# Problem 3 – Ascent

Doctor T. likes to communicate with encrypted messages with his lackeys. But the lackeys are relatively stupid, so you need to help them understand the Doctor’s messages, by decrypting his encrypted data.

You have been tasked to decrypt several messages. You will be given the messages as several input lines. You must read and decrypt messages until you receive the command “**Ascend**”.

When you receive a given message you must **decrypt** it and **print** it **immediately**. The messages have several encoded strings in them, which you must find. The encoded string consist of – a **single comma** (**,**) or an **underscore** (**\_**), a **sequence of English Alphabet letters**, and a **digit** at the end of it. **Examples:** “**,asd9**”, “**\_z8**”, “**,xzc9**“.

The decoding of a message is **done** by **adding** or **subtracting** the **digit** in it, from the **ASCII** **code** of **every character** in the **sequence of letters**. If the front character is a **comma** (**,**) you **add** the **digit** to the **ASCII codes** of the characters, if it’s an **underscore** (**\_**), you **subtract** it.

When you find an **encoded string**, you must “**memorize**” it. In other words, you find the encoded string “**\_qlfh5**”,   
and decode it – it results in “**lgac**”. If you find the same encoded string (**\_qlfh5**), **again**, in a message **afterwards**, you must **replace** it with the **decoded value** corresponding to it (**lgac**), **BEFORE** beginning to decode the message itself.   
**Example:** “**abcdefg,\_qlfh52hijklmn**” = “**abcdefg,lgac2hijklmn**” = “**abcdefgnicehijklm**”.

You know that “**\_qlfh5**” is “**lgac**”, so the message would turn into “**abcedfg,lgac2hijklmn**” which after decoding turns into “**abcdefgnicehijklm**”.

### Input

* The input comes in the form of input lines containing the encoded messages.
* When you receive the command “**Ascend**” the input sequence ends.

### Output

* You must print every message, **immediately** after you’ve decoded it.

### Constrains

* The input lines may consist of any **ASCII character**.
* Upon **replacing** with **memorized** patterns, you must do it in **the order** in which you’ve **found the patterns**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| ,plcqrkf3 \_twn2z <3  Ascend | softuni rulz <3 |
| This \_vet4 is a ,I2\_iller0  Ya ,eswx2 listenin \_zu6 \_Vjl9 ,,I22iller  Ascend | This rap is a Killer  Ya guyz listenin to Mac Miller |