# GE23131-Programming Using C-2024

## Quiz navigation

| | |
|---|---|
| **Status** | Finished |
| **Started** | Tuesday, 14 January 2025, 5:57 PM |
| **Completed** | Tuesday, 14 January 2025, 6:07 PM |
| **Duration** | 10 mins 27 secs |

**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:**

1<=T<=100

1<=N<=10^12

**Input**

·      The test case contains a single integer N.

**Output**

For each test case, print a single line containing the string "1" if you can make exactly N rupees or "0" otherwise.

SAMPLE INPUT

1

SAMPLE OUTPUT

1

SAMPLE INPUT

2

SAMPLE OUTPUT

0

**Answer:**  (penalty regime: 0 %)

Reset answer

```c
/*
 * Complete the 'myFunc' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER n as parameter.
 */

int myFunc(int n)
{
    if(n==1)
    return 1;
    if(n%10==0)
    if(myFunc(n/10)==1)
    return 1;
    if(n%20==0)
    if(myFunc(n/20)==1)
    return 1;
    return 0;
}
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✓ | printf("%d", myFunc(1)) | 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! ✓

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

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

```c
1    /*
2     * Complete the 'powerSum' function below.
3     *
4     * The function is expected to return an INTEGER.
5     * The function accepts following parameters:
6     *  1. INTEGER x
7     *  2. INTEGER n
8     */
9    #include<stdio.h>
10   #include<ctype.h>
11   #include<math.h>
12   int powerSum(int x, int m, int n)
13   {
14       int tmp;
15       tmp = pow(m, n);
16       if(tmp == x)
17       return 1;
18       if(tmp > x)
19       return 0;
20       return powerSum(x, m + 1, n)+ powerSum(x - tmp, m + 1, n);
21
22   }
```

| | Test | Expected | Got | |
|---|---|---|---|---|
| ✓ | printf("%d", powerSum(10, 1, 2)) | 1 | 1 | ✓ |

Passed all tests! ✓