Description

The well-known Fibonacci sequence is: $F_i = F_{i-1} + F_{i-2}$ for $i \geq 2$, $F_0 = 0$, $F_1 = 1$. Tom discovers that the Fibonacci number grows very quickly, for example $F_{40} = 102334155$. To make further discovery of the Fibonacci numbers, Tom takes the following steps:

1. Take the first $\boldsymbol{n}$ Fibonacci numbers (exclude F_0) $S_1 = \{F_1, F_2, \ldots, F_n\}$
2. Modulo each Fibonacci number by a positive integer Q, i.e. $A_i = F_i \% Q$ and obtain a new sequence $S_2 = \{A_1, A_2, \ldots, A_n\}$
3. Sort the numbers in S_2 from small to large and obtain sequence S_3
   $S_2 = \{A_1, A_2, \ldots, A_n\} \rightarrow S_3 = \{c_1, c_2, \ldots, c_n\}$
4. For numbers in sequence S_3, calculate the weighted sum modular Q
   $(\sum_{k=1}^{n} k \cdot c_k) \% Q = (1 \cdot c_1 + 2 \cdot c_2 + 3 \cdot c_3 + \cdots + n \cdot c_n) \% \boldsymbol{Q}$

Can you write a program to calculate the result?

Input

The input contains multiple test cases. The first line of the input is a number T ($1 \leq T \leq 100$), indicating the number of test cases. Each test case contains two integers n ($2 \leq n \leq 5{,}000{,}000$) and Q ($2 \leq Q \leq 1000{,}000{,}000$) in one line.

Output

For each test case, print the weighted sum in a separate line.

Sample

Sample input

```
4
5 100
5 3
15 13
5000000 1000000000
```

Sample output

```
15 13
5000000 1000000000
```

Sample output

```
46
2
11
973061125
```

Explanation: In the second sample: the first 5 Fibonacci numbers are {1, 1, 2, 3, 5}, after modular 3 it becomes {1, 1, 2, 0, 2} and after sorting it is {0,1,1,2,2}, hence the weighted sum is

$$0 \cdot 1 + 1 \cdot 2 + 1 \cdot 3 + 2 \cdot 4 + 2 \cdot 5 = 23$$

After modular 3 it is 23 % 3 = 2.

Constraint and hint

radix sort

C++ 17
GCC 10.2.0

C++ (NOI)
GCC 4.8.4 (NOILinux 1.4.1)

C++ 11 (NOI)
GCC 4.8.4 (NOILinux 1.4.1)

C++ 11 (Clang)
Clang 12.0.1

C++ 17 (Clang)
Clang 12.0.1

C
Clang 12.0.1

C (NOI)

1

Or upload the code file: Choose File No file chosen