C. One Move from Towers of Hanoi

For this problem, we are concerned with the classic problem of Towers of Hanoi. In this problem there are three posts and a collection of circular disks. Let's call the number of disks *n*. The disks are of different sizes, with no two having the same radius, and the one main rule is to never put a bigger disk on top of a smaller one. We will number the disks from 1 (smallest) to *n* (biggest) and name the posts A, B, and C. If all the disks start on post A, and the goal is to move the disks to post C by moving one at a time, again, never putting a bigger one on top of a smaller one, there is a well-known solution that recursively calls for moving *n*–1 disks from A to B, then directly moves the bottom disk from A to C, then recursively calls for moving the *n*–1 disks from B to C.

Pseudocode for a recursive solution to classic Towers of Hanoi problem:

```
move(num_disks, from_post, spare_post, to_post)
  if (num_disks == 0)
    return
  move(num_disks - 1, from_post, to_post, spare_post)
  print ("Move  disk ", num_disks, " from ",
    from_post, " to ", to_post)
  move(num_disks - 1, spare_post, from_post, to_post)
```

The problem at hand is determining the k^{th} move made by the above algorithm for a given *k* and *n*.

Input:

Input will be two integers per line, *k* and *n*. End of file will be signified by a line with two zeros. All input will be valid, *k* and *n* will be positive integers with *k* less than 2^n so that there is a k^{th} move, and *n* will be at most 60 so that the answer will fit in a 64-bit integer type.

Output:

Output the requested k^{th} move made by the above algorithm. Follow this format exactly: "`Case`", one space, the case number, a colon and one space, and the answer for that case given as the number of the disk, the name of the from_post, and the name of the to_post with one space separating the parts of the answer. Do not print any trailing spaces.

Sample Input	Sample Output
1 3 5 3 8 4 0 0	Case 1: 1 A C Case 2: 1 B A Case 3: 4 A C