

Week 12 - 1:

ROLL NO.:240801052

Name: deepak manolal m

Q1) A binary number is a combination of 1s and 0s. Its nth least significant digit is the nth digit starting from the right starting with 1. Given a decimal number, convert it to binary and determine the value of the the 4th 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 4th 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 4th least significant digit in the binary representation of number.

Constraints

$0 \leq \text{number} < 231$

Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The only line contains an integer, number.

Sample Input

STDIN Function

-----

32 → number = 32

Sample Output

0

Explanation

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

CODE:

<b>Status</b>	Finished
<b>Started</b>	Wednesday, 15 January 2025, 4:55 PM
<b>Completed</b>	Wednesday, 15 January 2025, 5:03 PM
<b>Duration</b>	8 mins 10 secs

```

1  /*
2   * Complete the 'fourthBit' function below.
3   *
4   * The function is expected to return an INTEGER.
5   * The function accepts INTEGER number as parameter.
6   */
7
8  int fourthBit(int num)
9  {
10
11     int i=4,sum=0;
12     while(i--)
13     {
14         if(num%2==0)
15         {
16             sum=0;
17         }
18         else
19         {
20             sum=1;
21         }
22         num=num/2;
23     }
24     return sum;
25 }

```

OUTPUT:

	Test	Expected	Got	
✓	printf("%d", fourthBit(32))	0	0	✓
✓	printf("%d", fourthBit(77))	1	1	✓

Passed all tests! ✓

Q2) Determine the factors of a number (i.e., all positive integer values that evenly divide into a number) and then return the *pth* element of the list, sorted ascending. If there is no *pth* element, return 0.

Example

$n = 20$

$p = 3$

The factors of 20 in ascending order are {1, 2, 4, 5, 10, 20}. Using 1-based indexing, if  $p = 3$ , then 4 is returned. If  $p > 6$ , 0 would be returned.

Function Description

Complete the function `pthFactor` in the editor below.

`pthFactor` has the following parameter(s):

`int n`: the integer whose factors are to be found

`int p`: the index of the factor to be returned

Returns:

`int`: the long integer value of the *pth* integer factor of *n* or, if there is no factor at that index, then 0 is returned

Constraints

$1 \leq n \leq 10^{15}$

$1 \leq p \leq 10^9$

Input Format for Custom Testing

Input from `stdin` will be processed as follows and passed to the function.

The first line contains an integer *n*, the number to factor.

The second line contains an integer *p*, the 1-based index of the factor to return.

Sample Input

STDIN Function

-----

10 →  $n = 10$

3 →  $p = 3$

Sample Output

5

## Explanation

Factoring  $n = 10$  results in  $\{1, 2, 5, 10\}$ . Return the  $p = 3$ rd factor, 5, as the answer.

## CODE:

```
1  /*
2  * Complete the 'pthFactor' function below.
3  *
4  * The function is expected to return a LONG_INTEGER.
5  * The function accepts following parameters:
6  * 1. LONG_INTEGER n
7  * 2. LONG_INTEGER p
8  */
9
10 long pthFactor(long n, long p)
11 {
12     int count=0,i=1;
13     while(count!=p)
14     {
15         if(n%i==0)
16             count++;
17         else if(i>n)
18             return 0;
19         i++;
20     }
21     return i-1;
22 }
```

## OUTPUT:

	Test	Expected	Got	
✓	printf("%ld", pthFactor(10, 3))	5	5	✓
✓	printf("%ld", pthFactor(10, 5))	0	0	✓
✓	printf("%ld", pthFactor(1, 1))	1	1	✓

Passed all tests! ✓