

## Problem A. One-time passwords

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            2 seconds  
Memory limit:         256 megabytes

Nowadays two-factor authentication, when user is required to use primary password and one or more one-time passwords, is becoming more widespread. Consider one of possible ways to generate such kind of passwords.

Let  $F(Q)$  be a number of positive integers not greater than  $Q$ , which can be represented as  $2^x - 2^y$  when  $x, y$  are non-negative integer numbers. Consider all possible numbers  $Q$  such as  $F(Q) = N$  and sort them in ascending order by number of one-bits in their binary representation. If two numbers have the same number of one-bits in binary representation, they should be compared by their values. Proposed algorithm chooses  $K$ -th number in this sorted sequence.

You are required to find one-time passwords for  $T$  authentication sessions.

### Input

First line contains an integer number  $T$  — number of authentication sessions. Next  $T$  lines contain two numbers  $N_i$  and  $K_i$  each — parameters of one-time password generation algorithms.

$$1 \leq T \leq 10^5$$
$$1 \leq N_i, K_i \leq 10^{18}$$

### Output

$T$  lines containing one integer number each — one-time password for corresponding authentication session. Each password should be computed in modulo  $10^9 + 7$ . If it is impossible to generate one time password, -1 should be printed.

### Example

standard input	standard output
1	42
16 10	