# Problem F
# Basic Text Search Engine

**Time Limit: 3 seconds**
**Memory Limit: 512 Megabytes**

**Problem description**

You are asked to build an application to search for documents by an input query (q) from the dataset (D). Note that the text search problem is not an exact search problem. You can understand that the system will try to find out the text closest to the input query.

The system will build a dictionary from the available words in the dataset (array of documents).

For example:

The dataset defined as:

D = {"I love music", "computer can play music" , "music is just music" , "computer can do everything" , "I love you"};

We will be able to harvest a dictionary of words.

vocab = {"I", "love", "music", "computer", "can", "play", "is", "just", "do", "everything", "you"}.

The words that appear first in the dataset are also first in the vocab.

Usually, search engines see the documents as vectors. The process of converting documents into vectors is called vectorization. The vector space is determined by the vocab. The below table illustrates the result of the vectorization process.

| | I | Love | music | computer | can | play | is | just | do | everything | you |
|---|---|---|---|---|---|---|---|---|---|---|---|
| D[1] | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| D[2] | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| D[3] | 0 | 0 | 2 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| D[4] | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| D[5] | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

The elements of the vector represent the number of occurrences of the corresponding words in the vocab. The same treatment also be applied to the input query.

In order to compute the similarity of 2 given documents A and B, we can use the Cosine similarity as:

$$sim(A,B) = \frac{\sum_{i=1}^{|vocab|} A_i B_i}{\sqrt{\sum_{i=1}^{|vocab|} A_i^{\,2}}\sqrt{\sum_{i=1}^{|vocab|} B_i^{\,2}}}$$

Where $|vocab|$ denotes the size of the vocab.

The search engine has to rank the documents by their similarities to the input query. The system displays the document with higher similarity first. If 2 documents have same similarity degree, the first appear in the dataset will be displayed first.

**Input**

The first line is the input query.

Second line is the number d of documents in the dataset – where d is an integer value and 0 < d <= 1000.

Other lines are the documents in the dataset.

**Output**

Every output lines in format: document (similarity). Where similarity is a decimal value which has an absolute or relative error of less than $10^{-2}$

Example:

| Input | Output |
|---|---|
| I love dog<br>5<br>I love music<br>computer can play music<br>music is just music<br>computer can do everything<br>I love you | I love music (0.81)<br>I love you (0.81)<br>music is just music (0.00)<br>computer can do everything (0.00)<br>computer can play music (0.00) |

| Input | Output |
|---|---|
| music computer music music<br>5<br>I love music<br>computer can play music<br>music is just music<br>computer can do everything<br>I love you | music is just music (0.77)<br>computer can play music (0.63)<br>I love music (0.54)<br>computer can do everything (0.15)<br>I love you (0.00) |

| Input | Output |
|---|---|
| nothing here<br>5<br>I love music<br>computer can play music<br>music is just music<br>computer can do everything<br>I love you | I love music (0.00)<br>computer can play music (0.00)<br>music is just music (0.00)<br>computer can do everything (0.00)<br>I love you (0.00) |

A relax page, open to next page for the next challenge in your journey to the TOP.