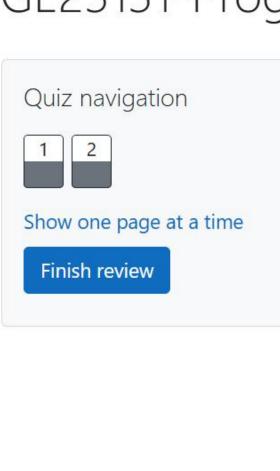
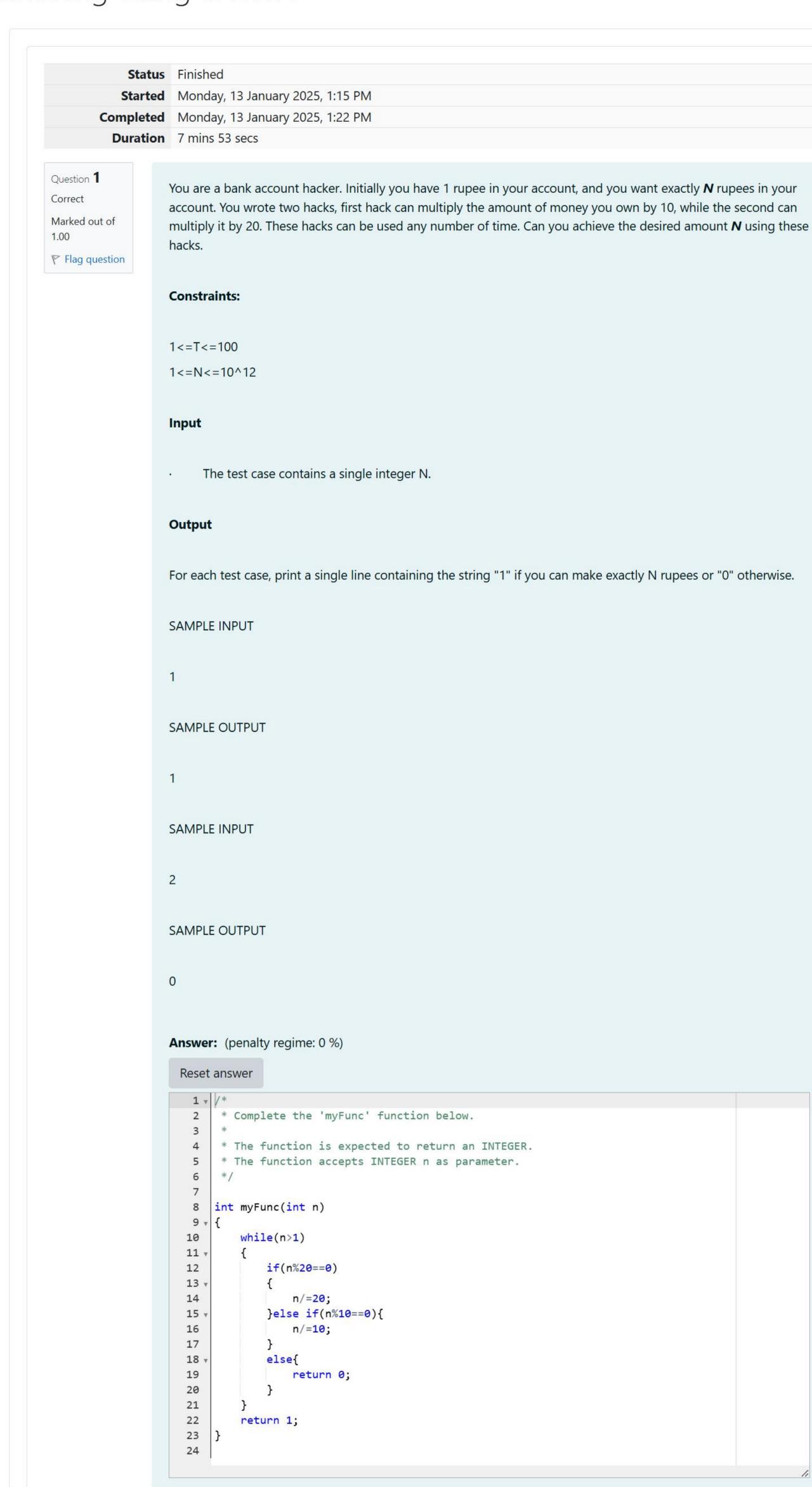
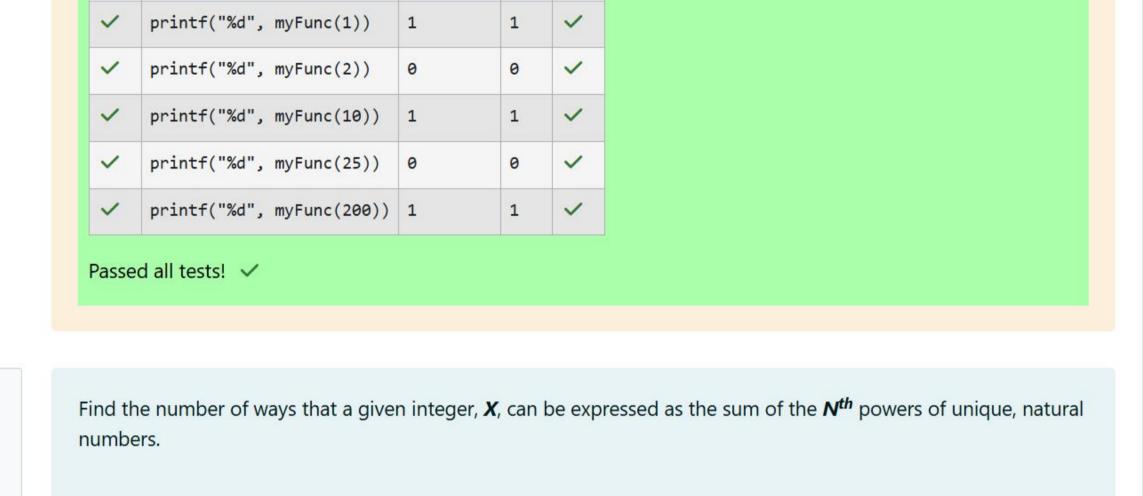
REC-CIS

GE23131-Programming Using C-2024







Expected Got

```
♥ Flag question
```

Question 2

Marked out of

Correct

1.00

Function Description

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

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

powerSum has the following parameter(s):

possible combinations.

solution is $2^2 + 3^2$.

Test

X: the integer to sum to

N: the integer power to raise numbers to

Input Format

Constraints

The second line contains an integer N.

The first line contains an integer X.

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

Sample Input 0

 $1 \le X \le 1000$

 $2 \le N \le 10$

10

2 Sample Output 0

1

unique numbers.

Explanation 0

 $10 = 1^2 + 3^2$ This is the only way in which 10 can be expressed as the sum of unique squares.

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

Sample Input 1

100

2

3

Explanation 1

Sample Output 1

 $100 = (10^2) = (6^2 + 8^2) = (1^2 + 3^2 + 4^2 + 5^2 + 7^2)$

Sample Input 2

Sample Output 2

100

100 can be expressed as the sum of the cubes of 1, 2, 3, 4.

1

3

(1 + 8 + 27 + 64 = 100). There is no other way to express 100 as the sum of cubes.

Explanation 2

Answer: (penalty regime: 0 %)

Reset answer

2 * Complete the 'powerSum' function below.
3 *
4 * The function is expected to return an II

```
* The function is expected to return an INTEGER.
     * The function accepts following parameters:
     * 1. INTEGER x
     * 2. INTEGER n
     */
8
   int powerSum(int x, int m, int n)
11 v {
        if(x==0)return 1;
12
        if(x<0)return 0;
13
        int count=0;
14
        for (int i=m;;i++)
15
16
            int power=1;
17
            for(int j=0;j<n;j++)</pre>
18
        power*=i;
19
        if(power>x)break;
20
        count+=powerSum(x-power,i+1,n);
21
22
23
        return count;
24 }
                                    Expected Got
    Test
```

```
Test Expected Got

v printf("%d", powerSum(10, 1, 2)) 1 1 v

Passed all tests! v

Finish review
```

REC-CIS

GE23131-Programming Using C-2024 Quiz navigation Show one page at a time Finish review

Question 1 Correct Marked out of 1.00 Flag question

Status Finished Started Monday, 23 December 2024, 5:33 PM Completed Friday, 20 December 2024, 1:55 PM **Duration** 3 days 3 hours

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: $23^{10} = 2^4 + 2^2 + 2^1 + 2^0 = (10111)_2$. The value of the 4th index from the right in the binary representation is 0. **Function Description**

Complete the function fourthBit in the editor below.

Returns:

Constraints

fourthBit has the following parameter(s):

int number: a decimal integer

int: an integer 0 or 1 matching the 4th least significant digit in the binary representation of number.

Input Format for Custom Testing Input from stdin will be processed as follows and passed to the function.

 $0 \le \text{number} < 2^{31}$ The only line contains an integer, number.

Sample Case 0 Sample Input 0 STDIN Function

_____ $32 \rightarrow \text{number} = 32$ **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 $77 \rightarrow \text{number} = 77$ **Sample Output 1** 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 * Complete the 'fourthBit' function below. 3 * The function is expected to return an INTEGER. * The function accepts INTEGER number as parameter. 6

7

9

10

int fourthBit(int number)

int binary[32];

number/=2;

i++;

binary[i]=number%2;

return binary[3];

int n: the integer whose factors are to be found

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

The first line contains an integer n, the number to factor.

int p: the index of the factor to be returned

Expected Got

return the pth element of the list, sorted ascending. If there is no pth element, return 0.

1

Determine the factors of a number (i.e., all positive integer values that evenly divide into a number) and then

/

int i=0; 11 while (number>0) 12 13 14 15 16 17 if (i>=4) 18 19 20 21 22 else 23 return 0; 24 Test printf("%d", fourthBit(32)) 0 printf("%d", fourthBit(77)) 1 Passed all tests! <

Example Flag question 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):

Question 2

Marked out of

Correct

1.00

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

> The second line contains an integer p, the 1-based index of the factor to return. Sample Case 0 Sample Input 0 STDIN Function

 $10 \rightarrow n = 10$

Sample Output 0

5

 $3 \rightarrow p = 3$

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

 $10 \rightarrow n = 10$ $5 \rightarrow p = 5$ **Sample Output 1**

0

1

STDIN Function

Explanation 1 Sample Case 2

Sample Input 2 STDIN Function $1 \rightarrow n = 1$ $1 \rightarrow p = 1$

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 1 | /* 3 4

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 Output 2**

* 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 8 */ long pthFactor(long n, long p) 10 11 | { int count = 0; 12 for(long i=1;i<=n;++i)</pre> if (n%i==0) count++; if(count==p) return i; return 0; **Expected Got** Test printf("%ld", pthFactor(10, 3)) 5

13 14 15 16 v 17 18 19 20 21 22 } printf("%ld", pthFactor(10, 5)) 0 printf("%ld", pthFactor(1, 1)) 1 ~ Passed all tests! < Finish review