

You are updating the username policy on your company's internal networking platform. According to the policy, a username is considered valid if all the following constraints are satisfied:

- The username consists of **8** to **30** characters inclusive. If the username consists of less than **8** or greater than **30** characters, then it is an invalid username.
- The username can only contain alphanumeric characters and underscores (`_`). Alphanumeric characters describe the character set consisting of *lowercase* characters [`a – z`], *uppercase* characters [`A – Z`], and digits [`0 – 9`].
- The *first* character of the username must be an *alphabetic* character, i.e., either *lowercase* character [`a – z`] or *uppercase* character [`A – Z`].

For example:

Username	Validity
<b>Julia</b>	INVALID; Username length < 8 characters
<b>Samantha</b>	VALID
<b>Samantha_21</b>	VALID
<b>1Samantha</b>	INVALID; Username begins with non-alphabetic character
<b>Samantha?10_2A</b>	INVALID; '?' character not allowed

Update the value of *regularExpression* field in the *UsernameValidator* class so that the regular expression only matches with valid usernames.

### Input Format

The first line of input contains an integer *n*, describing the total number of usernames. Each of the next *n* lines contains a string describing the username. The locked stub code reads the inputs and validates the username.

### Constraints

- $1 \leq n \leq 100$
- The username consists of any printable characters.

### Output Format

For each of the usernames, the locked stub code prints `Valid` if the username is valid; otherwise `Invalid` each on a new line.

### Sample Input 0

```
8
Julia
Samantha
Samantha_21
1Samantha
Samantha?10_2A
JuliaZ007
Julia@007
_Julia007
```

### Sample Output 0

```
Invalid
Valid
Valid
Invalid
Invalid
Valid
Invalid
Invalid
```

### Explanation 0

Refer diagram in the challenge statement.

