Recurrence Equation Finder

Many problems have solutions involving linear recurrence equations of the form **f(n) = a · f(n-1) + b · f(n-2) (n ≥ 2)**. Usually the coefficients **a** and **b** are between 0 and 10, so it would be useful to have a program which checks if some given values can be produced by such a recurrence equation. Since the growth of the values **f(n)** can be exponential, we will consider the values modulo some integer constant **k**.

More specifically you will be given **f(0)**, **f(1)**, **k** and some value pairs (**i** , $\mathbf{x_i}$), where $\mathbf{x_i}$ is the remainder of the division of **f(i)** by **k**.

You have to determine coefficients **a** and **b** for the recurrence equation **f** such that for each given value pair (**i**, $\mathbf{x_i}$) the equation $\mathbf{x_i = f(i) \bmod k}$ holds.

Hints

You can write the recurrence equation as follows:

$$\begin{pmatrix} ab \\ 10 \end{pmatrix} \cdot \begin{pmatrix} f(n-1) \\ f(n-2) \end{pmatrix} = \begin{pmatrix} f(n) \\ f(n-1) \end{pmatrix}$$

Let $A := \begin{pmatrix} ab \\ 10 \end{pmatrix}$

Then, $A^n \cdot \begin{pmatrix} f(1) \\ f(0) \end{pmatrix} = \begin{pmatrix} f(n+1) \\ f(n) \end{pmatrix}$. These equations also apply if everything is calculated modulo **k**.

To speed up the calculation of A^n, the identity $A^n = (A^{n \text{ div } 2})^2 \cdot A^{n \bmod 2}$ may be used. Also, $(a \cdot b) \bmod c = ((a \bmod c) \cdot (b \bmod c)) \bmod c$.

Input

The first line of the input contains a number **T ≤ 20** which indicates the number of test cases to follow.

Each test case consists of 3 lines. The first line of each test case contains the three integers **f(0)**, **f(1)** and **k**, where **2 ≤ k ≤ 10000** and **0 ≤ f(0),f(1) < k**. The second line of each test case contains a number **m ≤ 10** indicating the number of value pairs in the next line. The third line of each test case contains **m** value pairs (**i,**$\mathbf{x_i}$), where **2 ≤ i ≤ 1000000000** and **0 ≤** $\mathbf{x_i}$ **< k**.

Output

For each test case print one line containing the values **a** and **b** separated by a space character, where **0 ≤ a,b ≤ 10**. You may assume that there is always a unique solution.

Example

Input:

```
2
1 1 1000
```

3
2 2 3 3 16 597
0 1 10000
4
11 1024 3 4 1000000000 4688 5 16

Output:
1 1
2 0