Problem A: Fractional Lotion

Acknowledgement: This problem is from the CTU Open Contest 2013

Freddy practices various kinds of alternative medicine, such as homeopathy. This practice is based on the belief that successively diluting some substances in water or alcohol while shaking them thoroughly produces remedies for many diseases.

This year, Freddy's vegetables appear to have caught some disease and he decided to experiment a little bit and investigate whether homeopathy works for vegetables too. As Freddy is also a big fan of mathematics, he does not strictly insist that the substances have small concentrations, but he instead requires the concentrations to be reciprocals of integers ($1/n$). In experiments, some of the vegetables really got much better.

Seeing Freddy's successes, a fellow gardener also wants to try one of these potions and asks for a flask. Freddy has one flask of the potion in concentration $1/n$ and does not want to give it all out. Your task is to find out in how many ways the potion can be split into two flasks and diluted so that the resulting potions both have the same volume as the original one and the resulting concentrations also are reciprocals of integers — we do not want to end up with useless fluid, do we?

Input Specification

Each line of the input describes one test case. The line contains the expression "`1/`n" representing the original concentration. You are guaranteed that $1 \leq n \leq 10\,000$. There are no spaces on the line.

Output Specification

For each test case, output a single line with the total number of distinct pairs $\{x, y\}$ of positive integers satisfying $1/x + 1/y = 1/n$. Pairs differing only in the order of the two numbers are not considered different.

Sample Input

```
1/2
1/4
1/1
1/5000
```

Output for Sample Input

```
2
3
1
32
```

This problem was inspired by Project Euler

Problem B: Furry Nuisance

In order to protect himself from evil bunnies, Freddy decided to install an automatic system to detect them in pictures from surveillance cameras. Sophisticated software detects important points in the picture and lines between them. Unfortunately, the terrain in the pictures is quite varied and lot of the points and lines are actually not bunnies.

You have made the following observation: Each bunny has four paws and a body joining them. Based on this observation, write a program to decide whether a given picture can possibly contain a bunny.

Input Specification

The input contains several test cases. The first line of each test case contains two integers n and m ($0 \leq n \leq 10\,000$, $0 \leq m \leq 20\,000$), giving the number of points and lines in the image, respectively. Each of the m following lines contains two distinct integers x and y ($1 \leq x, y \leq n$), indicating that the points x and y are directly joined by a line. You may assume that each pair of points is joined by at most one direct line and that no point is directly joined with itself.

Output Specification

For each input instance, output "`YES`" if the picture can contain a bunny, and "`NO`" otherwise. The picture can contain a bunny if it is possible to remove some of the points and lines so that the resulting image is connected and has exactly 4 paws.

The image is said to be connected if (and only if) each two points are joined with each other by one or more successive lines. A paw is a point which is directly joined with exactly one other point.

Sample Input

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

Output for Sample Input

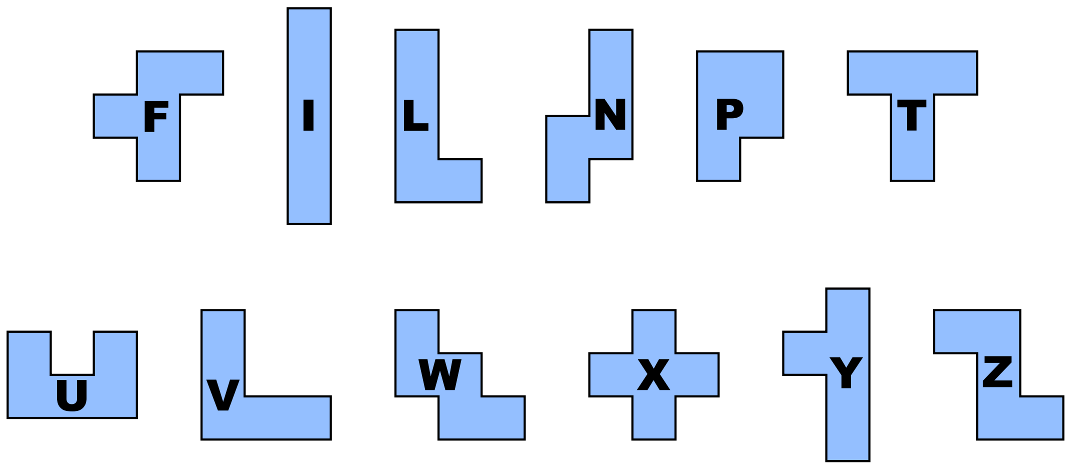
```
NO
YES
```

Problem C: Flower Pots

Freddy is planning a garden party for this weekend. Many old classmates are going to pay him a visit and Freddy is thinking about some elegant and unexpected improvement of his small horticultural oasis. Inspired by the creations in his favorite gardening journals he decided to install two flower arrangements on the opposite sides of the walkway leading to his garden. To match the other plants in the vicinity, one arrangement has to be in light bronze yellow and the other in dark pastel red color. The pots in which the flowers will be placed have to be painted with the same matching colors.

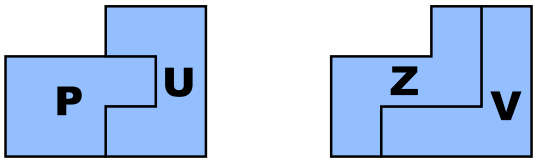
Freddy knows that it is much cheaper and faster to buy new pots than to repaint existing ones. Thus, being pressed by fast-approaching date of the party, Freddy has decided to purchase flower pots from a small company which not only can deliver pots painted with exact colors needed but they can deliver them today afternoon. The owner of the company is a Dutch artist who specializes herself in designs far from ordinary. Her very special pots are manufactured in so-called pentomino shapes and they can be flipped over so that their top side can serve as the bottom side and vice versa.

A pentomino-shaped flower pot is made of five squares welded together so that sides of any two neighboring squares always touch each other along the whole edge. In the actual pots, each square is about ten square feet, but the size does not matter for this problem, it is only important that all squares in all shapes are of the same size. There are exactly 12 possible shapes, they are listed in the figure below and each shape is traditionally named by a letter to which it bears some resemblance. (with Freddy's favorite shape being **F**, of course)



Freddy is going to buy two yellow pots and two red pots. For aesthetical reasons, he wants both arrangements to be of equal shape. Two pots of the same color will be put on the lawn closely together so that the divisions between them will not be visible and only the outline of the whole arrangement will be important for judging the equality of shapes. In the resulting arrangements, no pots can overlap.

Freddy has to choose the pots carefully because some pairs are clearly incompatible from this point of view (such as W+F and I+I), some other may be compatible but it might not be immediately obvious that they really are (such as P+U and V+Z and their possible arrangement in the figure below). Therefore, Freddy asks you for a help. You are given two yellow pots and two red pots and your program should decide if two arrangements with the same outline can be created.



Input Specification

The input contains several test cases. Each test case consists of one line. The line starts with two letters that specify the shapes of two yellow pots, then there is one space and other two letters giving the shapes of two red pots. All four letters are in uppercase and each of them is one of the 12 valid letters listed above.

Output Specification

For each test case, output a single line of text. The line should contain “YES” if an arrangement exists which can be composed from both pairs of yellow pots and red pots. If there is no such arrangement, the line should contain “NO”.

Sample Input

```
II PP
WF II
VZ UP
```

Output for Sample Input

```
YES
NO
YES
```

Problem D: Frozen Rose-Heads

The winter is coming and all the experts are warning that it will be the coldest one in the last hundred years. Freddy needs to make sure that his garden does not sustain any damage. One of the most important tasks is to make sure that no water remains in his large watering system.

All the water comes from a central node and is distributed by pipes to neighboring nodes and so on. Each node is either a sprinkler (rose head) with no outgoing pipe or an internal node with one or more outgoing pipes leading to some other nodes. Every node has exactly one incoming pipe, except for the central node which takes the water directly from a well and has no incoming pipe. Every pipe has a valve that stops all the water going through the pipe. The valves are of different quality and age, so some may be harder to close than others.

Freddy knows his valves well and has assigned a value to each pipe representing the amount of effort needed to close the corresponding valve. He asks you to help him count the minimum effort needed to close some valves so that no water goes to the sprinklers.

Input Specification

The input contains several test cases. Each test case starts with a line with two integers, the number of nodes n ($2 \leq n \leq 1\,000$), and the number of the central node c ($1 \leq c \leq n$). Each of the next $n-1$ lines represents one pipe and contains three integers, u, v ($1 \leq u, v \leq n$) and w ($1 \leq w \leq 1\,000$), where u and v are the nodes connected by a pipe and w is the effort needed to close the valve on that pipe. You may assume that every node is reachable from the central node.

Output Specification

For each test case, output a single line containing the minimum sum of efforts of valves to be closed, such that the central node gets separated from all sprinklers.

Sample Input

```
3 1
2 1 5
1 3 4
7 7
7 6 10
7 5 10
6 4 1
6 3 1
5 2 1
5 1 2
```

Output for Sample Input

```
9
5
```

This problem was adapted from Stanford Local Contest

Problem E: False Sense of Security

Freddy discovered a new procedure to grow much bigger cauliflowers. He wants to share this finding with his fellow gardener Tommy but he does not want anyone to steal the procedure. So the two gardeners agreed upon using a simple encryption technique proposed by M. E. Ohaver.

The encryption is based on the Morse code, which represents characters as variable-length sequences of dots and dashes. The following table shows the Morse code sequences for all letters:

A	`.-`	H	`....`	O	`---`	V	`...-`
B	`-...`	I	`..`	P	`.--.`	W	`.--`
C	`-.-.`	J	`.---`	Q	`--.-`	X	`-..-`
D	`-..`	K	`-.-`	R	`.-.`	Y	`-.--`
E	`.`	L	`.-..`	S	`...`	Z	`--..`
F	`..-.`	M	`--`	T	`-`		
G	`--.`	N	`-.`	U	`..-`		

Note that four possible dot-dash combinations are unassigned. For the purposes of this problem we will assign them as follows (note these are not the assignments for actual Morse code):

underscore ("_")	`..--`	period (".")	`---.`
comma (",")	`.-.-`	question mark ("?")	`----`

In practice, characters in a message are delimited by short pauses, typically displayed as spaces. Thus, the message ACM_GREATER_NY_REGION is encoded as:
`.- -.-. -- ..-- --. .-. . .- - . .-. ..-- -. -.-- ..-- .-. . --. .. --- -.`

The Ohaver's encryption scheme is based on mutilating Morse code, namely by removing the pauses between letters. Since the pauses are necessary (because Morse is a variable-length encoding that is not prefix-free), a string is added that identifies the number of dots and dashes in each character. For example, consider the message "`.--.-.--`". Without knowing where the pauses should be, this could be "`ACM`", "`ANK`", or several other possibilities. If we add length information, such as "`.--.-.-- 242`", then the code is unambiguous.

Ohaver's scheme has three steps, the same for encryption and decryption:

1. Convert the text to Morse code without pauses but with a string of numbers to indicate code lengths.
2. Reverse the string of numbers.
3. Convert the dots and dashes back into the text using the reversed string of numbers as code lengths.

As an example, consider the encrypted message “`AKADTOF_IBOETATUK_IJN`”. Converting to Morse code with a length string yields:
`.--.-.--..----..-...--..-...---.-.--..--.-..--...----. 232313442431121334242`
By reversing the numbers and decoding, we get the original message “`ACM_GREATER_NY_REGION`”.

The security of this encoding scheme is not too high but Freddy believes it is sufficient for his purposes. Will you help Freddy to implement this encoding algorithm and to protect his sensitive information?

Input Specification

The input will consist of several messages encoded with Ohaver's algorithm, each of them on one line. Each message will use only the twenty-six capital letters, underscores, commas, periods, and question marks. Messages will not exceed 1000 characters in length.

Output Specification

For each message in the input, output the decoded message on one line.

Sample Input

```
FENDSVTSLHW.EDATS,EULAY
TRDNWPLOEF
NTTTGAZEJUIIGDUZEHKUE
QEWOISE.EIVCAEFNRXTBELYTGD.
?EJHUT.TSMYGW?EJHOT
DSU.XFNCJEVE.OE_UJDXNO_YHU?VIDWDHPDJIKXZT?E
ADAWEKHZN,OTEATWRZMZN_IDWCZGTEPION
```

Output for Sample Input

```
FALSE_SENSE_OF_SECURITY
CTU_PRAGUE
TWO_THOUSAND_THIRTEEN
QUOTH_THE_RAVEN,_NEVERMORE.
TO_BE_OR_NOT_TO_BE?
THE_QUICK_BROWN_FOX_JUMPS_OVER_THE_LAZY_DOG
ADAPTED_FROM_ACM_GREATER_NY_REGION
```