Digital Pakistan Speed Programing Competition Online Qualifier Round

Instructions

- Do not open the booklet unless you are explicitly told to do so. You can only read these instructions below.
- If you have any question regarding the problems, seek a clarification from the judges using DOMJudge.
- Before submitting a run, make sure that it is executable via command line. For Java, it must be executable via "javac" and for GNU C++ via "g++". Java programmers need to remove any "package" statements and source code's file name must be the same as of main class. C++ programmers need to remove any getch() / system("pause") like statements.
- Do not attach input files while submitting a run, only submit/attach source code files, i.e., *.java or *.cpp or *.py.
- Language supported: C/C++, Java and Python3
- Source code file name should not contain white space or special characters.
- You must take input from Console i.e.: Standard Input Stream (stdin in C, cin in C++, System.in in Java, stdin in Python)
- You must print your output to Console i.e.: Standard Output Stream (stdout in C, cout in C++, System.out in Java)
- Please, don't create/open any file for input or output.
- Please strictly meet the output format requirements as described in problem statements, because your program will be auto judged by computer. Your output will be compared with judge's output byte-by-byte and not tolerate even a difference of single byte. So, be aware! **Pay special attention to spaces, commas, dots, newlines, decimal places, case sensitivity etc.**
- All your programs must meet the time constraint specified.
- The decision of judges will be absolutely final.

Problem 03: In-order Binary Integer

Time Limit: 1 sec

Nomi is learning about **binary trees** and has recently developed a fascination with **binary numbers**. As an experiment, he devised a unique transformation process. First, he converts a given positive integer into its binary representation. He then constructs a binary tree using the digits of this binary string. The rule he follows is simple: **the root of the tree is always digit one** (reading a binary string from left to right). He inserted zero digit as the **left child** of the current node and one digit as the **right child**. The current node is always the newly inserted node.

After building the tree, Nomi performs an **in-order traversal** (left, root, right) to form a new binary string. He then converts this new binary string back into its decimal form. Now, Nomi wants to verify whether his transformation process works correctly. He has hired you to write a program that takes a positive integer and returns the final decimal number obtained after performing his transformation.

Input Format

The only input is an integer within the 64-bit range.

Output Format

The only output is an integer. The output integer will be the decimal representation of in-order binary string of the original integer.

Constraints

- Input integer is greater than zero
- Input integer is less than 64-bit integer

```
1
 \
  1
   \
    1
     \
      1
       \
        1
       /
      0
       \
        1
       /
      0
     /
    0
   /
  0
```

Binary Tree of 1000 (decimal), 1111101000 (binary)

Sample	**Input**	**Output**	**Explanation**
1	**10**	**3**	**10** in binary is **1010**. The in-order traversal will be **0011**, which is a binary of **3**.
2	**1000**	**963**	The binary of **1000** is **1111101000**. After forming the binary tree, the in-order binary string is **1111000011**, which is a binary of **963**