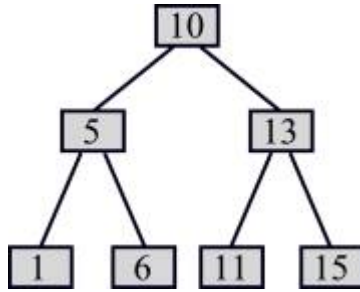BST (Binary Search Tree) is an efficient data structure for searching. In a BST all the elements of the left sub-tree are smaller and those of right sub-tree are greater than the root. A typical example of BST is

Normally, we construct BST by successively inserting an element. In that case, the ordering of elements has great impact on the structure of the tree. Look at the following cases:

In this problem, you have to find the order of 1 to N integers such that the BST constructed by them has height of at most H. The height of a BST is defined by the following relation

1. BST having no node has height 0.

2. Otherwise, it is equal to the maximum of the height of the left sub-tree and right sub-tree plus 1.

Again, several orderings can satisfy the criterion. In that case we prefer the sequence where smaller numbers come first. For example, for $N = 4$, $H = 3$ we want the sequence 1 3 2 4 rather than 2 1 4 3 or 3 2 1 4.

Input

Each test case starts with two positive integers N ($1 \le N \le 10000$) and H ($1 \le H \le 30$). Input is terminated by $N = 0$, $H = 0$. This case should not be processed. There can be at most 30 test cases.

Output

Output of each test case should consist of a line starting with 'Case #:   ' where # is the test case number. It should be followed by the sequence of N integers in the same line. There must not be any trailing space at the end of the line. If it is not possible to construct such tree then print Impossible. (without the quotes).

Sample Input

```
4 3
4 1
6 3
0 0
```

Sample Output

```
Case 1: 1 3 2 4
Case 2: Impossible.
Case 3: 3 1 2 5 4 6
```