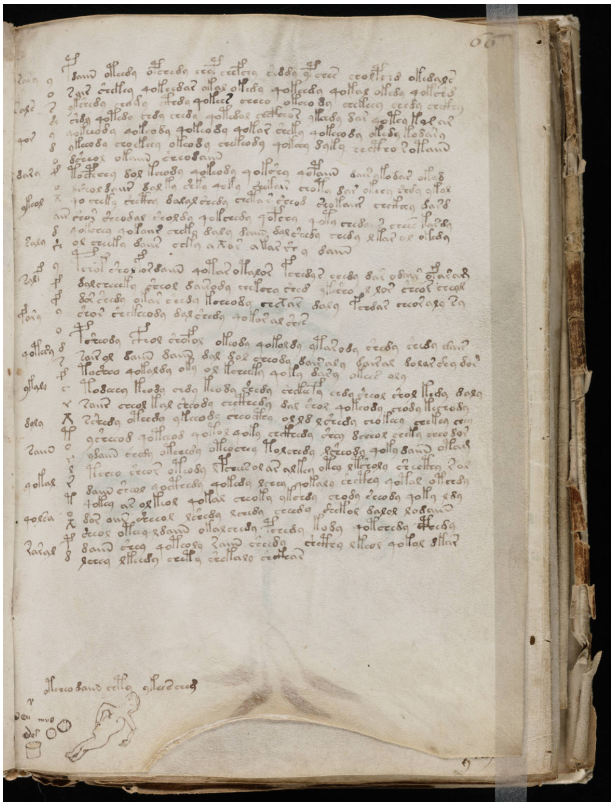
Challenge 2 - The Voynich Manuscript

« Prev Next »

The Voynich manuscript is an illustrated codex hand-written in an unknown writing system.



The vellum on which it is written has been carbon-dated to the early 15th century, and it may have been composed in Northern Italy during the Italian Renaissance.

The manuscript is named after Wilfrid Voynich, a Polish book dealer who purchased it in 1912.

The Voynich manuscript has been studied by many professional and amateur cryptographers, including American and British codebreakers from both World War I and World War II. No one has yet succeeded in deciphering the text, and it has become a famous case in the history of cryptography. The mystery of the meaning and origin of the manuscript has excited the popular imagination, making the manuscript the subject of novels and speculation. None of the many hypotheses proposed over the last hundred years has yet been independently verified.

(On the right, a photograph of a page of this mysterious manuscript, from the Beinecke Rare Book & Manuscript Library, Yale University)

In recent times, it has been declared that the corpus of this manuscript approximately follows Zipf's law, which states that given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the

frequency table. Thus the most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, etc.

So, as the lead computer scientist at Yale University Library, you are determined to put an end to this mystery and solve whether the Voynich Manuscript is really written in a natural language or whether it is some kind of encrypted book.

In order to do so, you will develop a code that is able to determine which fragments of a corpus follow Zipf's law and see which are the most frequent words and how frequently they appear.

Input

The downloadable corpus text of a sample manuscript with words separated by spaces.

In the first line, the number of fragments to check from the manuscript, N.

Each case is described in a line with two ordinal integers, A and B, that determine the first and the last word of the fragment to check.

Output

For each case, the output is the string "Case #t:" followed by the first word and the number of literal appearances (frequency) separated by a space, the second word and its frequency, and the third word and its frequency, separated by commas.

We can assure you that the three words that appear most frequently will not appear the same number of times as each other.

Check the output example.

Limits

1 <= N <= 5000

1 <= A,B <= 35000

Sample input

Considering the following manuscript:

Blue White Aqua Azure Beige Lavender White Black Blue Lavender Blue Brown Khaki Blue Brown Blue White Khaki Blue Cyan Brown Gold White Blue Lavender White Lavender White Ivory Khaki Lavender Beige Lime Magenta

And the following input

```
2
11 22
5 28
```

Sample output

```
Case #1: Blue 4,Brown 3,Khaki 2
Case #2: Blue 6,White 5,Lavender 4
```

Test your code

You can test your program against both the input provided in the test phase and the input provided in the submit phase. A nice output will tell you if your program got the right solution or not. You can try as many times as you want to. Be careful with extra whitespaces, the output should be exactly as described.

Test your program against the input provided in the test phase

Download test input

Program output:

Seleccionar archivo Ningún archivo seleccionado

Submit test output

Test your program against the input provided in the submit phase

Download input

Program output:

Seleccionar archivo Ningún archivo seleccionado

Submit output

During the submit phase, in some problems, we might give your program harder inputs. As with the test token, a nice output will tell you if your program got the right solution or not. You can try as many times as you need.

In the actual contest you first need to solve the test phase before submitting the code, you must provide the source code used to solve the challenge and you can only submit once (once your solution is submitted you won't be able to amend it to fix issues or make it faster).

If you have any doubts, please check the info section.

« Prev Next »