CCO '02 P1 - Spam

Time limit: 2.0s **Memory limit:** 64M

Canadian Computing Competition: 2002 Stage 2, Day 1, Problem 1

Unsolicited email (spam) is annoying and clutters your mailbox. You are to write a spam filter - a program that reads email messages of regular ASCII characters and tries to determine whether or not each message is spam.

How can we determine whether or not a message is spam? Spam contains words and phrases that are not common in genuine email messages. For example, the phrase

```
MAKE MONEY FAST, HONEY!!
```

is in all-uppercase, contains the word `money` and ends with a double exclamation mark.

One way to create a spam filter is to read through many spam and non-spam messages and to come up with a set of rules that will classify any particular message as spam or not. This process can be tedious and error prone to do manually. Instead you will write a program to automate the process.

A useful step in automatic classification is to split the text up into a set of **trigrams**. A trigram is a sequence of three adjacent characters that appear in the message. A trigram is case sensitive. The example above is composed of the trigrams:

```
MAK
AKE
KE
E M
 MO
MON
ONE
NEY
EY
Y F
 FA
FAS
AST
ST,
T,
, H
 HO
HON
ONE
NEY
EY!
Y!!
```

If we examine a sample of spam and non-spam messages we find that some trigrams are more common in spam; whereas others are more common in non-spam. This observation leads to a classification method:

- Examine a sample consisting of a large number of spam messages. Count the number of times that each trigram occurs. In the example above, there are 20 distinct trigrams; the trigrams `ONE` and `NEY` occur twice each and the remaining 18 trigrams occur once each. (Trigrams that do not occur are considered to occur 0 times.) More formally, for each trigram t we compute the frequency $f_{\text{spam}}(t)$ with which it occurs in the sample of spam.
- Examine a sample consisting of a large number of non-spam messages. Compute $f_{\text{non-spam}}(t)$, the frequency with which each trigram t appears in the sample of non-spam.
- For each message to be filtered, compute $f_{\text{message}}(t)$ for each trigram t.
- If f_{message} resembles f_{spam} more closely than it resembles $f_{\text{non-spam}}$ it is determined to be spam; otherwise it is determined to be non-spam.

- A **similarity** measure determines how closely f_1 and f_2 resemble one another. One of the simplest measures is the cosine measure:

$$\text{similarity}(f_1, f_2) = \frac{\sum_t f_1(t) \times f_2(t)}{\sqrt{\sum_t [f_1(t)]^2} \times \sqrt{\sum_t [f_2(t)]^2}}$$

  Then we say a message is spam if:

$$\text{similarity}(f_{\text{message}}, f_{\text{spam}}) > \text{similarity}(f_{\text{message}}, f_{\text{non-spam}})$$

Input Specification

The first line of input contains three integers: s the number of sample spam messages to follow; n the number of sample non-spam messages to follow; c the number of messages to be classified as spam or non-spam, based on the trigram frequencies of the sample messages. Each message consists of several lines of text and is terminated by a line containing `ENDMESSAGE`. This line will not appear elsewhere in the input, and is not considered part of the message.

Output Specification

For each of the c messages, your program will output two lines. On the first line, output $\text{similarity}(f_{\text{message}}, f_{\text{spam}})$ and $\text{similarity}(f_{\text{message}}, f_{\text{non-spam}})$. On the second line print the classification of the message (`spam` or `non-spam`). Round the numbers to five decimal digits.

When forming trigrams, we never include a newline character. We don't include trigrams that span multiple lines, either. So in the first spam message of Sample Input 1, the only trigrams are:

```
AAA
BBB
BB
B
  C
 CC
CCC
```

Sample Input 1

```
2 1 1
AAAA
BBBB  CCCC
ENDMESSAGE
BBBB
ENDMESSAGE
AAAABBBB
ENDMESSAGE
AAABB
ENDMESSAGE
```

Sample Output 1

```
0.21822 0.73030
non-spam
```

Sample Input 2

1 1 2

DOES THIS SOUND LIKE YOU?

* Tired of Mounting Credit Card Debt?

* Frustrated by Creditor Harrassment?

* Bogged down by Medical Expenses?

* Just Plain Tired of the Financial Insanity?

HERES WHAT WE CAN DO...

* Reduce your debts by up to 60%!

* Reduce or Eliminate Interest!

* Preserve or Rebuild your Credit

* Stop the Harrassing Phone Calls!

CLICK HERE TO GET OUT OF DEBT

Did you know that you could reduce all of your unsecured debt by up to 60% and consolidate it into ONE monthly payment WITHOUT taking out another loan?!

Let US deal with your creditors, we'll negotiate a reduced payback and combine all of your debt into one simple payment saving you thousands

of dollars! Take 90 seconds to fill out the simple quote form to see how much less you could be paying. It's fast, free and there is no obligation to apply!

CLICK HERE FOR A FREE QUOTE

Please know that we do not want to send you information regarding our special offers if you do not wish to receive it. If you would no longer like us to contact you or feel that you have received this email in error, please click here to unsubscribe.

ENDMESSAGE

If any of them do Java stuff, they might be interested in Soot, our research compiler. A new release is due out any day now.

http://www.sable.mcgill.ca/soot/

Of course, there are other similar things out there. The Flex compiler at MIT is quite nice. It's a native code compiler, whereas Soot outputs bytecode.

I guess Chambers has some stuff too, but I haven't played with it. He apparently uses Soot for his courses.

Other stuff we have (www.sable.mcgill.ca):

SableCC, a better yacc in Java

Ashes, a bunch of Java benchmarks

SableVM, would have been a nice VM, but no really usable versions exist yet

an mostly up-in-the-air profiling and visualization thingie

We don't have any non-Java stuff. Laurie has some old C stuff, but I

don't know that it's very general-purpose.

Ondrej

ENDMESSAGE

Let us help you collect what your children are due!

Contact us now for more information.

You have absolutely nothing to lose!!!

Please call 1 877 306 6599 8am to 5pm CST Mon-Sat.

Consultants are waiting for your call...

++

This ad is pro duced and sent out by:

Unive rsal Adve rtising Syste ms

ENDMESSAGE

Thanks for giving GB the good word on Web traps and treatment thereof.

However, I am puzzled (as usual). I have occasionally encountered similar

garbage which is difficult to dump but I almost never approach the Web via

IE

but use Netscape Communicator to get to Google and so on. GB declares that

she, also, uses Netscape. Do IE and Net ever talk to each other?

We seem to have developed the old problem of the task bar (?) being vertical

rather than horizontal. I looked up you past advice, e.g about the RH mouse

button, but haven't had success in getting the bar back down to the bottom.

It's certainly not a serious problem but sometimes I have to do a a lot of

fiddling to get at the close button and scroll control.

I have finally got around to burning some pictures onto CD's. I had a lot

of trouble matching up what it said in the "manual" with what it said on the

screen but now I ignore the manual and things work fine. I have had

absolutely no Adobe seizures in the course of these burns. What Adobe

doesn't seem to like is combined operations involving the printer. There it

often quite cold at various stages in the procedure.

Has Judy moved from the active to the nail-biting phase of the comp exam?

Pass on our regards.

ENDMESSAGE

Sample Output 2

```
0.28761 0.20595
spam
0.44314 0.49243
non-spam
```