# GE23131-Programming Using C-2024

| | |
|---|---|
| **Status** | Finished |
| **Started** | Sunday, 12 January 2025, 11:21 PM |
| **Completed** | Sunday, 12 January 2025, 11:31 PM |
| **Duration** | 9 mins 25 secs |

**Question 1**

Correct

Marked out of 1.00

☐⚑ Flag question

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

**Example**

number = 23

·        Convert the decimal number 23 to binary number: $23^{10} = 2^4 + 2^2 + 2^1 + 2^0$   = (10111

·        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 4th least significant digit in the binary representation of nu

**Constraints**

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

**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 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 4th index from the right in the binary representation is 0.

**Sample Input 1**

STDIN   Function

----- --------

77   →   number = 77

**Sample Output 1**

1

**Explanation 1**

· Convert the decimal number 77 to binary number: $77_{10} = (1001101)_2$.

· The value of the 4th index from the right in the binary representation is 1.

**Answer:** (penalty regime: 0 %)

[ Reset answer ]

| | Test | Expected | Got | |
|---|---|---|---|---|
| | `printf("%d", fourthBit(32))` | 0 | 0 | |
| | `printf("%d", fourthBit(77))` | 1 | 1 | |

Passed all tests!

---

Question **2**

Correct

Marked out of 1.00

☐⚑ Flag question

Determine the factors of a number (i.e., all positive integer values that evenly divide into a nu
$p^{th}$ element of the list, sorted ascending. If there is no $p^{th}$ 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, th
would be returned.

**Function Description**

Complete the function pthFactor in the editor below.

pthFactor has the following parameter(s):

Returns:

int: the long integer value of the p<sup>th</sup> integer factor of n or, if there is no factor at that index, th

**Constraints**

$1 \le n \le 10^{15}$
$1 \le p \le 10^9$

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 Case 0**
**Sample Input 0**

```
STDIN    Function
-----    --------
10    →   n = 10
3     →   p = 3
```

**Sample Output 0**

5

**Explanation 0**

Factoring n = 10 results in {1, 2, 5, 10}. Return the p = 3<sup>rd</sup> factor, 5, as the answer.

**Sample Case 1**
**Sample Input 1**

```
STDIN    Function
-----    --------
10    →   n = 10
5     →   p = 5
```

**Sample Output 1**

0

**Explanation 1**

Factoring n = 10 results in {1, 2, 5, 10}. There are only 4 factors and p = 5, therefore 0 is return

**Sample Case 2**
**Sample Input 2**

```
STDIN    Function
-----    --------
1     →   n = 1
```

**Sample Output 2**

1

**Explanation 2**

Factoring n = 1 results in {1}. The p = 1st factor of 1 is returned as the answer.

**Answer:** (penalty regime: 0 %)

Reset answer

| 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!

Save the state of the flags