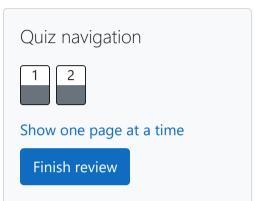
# GE23131-Programming Using C-2024





## Question 1

Correct

Marked out of 1.00

Flag question

You are a bank account hacker. Initially you have 1 rupee in your account, and you want exactly **N** rupees in your account. You wrote two hacks, first hack can multiply the amount of money you own by 10, while the second can multiply it by 20. These hacks can be used any number of time. Can you achieve the desired amount **N** using these hacks.

#### **Constraints:**

## Input

The test case contains a single integer N.

#### Output





```
SAMPLE INPUT
1
SAMPLE OUTPUT
1
SAMPLE INPUT
2
SAMPLE OUTPUT
0
Answer: (penalty regime: 0 %)
 Reset answer
       * Complete the 'myFunc' function below.
       * The function is expected to return an INTEGER.
       * The function accepts INTEGER n as parameter.
   8 | int myFunc(int n) {
```

// If n is loss than 1 it's impossible to achieve

```
ΙZ
13
        // Working backwards from n to 1 by dividing by 10 or 20
14
        while (n > 1) {
15 ▼
            // If divisible by 20, divide by 20
16
17 🔻
            if (n % 20 == 0) {
                n /= 20;
18
19
            // If divisible by 10, divide by 10
20
            else if (n % 10 == 0) {
21 *
22
                n /= 10;
23
            // If not divisible by either 10 or 20, it's impossible
24
25 ▼
            else {
26
                return 0;
27
28
29
        // If we reached exactly 1, it's possible
30
        return (n == 1);
31
32 }
```

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

Passed all tests! ✓



Marked out of 1.00

Flag question

**N** 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**

$$1 \le X \le 1000$$

$$2 \le N \le 10$$

### **Output Format**



# 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

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

1 |

2 \* Complete the 'powerSum' function below.

3

```
* 2. INTEGER n
 8
   #include <stdio.h>
10
    #include <math.h>
11
12
13 v int powerSum(int x, int m, int n) {
        if (x == 0)
14
15
          return 1;
        if (x < 0 | | pow(m, n) > x)
16
17
           return 0;
18
        return powerSum(x - pow(m, n), m + 1, n) + powerSum(x, m + 1,  
19
20
```

	Test	Expected	Got	
~	printf("%d", powerSum(10, 1, 2))	1	1	~

Passed all tests! <

Finish review

