**Week-12**

**1.**

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

**Source Code**

int myFunc(int n)

{

while(n>1) {

if(n%20==0){

n/= 20;

}

else if(n%10==0){

n/=10;

}

}

else{

return 0;

}

}

return (n==1);

else{

return 0;

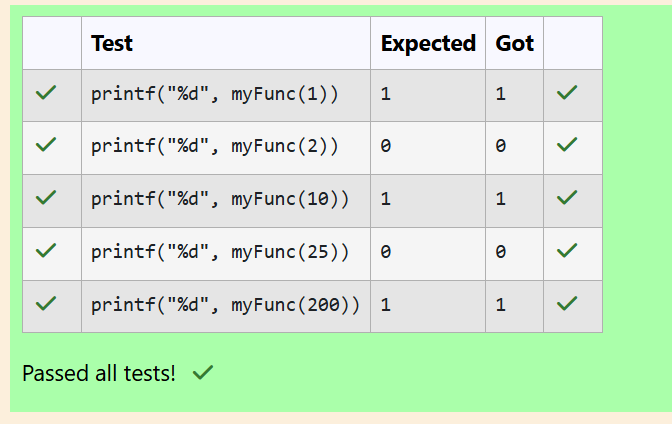
}

}

return (n==1);

}

**Output**



**2.**

Find the number of ways that a given integer, X, can be expressed as the sum of the Nth 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 2² + 3².

**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.

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 = 12 + 32

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 = (102) = (6² + 8²) = (1² + 3² + 4² + 5² + 7²)

**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.

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

**Source Code**

#include<stdio.h>

#include <math.h>

int powerSum(int x, int m, int n)

{

int power=m;

for (int i=1;i<n;i++) {

power\*=m;

}

if(power==x)

return 1;

if (power>x)

return 0;

return powerSum (x-power, m+1, n)+powerSum(x,m+1,n);

}

**Output**

A screenshot of a computer

Description automatically generated