GCD Strings

Attempted by: 1461 / Accuracy: 48% / Maximum Score: 20 / 22 Votes

Tag(s): Easy, Math, Recursion

Let $P[0 \ldots N-1]$ be a binary string of length *N*. Then let's define $S^{\infty}(P)$ as an infinite string with $S^{\infty}[i] = P[i\%N] \; \forall \; i \geq 0$ (informally, $S^{\infty}(P)$ is the concatenation of *P* with itself an infinite number of times).

Define the **GCD-string** of two integers a, b, with $a \geq b$ to be a binary string of length *a* that satisfies the following:

- $g(a, b)$ = $100 \ldots 000$ (*1* followed by $a - 1$ zeros) if *a* is divisible by *b*
- $g(a, b)$ = First *a* characters of $S^{\infty}(g(b, a \; mod \; b))$ otherwise

We can define $F(a, b)$ to be the value of the integer represented by the binary string $g(a, b)$ in **base-2**. Given *T* pairs of integers (x, y), compute $F(x, y) \; mod \; 10^9 + 7$ for each pair.

Input Format:

The first line will contain the number of test cases *T*.
Each test case can be described with a single line containing two integers x, y.

Output Format:

Output *T* numbers, the answers to each problem.

Constraints

For all subtasks:
$T \leq 10^4$
$1 \leq y \leq x$

File 1 (70 pts)
$x \leq 100$

File 2 (30 pts)
$x \leq 10^9$

SAMPLE INPUT

```
5
3 1
3 2
5 2
10 4
100 3
```

SAMPLE OUTPUT

```
4
5
21
546
986497880
```


Explanation

The base 2 results for the first four samples are as follows

1. 100
2. 101
3. 10101
4. 1000100010

Time Limit:	2.0 sec(s) for each input file.
Memory Limit:	256 MB
Source Limit:	1024 KB
Marking Scheme:	Marks are awarded when all the testcases pass.
Allowed Languages:	C, C++, C++14, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Swift, Swift-4.1, Visual Basic

CODE EDITOR

Enter your code or Upload your code as file. Save Python 3 (python 3.5.2)

```
'''
# Sample code to perform I/O:

name = input()                  # Reading input from STDIN
print('Hi, %s.' % name)         # Writing output to STDOUT

# Warning: Printing unwanted or ill-formatted data to output will cause the test
```

```
'''

# Write your code here

```

3

LIVE EVENTS

1:1

Provide custom input

COMPILE & TEST SUBMIT

Your Rating: Like 1 Share Tweet

PROGRAMMERS WHO SOLVED THIS PROBLEM ALSO SOLVED

A Tryst With Chess

Attempted By: **1449** / Accuracy: **81**

Its Confidential

Attempted By: **440** / Accuracy: **51**

N-Queens

Attempted By: **1811** / Accuracy: **89**