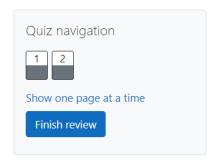
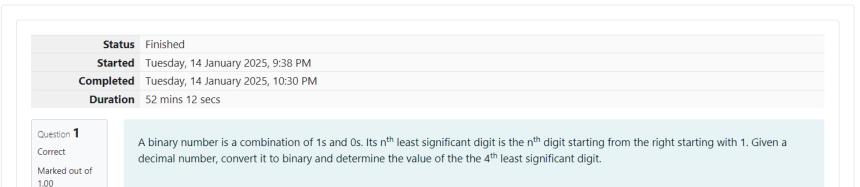
# GE23131-Programming Using C-2024





Example

Flag question

number = 23

- Convert the decimal number 23 to binary number:  $23^{10} = 2^4 + 2^2 + 2^1 + 2^0 = (10111)_2$ .
- 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.

**REC-CIS** 

▲ KAAVIYASRI PS 2024-CSE K2



dimensiona...

> Week-10-Character

DIFFICUSIONAL AND INITIO

- Arrays and Strings
- > Assessment-10-

Character Arrays and Strings

- > Week-11-String **Handling Functions**
- > Assessment-11-String

Handling Functions

**∨** Week-12-User-Defined **Functions** 

#### **■ Coding**

> Assessment-12-User-

**Defined Functions** 

> Week-12-Recursive

Functions

> Assessment-12-

Recursive Functions

> Week-13-Passing Arrays and Strings to Functions

> Assessment-13-Passing

Arrays and Strings to

Functions

> Week-14-Structures and

Unions

> Week-15-Pointers

#### **REC-CIS** DIFFICUSIONAL AND WIGHT

dimensiona...

> Week-10-Character

Arrays and Strings

> Assessment-10-

Character Arrays and Strings

> Week-11-String

**Handling Functions** 

> Assessment-11-String **Handling Functions** 

**∨** Week-12-User-Defined

Functions

#### Coding

> Assessment-12-User-

**Defined Functions** 

> Week-12-Recursive

**Functions** 

> Assessment-12-

**Recursive Functions** 

> Week-13-Passing Arrays

▲ KAAVIYASRI PS 2024-CSE K2 ∨





and Strings to Functions

> Assessment-13-Passing

Arrays and Strings to Functions

> Week-14-Structures and Unions

> Week-15-Pointers

#### REC-CIS

▲ KAAVIYASRI PS 2024-CSE K2 ∨

Dimensional and Multidimensional Arrays

> Assessment-09-Two-Dimensional and Multidimensiona...

> Week-10-Character Arrays and Strings

> Assessment-10-Character Arrays and Strings

> Week-11-String Handling Functions

> Assessment-11-String Handling Functions

➤ Week-12-User-Defined Functions

# Coding

> Assessment-12-User-Defined Functions

> Week-12-Recursive Functions

> Assessment-12-Recursive Functions

> Week-13-Passing Arrays and Strings to Functions

> Assessment-13-Passing Arrays and Strings to Functions

> Week-14-Structures and

list, sorted ascending. If there is no p'' element, return 0. Marked out of Flag question Example n = 20p = 3The 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

 $10 \rightarrow n = 10$ 

 $3 \rightarrow p = 3$ 

# 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 → 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
        * Complete the 'pthFactor' function below.
       * The function is expected to return a LONG_INTEGER.
       * The function accepts following parameters:
        * 1. LONG INTEGER n
        * 2. LONG INTEGER p
       long pthFactor(long n, long p)
  10
```

```
11 v {
12
        int count=0;
        for(long i=1;i<=n;++i)</pre>
13
        if(n%i==0){
14 🔻
15
            count++;
            if(count==p)
17
            return i;
18
19
        return 0;
20 }
```

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

✓ pı	rintf("%ld",	pthFactor(10, 5))	0	0	~	
✓ рі	rintf("%ld",	pthFactor(1, 1))	1	1	<b>~</b>	

Passed all tests! <

Finish review