

## A power B

You are given a function `add(a, b)` that computes the value of  $a+b \bmod m$ , i.e the remainder when  $a+b$  is divided by the number  $m$ .

You do not know the value of  $m$

. It is guaranteed that the hidden number  $m$  is between 1 and 109 (inclusive).

Complete the function `power(a, b)` that should compute the value of  $ab \bmod m$ , i.e. the remainder when  $ab$  is divided by the number  $m$  using the given function `add(a, b)`.

You cannot call the function `add(a, b)` more than 104 times in your code.

Whenever you call the function `add(a, b)`, both the arguments,  $a, b$  must be **less than** 109

Refer the given [solution template](#), and fill in the function `power(a, b)`.

It is possible to figure out the value of  $m$

, can you write a program that does not take more than  $\lg m$  calls to the function in order to calculate  $m$

?

Can you think of ways to calculate  $ab$

using less than  $a * b$  queries without calculating the value of  $m$

?

## Instructions

Refer the given [solution template](#), and fill in the function `power(a, b)`, do not modify any of the functions given there. Use the function `add(a, b)` to perform addition modulo  $m$ .

You have to implement the function `pwr(a, b)` using the function `add(a, b)`

## Constraints

- $1 \leq b \leq 100$
- $1 \leq a \leq 100$
- $1 \leq m \leq 109$

- (the hidden number)
- $0 \leq$

arguments of the function `add`  $\leq 109$ , i.e whenever you call the function, the values of the arguments should be between 0 and 109

- (inclusive).

## Sample Solution

[sample solution](#).

## Local Tester

[Local Tester](#) is a python program that you can use to test your program locally.

Copy your function and replace it with the empty one declared in the tester and run it on your computer as you would run any python program.

## Explanation

The given sample solution shows how you can utilise the `add` function to compute  $2a \bmod m$  using 1 call to the given `add` function. It is given only to demonstrate how to use the function `add` to compute other functions.

## Judging

If the output of the function is not correct, or you called the `add` function more than 104 times then you will receive a wrong answer.