



## Problem 4 – Treasure Map

Believe it or not you are just about to receive a letter from a ridiculously rich uncle of yours who happened to have lived in the very same city as you. The special thing about that letter is that it is actually a map which leads you to a secret treasure you have inherited. The map consists of list of addresses you must visit in order to get to the treasure... Though the “old fox” is not stupid so he has also written a password next to each address which you must tell when you reach there. Easy-peasy ... yeah, sure! The only problem is that your uncle’s little daughter has found the letter by accident and drew some extra symbols on it before it was sent to you. Fortunately, we have some hints for you on how to read the letter.

### Instructions

Your uncle has written everything in the middle of the paper so you need to **search** for the **most inner valid instruction**. If the matches are even number **take the second** one from the **central pair** (the one on the right).

- All valid instructions are **locked between** the symbols “#” and “!”. Each of them **can stay before or ahead** of the instruction block but they **should not be repeating!** E.g.:

**“#{instruction pattern}!” or “!{instruction pattern}#”**

Inside the locking symbols you should find:

- The **first four letter word** - the name of the street (only letters, should not be preceded or followed by a digit or a letter)
- The **last** street number - password **pair** in the format: **“{street number}-{password}”** where the **street number** consists of **3 digits** and the **password** consists of **4 OR 6 digits**

### Input

- On the first line, you will receive an **integer N**
- On the next **N** lines, you will receive **strings** that may contain **any ASCII symbol**

### Output

- You must **print** on the console **all read** instructions, **each on new line** in the format:  
**“Go to str. {address name} {street number}. Secret pass: {password}.”**

### Constraints

- The first line containing digits **won’t be longer** than **60** symbols
- Each address name - street number **pair** will be **unique**
- There will not be any **nested valid** instructions
- Time limit: 0.3 sec. Memory limit: 16 MB.

### Examples

Input	Output
2 #a asda 90a 164-1854ipo!21#w!i1p#Chen 800-	Go to str. Chen 800. Secret pass: 1989. Go to str. Sivu 387.

<p>1989!912!90!d09# #Sivu 111-198990ads387-0990!</p>	<p>Secret pass: 0990.</p>
<p>2 .`#2#asio800-1989!9#ioio 123-3211!! a aJax asd asdf asio 007-1914#+#sh shi shak 89 12 sa 980.6573! !jdmu 890-890899# ?phpjfa9f0a09!kop.uoi-haya &lt;&gt;918-3123+#\x\k;</p>	<p>Go to str. aJax 007. Secret pass: 1914. Go to str. haya 918. Secret pass: 3123.</p>
<p>3 #189-3123ahij!ds.,/21a!vnkaa pehA ;lf984-123129 ao 276-0001#shjlk!da901-13!2#!asdf i #sd.#swe.,jkas-112-0131!# oio Perl.123-1234!!jj ko staa mo staa 777-7171adfsa#io .,`213.31245.0-9gFk!!!!1-1####flks0--+\&lt;!@AsfA kit 444-444111#1&gt;&gt;#oppo154-2017ak!jik tak</p>	<p>Go to str. pehA 276. Secret pass: 0001. Go to str. Perl 123. Secret pass: 1234. Go to str. AsfA 444. Secret pass: 444111.</p>