## \*Winning Ticket

Lottery is exciting. What is not, is checking a million tickets for winnings only by hand. So, you are given the task to create a program which automatically checks if a ticket is a winner.

You are given a **collection of tickets separated by commas and spaces**. You need to check every one of them if it has a winning combination of symbols.

**A valid ticket should have exactly 20 characters**. The winning symbols are '**@**', '**#**', '**$**' and '**^**'. But in order for a ticket to be a winner the symbol should uninterruptedly repeat for at least **6 times** in both the **tickets left half** and the **tickets right half**.

For example, a valid winning ticket should be something like this:

"**Cash$$$$$$Ca$$$$$$sh**"

The left half "**Cash$$$$$$**" contains "**$$$$$$**", which is also contained in the tickets right half "**Ca$$$$$$sh**". A winning ticket should contain symbols repeating up to 10 times in both halves, which is considered a Jackpot (for example: "**$$$$$$$$$$$$$$$$$$$$**").

**Input**

The input will be read from the console. The input consists of a **single line** containing all tickets **separated by commas and one or more white spaces** in the format:

* **"{ticket}, {ticket}, … {ticket}"**

**Output**

Print the result for every ticket in the order of their appearance, each on a separate line in the format:

* **Invalid ticket - "invalid ticket"**
* **No match - "ticket "{ticket}" - no match"**
* **Match with length 6 to 9 - "ticket "{ticket}" - {match length}{match symbol}"**
* **Match with length 10 - "ticket "{ticket}" - {match length}{match symbol} Jackpot!"**

**Constrains**

* Number of tickets will be in range [0 … 100]

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| Cash$$$$$$Ca$$$$$$sh | ticket "Cash$$$$$$Ca$$$$$$sh" - 6$ |
| $$$$$$$$$$$$$$$$$$$$, aabb , th@@@@@@eemo@@@@@@ey | ticket "$$$$$$$$$$$$$$$$$$$$" - 10$ Jackpot!  invalid ticket  ticket "th@@@@@@eemo@@@@@@ey" - 6@ |
| validticketnomatch:( | ticket "validticketnomatch:(" - no match |