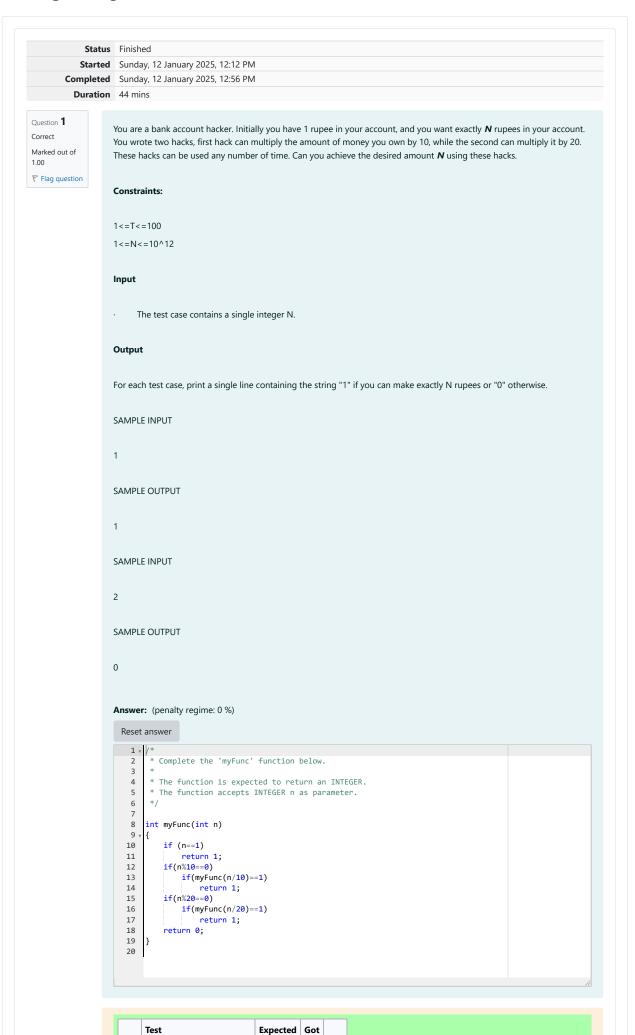
# GE23131-Programming Using C-2024





printf("%d", myFunc(1))

~	printf("%d", myFunc(2))	0	0	~
~	<pre>printf("%d", myFunc(10))</pre>	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  ${\it N}$ .

#### Constraints

 $1 \le X \le 1000$ 

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

```
Explanation 1

100 = (10<sup>2</sup>) = (6<sup>2</sup> + 8<sup>2</sup>) = (1<sup>2</sup> + 3<sup>2</sup> + 4<sup>2</sup> + 5<sup>2</sup> + 7<sup>2</sup>)

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
```

```
* Complete the 'powerSum' function below.
 2
      * The function is expected to return an INTEGER.
* The function accepts following parameters:
4
     * 1. INTEGER x
* 2. INTEGER n
 6
8
9
     int powerSum(int x, int m, int n)
10
11 •
12
          int tmp;
          tmp=1;
for(int i=1;i<=n;i++)</pre>
13
14
15
16
              tmp=tmp*m;
          if(tmp==x)
18
          return 1;
if(tmp>x)
19
20
          return 0;
return powerSum(x,m+1,n)+powerSum(x-tmp,m+1,n);
21
22
23
```

```
Test Expected Got

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

Passed all tests! ✓
```

Finish review