Illustration Cropping

Professor Tibor von Minerale is preparing to give a lecture on balanced binary search trees. Recall that these are binary trees with two properties:

- *Balanced tree:* For every node, the height of its left subtree and the height of its right subtree differ by at most 1. For instance in Figure 1, the left and right subtrees of node 7 have heights 2 and 1, respectively. If a node does not have a left (or right) subtree then that subtree is considered to have height 0.
- *Search tree:* Each node has a value. The value of a node is greater than all the values in the left subtree of the node, and smaller than all the values in the right subtree of the node. For instance in Figure 1, the left subtree of node 7 contains the values 4, 5 and 6 which are all smaller than 7.

Tibor wants to include an illustration of a balanced binary search tree on his lecture slides. He got a picture of such a tree from a colleague. This tree has n nodes with the values 1 to n. However, it turns out to be too big to fit on his lecture slides so he would like to make it smaller. In particular, he would like to erase some nodes from the tree such that it has exactly k remaining nodes. Whenever he erases a node, he also erases the subtrees of that node. Of course, the resulting tree must still be a balanced binary search tree.

For pedagogical purposes, Tibor would like the node values in his final tree to be small. Therefore, he wants the list of the k remaining node values to be the lexicographically smallest possible. For example he would prefer a tree containing values $2, 5, 9$ over a tree containing values $2, 6, 7$.

As Tibor is far too busy doing more important things, he tasks you of finding which nodes to erase.

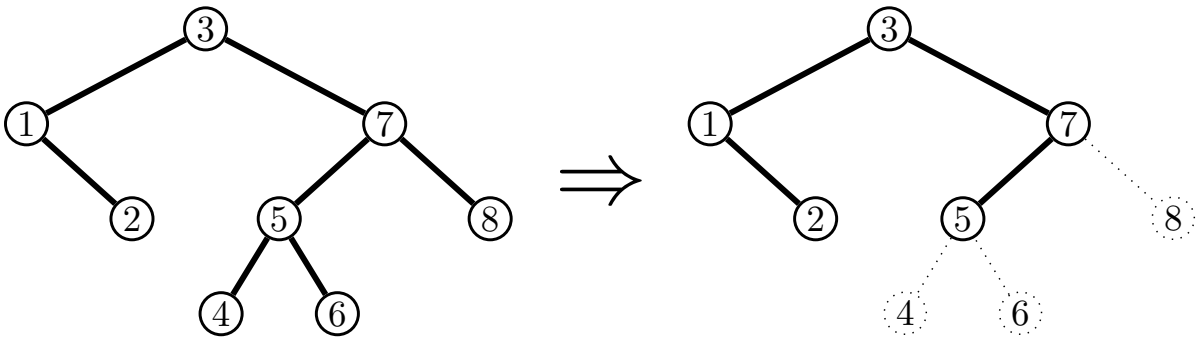

Figure 1: Illustration of Example 2 and its solution.

Input

The input consists of:

- One line with two integers n and k ($2 \leq n \leq 500\,000$, $1 \leq k \leq n-1$), the number of nodes in the tree and the number of nodes to keep.
- n lines, the i-th of which contains an integer p_i ($1 \leq p_i \leq n$ or $p_i = -1$), the parent of the node with value i or -1 if the node with value i is the root.

It is guaranteed that the given tree is a balanced binary search tree.

Output

Output a single line with a binary string of length n. The i-th character should be '1' if the node with value i should be kept, and '0' if it should be erased.

Examples

input	output
3 1 2 -1 2	010
8 5 3 1 -1 5 7 5 3 7	11101010