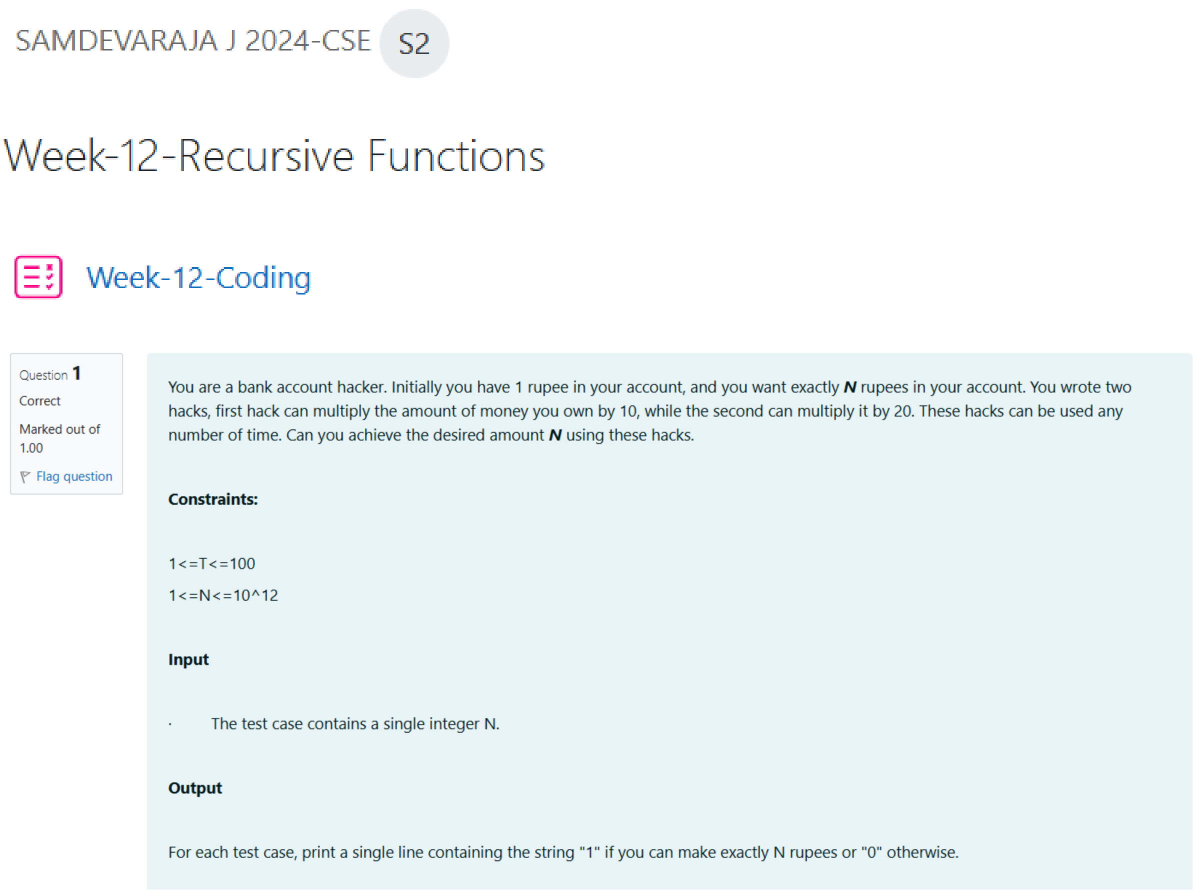

SAMDEVARAJA J 2024-CSE S2

Week-12-Recursive Functions

Week-12-Coding

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

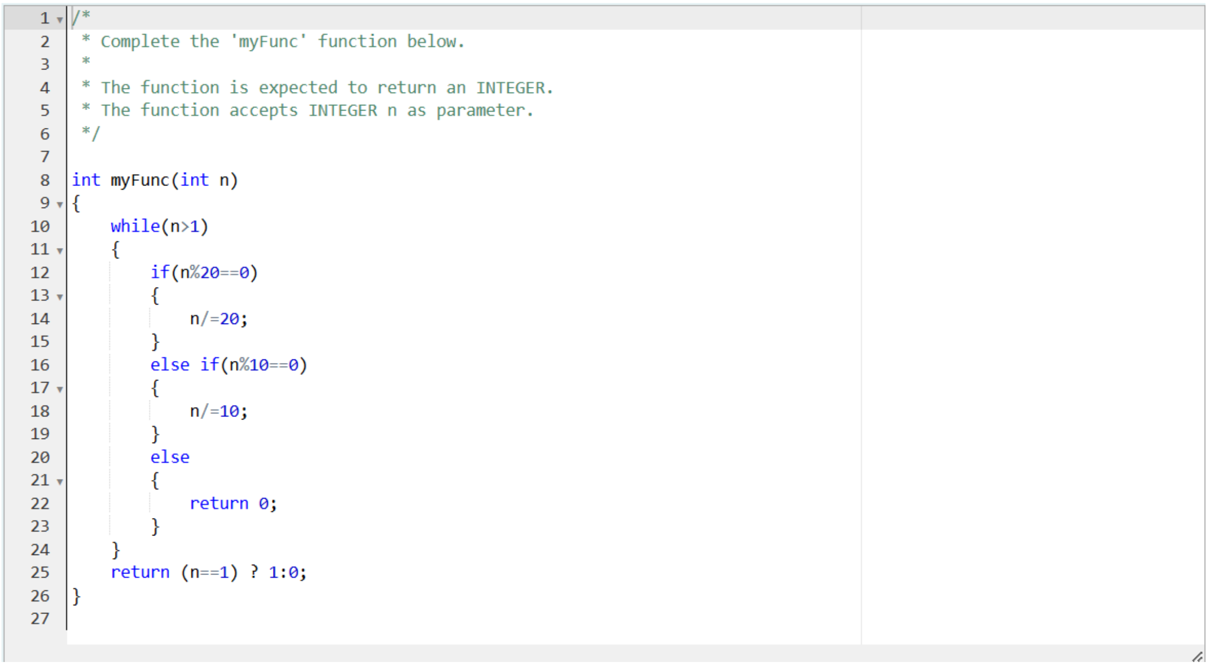
Input

· The test case contains a single integer N.

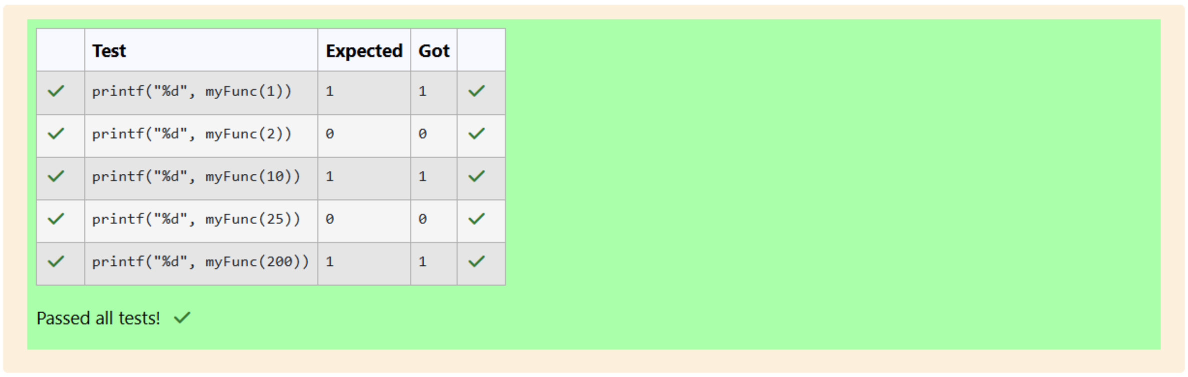
Output

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

Source code:

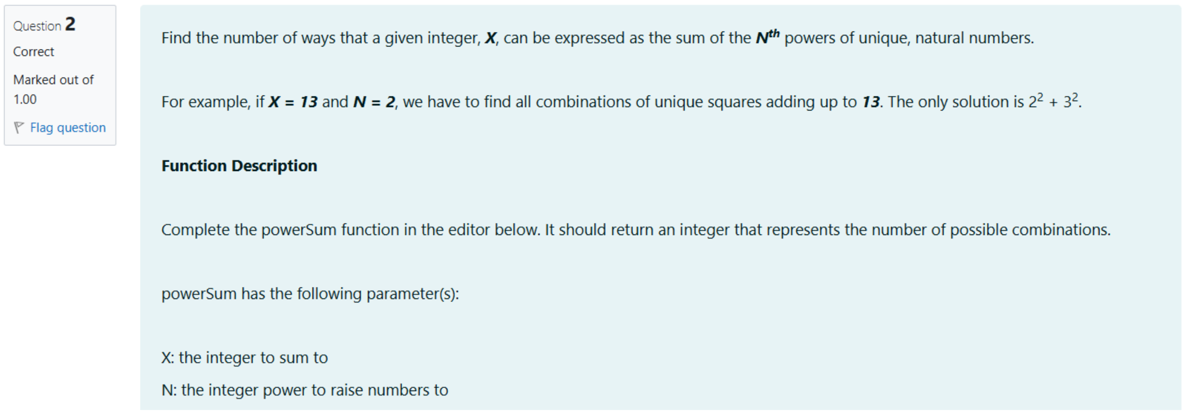

```c
/*
 * Complete the 'myFunc' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER n as parameter.
 */

int myFunc(int n)
{
    while(n>1)
    {
        if(n%20==0)
        {
            n/=20;
        }
        else if(n%10==0)
        {
            n/=10;
        }
        else
        {
            return 0;
        }
    }
    return (n==1) ? 1:0;
}

```

Result:



	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! ✓



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

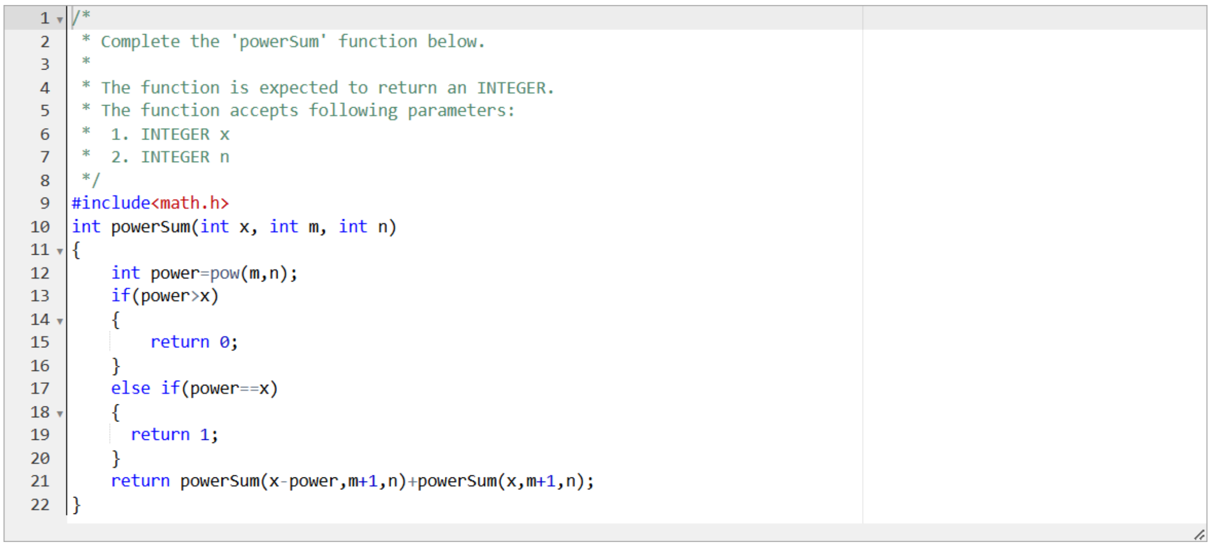
Complete the powerSum function in the editor below. It should return an integer that represents the number of possible combinations.

powerSum has the following parameter(s):

X: the integer to sum to

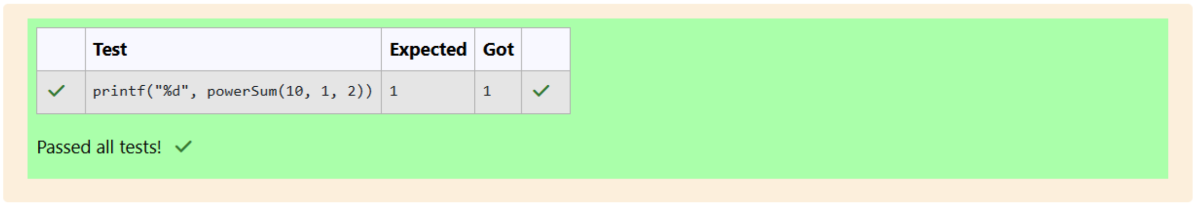
N: the integer power to raise numbers to

Source code:



```
/*
 * Complete the 'powerSum' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts following parameters:
 *  1. INTEGER x
 *  2. INTEGER n
 */
#include<math.h>
int powerSum(int x, int m, int n)
{
    int power=pow(m,n);
    if(power>x)
    {
        return 0;
    }
    else if(power==x)
    {
      return 1;
    }
    return powerSum(x-power,m+1,n)+powerSum(x,m+1,n);
}
```

Result:



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

Passed all tests! ✓