Reverse Bits

Try solving the Reverse Bits problem.

We'll cover the following

Statement

Given an integer n, we need to calculate the 32-bit unsigned integer it would be if we reversed its bits. When we say "reverse" we don't mean flipping the 0s to 1s and vice versa, but simply reversing the order in which they appear – from left-to-right to right-to-left. We need to return the integer the reversed bits result in.

Constraints:

- $0 \leq n \leq 2^{31}$

> In Java, an unsigned integer type does not exist. In such cases, the input is given as a signed integer type.

Examples

Sample example 1

Input

n	16 (00000000 00000000 00000000 00010000)

Output

Reversed	134217728 (00001000 00000000 00000000 00000000)

1 of 2

Sample example 2

Input

n	80 (00000000 00000000 00000000 01010000)

Output

Reversed	167772160 (00001010 00000000 00000000 00000000)

Understand the problem

Let's take a moment to make sure you've correctly understood the problem. The quiz below helps you check if you're solving the correct problem:

Reverse Bits

1 What is the output if we have 5 as an unsigned 32-bit integer and reverse the bits?

A) 4294967290

B) 2684354560

C) — 5

D) — 2684354560

Submit Answer

Question 1 of 2
0 attempted

Reset Quiz

Figure it out!

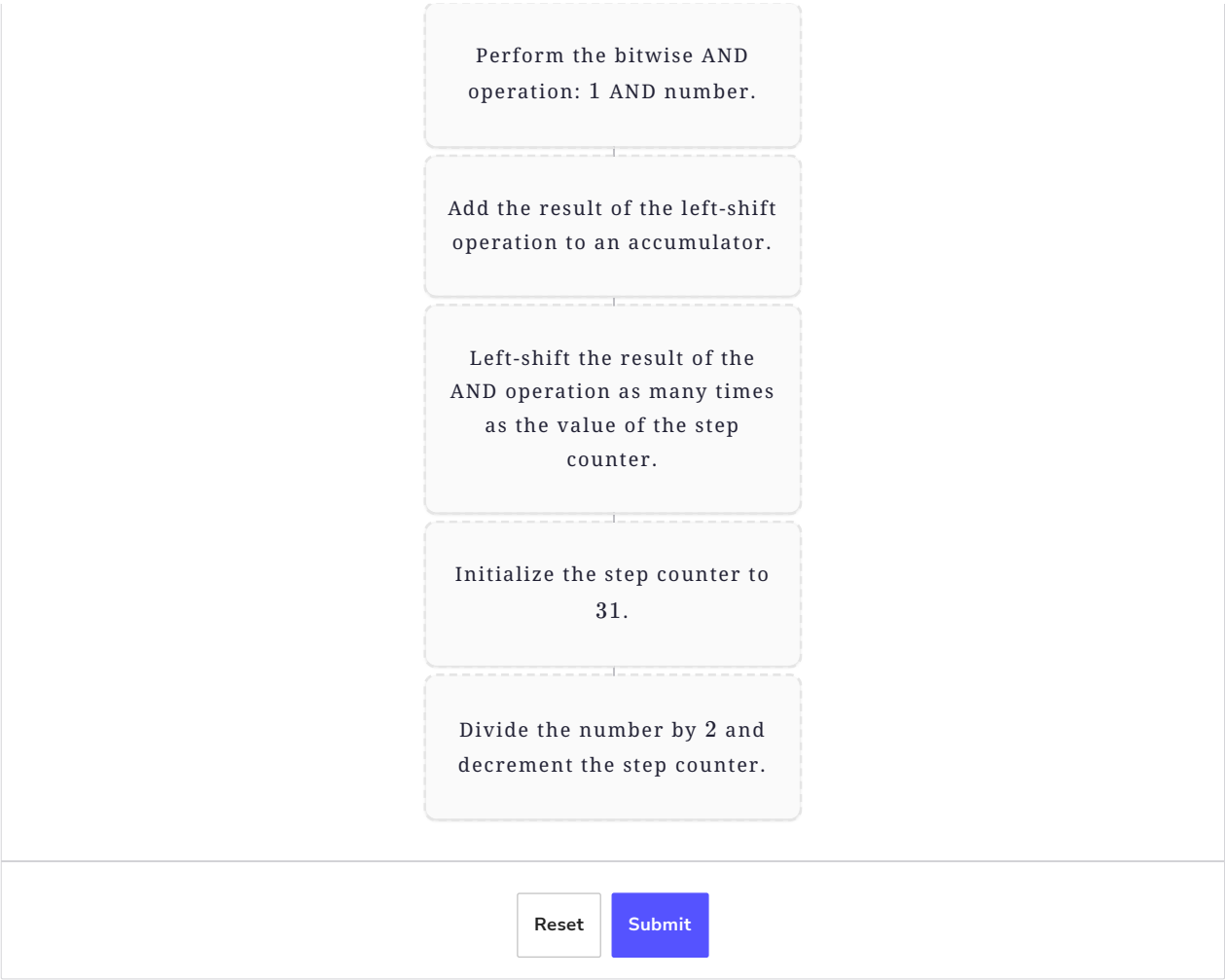
We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

> **Note:** As an additional challenge, we have intentionally hidden the solution to this puzzle.

Drag and drop the cards to rearrange them in the correct sequence.

Return the accumulator.

Loop while the number is greater than 0.



Try it yourself

Implement your solution in the following coding playground.

> **Note:** We have left the solution to this challenge as an exercise for you. You may try to translate the logic of the solved puzzle into a coded solution.

80

Reverse Bits

Back

Solution: Two Single ...

Next

Challenge Yourself: Int...

Mark as Completed