REC-CIS

GE23131-Programming Using C-2024

Quiz navigation

1 2

Show one page at a time

Finish review

Status	Finished
Started	Thursday, 16 January 2025, 7:41 AM
Completed	Thursday, 16 January 2025, 7:46 AM
Duration	4 mins 31 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 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

Constraints

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

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

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

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

Answer: (penalty regime: 0 %)

Reset answer

```c
/*
 * Complete the 'fourthBit' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER number as parameter.
 */

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;
}
```

C-CIS

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

Passed all tests! ✓

Correct

Marked out of 1.00

Flag question

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):

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

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 = 3rd factor, 5, as the answer.

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 returned as the answer.

Sample Case 2

Sample Input 2

STDIN Function

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

1 → n = 1

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

```
9
10 long pthFactor(long n, long p)
11 {
12     int count=0;
13     for(long i=1;i<=n;++i)
14     {
15         if(n%i==0)
16         {
17             count++;
18             if(count==p)
19             {
20                 return i;
21             }
22         }
23     }
24     return 0;
25 }
```

	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! ✓