[B] The Rotating Digits

Program:	`rotate.(cpp|java)`
Input:	`rotate.in`
Balloon Color:	`Yellow`

Description

A single rotation of digits in an integer number would remove the last digit and place it as the first digit, while shifting all other digits one position to the right. For example, if the integer is 12345, a single rotation would make the integer 51234, and two rotations would make it 45123.

Given an integer number consisting of exactly three digits, and a number of rotations, your task is to create the integer formed by the number of rotations given.

Input Format

The input starts with a number T ($1 \leq T \leq 1000$) that represents the number of test cases in the file. Each test case consists of a line that contains two integer, the initial number N ($100 \leq N \leq 999$), and the number of rotations R ($0 \leq R \leq 100$).

Output Format

The output for each test case is in this form:

`k. ans`

where ***k*** represents the test case number (starting at 1), and ***ans*** is the integer number formed after the rotations. Note that the digit 0 needs to be maintained when rotating, for example, rotating the integer 100 once yields a number that is represented as 010.

Sample Input / Output

`rotate.in`

```
2
123 1
489 2
```

`OUTPUT`

```
1. 312
2. 894
```