Code Jam 2013 - Round 2

Many Prizes

Problem

We're going to run a tournament with $2^{\mathbf{N}}$ teams, and give out **P** identical prizes to the teams with ranks `0..`**`P`**`-1`.

The teams are numbered 0 through $2^{\mathbf{N}}$-1. When team i and team j play against each other in a game, team i will win iff `i<j`.

The teams for a tournament are organized in some order, called the tournament's *tournament list*, which contains all $2^{\mathbf{N}}$ teams in the tournament. The tournament list will affect which teams play each other, and in what order.

Your job will be to find the largest-numbered team that is *guaranteed* to win a prize, independent of how the tournament list is ordered; and to find the largest-numbered team that *could* win a prize, depending on how the tournament list is ordered.

Tournament Resolution

The tournament is conducted in **N** *rounds*.

Each team has a *record*: the list of the results of the games it has played so far. For example, if a team has played three games, and won the first, lost the second and won the third, its record is `[W, L, W]`. If a team has played zero games, its record is `[]`.

In each round, every team plays a game against a team with the same record. The first team in the tournament list with a particular record will play against the second team with that record; the third team with the same record will play against the fourth; and so on.

After **N** rounds, each team has a different record. The teams are ranked in reverse lexicographical order of their records; so `[W, W, W] > [W, W, L] > [W, L, W] ... > [L, L, L]`.

Here is an example of a tournament with **N**=3, and the tournament list `[2, 4, 5, 3, 6, 7, 1, 0]`, where the columns represent different rounds, and the teams are grouped by their records. The winner of each game in the example has been marked with a `*`.

```
Round 1     Round 2     Round 3     Final Result
                                    (best rank at top)
[]          [W]         [W,W]
2  *        2  *        2           0  [W,W,W]
4           3           0  *        2  [W,W,L]
                        [W,L]
5           6           3  *        3  [W,L,W]
3  *        0  *        6           6  [W,L,L]
            [L]         [L,W]
6  *        4  *        4           1  [L,W,W]
7           5           1  *        4  [L,W,L]
                        [L,L]
1           7           5  *        5  [L,L,W]
0  *        1  *        7           7  [L,L,L]
```

If we give out 4 prizes (**N**=3, **P**=4), the prizes will go to teams 0, 2, 3 and 6.

The largest-numbered team that was guaranteed to win a prize with **N**=3, **P**=4, independent of the order of the tournament list, was team `0`: this tournament list demonstrated that it's possible for team `1` *not* to win a prize, and it turns out that team `0` will always win one, regardless of the order of the tournament list.

The largest-numbered team that could win a prize with **N**=3, **P**=4, depending on how the tournament list was ordered, was team `6`: this tournament list demonstrated that it's possible for team `6` to win a prize, and it turns out that team `7` will never win one, regardless of the order of the tournament list.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case consists of two space-separated integers: **N**, which indicates the tournament has $2^{\mathbf{N}}$ teams, and **P**, the number of prizes.

Output

For each test case, output one line containing "Case #x: y z", where x is the case number (starting from 1), y is the largest-numbered team that is *guaranteed* to win a prize, independent of how the tournament list is ordered; and z is the largest-numbered team that *could* win a prize, depending on how the tournament list is ordered.

Limits

Time limit: 30 seconds per test set.
Memory limit: 1GB.
$1 \leq \mathbf{T} \leq 100$.
$1 \leq \mathbf{P} \leq 2^{\mathbf{N}}$.

Small dataset (Test set 1 - Visible)

$1 \leq \mathbf{N} \leq 10$.

Large dataset (Test set 2 - Hidden)

$1 \leq \mathbf{N} \leq 50$.

Sample

Sample Input

```
3
3 4
3 5
3 3
```

Sample Output

```
Case #1: 0 6
Case #2: 2 6
Case #3: 0 4
```