4th Bit

View | Try | Insights

Recommended Time: 11 mins | Points: 50 | 7 test cases (3 samples)

Skills: Problem Solving (Basic)

Coding | EASY | Bit Manipulation | Algorithms | Problem Solving

A binary number is a combination of *1*s and *0*s. Its n^{th} least significant digit is the the n^{th} digit starting from the right starting with *1*. Given a decimal number, convert it to binary and determine the value of the the 4^{th} least significant digit.

Example

number = 23

- Convert the decimal number 23 to binary number: $23_{10} = 2^4 + 2^2 + 2^1 + 2^0 = (10111)_2$.
- The value of the 4^{th} index from the right in the binary representation is *0*.

Function Description

Complete the function *fourthBit* in the editor below.

fourthBit has the following parameter(s):

int number: a decimal integer

Returns:

int: an integer *0* or *1* matching the 4^{th} least significant digit in the binary representation of *number*.

Constraints

- $0 \leq number < 2^{31}$

▶ **Input Format for Custom Testing**

▼ **Sample Case 0**

Sample Input 0

```
STDIN  Function
-----  --------
32   → number = 32
```

Sample Output 0

```
0
```

Explanation 0

- *Convert the decimal number 32 to binary number:* $32_{10} = (100000)_2$.
- The value of the 4^{th} index from the right in the binary representation is *0*.

▶ **Sample Case 1**