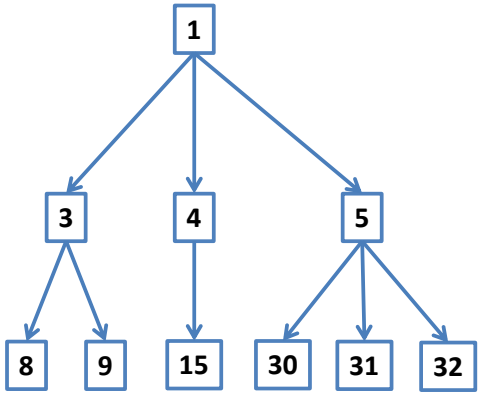
Problem 3: Count Your Cousins

Source file: `cousins.(cpp|java)`
Input file: `cousins.in`
Output file: `cousins.out`

A tree can be formed from a strictly increasing sequence of integers as follows:

- The first integer in the sequence is the root of the tree
- The next set of *consecutive* integers in the sequence describes the children of the root. The first of these will be greater than root+1.
- From there, each set of *consecutive* integers describes the children of the lowest numbered node that does not yet have children.
- Non-consecutive integers mark a break between one set of children and the next

For example, the sequence: 1 3 4 5 8 9 15 30 31 32 would produce the family tree:



Two nodes are considered to be *cousins* if they have different parents, but the same grandparent. Given a tree and a particular node of that tree, count the number of cousins of the node. For example, node 15 has 5 cousins, but node 4 has no cousins.

Input File

There will be multiple test cases in the input. Each test case will consist of two lines.

The first line will contain two integers, n ($1 \le n \le 1,000$) and k ($1 \le k \le 1,000,000$), where n is the number of nodes in the tree, and k is the particular node of interest. On the following line will be n integers, all in the range from 1 to 1,000,000, and guaranteed to be strictly increasing. These n integers describe the tree, in the manner described above. The value k is guaranteed to be one of the n integers. End of input will be indicated by a line with two 0s.

Output

For each test case, output a single integer, indicating the number cousins of node k.

Sample Input File

```
10 15
1 3 4 5 8 9 15 30 31 32
12 9
3 5 6 8 9 10 13 15 16 22 23 25
10 4
1 3 4 5 8 9 15 30 31 32
0 0
```

Sample Output

```
5
1
0
```