Problem 6: Numbering Roads

(Medium)

(Adapted from UVa 11723)

In my country, streets don't have names, each of them is just given a number as a name. These numbers are supposed to be unique but that is not always the case. The local government allocates some integers to name the roads and in many cases the number of integers allocated is less that the total number of roads. In that case to make road names unique some single character suffixes are used. So roads are named as 1, 2, 3, 1A, 2B, 3C, etc. Of course the number of suffixes is also always limited to 26 (A, B, . . . , Z). For example if there are 4 roads and 2 different integers are allocated for naming then some possible assignments of names can be:

No. 1	No. 2	No. 3	No. 4
1	2	1A	2B
1	2	1A	2C
3	4	3A	4A
1	2	1B	1C

Given the number of roads (R) and the numbers of integers allocated for naming (N), your job is to determine the minimum number of different suffixes that will be required (of all possible namings) to name the streets, assuming that no two streets can have the same name.

Input Format

The input file contains some lines of input. Each line contains two integers R and N in that order. Here R is the total number of streets to be named and N denotes the number of integers allocated for naming.

End of input is determined by two consecutive zeros (`0 0`) as input.

Constraints

- The input file can contain up to 10002 lines of inputs.
- $1 \leq N, R \leq 10^4$

Output Format

For each line of input produce one line of output. This line contains the serial of output followed by an integer D which denotes the minimum number of suffixes required to name the streets. If it is not possible to name all the streets, print `impossible` instead.

Sample Input

```
8 5
100 2
0 0
```

Sample Output

```
Case 1: 1
Case 2: impossible
```