Coding

✓ Done

Re-attempt quiz

Attempts allowed: 4

Time limit: 1 hour 30 mins

Grading method: Highest grade

Your attempts

Attempt 1

Status	Finished
Started	Thursday, 16 January 2025, 6:15 AM
Completed	Thursday, 16 January 2025, 6:19 AM
Duration	4 mins 41 secs

Review

GE23131-Programming Using C-2024

Status	Finished
Started	Thursday, 16 January 2025, 6:15 AM
Completed	Thursday, 16 January 2025, 6:19 AM
Duration	4 mins 41 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 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

- 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 4th least significant digit in the binary representation of number.

Constraints

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

Input Format for Custom Testing

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

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 Case 1

Sample Input 1

STDIN Function

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

77 → number = 77

Sample Output 1

1

Explanation 1

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

Answer: (penalty regime: 0 %)

Reset answer

```c
/*
 * Complete the 'fourthBit' f
 *
 * The function is expected t
 * The function accepts INTEG
 */

int fourthBit(int number)
{
    int binary[32];
    int i=0;
    while(number>0)
    {
        binary[i]=number%2;
        number/=2;
        i++;
    }
    if(i>=4)
    {
        return binary[3];
    }
    else
    return 0;
}
```

	Test	Exp
✓	`printf("%d", fourthBit(32))`	0
✓	`printf("%d", fourthBit(77))`	1

Passed all tests! ✓

Answer: (penalty regime: 0 %)

Reset answer

```
1
2  Bit' function below.
3
4  cted to return an INTEGER.
5   INTEGER number as parameter.
6
7
8  r)
9
10
11
12
13
14 r%2;
15
16
17
18
19
20 ];
21
22
23
24
```

	Test	Exp
✓	`printf("%d", fourthBit(32))`	0
✓	`printf("%d", fourthBit(77))`	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 number) and then return the 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, 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):

found

int p: the index of the factor to be returned

Returns:

int: the long integer value of the p^{th} integer factor of n or, if there is no factor at that index, then 0 is returned

Constraints

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

$1 \leqslant p \leqslant 10^{9}$

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

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

```
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 = 3rd 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

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

```
/*
 * Complete the 'pthFactor' f
 *
 * The function is expected t
 * The function accepts follo
 *  1. LONG_INTEGER n
 *  2. LONG_INTEGER p
 */

long pthFactor(long n, long p
{
    int count=0;
    for(long i=1;i<=n;i++)
    {
        if(n%i==0)
        {
            count++;
            if(count==p)
            {
                return i;
            }
        }
    }
```

```
3
4   nction is expected to return a
5   nction accepts following paran
6   NG_INTEGER n
7   NG_INTEGER p
8
9
10  actor(long n, long p)
11
12  ount=0;
13  ong i=1;i<=n;i++)
14
15  f(n%i==0)
16
17      count++;
18      if(count==p)
19      {
20          return i;
21      }
22
23
24  n 0;
25
```

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

Passed all tests! ✓

Finish review