Bintree

Submit solution

All submissions
Best submissions

Points: 100 (partial)
Time limit: 1.0s
Memory limit: 256M

Author:
admin

Problem type

Allowed languages
C, C++

Problem: Symmetric Numbering of a Complete Binary Tree

You are given a complete binary tree with `n` vertices. The tree has the following properties:

- It has exactly one root.
- Every node is either a **leaf** (has no children) or an **internal node** (has exactly two children).
- All **leaves are at the same depth**.
- The number `n + 1` is a power of 2.

The tree is numbered **symmetrically**:

- For each subtree, the left subtree is numbered first recursively.
- Then the current root is numbered.
- Finally, the right subtree is numbered recursively.

This is essentially the **inorder traversal** of the tree.

Movement Instructions

You are given a starting node and a string `s` which consists of instructions:

- `'L'`: move to the **left child**
- `'R'`: move to the **right child**
- `'U'`: move to the **parent**

If a move cannot be executed (e.g., going left from a leaf or going up from the root), that move is skipped.

- A line with `n x`
- and a line containing string s
  - `n` : number of nodes in the tree, and `n + 1` is a power of 2
  - `x` : the starting vertex (1 ≤ x ≤ n)
  - `s` : a string of instructions, only containing `'L'`, `'R'`, `'U'` length of string <=3*10^5

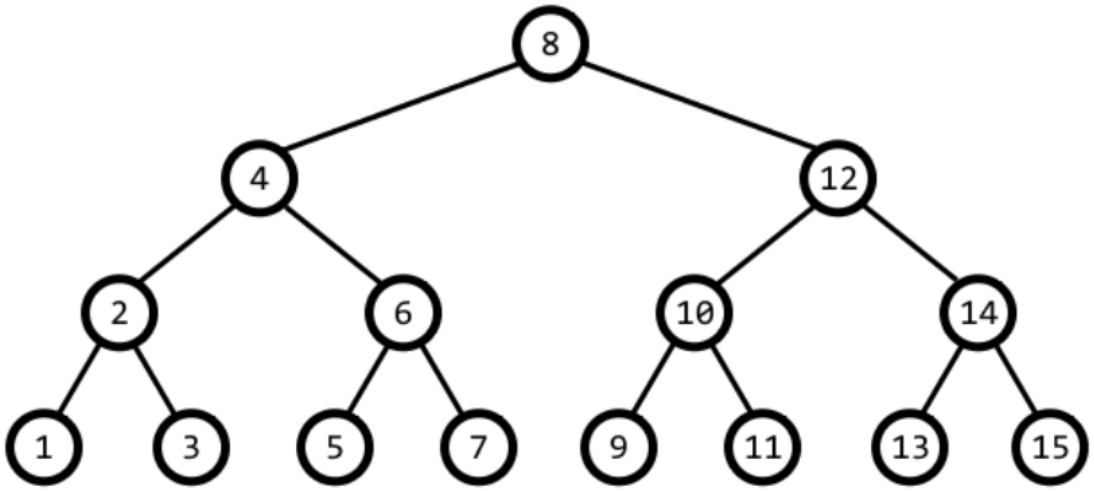
Output

Print the resulting node number after applying the movement string `s` to the starting node `x`.

Constraints

Batch 1 (20 marks): (1 ≤ n ≤ 10^6)

Batch 2 (80 marks): (1 ≤ n ≤ 10^18)



Example

Input

15 4

UURL

Output

10

Input

15 8

Output

5

Clarifications

Report an issue

No clarifications have been made at this time.