

7401 Binary Tree

The Old Frog King lives on the root of an infinite tree. According to the law, each node should connect to exactly two nodes on the next level, forming a full binary tree.

Since the king is professional in math, he sets a number to each node. Specifically, the root of the tree, where the King lives, is 1. Say $f_{root} = 1$.

And for each node u , labels as f_u , the left child is $f_u \times 2$ and right child is $f_u \times 2 + 1$. The king looks at his tree kingdom, and feels satisfied.

Time flies, and the frog king gets sick. According to the old dark magic, there is a way for the king to live for another N years, only if he could collect exactly N soul gems.

Initially the king has zero soul gems, and he is now at the root. He will walk down, choosing left or right child to continue. Each time at node x , the number at the node is f_x (remember $f_{root} = 1$), he can choose to increase his number of soul gem by f_x , or decrease it by f_x .

He will walk from the root, visit exactly K nodes (including the root), and do the increasement or decrease as told. If at last the number is N , then he will succeed.

Noting as the soul gem is some kind of magic, the number of soul gems the king has could be negative.

Given N , K , help the King find a way to collect exactly N soul gems by visiting exactly K nodes.

Input

First line contains an integer T , which indicates the number of test cases.

Every test case contains two integers N and K , which indicates soul gems the frog king want to collect and number of nodes he can visit.

Restrictions:

- $1 \leq T \leq 100$.
- $1 \leq N \leq 10^9$.
- $N \leq 2^K \leq 2^{60}$.

Output

For every test case, you should output 'Case # x :' first, where x indicates the case number and counts from 1.

Then K lines follows, each line is formatted as ' $a b$ ', where a is node label of the node the frog visited, and b is either '+' or '-' which means he increases / decreases his number by a .

It's guaranteed that there are at least one solution and if there are more than one solutions, you can output any of them.

Sample Input

```
2
5 3
10 4
```

Sample Output

Case #1:

1 +

3 -

7 +

Case #2:

1 +

3 +

6 -

12 +