REC-CIS

Quiz navigation Show one page at a time Finish review

Question 1 Correct Marked out of

1.00 question

GE23131-Programming Using C-2024

Status Finished

Duration 6 mins 41 secs

Example

number = 23

 $= (10111)_2.$

Started Monday, 13 January 2025, 10:50 AM

A binary number is a combination of 1s and 0s. Its nth least significant digit is

convert it to binary and determine the value of the the 4th least significant digit.

Convert the decimal number 23 to binary number: $23^{10} = 2^4 + 2^2 + 2^1 + 2^2 + 2^3 + 2^4 +$

The value of the 4th index from the right in the binary representation is 0.

Convert the decimal number 32 to binary number: $32_{10} = (100000)_2$.

Convert the decimal number 77 to binary number: $77_{10} = (1001101)_2$.

* Complete the 'fourthBit' function below.

* The function is expected to return an INTEGER.

* The function accepts INTEGER number as parameter.

Expected Got

Determine the factors of a number (i.e., all positive integer values that evenly

divide into a number) and then return the pth element of the list, sorted

The factors of 20 in ascending order are {1, 2, 4, 5, 10, 20}. Using 1-based

int: the long integer value of the pth integer factor of n or, if there is no factor at

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

The second line contains an integer p, the 1-based index of the factor to return.

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

indexing, if p = 3, then 4 is returned. If p > 6, 0 would be returned.

Complete the function pthFactor in the editor below.

pthFactor has the following parameter(s):

int p: the index of the factor to be returned

that index, then 0 is returned

Input Format for Custom Testing

int n: the integer whose factors are to be found

0

1

✓

/

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

The value of the 4th index from the right in the binary representation is 0.

the nth digit starting from the right starting with 1. Given a decimal number,

Completed Monday, 13 January 2025, 10:57 AM

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 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 $32 \rightarrow number = 32$ **Sample Output 0** 0 **Explanation 0**

Sample Case 1 Sample Input 1 STDIN Function $77 \rightarrow \text{number} = 77$ **Sample Output 1** 1

Explanation 1

Reset answer

*/

12 √ int mymain(){

Test

Passed all tests! <

Example

n = 20

p = 3

Returns:

Constraints

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

 $1 \le p \le 10^9$

Sample Case 0

Sample Input 0

Function Description

1 | /*

2

5 6

9 🔻 {

10 11

13

14

15

16 17

Answer: (penalty regime: 0 %)

int fourthBit(int number)

int number;

return 0;

return (number>>3)&1;

scanf("%d",&number);

printf("%d", fourthBit(32))

ascending. If there is no pth element, return 0.

printf("%d", fourthBit(77)) 1

printf("%d\n",fourthBit(number));

Question **2** Correct Marked out of 1.00 ▼ Flag question

5

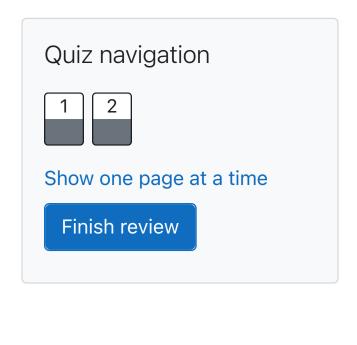
STDIN Function $10 \rightarrow n = 10$ $3 \rightarrow p = 3$ **Sample Output 0** 0

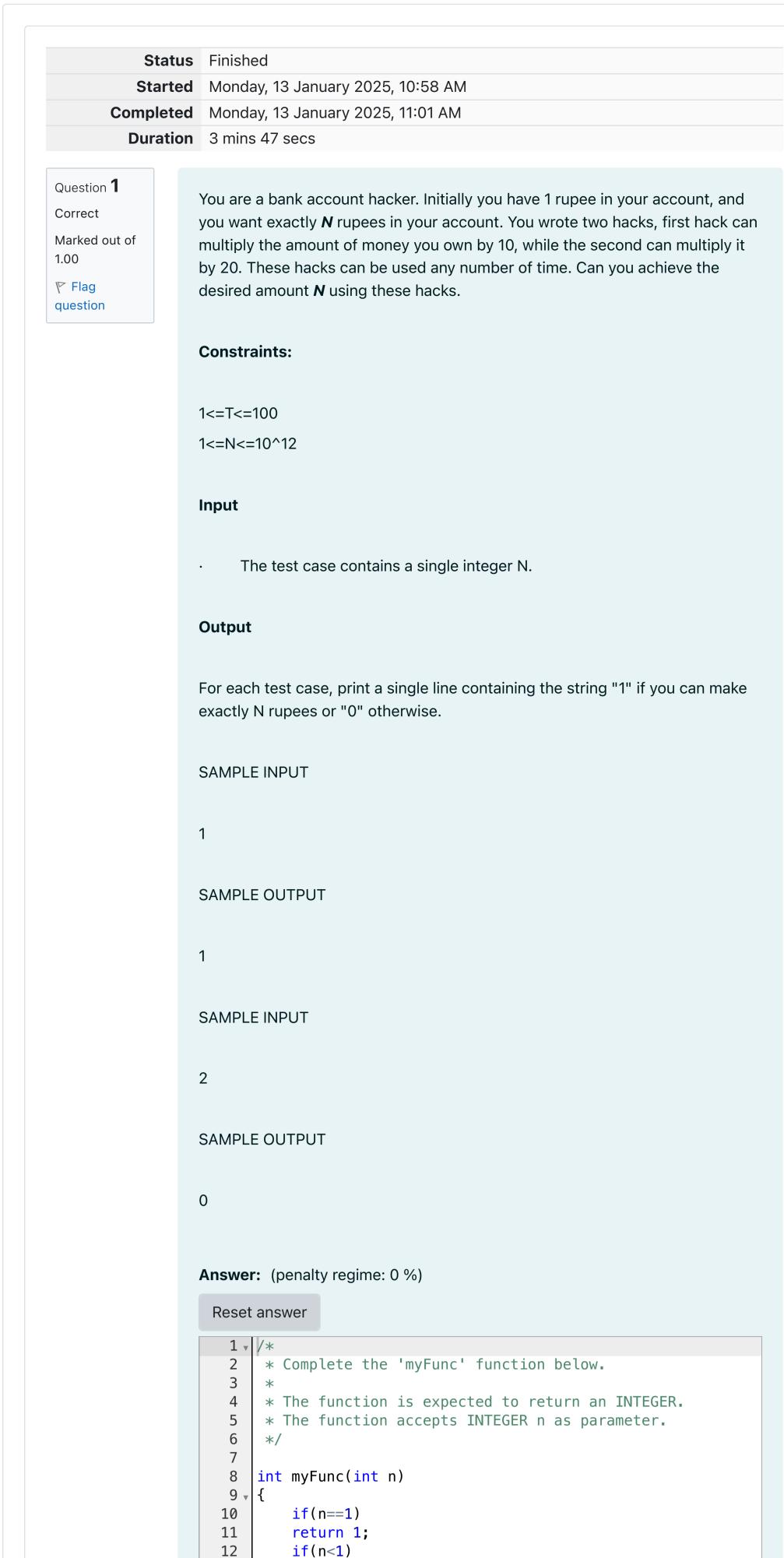
Explanation 0 Factoring n = 10 results in $\{1, 2, 5, 10\}$. Return the $p = 3^{rd}$ factor, 5, as the answer. **Sample Case 1 Sample Input 1** STDIN Function $10 \rightarrow n = 10$ $5 \rightarrow p = 5$ **Sample Output 1 Explanation 1 Sample Case 2**

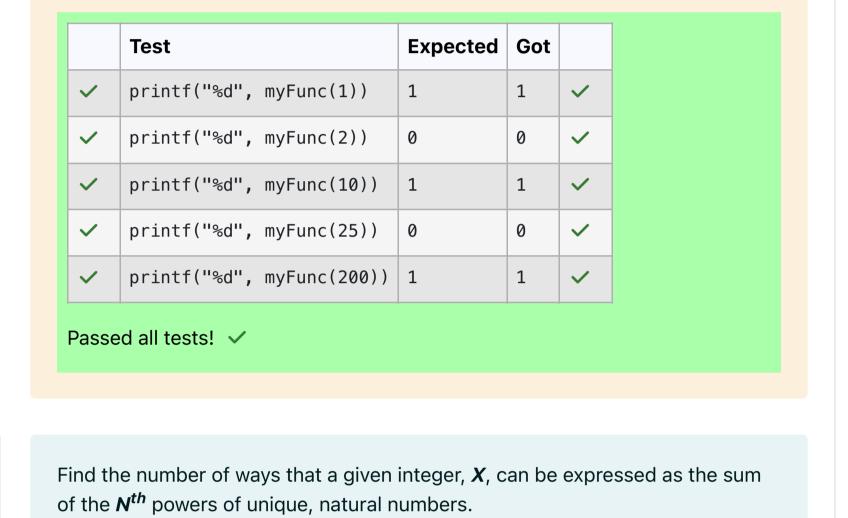
Factoring n = 10 results in $\{1, 2, 5, 10\}$. There are only 4 factors and p = 5, therefore 0 is returned as the answer. **Sample Input 2** STDIN Function $1 \rightarrow n = 1$ $1 \rightarrow p = 1$ **Sample Output 2 Explanation 2 Answer:** (penalty regime: 0 %) Reset answer 1 | /* * Complete the 'pthFactor' function below. 3 * The function is expected to return a LONG_INTEGER. * The function accepts following parameters: 5 * 1. LONG_INTEGER n 7 * 2. LONG_INTEGER p 8 */ long pthFactor(long n, long p) 10 11 ▼ { 12 int count = 0; for (int i=1;i<=n;i++){</pre> 13 ▼ if(n%i==0){ 14 ▼ 15 count ++; if(count==p){ 16 • return i; 17 18 19 20 return 0; 21 22 23 24 v int mymain(){ int n,p; 25 scanf("%d %d",&n,&p); 26 27 printf("%ld\n",pthFactor(n,p)); return 0; 28 } 29

Factoring n = 1 results in $\{1\}$. The p = 1st factor of 1 is returned as the answer. **Expected Got** Test printf("%ld", pthFactor(10, 3)) | 5 5 printf("%ld", pthFactor(10, 5)) 0 printf("%ld", pthFactor(1, 1)) **/** Passed all tests! < Finish review

GE23131-Programming Using C-2024







♥ Flag
 question

Question **2**

Marked out of

Correct

1.00

13

14

15

16

1718

1920

return 0;

return 1;

return 1;

return 0;

if(n%10==0 && myFunc(n/10))

if(n%20==0 && myFunc(n/20))

squares adding up to 13. The only solution is $2^2 + 3^2$.

Function Description

For example, if X = 13 and N = 2, we have to find all combinations of unique

that represents the number of possible combinations.

Complete the powerSum function in the editor below. It should return an integer

N: the integer power to raise numbers to Input Format

The first line contains an integer **X**.

powerSum has the following parameter(s):

X: the integer to sum to

The second line contains an integer *N*.

Constraints

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

 $1 \le X \le 1000$

Output Format

 $2 \le N \le 10$

Sample Input 0

Sample Output 0

10

2

represented as the sum of squares of unique numbers. $10 = 1^2 + 3^2$

Explanation 0

This is the only way in which *10* can be expressed as the sum of unique

If X = 10 and N = 2, we need to find the number of ways that 10 can be

Sample Input 1

squares.

Sample Output 1

3

100

2

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

Sample Input 2

Sample Output 2

Explanation 1

100 3

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

Explanation 2

1

2

Answer: (penalty regime: 0 %)

cubes.

Reset answer

1 • /*

* Complete the 'powerSum' function below.

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

3 * The function is expected to return an INTEGER. 4 * The function accepts following parameters: * 1. INTEGER x 7 * 2. INTEGER n 8 */ 9 int powerSum(int x, int m, int n) 10 11 ▼ { int power=1; 12 for(int i=0;i<n;i++)</pre> 13 14 power*=m; if(power==x) return 1; 15 16 if(power>x) return 0; return powerSum(x-power, m+1, n)+powerSum(x, m+1, n); 17 18

