Question One [50 marks]

File names

- Use loadshed.c if you are writing your program in C.
- Use loadshed.cpp if you are writing your program in C++.
- Use loadshed.java if you are writing your program in Java.
  Note that case matters.

Problem Description

Eskom has adopted a new strategy where they provide electricity to customers via a series of **power stations** arranged in a chain.
All power originates from Eskom's main supply and passes sequentially through the stations:
$S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow \ldots \rightarrow S_n$.

Each station has two possible states:

- **ON** – the station allows power to flow through to the next station.
- **OFF** – the station stops power from flowing through.

Initially, **all stations are OFF**, meaning only the first station (S_0) receives power from Eskom, and your house (plugged into the Nth station) has **no power**.

When a **load-shedding announcement** is made, each station currently **receiving power** will **toggle its state** (switching ON → OFF or OFF → ON).
Power will then flow through the chain according to these new states.

The announcements happen multiple times.
You need to determine whether your house (plugged into the Nth station) has power after the Kth announcement.

Your house has electricity **if and only if** all the stations between Eskom and your house (including your own) are in the ON state and are receiving power.

Example

Consider **N = 2** (two stations) and **K = 3** announcements:

Step	**Stations (S_1, S_2)**	**Description**
Start	OFF, OFF	Only S_0 has power
After 1st announcement	ON, OFF	S_1 toggles ON and gives power to S_2
After 2nd announcement	OFF, ON	S_1 toggles OFF (cuts power to S_2)
After 3rd announcement	ON, ON	S_1 toggles ON again and powers S_2

Since both S_1 and S_2 are ON after 3 announcements, your house **has power**.
Therefore, the output is ON.

Input and Output

Program input and output will make use of **file I/O**.

- Input is read from a file named **ts1_input.txt**, where each line contains two integers **N** and **K**, separated by a space.
- Output must be written to a file named **output.txt**, where each line corresponds to one test case, in the format:

Case #x: ON

or

Case #x: OFF

Sample Input (ts1_input.txt)

2 3

2 2

1 1

Sample Output (output.txt)

Case #1: ON

Case #2: OFF

Case #3: ON

Constraints

- $1 \le N \le 10$
- $0 \le K \le 10^{18}$
- There may be multiple test cases (up to 1000 lines).
- You must use **arbitrary precision integers** (e.g. BigInteger in Java) since K can be large.