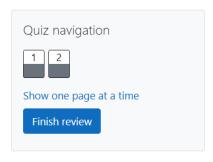
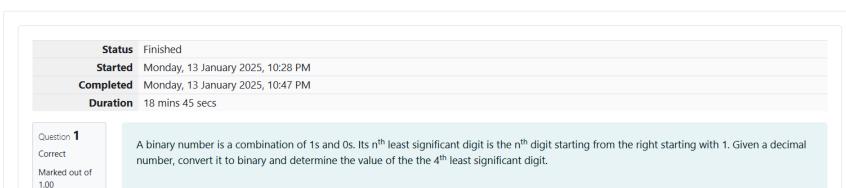
# GE23131-Programming Using C-2024





#### Example

Flag question

number = 23

- Convert the decimal number 23 to binary number:  $23^{10} = 2^4 + 2^2 + 2^1 + 2^0 = (10111)_2$ .
- $\cdot$  The value of the 4<sup>th</sup> 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

 $0 \le \text{number} < 2^{31}$ 

#### input Format for Custom lesting

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 Case 1

## Sample Input 1

```
STDIN Function
-----
77 → number = 77
```

## Sample Output 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
```

```
1 v /*
     * Complete the 'fourthBit' function below.
 3
     * The function is expected to return an INTEGER.
 4
 5
     * The function accepts INTEGER number as parameter.
 6
    int fourthBit(int number)
 8
 9
      int binary[32];
10
11
      int i=0;
12
      while(number>0)
13 1
14
          binary[i]=number%2;
15
          number/=2;
16
          i++;
17
18
      if(i>=4)
19
20
          return binary[3];
21
22
      else
23
      return 0;
24 }
```

		Test	Expected	Got	
	~	<pre>printf("%d", fourthBit(32))</pre>	0	0	<b>~</b>
	~	printf("%d", fourthBit(77))	1	1	<b>~</b>

Passed all tests! <

Question **2**Correct

Marked out of

Determine the factors of a number (i.e., all positive integer values that evenly divide into a number) and then return the p<sup>th</sup> element of the list, sorted ascending. If there is no p<sup>th</sup> element, return 0.



n = 20

Example

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

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

 $1 \le p \le 10^9$ 

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

```
JIDIN TUILLION
10 \rightarrow n = 10
3 \rightarrow p = 3
Sample Output 0
5
Explanation 0
Factoring n = 10 results in \{1, 2, 5, 10\}. Return the p = 3^{rd} factor, 5, as the answer.
Sample Case 1
Sample Input 1
STDIN Function
10 \rightarrow n = 10
5 \rightarrow 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 returned as the answer.
Sample Case 2
Sample Input 2
STDIN Function
```

 $1 \rightarrow n = 1$ 

```
1 \rightarrow p = 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
```

```
2
     * Complete the 'pthFactor' function below.
 3
     * The function is expected to return a LONG_INTEGER.
 4
     * The function accepts following parameters:
 5
     * 1. LONG INTEGER n
 6
     * 2. LONG_INTEGER p
 7
 8
 9
10
    long pthFactor(long n, long p)
11 🔻 {
12
        int count=0;
13
        for(long i=1;i<=n;++i)</pre>
14
           if(n%i==0)
15
16
17
               count++;
               1f(count--p)
TO
19
20
                   return i;
21
22
23
24
        return 0;
25 }
```

	Test	Expected	Got	
~	<pre>printf("%ld", pthFactor(10, 3))</pre>	5	5	<b>~</b>
~	<pre>printf("%ld", pthFactor(10, 5))</pre>	0	0	<b>~</b>
~	<pre>printf("%ld", pthFactor(1, 1))</pre>	1	1	<b>~</b>

Passed all tests! ✓

Finish review