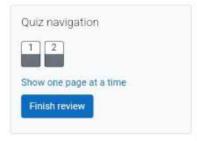
GE23131-Programming Using C-2024



Status Finished
Started Friday, 27 December 2024, 3:50 PM
Completed Friday, 27 December 2024, 4:00 PM
Duration 9 mins 10 secs

Question 1
Correct
Marked out of 1.00
F Flag

question

A binary number is a combination of 1s and 0s. Its nth least significant digit is the nth digit starting from the right starting with 1. Given a decimal number, convert it to binary and determine the value of the the 4th least significant digit.

Example

number = 23

- Convert the decimal number 23 to binary number: 23¹⁰
 = 2⁴ + 2⁷ + 2¹ + 2⁰
 = (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 ≤ number < 2³¹

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: 32₁₀ = (100000)₂.
- $\,\cdot\,\,$ 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

3

Explanation 1

- Convert the decimal number 77 to binary number. 77₁₀ = (1001101)₂.
- The value of the 4th index from the right in the binary representation is 1.

Answer: (penalty regime: 0 %)

```
Reset answer
```

```
* Complete the 'fourthBit' function belo
 3
    * The function is expected to return an
* The function accepts INTEGER number as
 4
    int fourthBit (int number)
10
11
12
             int binary [32];
             int i=0;
while (number>0)
13
14
15 -
16
                 binary[i]=number%2;
17
                 number/=2;
18
19
             if(i>=4)
20
21 +
22
                 return binary[3];
23
24
             else
25
             return 0;
26
27
28
```

```
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
F 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): 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

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

— — 10 10 → n = 10 3 → p = 3

STDIN Function

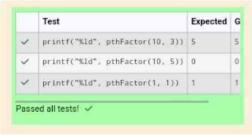
Sample Output 0

5

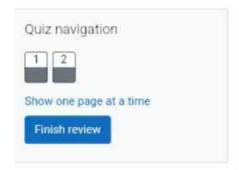
Explanation 0

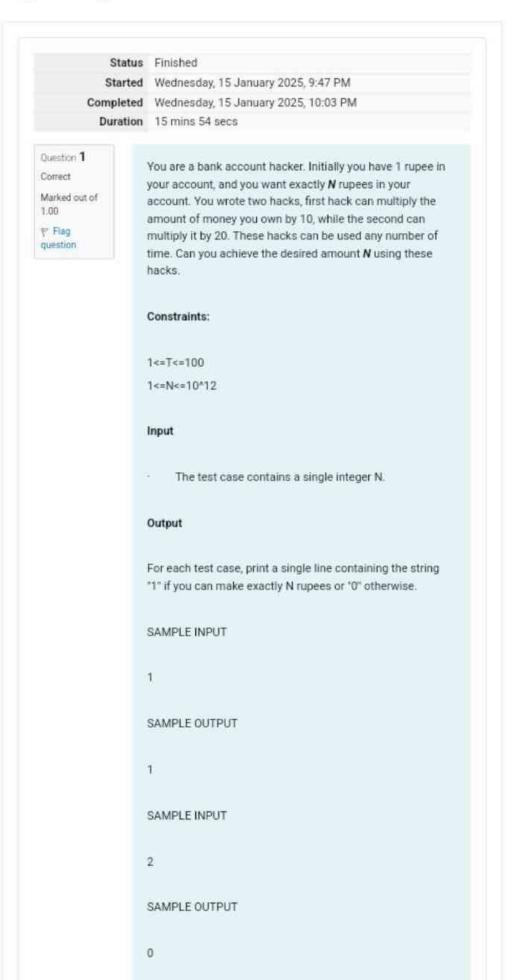
Factoring n = 10 results in {1, 2, 5, 10}. Return the $p = 3^{rd}$ factor, 5, as the answer.

Sample Case 1 Sample Input 1 STDIN Function 10 - n = 10 5 → p = 5 Sample Output 1 **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 1 - p=1 Sample Output 2 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 belo * The function is expected to return a L * The function accepts following paramet * 1. LONG_INTEGER n * 2. LONG_INTEGER p 4 5 8 10 long pthFactor(long n, long p) int count=0; 12 for(long i=1;i<=n;++i) 13 14 + 15 if(n%i==0) 16 + { count++; 18 if(count==p) 19 -{ 20 return i; 21 23 24 return 0;



GE23131-Programming Using C-2024





Answer: (penalty regime: 0 %)

Reset answer

```
int myFunc(int n)
        if(n==1){
 6
             return 1;
 8 +
9
       if(n%10!=0&&n%20!=0){
    return 0;
10
       if((n%10==0&&myFunc(n/10)==1)||(n%20==0 return 1;
11 -
13
14
       return 0;
15
     int Main(){
17
18
          int n;
scanf("%d",&n);
printf("%d\n",myFunc(n));
19
20
          return 0;
22
```

	Test	Expected	Got	
~	printf("%d", myFunc(1))	1	1	V
~	printf("%d", myFunc(2))	0	0	V
/	printf("%d", myFunc(10))	1	1	V
~	printf("%d", myFunc(25))	0	0	V
/	printf("%d", myFunc(200))	1	1	V

Question 2 Correct Marked out of 1.00

ked out of

F Flag question Find the number of ways that a given integer, \mathbf{X} , can be expressed as the sum of the \mathbf{N}^{th} powers of unique, natural numbers.

For example, if X = 13 and N = 2, we have to find all combinations of unique squares adding up to 13. The only solution is $2^2 + 3^2$.

Function Description

Complete the powerSum function in the editor below, It should return an integer that represents the number of possible combinations,

powerSum has the following parameter(s):

X: the integer to sum to

N: the integer power to raise numbers to

Input Format

The first line contains an integer X.

The second line contains an integer N.

Constraints

1 ≤ X ≤ 1000

2 ≤ N ≤ 10

Output Format

Output a single integer, the number of possible combinations calculated.

Sample Input 0 10 2 Sample Output 0 1 Explanation 0 If X = 10 and N = 2, we need to find the number of ways that 10 can be represented as the sum of squares of unique numbers. $10 = 1^2 + 3^2$ This is the only way in which 10 can be expressed as the sum of unique squares. Sample Input 1 100 2 Sample Output 1 3 Explanation 1 $100 = (10^2) = (6^2 + 8^2) = (1^2 + 3^2 + 4^2 + 5^2 + 7^2)$ Sample Input 2 100 3 Sample Output 2 1 Explanation 2 100 can be expressed as the sum of the cubes of 1, 2, 3, 4. (1 + 8 + 27 + 64 = 100). There is no other way to express 100

as the sum of cubes.

```
Answer: (penalty regime: 0 %)
 Reset answer
      #include<stdio.h>
      #includesmath.h>
   2
   3
   4
   5 int powerSum(int x, int m, int n)
   6 + {
   7 .
          if(x<0){
              return 0;
   8
   9
  10 +
          if(x==0){
  11
              return 1;
  12
  13 v
          if(m>x){
  14
             return 0;
  15
  16
          int p-pow(m,n);
          return powerSum(x-p,m+1,n)+powerSum(x
  17
  18 }
  19 - int Main(){
  20
          int x,n;
  21
          scanf("%d",&x);
          scanf("%d",&n);
  22
         printf("%d\n",powerSum(x,1,n));
  23
          return 0;
  24
  25 }
```

Test	Expected
<pre> ✓ printf("%d", powerSum(10, 1, 2))</pre>	1

Finish review