Code Jam 2018 - Practice Session

Bathroom Stalls

Problem

A certain bathroom has **N** + 2 stalls in a single row; the stalls on the left and right ends are permanently occupied by the bathroom guards. The other **N** stalls are for users.

Whenever someone enters the bathroom, they try to choose a stall that is as far from other people as possible. To avoid confusion, they follow deterministic rules: For each empty stall S, they compute two values L_S and R_S, each of which is the number of empty stalls between S and the closest occupied stall to the left or right, respectively. Then they consider the set of stalls with the farthest closest neighbor, that is, those S for which $\min(L_S, R_S)$ is maximal. If there is only one such stall, they choose it; otherwise, they choose the one among those where $\max(L_S, R_S)$ is maximal. If there are still multiple tied stalls, they choose the leftmost stall among those.

K people are about to enter the bathroom; each one will choose their stall before the next arrives. Nobody will ever leave.

When the last person chooses their stall S, what will the values of $\max(L_S, R_S)$ and $\min(L_S, R_S)$ be?

Input

The first line of the input gives the number of test cases, **T**. **T** lines follow. Each line describes a test case with two integers **N** and **K**, as described above.

Output

For each test case, output one line containing `Case #x: y z`, where `x` is the test case number (starting from 1), `y` is $\max(L_S, R_S)$, and `z` is $\min(L_S, R_S)$ as calculated by the last person to enter the bathroom for their chosen stall S.

Limits

1 ≤ **T** ≤ 100.
1 ≤ **K** ≤ **N**.
Time limit: 60 seconds per test set.
Memory limit: 1GB.

Small Dataset 1 (Test set 1 - Visible)

1 ≤ **N** ≤ 1000.

Small Dataset 2 (Test set 2 - Visible)

1 ≤ **N** ≤ 10^6.

Large Dataset (Test set 3 - Hidden)

$1 \le \mathbf{N} \le 10^{18}$.

Sample

Sample Input	Sample Output
5 4 2 5 2 6 2 1000 1000 1000 1	Case #1: 1 0 Case #2: 1 0 Case #3: 1 1 Case #4: 0 0 Case #5: 500 499

In Sample Case #1, the first person occupies the leftmost of the middle two stalls, leaving the following configuration (`O` stands for an occupied stall and `.` for an empty one): `O.O..O`. Then, the second and last person occupies the stall immediately to the right, leaving 1 empty stall on one side and none on the other.

In Sample Case #2, the first person occupies the middle stall, getting to `O..O..O`. Then, the second and last person occupies the leftmost stall.

In Sample Case #3, the first person occupies the leftmost of the two middle stalls, leaving `O..O...O`. The second person then occupies the middle of the three consecutive empty stalls.

In Sample Case #4, every stall is occupied at the end, no matter what the stall choices are.

In Sample Case #5, the first and only person chooses the leftmost middle stall.