

Status	Finished
Started	Wednesday, 15 January 2025, 12:07 AM
Completed	Wednesday, 15 January 2025, 12:10 AM
Duration	3 mins 20 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

Reset answer

```
1 /*
2  * Complete the 'myFunc' function below.
3  *
4  * The function is expected to return an INTEGER.
5  * The function accepts INTEGER n as parameter.
6  */
7
8 int myFunc(int n)
9 {
10     return n==1 || n%10==0;
11 }
12
```

	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	✓

of
tion

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

$1 \leq X \leq 1000$

$2 \leq N \leq 10$

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

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

Passed all tests! ✓