Check factorization

Your friend build a fantastic factoring algorithm, and challenge you to check his results.

Assume that number $N = p_0^{e_0} \times p_1^{e_1} \times \cdots p_k^{e_k}$, where p_i are prime numbers, and e_i are positive intergers.

You will be given a prime factorization of N, you'll have to print $N \pmod m$.

Input

The first line of the input consist of a single integer number *t* which determines the number of tests.

Each test is on 2 separate lines.

In each test,

- on the first line, there is two integer numbers k, and m.
- on the second line, there is $2(k+1)$ integer numbers p_i and e_i, with p_i a prime number.

Constraints

- $0 < t \leqslant 256$;
- $0 \leqslant k \leqslant 1000$;
- $0 < m \leqslant 2\times10^9$;
- $1 < p_i < 2\times10^9$, a prime number ;
- $0 < e_i < 2\times10^9$.

Output

For each test case, print $N \pmod m$.

Example

Input:

```
3
0 1000
17,1
2 100
2,1 5,1 7,2
1 1000
3,1 1000000007,1
```

Output:

```
17
90
21
```

Explanation

For the first test case, $N = 17^1$, and $N \pmod {1000} = 17$.

For the second test case, $N = 2^1 \times 5^1 \times 7^2 = 490$, and $N \pmod {100} = 90$.

For the third test case, $N = 3^1\times 1000000007^1 = 3000000021$, and $N \pmod {1000} = 21$.