Torneio Inter-Universitário de Programação

Primeira Prova
14 de Abril de 2010 — 17h00–20h00

http://mooshak.di.uminho.pt/

Departamento de Informática
Universidade do Minho

Problems

A The Cybernetic Poet

B Bifid Ciphers

C Stacking Boxes

D Repetitions

E Gaussian Primes

Scientific Committee

- Pedro Rangel Henriques ..... prh@di.uminho.pt
- José Bernardo Barros ..... jbb@di.uminho.pt
- José João Almeida ..... jj@di.uminho.pt
- Rui Mendes ..... rcm@di.uminho.pt
- Alberto Simões ..... ambs@di.uminho.pt

General Information

1. The contest follows ICPC rules (to know more, go to `http://icpc.baylor.edu/`).
2. It has a duration of 3 hours for the 5 problems.
3. The programs must read from the standard-input and write to the standard-output.
4. The generated test inputs use the following norms:
   - There are no spaces at the end of the lines and each line ends with a newline (`\n`).
   - No multiple spacing is being used, unless it is explicit mentioned on the problem description.
5. The generated test outputs follow the same rules.
6. Problems timeouts are (the machine that runs the tests has a 2GHz processor):

   A 1 sec;

   B 1 sec;

   C 6 sec;

   D 3 sec;

   E 1 sec.
7. Each team (3 persons maximum) should use one PC and it is strictly forbidden to use any online resource other than Mooshak or what is given in the local PC disk.
8. Further information can be obtained by clicking on the Help link on your Mooshak account screen.
9. Please ask your questions to the jury using the Questions button.

Available Compilers

Language	Compiler	Version	Command Line	Extension
C	gcc	4.1.2	gcc -Wall -lm $file	.c
C++	gcc	4.1.2	g++ -Wall $file	.cpp
Objective-C	gcc	4.1.2	gcc -fconstant-string-class=NSConstantString (...) $file -lobjc -lgnustep-base	.m
Java	jdk	1.6.0_07	javac -nowarn $file	.java
Pascal	Free Pascal	2.2.2	fpc -v0w -oprog $file	.pas
Haskell	GHC	6.8.3	ghc $file	.hs
OCaml	ocamlopt	3.10.2	ocamlopt unix.cmxa str.cmxa nums.cmxa -pp camlp4o $file	.ml
Python	Python	2.4.3	python $file	.py
Perl	Perl	5.8.8	perl $file	.pl

The Cybernetic Poet

AMS

Introduction

Computer scientists are usually not proficient at writing poems[1]. They know the concept, know what are rhymes and know how to write programs. They simply don't have the sensibility. Your mission is to to write a poem generator.

Problem

We want to generate poems with the usual rhyme mechanism: A B A B (the first line rhymes with the third, the second with the fourth). Also, we will use a simplified metric, that is, the four sentences need to have the same number of syllables.

Given the number of syllables (metric) required and a dictionary (with word metrics and rhyme information) you should generate a random poem.

Input

The first line is the metric. For instance, the number 7 means that each line should have 7 syllables (metric ignores all syllables after the stressed syllable but we will overlook that).

The next line has the number of words in the dictionary. The remaining lines contain the words where syllables are sepparated by dashes (the minus sign). After a sepparating colon follows the rhyme information.

Example:

```
ca-va-lo:alo
```

means that this word has a metric of three, and rhymes with other words with the "alo" as the rhyme information.

The input information was tested for, at least, one possible poem (in fact, 1 poem can be shuffled in different ways, so there is usually more than one solution).

Constraints

- metric ranges from 5 to 15;
- word list can get up to 1 000 words;
- words can have 1 to 10 syllables;

[1]There seems to be an exception among those who work with co-algebras.

Output

The output is a poem, with A B A B rhyme, where all sentences have the same number of syllables (defined by the metric value), and where no word can be repeated. Furthermore, the maximum number of words per line is the metric divided by two rounding up (thus if the metric is 7, the maximum number of words is 4).

Samples

Input

```
5
12
ca-va-lo:alo
ga-to:ato
ja-var-do:ardo
bran-co:anco
com:com
par-do:ardo
ma-ca-co:aco
fran-co:aco
pre-to:eto
fran-co:anco
fres-co:esco
mei-o:eio
```

Output

```
cavalo branco
com gato pardo
macaco franco
meio javardo
```

Bifid Ciphers

RCM

B

Introduction

The Bifid cipher is an aperiodic fractionating cipher. Given a keyword, the first step is to create a 5×5 matrix. As there are 26 letters, the letters `v` and `w` are combined. The matrix is constructed by first laying the keyword (after removing duplicates of the same letter) and then the remaining letters of the alphabet one line at a time. Consider the keyword "profitable". This yields the matrix displayed in table 1. If the keyword has several occurences of the same letter, the first one is used. Thus, using the keyword "privateer" is equivalent to using "private".

	0	1	2	3	4
0	p	r	o	f	i
1	t	a	b	l	e
2	c	d	g	h	j
3	k	m	n	q	s
4	u	v	x	y	z

Tabela 1: The matrix when the keyword is "profitable"

Encrypting a message is performed in two steps. First, using each letter of the plaintext, we create two strings using the coordinates of the letter in the matrix: the number of the line goes to the first string and the number of the column goes to the second one. Thus, if the plaintext message was "Invasion starts tomorrow. Bring the heavy guns." we first remove all spaces and punctuation and convert all letters to lowercase letters. Then we perform the first step:

```
invasionstartstomorrovbringtheheavyguns
03413003311013103000041003212121144243342114422401104021211212142203434113202 4
42114422401104021211212142203434113202 4
```

The second step involves concatenating both strings and then using the matrix to obtain the letters corresponding to the coordinates. We take every two numbers and get the corresponding letter from the matrix. By following this procedure, we finally obtain the ciphertext "fvkfmtltkpitfdddexysdexjrtuddbbegfyvlcj".

```
03413003311013103000041003212121144243342114422401104021211212142203434113202 4
 f v k f m t l t k p i t f d d d e x y s d e x j r t u d d b b e g f y v l c j
```

Problem

Your problem is to decrypt a message. You are given the keyword and the ciphertext and have to discover the plaintext.

Input

The first line of input is the keyword, the following line contains the ciphertext.

Constraints

- The maximum size of the keyword is 20;
- The maximum size of the ciphertext is 5000.

Output

The output is the plaintext.

Samples Input

Input

```
wanderer
lvshrsoxmptdxvosoxmovgvuzamrzrateaeamrz
```

Output

```
itvasthebestoftimesitvasthevorstoftimes
```

Stacking Boxes

RCM

C

Introduction

Robert was trapped inside a warehouse full of boxes. The good news are that the warehouse has windows. The bad news are that they are extremely high! As usual in these cases, there are no stairs in sight nor anything that can help Robert reach the window. Well, except boxes of course!

Did I tell you that Robert is obsessed with mathematical problems? Well, in fact Robert is one of these geeks that have virtually no strength at all and, as such, cannot lift boxes. So he fired up his laptop and turned his predicament into a programming problem: how high could a strong person build a stack of boxes given the dimensions of all the boxes and knowing that any box can be rotated in any way. This means that a box of $2 \times 3 \times 5$ could be used as a box of $3 \times 5 \times 2$ or $2 \times 5 \times 3$.

Problem

Your problem is to find out which boxes can be stacked to form the highest pile possible. For safety reasons, for a box with dimensions $x_1 \times y_1 \times z_1$ to be on top of box with dimensions $x_2 \times y_2 \times z_2$ then either $x_1 < x_2$ and $y_1 < y_2$ or $x_1 < y_2$ and $y_1 < x_2$.

Input

The input will consist of one line containing one integer N giving the number of boxes and N lines, each one containing the dimensions of one box given by three (space separated) integers x y z.

Constraints

$1 \leq N \leq 1200$;
$1 < x, y, z \leq 5000$.

Output

The output should be a line with an integer representing the height of the tallest stack that can be built out with the available boxes.

Samples

Sample 1

In this case the boxes number 10, 6, 9 and 4 were used to make a pile 16 meters high.

Input

```
10
1 3 3
3 1 4
4 1 4
4 5 4
2 5 4
2 4 2
4 2 1
5 4 4
5 4 3
1 1 3
```

Output

```
16
```

Sample 2

In this case the boxes number 10, 7, 8, 4, 5, 3, 1 and 6 were used to make a pile 89 meters high.

Input

```
10
17 14 3
12 7 17
15 13 14
8 15 11
12 9 16
7 17 19
11 5 2
9 11 3
7 1 15
2 11 1
```

Output

```
89
```

Repetitions

JBB

Introduction

Given a finite sequence, we want to determine the maximal subsequence that repeats itself. That is, we want to find the longest repeating subsequence.

Problem

Given a sequence S of N characters we want to determine the longest subsequence T of S that is repeated in S. Formally, we want to determine if there exists T of length K such that

$$S = A + T + B + T + C$$

where + stands for sequence concatenation and such that there exists no other subsequence T' of S of length K' such that

$$S = A' + T' + B' + T' + C' \wedge K' > K$$

Input

Your program should read a string of N characters.

Constraints

$1 \leq N \leq 1000$

Output

The program should print the length of the longest repeating subsequence followed by the sequence itself or a line with a single zero in case there is no repeating subsequence. If there are several repeating subsequences of the same size, the first one should be printed.

Samples

Input

```
yxzxzxyxzz
```

Output

```
3 yxz
```

Input

```
xyz
```

Output

```
0
```

Input

```
aabbccdd
```

Output

```
1 a
```

Input

```
aaaaaa
```

Output

```
3 aaa
```

Gaussian Primes

AMS and RCM

E

Introduction

A Gaussian integer is a number of the form $a + bi$ with $a, b \in \mathbb{Z}$. A Gaussian prime is a Gaussian integer which is only divisible by itself and the unit[1]. Note that not all integer primes are Gaussian primes. For instance 5 is not a Gaussian prime while 7 is[2].

$$\begin{aligned} 5 &= (2+i)(2-i) \\ 7 &= 7 \times 1 = 7i \times -i = -7i \times i = -7 \times -1 \end{aligned}$$

Problem

Your task is to determine if a given Gaussian integer is a Gaussian prime.

Input

The first line contains the number of Gaussian integers N. The following lines contain two numbers a and b that form the Gaussian integer $a + bi$.

Constraints

- $1 \leq N \leq 500$
- $-200 \leq a, b \leq 200$

Output

The output is a line for each Gaussian integer with the word yes or no.

[1] Recall that the unit is both 1, i, -1 and $-i$

[2] As you can only divide it by the unit

Samples

Input

```
8
2 3
2 0
0 -2
3 0
0 3
5 0
7 0
12 -13
```

Outut

```
yes
no
no
yes
yes
no
yes
yes
```