02. Message Decrypter

Create a program that checks if inputs have a valid message and decrypt it. On the first line, you will receive a number that indicates how many inputs you will receive on the following lines.

A message is valid when:

* There is nothing else before and after it
* It starts with a tag, which is surrounded by either "$" or "%" (but not both at the same time). The tag itself has to be minimum 3 characters long, start with an uppercase letter, followed only by lowercase letters
* There is a colon and a single white space after the tag
* There are 3 groups consisting of numbers between "[" and "]", followed by a pipe ("|")

Example for a valid message:

"$Request$: [73]|[115]|[32]|"

You must check if the message is valid, and if it is - decrypts it. If it isn't - print the following message:

"Valid message not found!"

Decrypting a message means taking all numbers and turn them into ASCIT symbols. After successful decrypt, print it in the following format:

"{tag}: {decryptedMessage}"

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 message.

Output

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

Input

Example for a valid message:

"$Request$: [73]|[115]|[32]|"

Output

· The possible outputs are:

* "{tag}: {decryptedMessage}"
* "Valid message not found!"

Вход

4

$Request$: [73]|[115]|[105]|

%Taggy$: [73]|[73]|[73]|

%Taggy%: [118]|[97]|[108]|

$Request$: [73]|[115]|[105]|[32]|[75]|

Изход

Request: Isi

Valid message not found!

Taggy: val

Valid message not found!

Вход

3

This shouldnt be valid%Taggy%:

[118]|[97]|[108]|

$tAGged$: [97][97][97]|

$Request$: [73]|[115]|[105]|true

Изход

Valid message not found!

Valid message not found!

Valid message not found!