Problem 269: Voynich Manuscript

Difficulty: Hard

Author: Gary Hoffmann, Denver, Colorado, United States

Originally Published: Code Quest 2025

Problem Background

The Voynich Manuscript is an enigmatic, hand-written manuscript dating from the 15th century. Named for its owner in the 20th century, Wilfrid Voynich, the document is often described as the most mysterious book in the world. It is filled with an as-of-yet undeciphered script known as "Voynichese" that does not appear to match any known language. This writing has stumped linguists, cryptographers, and codebreakers alike, and it remains unknown whether the text is a constructed language, some form of code or cipher, or if the entire document is simply a hoax perpetrated by Voynich himself.

Problem Description

Your team has been tasked with building a tool that can help to analyze blocks of text using an approach previously used to research the Voynich Manuscript.[1] This approach attempts to determine the informational value of each word that appears within the text by analyzing its frequency within specific sections of the text, compared against its frequency in sections of randomly shuffled versions of the text. For example, a math textbook might have the word "calculus" appear more frequently in some chapters than others; we can assume that's a key topic in the textbook.

Your team will be presented with a large block of text to analyze. The text will be divided into several sections, each containing an equal number of words. Before beginning your analysis, you should remove all punctuation and numbers from the original text (don't replace these characters with spaces; this may cause words you would read as separate words to be merged into a single word) and convert all letters to lowercase.

To calculate a word's entropy within a section of text, count the number of times the word appears in the relevant section (S) and how many times it appears in the entirety of the text, across all sections (N). Then input those values into this formula to calculate E_S, the entropy of the word for that section.

$$E_s = -\frac{S}{N}\log_2\frac{S}{N}$$

[1] Montemurro MA, Zanette DH (2013) Keywords and Co-Occurrence Patterns in the Voynich Manuscript: An Information-Theoretic Analysis. PLoS ONE 8(6): e66344. https://doi.org/10.1371/journal.pone.0066344

If a word does not appear in a particular section, its entropy for that section is zero (0). Once you've calculated a word's entropy within each section, add those values together to determine the word's overall entropy for the text ($E_{original}$).

Part of this analysis also requires calculating the average entropy for each word across an infinite number of randomly shuffled versions of the text ($E_{shuffled}$); that is, texts using the same words, but in a randomly determined order. Since you don't have an infinite amount of time, we've already calculated these values for you, and they will be provided in the input.

To calculate a word's informational value, count the number of times it appears across all sections of the text (again, N) and the total number of words in the text (C). Use those values along with the calculated ($E_{original}$) and provided ($E_{shuffled}$) entropy values to complete this formula:

$$I = \frac{N}{C}\left(E_{shuffled} - E_{original}\right)$$

Finally, compare the informational values of each word and determine the top three keywords in each text; a higher value indicates a more important word.

Important note: If your programming language does not provide the native means to calculate a base-2 logarithm, you can calculate it using the natural logarithm (which it should support):

$$\log_2 X = \ln X \,/\, \ln 2$$

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 two positive integers, separated by spaces:
    - S, the number of sections within the text to be evaluated
    - W, the number of words for which the value of $E_{shuffled}$ has been calculated.
- S lines, each containing a single section of the text to be analyzed. Lines may contain any printable characters, but will each contain the same number of words (within a test case). Lines will contain fewer than 2,000 characters.
- W lines, each containing a word that appears in the text (in lowercase letters), a space, and a decimal number representing that word's $E_{shuffled}$ value. Words that appear in the text that are not listed in this section have an $E_{shuffled}$ value of 0.0. Words will be listed in alphabetical order.

Due to its length, the sample input will not be replicated here. Please download the sample input from the contest website.

Sample Output

For each test case, your program must print the three words in the text with the highest informational values, on a single line and separated by spaces, in descending order of informational value. Print words using lowercase letters only.

```
chocolate pasta broccoli
```