E. On Iteration of One Well-Known Function

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Of course, many of you can calculate $\varphi(n)$ — the number of positive integers that are less than or equal to n, that are coprime with n. But what if we need to calculate $\varphi(\varphi(\ldots\varphi(n)))$, where function φ is taken k times and n is given in the canonical decomposition into prime factors?

You are given n and k, calculate the value of $\varphi(\varphi(\ldots\varphi(n)))$. Print the result in the canonical decomposition into prime factors.

Input

The first line contains integer m $(1 \le m \le 10^5)$ — the number of distinct prime divisors in the canonical representaion of n.

Each of the next m lines contains a pair of space-separated integers p_i, a_i $(2 \le p_i \le 10^6;\ 1 \le a_i \le 10^{17})$ — another prime divisor of number n and its power in the canonical representation. The sum of all a_i doesn't exceed 10^{17}. Prime divisors in the input follow in the strictly increasing order.

The last line contains integer k $(1 \le k \le 10^{18})$.

Output

In the first line, print integer w — the number of distinct prime divisors of number $\varphi(\varphi(\ldots\varphi(n)))$, where function φ is taken k times.

Each of the next w lines must contain two space-separated integers q_i, b_i $(b_i \ge 1)$ — another prime divisor and its power in the canonical representaion of the result. Numbers q_i must go in the strictly increasing order.

Examples

input

```
1
7 1
1
```

output

```
2
2 1
3 1
```

input

```
1
7 1
2
```

output

```
1
2 1
```

input

```
1
2 100000000000000000
10000000000000000
```

output

```
1
2 900000000000000000
```

Note

You can read about canonical representation of a positive integer here: http://en.wikipedia.org/wiki/Fundamental_theorem_of_arithmetic.

You can read about function $\varphi(n)$ here: http://en.wikipedia.org/wiki/Euler's_totient_function.