Responses will be graded on correctness, code design, and code style. Cor- rectness refers to your code's ability to pass the automated test cases (when you click "Submit" in repl.it, you will receive a report stating how many correctness points you have earned). Note that your program's output must exactly match the specs given here for each problem to pass the test cases. Design refers to how well your code is written (i.e. is it clear, efficient, and elegant). Style refers to the readability of your code (commented, correct indentation, good variable names).

(1) (10 points) Write code that prompts a user for their name and then dis- plays "Hello, [name here]!" The flow should look like the following:

```
What is your name? Rachel Min
Hello, Rachel Min!
```

If the user does not enter anything but presses Enter anyways, you should re-prompt for the user's name. This flow should look like the following (note that there should be a space after any ? or :):

```
What is your name?
What is your name?
What is your name?
Rachel Min
Hello, Rachel Min!
```

(2) (40 points) Binary & Linear Search
In this problem, you will write binary and linear search. For both func- tions you are to return the index of the specified number, or -1 if the number is not in the input list. Also, even if you pass the tests, the TAs will go through the code and ensure you implemented linear search for linear search and binary search for binary search.

For example, searching [0, 5, 10] for 10 should return 2.
Searching [0, 5, 10] for 12 should return -1.

(a) In repl.it implement the linearSearch() method. This method should perform linear search on the given input list, and return an index, or -1 if the element doesn't exist.
(b) In repl.it implement the binarySearch() method. This method should perform binary search on the given input list, and return an index, or -1 if the element doesn't exist.

(3) (50 points) Word Analysis
In this problem, you will need to implement each of the following meth- ods: MIN, MAX, RANGE, AVERAGE, MODE. The first line of input represents the method to call. It will be one of the following: MIN, MAX, RANGE, AVERAGE, MODE. The second line will represent an integer n, which denotes the number of words in the list. The following n lines will each be a word. The

words will not necessarily be sorted. Your output should be a single line representing the results of the method. I included an example below.

input:
MAX
3
hello
there
wonderful
output:
9

(a) Implement the minWordLength() method, which should return the length of the smallest word in the list.
(b) Implement a method called maxWordLength(), which should return the length of the largest word in the list.
(c) Implement a method called wordLengthRange(), which should re- turn the range of lengths in the word list.
(d) Implement a method called averageWordLength(), which should re- turn the average word length in the word list. The method should return the average, accurate to exactly two decimal places (i.e. if the average length is 5, return 5.00).
(e) Implement a method called mostCommonWordLength(), which should return the most common length of the words in the list. If there is a tie, you should return -1. You may assume that the length of a word is at most 100 and at least 1.