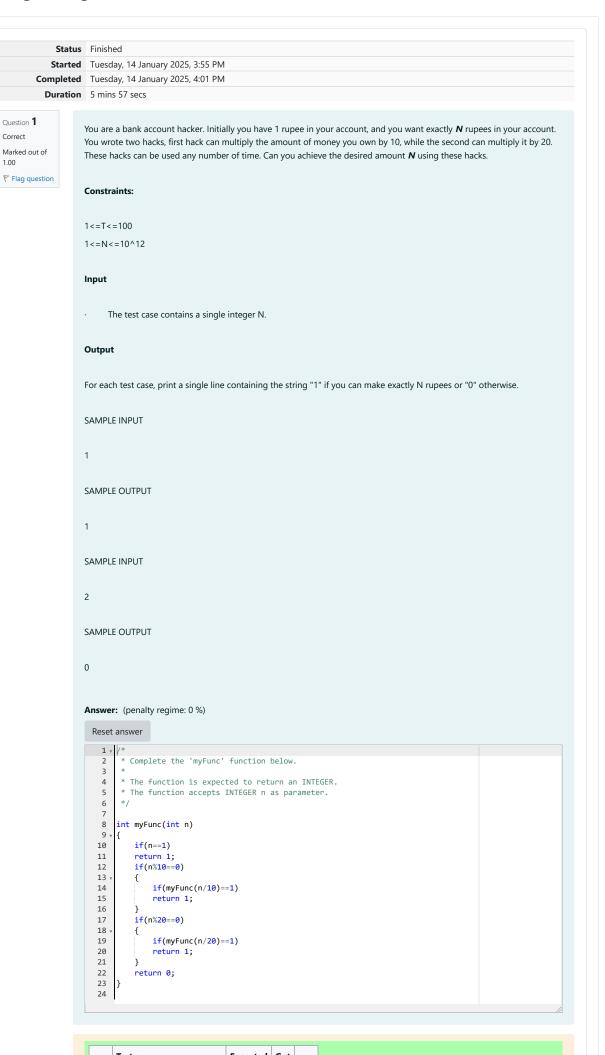
GE23131-Programming Using C-2024





	lest	Expected	GOT	
~	<pre>printf("%d", myFunc(1))</pre>	1	1	~
~	<pre>printf("%d", myFunc(2))</pre>	0	0	~
~	printf("%d", myFunc(10))	1	1	~
~	printf("%d", myFunc(25))	0	0	~
~	printf("%d", myFunc(200))	1	1	~

Passed all tests! ✓

Question **2**Correct
Marked out of 1.00

Flag question

Find the number of ways that a given integer, X, can be expressed as the sum of the 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
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
          * Complete the 'powerSum' function below.
    2
   3
        * The function is expected to return an INTEGER.
* The function accepts following parameters:
* 1. INTEGER x
    5
         * 2. INTEGER n
    8
    9
   10
        int powerSum(int x, int m, int n)
   11 、
   12
             int tmp;
            tmp=1;
for(int i=1;i<=n;i++)</pre>
   13
   14
   15 1
   16
                 tmp=tmp*m;
   17
   18
             if(tmp==x)
   19
   20
21
             if(tmp>x)
             return 0;
return powerSum(x,m+1,n)+powerSum(x-tmp,m+1,n);
   22
```

Finish review