CCC '11 S1 - English or French?

Time limit: 2.0s **Memory limit:** 256M

Canadian Computing Competition: 2011 Stage 1, Senior #1

You would like to do some experiments in *natural language processing*. Natural language processing (NLP) involves using machines to recognize human languages.

Your first idea is to write a program that can distinguish English text from French text.

After some analysis, you have concluded that a very reasonable way of distinguishing these two languages is to compare the occurrences of the letters `t` and `T` to the occurrences of the letters `s` and `S`. Specifically:

- if the given text has more `t` and `T` characters than `s` and `S` characters, we will say that it is (probably) English text;
- if the given text has more `s` and `S` characters than `t` and `T` characters, we will say that it is (probably) French text;
- if the number of `t` and `T` characters is the same as the number of `s` and `S` characters, we will say that it is (probably) French text.

Input Specification

The input will contain the number N $(0 < N < 10\,000)$ followed by N lines of text, where each line has at least one character and no more than 100 characters.

Output Specification

Your output will be one line. This line will either consist of the word English (indicating the text is probably English) or French (indicating the text is probably French).

Sample Input 1

```
3
The red cat sat on the mat.
Why are you so sad cat?
Don't ask that.
```

Output for Sample Input 1

```
English
```

Sample Input 2

```
3
Lorsque j'avais six ans j'ai vu, une fois,
une magnifique image,
dans un livre
```

Output for Sample Input 2

```
French
```

(Note: Sample Input 2 is the first sentence of *Le Petit Prince* by Antoine de Saint-Exupéry.)

Sample Input 3

```
4
Si je discernais ta voix encore
Connaissant ce coeur qui doute,
Tu me dirais de tirer un trait
Quoi que partir me coute.
```

Output for Sample Input 3

```
English
```

(Note: Sample Input 3 is added by DMOJ from *Le Fantôme de l'Opéra*.)

CCC '13 J2 - Rotating letters

Time limit: 2.0s **Memory limit:** 256M

Canadian Computing Competition: 2013 Stage 1, Junior #2

An artist wants to construct a sign whose letters will rotate freely in the breeze. In order to do this, she must only use letters that are not changed by rotation of 180 degrees: I, O, S, H, Z, X, and N.

Write a program that reads a word and determines whether the word can be used on the sign.

Input Specification

The input will consist of one word, all in uppercase letters, with no spaces. The maximum length of the word will be 30 letters, and the word will have at least one letter in it.

Output Specification

Output `YES` if the input word can be used on the sign; otherwise, output `NO`.

Sample Input 1

```
SHINS
```

Output for Sample Input 1

```
YES
```

Sample Input 2

```
NOISE
```

Output for Sample Input 2

```
NO
```

CCC '11 S2 - Multiple Choice

Time limit: 2.0s **Memory limit:** 256M

Canadian Computing Competition: 2011 Stage 1, Senior #2

Your teacher likes to give multiple choice tests. One benefit of giving these tests is that they are easy to mark, given an answer key. The other benefit is that students believe they have a one-in-five chance of getting the correct answer, assuming the multiple choice possibilities are `A`, `B`, `C`, `D` or `E`.

Write a program that your teacher can use to grade one multiple choice test.

Input Specification

The input will contain the number N $(0 < N < 10\,000)$ followed by $2N$ lines. The $2N$ lines are composed of N lines of student responses (with one of `A`, `B`, `C`, `D` or `E` on each line), followed by N lines of correct answers (with one of `A`, `B`, `C`, `D` or `E` on each line), in the same order as the student answered the questions (that is, if line i is the student response, then line $N + i$ will contain the correct answer to that question).

Output Specification

Output the integer C $(0 \leq C \leq N)$ which corresponds to the number of questions the student answered correctly.

Sample Input 1

```
3
A
B
C
A
C
B
```

Output for Sample Input 1

```
1
```

Sample Input 2

```
3
A
A
A
A
B
A
```

Output for Sample Input 2

```
2
```

CCC '12 J3 - Icon Scaling

Time limit: 0.5s **Memory limit:** 256M

Canadian Computing Competition: 2012 Stage 1, Junior #3

You have been asked to take a small icon that appears on the screen of a smart telephone and scale it up so it looks bigger on a regular computer screen.

The icon will be encoded as characters (`x` and `*`) in a 3×3 grid as follows:

```
*x*
 xx
* *
```

Write a program that accepts a positive integer scaling factor and outputs the scaled icon. A scaling factor of k means that each character is replaced by a $k \times k$ grid consisting only of that character.

Input Specification

The input will be a positive integer k such that $k < 25$.

Output Specification

The output will be $3k$ lines, which represent each individual line scaled by a factor of k and repeated k times. A line is scaled by a factor of k by replacing each character in the line with k copies of the character.

Sample Input

```
3
```

Output for Sample Input

```
***xxx***
***xxx***
***xxx***
   xxxxxx
   xxxxxx
   xxxxxx
***   ***
***   ***
***   ***
```

CCC '24 J2 - Dusa And The Yobis

Time limit: 3.0s **Memory limit:** 1G

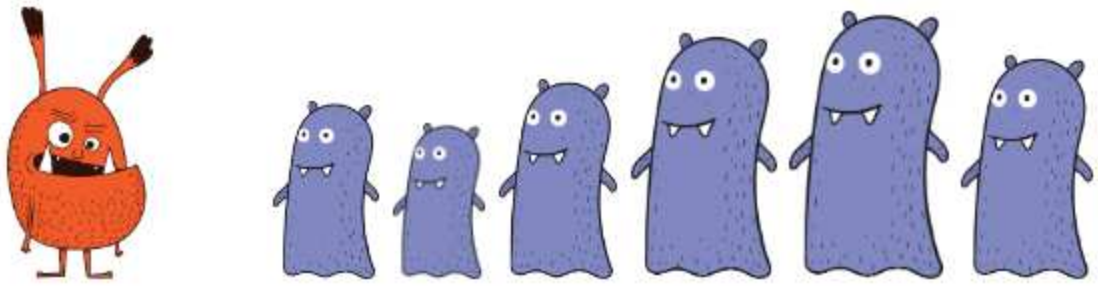
Canadian Computing Competition: 2024 Stage 1, Junior #2

Dusa eats Yobis, but only Yobis of a certain size.

If Dusa encounters a Yobi that is smaller than itself, it eats the Yobi, and absorbs its size. For example, if Dusa is of size 10 and it encounters a Yobi of size 6, Dusa eats the Yobi and expands to size $10 + 6 = 16$.

If Dusa encounters a Yobi that is the same size as itself or larger, Dusa runs away without eating the Yobi.

Dusa is currently facing a line of Yobis and will encounter them in order. Dusa is guaranteed to eventually encounter a Yobi that causes it to run away. Your job is to determine Dusa's size when this happens.



Input Specification

The first line of input contains a positive integer, D, representing Dusa's starting size.

The remaining lines of input contain positive integers representing the sizes of the Yobis in order.

Output Specification

Output the positive integer, R, which is Dusa's size when it eventually runs away.

Sample Input 1

```
5
3
2
9
20
22
14
```

Output for Sample Input 1

```
19
```

Explanation of Output for Sample Input 1

Dusa is large enough to eat the Yobi of size 3. This brings Dusa's size to 8. Dusa is large enough to eat the Yobi of size 2. This brings Dusa's size to 10. Dusa is large enough to eat the Yobi of size 9. This brings Dusa's size to 19. The Yobi of size 20 causes Dusa to run away.

Sample Input 2

```
10
10
3
5
13
```

Output for Sample Input 2

```
10
```

Explanation of Output for Sample Input 2

The Yobi of size 10 causes Dusa to run away, leaving its size unchanged.

CCC '14 J3 - Double Dice

Time limit: 2.0s **Memory limit:** 256M

Canadian Computing Competition: 2014 Stage 1, Junior #3

Antonia and David are playing a game. Each player starts with 100 points.

The game uses standard six-sided dice and is played in rounds. During one round, each player rolls one die. The player with the lower roll loses the number of points shown on the higher die. If both players roll the same number, no points are lost by either player.

Write a program to determine the final scores.

Input Specification

The first line of input contains the integer n $(1 \leq n \leq 15)$, which is the number of rounds that will be played. On each of the next n lines, will be two integers: the roll of Antonia for that round, followed by a space, followed by the roll of David for that round. Each roll will be an integer between 1 and 6 (inclusive).

Output Specification

The output will consist of two lines. On the first line, output the number of points that Antonia has after all rounds have been played. On the second line, output the number of points that David has after all rounds have been played.

Sample Input

```
4
5 6
6 6
4 3
5 2
```

Output for Sample Input

```
94
91
```

Explanation of Output for Sample Input

After the first round, David wins, so Antonia loses 6 points. After the second round, there is a tie and no points are lost. After the third round, Antonia wins, so David loses 4 points. After the fourth round, Antonia wins, so David loses 5 points. In total, Antonia has lost 6 points and David has lost 9 points.

CCC '16 J2 - Magic Squares

Time limit: 2.0s **Memory limit:** 256M

Canadian Computing Competition: 2016 Stage 1, Junior #2

Magic Squares are square arrays of numbers that have the interesting property that the numbers in each column, and in each row, all add up to the same total.

Given a 4×4 square of numbers, determine if it is a magic square.

Input Specification

The input consists of four lines, each line having 4 space-separated integers.

Output Specification

Output either `magic` if the input is a magic square, or `not magic` if the input is not a magic square.

Sample Input 1

```
16 3 2 13
5 10 11 8
9 6 7 12
4 15 14 1
```

Output for Sample Input 1

```
magic
```

Explanation for Output for Sample Input 1

Notice that each row adds up to 34, and each column also adds up to 34.

Sample Input 2

```
5 10 1 3
10 4 2 3
1 2 8 5
3 3 5 0
```

Output for Sample Input 2

```
not magic
```

Explanation for Output for Sample Input 2

Notice that the top row adds up to 19, but the rightmost column adds up to 11.

CCC '19 J3 - Cold Compress

Time limit: 1.0s **Memory limit:** 256M

Canadian Computing Competition: 2019 Stage 1, Junior #3

Your new cellphone plan charges you for every character you send from your phone. Since you tend to send sequences of symbols in your messages, you have come up with the following compression technique: for each symbol, write down the number of times it appears consecutively, followed by the symbol itself. This compression technique is called *run-length encoding*.

More formally, a block is a substring of identical symbols that is as long as possible. A block will be represented in compressed form as the length of the block followed by the symbol in that block. The encoding of a string is the representation of each block in the string in the order in which they appear in the string.

Given a sequence of characters, write a program to encode them in this format.

Input Specification

The first line of input will contain the number N, which is the number of lines that follow. The next N lines will contain at least one and at most 80 characters, none of which are spaces.

Output Specification

Output will be N lines. Line i of the output will be the encoding of the line $i + 1$ of the input. The encoding of a line will be a sequence of pairs, separated by a space, where each pair is an integer (representing the number of times the character appears consecutively) followed by a space, followed by the character.

Sample Input

```
4
+++===!!!!
777777......TTTTTTTTTTT
(AABBC)
3.1415555
```

Output for Sample Input

```
3 + 3 = 4 !
6 7 6 . 12 T
1 ( 2 A 2 B 1 C 1 )
1 3 1 . 1 1 1 4 1 1 4 5
```

Explanation of Output for Sample Input

To see how the first message (on the second line of input) is encoded, notice that there are 3 `+` symbols, followed by 3 `=` symbols, followed by 4 `!` symbols.

CCC '15 J3 - Rövarspråket

Time limit: 2.0s **Memory limit:** 256M

Canadian Computing Competition: 2015 Stage 1, Junior #3

In Sweden, there is a simple child's game similar to Pig Latin called Rövarspråket (Robbers Language).

In the CCC version of Rövarspråket, every consonant is replaced by three letters, in the following order:

- the consonant itself;
- the vowel closest to the consonant in the alphabet (e.g., if the consonant is `d`, then the closest vowel is `e`), with the rule that if the consonant falls exactly between two vowels, then the vowel closer to the start of the alphabet will be chosen (e.g., if the consonant is `c`, then the closest vowel is `a`);
- the next consonant in the alphabet following the original consonant (e.g., if the consonant is `d`, then the next consonant is `f`) except if the original consonant is `z`, in which case the next consonant is `z` as well.

Vowels in the word remain the same. (Vowels are `a`, `e`, `i`, `o`, `u` and all other letters are consonants.) Write a program that translates a word from English into Rövarspråket.

Input Specification

The input consists of one word entirely composed of lowercase letters. There will be at least one letter and no more than 30 letters in this word.

Output Specification

Output the word as it would be translated into Rövarspråket on one line.

Sample Input 1

```
joy
```

Output for Sample Input 1

```
jikoyuz
```

Sample Input 2

```
ham
```

Output for Sample Input 2

```
hijamon
```