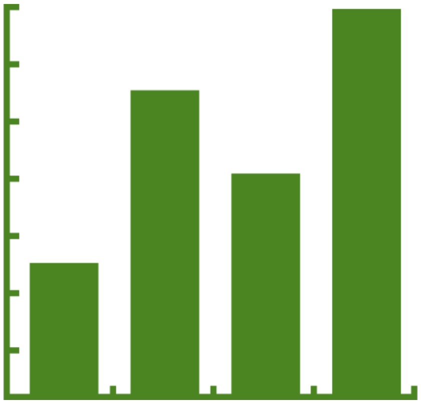
# Problem 130: Word by Word

Difficulty: Hard

Author: Richard Green, Whiteley, Hampshire, United Kingdom

Originally Published: Code Quest Australia 2019

## Problem Background



Cryptography and cryptanalysis are two opposing disciplines that had a profound effect on the course of history. Cryptography is the practice of creating codes, ciphers, and other means of hiding sensitive information from the public view. Since ancient times, military and government leaders have wanted to keep their communications secret to avoid giving any advantage to their enemies. Naturally, their enemies still want to get these communications, and were determined to not allow a simple code to stop them. The process of breaking codes and ciphers is known as cryptanalysis.

Cryptanalysts do a large part of their work using trial-and-error. They make assumptions about the message based on its encrypted appearance, the circumstances under which it was intercepted, who it was sent by and to whom it was being sent. This in turn allows them to make more guesses about what cipher was used, the key used to encrypt that specific message, and eventually what the actual content of the message is.

For example, during World War II, Alan Turing and other codebreakers at Bletchley Park had devised a way to break the German's Enigma cipher. Unfortunately, the Germans changed the key used to encrypt the cipher every day, and without knowing the key, the messages were still illegible. However, the Germans would consistently send a standard message to their troops around midnight every night; a weather report. Being a standardized report, the Allied codebreakers knew that the report would always start with the same word: "WETTER", the German word for "Weather." Using this assumption, Bletchley Park would attempt to break a message intercepted at midnight using all possible keys until one of the keys yielded the word "WETTER." Once this key was found - usually within a few hours - all messages sent that day could be read.

Some of these guesses will inevitably be wrong, but professional cryptanalysts have a wide range of tools to help them. Among these tools are huge amounts of analytical data about various languages - how often certain letters or groups of letters appear, how often certain words appear, which letters most often start and end words, and so on. All of this data can help cryptanalysts make more educated guesses about the messages they're trying to break.

## Problem Description

Your team is working with the Australian Signals Directorate to establish a new bank of statistical data on the English language (these databanks do have to be updated on a regular basis to account for shifts in how we use the language). You've been asked to write a tool that can analyze a large amount of text and count how often words of varying lengths appear. This data will then be stored in a government database for use in codebreaking.

Your tool must count the length of each word in the supplied text and keep a count of how many times a word of that length has appeared. When the entire text has been read, your tool must do the following:

- Calculate and print the average word length
  - To calculate the average, add together each individual word length value, then divide that sum by the total number of words.
- Determine and print the median word length
  - To calculate the median, create a sorted list of all of the individual word lengths. If the list contains an odd number of values, the median is the value in the exact middle of the list. If the list contains an even number of values, the median is the average of the two values in the middle of the list.
- Determine and print the mode(s)
  - The mode is the value that occurs most frequently within the list. If multiple word lengths are tied for the greatest number of occurrences, they are all considered the mode.
- Calculate and print the range
  - The range is the difference between the longest word length and the shortest
- Print a horizontal bar graph showing all word lengths present in the text
  - To print the bar graph, print one line for each word length value, starting from the smallest and ending with the largest. Do not skip any values; for example, if your text contains five- and seven-letter words, but no six-letter words, you should still include a line for a word length of 6. Each line should start with the word length, left-padded with spaces to two characters (that is, you would print 10 as '10' without any changes, but 5 would be printed ' 5', with a single space in front of the 5). Follow this with a pipe character (|), then with a number of lowercase x's equal to the number of times the word length appeared (if the word length did not appear, no x's should be printed).

## Sample Input

The first line of your program's input, received from the standard input channel, will contain a positive integer representing the number of test cases. Each test case will include:

- A line containing a positive integer, X, indicating the number of lines of text that follow.

- X lines, containing the lines of text to be analyzed. The text will contain upper- and lower-case letters, spaces, and standard ASCII punctuation.

```
1
4
Code Quest is really great and challenging.
We hope you have fun during the contest.
Good luck today! If you need any help, send us a clarification.
You can do it!
```

## Sample Output

For each test case, your program must output the following information:

- A line containing the text “Average: H”, where H is the average word length (rounded to one decimal place)
- A line containing the text “Median: I”, where I is the median word length (rounded to once decimal place)
- A line containing the text “Modes: J”, where J is a comma-delimited list of the modes of the list of word lengths. (e.g. “Modes: 4,5” if 4 and 5 are both modes, or just “Modes: 4” if 4 is the only mode)
- A line containing the text “Range: K”, where K is the range of the word lengths.
- K+1 lines containing a horizontal bar graph of the word lengths, as described above.

```
Average: 4.1
Median: 4.0
Modes: 3,4
Range: 12
 1|x
 2|xxxxxx
 3|xxxxxxxx
 4|xxxxxxxx
 5|xxx
 6|xx
 7|x
 8|
 9|
10|
11|x
12|
13|x
```