Problem L. Tower of Hanoi Movement - Easy

Time limit	1500 ms
Mem limit	1572864 kB
Code length Limit	50000 B
OS	Linux

The harder version of this problem can be found here

The Tower of Hanoi (also called the Tower of Brahma or Lucas' Tower and sometimes pluralized) is a mathematical game or puzzle. It consists of three rods and a number of disks of different sizes, which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.

The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

1. Only one disk can be moved at a time.
2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack.
3. No disk may be placed on top of a smaller disk.

With 3 disks, the puzzle can be solved in 7 moves. The minimal number of moves required to solve a Tower of Hanoi puzzle is $2^N - 1$, where n is the number of disks.

The description is retrieved from Wikipedia

From the description above, AVM wants to know the a^{th} step of optimal solution. The set of Tower of Hanoi consists of N disks and 3 rods (A as the source rod, B as the spare rod, and C as the target rod).

Input

The input file consists of several lines. The first line contains a single number T representing the number of test cases. The next T lines contains N and a representing the number of disk and the a^{th} move.

Output

The output file should contains T lines. The i-th line should contain $P : A => C$, the P^{th} disk, the rod of P^{th} disk before the movement, and the rod of P^{th} disk after the movement.

Constraint

$1 <= T <= 100$

$1 <= N <= 20$

$1 <= a <= 2^N - 1$

Example

Input	Output
3 2 3 3 5 3 7	1 : B => C 1 : B => A 1 : A => C

Explanation

The 2-disks Tower of Hanoi optimal solution is:

```
1 : A => B
2 : A => C
1 : B => C
```

Therefore, the first test case answer is

```
1 : B => C
```

The 3-disks Tower of Hanoi optimal solution is:

```
1 : A => C
2 : A => B
1 : C => B
3 : A => C
1 : B => A
2 : B => C
1 : A => C
```

Therefore, the second test case answer is

```
1 : B => A
```

and the last test case answer is

```
1 : A => C
```