Problem 11 – Little John

As you probably know Little John is the right hand of the famous English hero - Robin Hood. A little known fact is that Little John can't handle Math very well. Before Robin Hood left to see Marry Ann, he asked John to **count** his hay of arrows and send him an **encrypted** message containing the arrow's count**.** The message should be encrypted since it can be intercepted by the Nottingham's evil Sheriff. Your task is to help Little John before it is too late (0.10 sec).

You are given **4 input** strings (hay). Those strings **may or may not** contain arrows. The arrows can be of different type as follows:

- ">----->" – a small arrow
- ">>----->" – a medium arrow
- ">>>----->>" – a large arrow

Note that the **body** of each arrow will always be **5 dashes long**. The **difference** between the arrows is in their **tip** and **tail**. The given 3 types are the only ones you should count, the **rest should be ignored** (Robin Hood does not like them). You should start searching the hays **from the largest** arrow type down **to the smallest** arrow type.

After you find the **count** of each arrow type you should **concatenate** them into one number in order: small, medium, large arrow (even if the arrow count is 0). Then you **convert** the number in **binary** representation, **reverse** it and **concatenate it again** with the initial binary representation of the number. You **convert** the final binary number again **back to decimal**. This is the encrypted message you should send to Robin Hood.

Input

The input will be read from the console. The **data** will be received from 4 input **lines containing strings**.

Output

The output should be a decimal number, representing the encrypted count of arrows.

Constraints

- The input strings will contain any ASCII character.
- Allowed working time: 0.1 seconds. Allowed memory: 16 MB.

Examples

Input	Output
>>>----->>abc>>----->> >>>----->> >----->s >>----->	14535 *The count is: 1 small, 1 medium and 3 large arrows* *113(dec) = 1110001(bin) -> reversed is 1000111(bin)* *1110001 1000111(bin) = 14535(dec)*