Answer: (penalty regime: 0 %)

Reset answer

```c
/*
 * Complete the 'myFunc' function below.
 *
 * The function is expected to return an
 * The function accepts INTEGER n as para
 */
#include<stdio.h>
int myFunc(long long n)
{
    while(n>1){
        if(n%10!=0){
            return 0;
        }
        if(n%20==0){
            n/=20;
        }else {
            n/=10;
        }
    }
    return (n==1)?1:0;

}
int my_main()
{
    long long n;
    scanf("%lld",&n);
    if(myFunc(n)){
        printf("1\n");
    }else{
        printf("0\n");
    }
    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! ✓

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
/*
 * Complete the 'powerSum' function below
 *
 * The function is expected to return an
 * The function accepts following paramet
 *  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;
    }
    if(power==x)
    {
        return 1;
    }
    return powerSum(x-power,m+1,n)+powerS
}
```

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

Passed all tests! ✓