 4: R5+1-1+3

Input 5: R5-1+1-3

Expected output: false

**Test Case 9:**

Reason: +, -, and R must have numbers between them

Input 1: R5-+5-0

Input 2: R5+-5+0

Input 3: R+0-0

Expected output: false

**Test Case 10:**

Reason: can have multiple batches

Input 1: R100+50-50R1-1+0R100000-50000+50000

Expected output: True

**Test Case 11:**

Reason: need an R at the start of every batch

Input 1: 0+0-0

Input 2: 5+2-3

Input 3: R5+2-3 5+2-3R5+2-3

Input 4: R5+2-3 +2-3R5+2-3

Expected output: false

**Test Case 12:**

Reason: a + and a - are required inside every batch of results

Input 1: R5-3+2R2+2+0

Input 2: R5-3+2R2-2-0

Expected Output: false

**Test Case 13:**

Reason: batch results must match and make sense

Input 1: R10-1+0R10+2+17

Input 2: R5-7+0R2-0-0

Expected Output: false

**Test Case 14:**

Reason: total results of the string must match and make sense

Input 1: R10-10+0R10-9+1R10+4-5

Input 2: R5-7+0R2-0-0

Expected Output: false

**Test Case 15:**

Reason: no blank space or characters in between batches

Input 1: R10-10+0 R10+2+8

Input 2: R5-5+0RR2-0-2

Input 3: R5-5+0R=R2-0+2

Input 4: R5-5+0RasfsdjkR2-0+2

Expected Output: false

**Test Case 16:**

Reason: all -, +, R signs must have numbers between them

Input 1: R10-9+1R10+10-R1+1-1

Input 2: R10-+10R10+10-R1+1-1

Input 3: R10-0+10R10+-10R1+1-1

Expected Output: false

**Test Case 17:**

Reason: zero cases reported are not valid in a batch

Input: R2-1+1R0+0-0

Input: R0+0-0

Expected Output: false

**Test Case 18:**

Reason: empty string is not valid

Input: “”

Expected Output: false