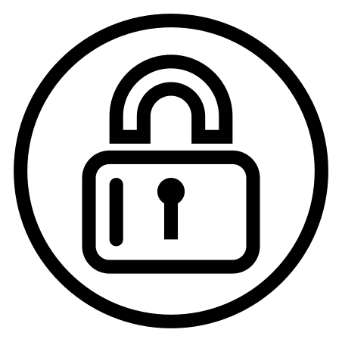
# Problem 2 - Encrypting Password



Create a program, that **checks** if **inputs** are a **valid password** and **encrypt** it. On the **first** line you will **receive** a **number** that **indicates** how **many** **inputs** you will **receive** on the **next** lines**.**

A password is **valid** when:

* It **starts** with a **group** of  **symbols** and **ends** with the **same symbols (the same length) - All symbols are possible**
* There is a **greater than sign (">")** after the first group and a **less than sign ("<")** before the last one
* In between the greater than sign and the less than sign there are **four** **groups** (each of **three** characters), separated by pipe ("**|**")
  + The first group consists only of **numbers**
  + The second group – only **lower case letters**
  + The third one – only **upper case letters**
  + The fourth one – all **symbols except "<" and ">"**

**Example for a valid message:**

**"$$$>312|dfe|KFE|@!#<$$$"**

You must **check** if the **password** is **valid** and if it **is** - **encrypt** it, if it **isn’t** - **print** the following **message**:

**"Try another password!"**

**Encrypting** a **password** means to **take** **all** **numbers, letters and symbols from the middle four groups** and **concatenatе** them. After successful encrypt, print it in the following format:

**"Password: {encrypted password}"**

## Input

* On the **first** line - **n** - the count of inputs.
* On the **next** **n** lines - **input** that you have to **check** if it has a **valid** **password**.

## Output

* Print all results from each input, each on a new line.

## Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| 3  ##>00|no|NO|!!!?<###  ##>123|yes|YES|!!!<##  $$<111|noo|NOPE|<<>$$ | Try another password!  Password: 123yesYES!!!  Try another password! | The first one doesn’t start and end with the same amount of '#' and the count of characters in each group is different than 3. The second one is correct. The third one uses the wrong '<' and '>' and the group containing "<<" can contain everything except '<' and '>'. |
| 5  aa>111|mqu|BAU|mqu<aa  ()>111!aaa!AAA!^&\*<()  o>088|abc|AAA|\*\*\*<o  asd>asd|asd|ASD|asd<asd  \*>088|zzzz|ZzZ|123<\* | Password: 111mquBAUmqu  Try another password!  Password: 088abcAAA\*\*\*  Try another password!  Try another password! |  |

## JS Examples

The input will be provided as an array of strings.

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| (["3",  "##>00|no|NO|!!!?<###",  "##>123|yes|YES|!!!<##",  "$$<111|noo|NOPE|<<>$$"]) | Try another password!  Password: 123yesYES!!!  Try another password! | The first one doesn’t start and end with the same amount of '#' and the count of characters in each group is different than 3. The second one is correct. The third one uses the wrong '<' and '>' and the group containing "<<" can contain everything except '<' and '>'. |
| (["5",  "aa>111|mqu|BAU|mqu<aa",  "()>111!aaa!AAA!^&\*<()",  "o>088|abc|AAA|\*\*\*<o",  "asd>asd|asd|ASD|asd<asd",  "\*>088|zzzz|ZzZ|123<\*"]) | Password: 111mquBAUmqu  Try another password!  Password: 088abcAAA\*\*\*  Try another password!  Try another password! |  |