Security System

Jose is a very rich man who loves to collect precious pieces. As he is not naive, all their riches are well kept in banks he trust. However, from time to time, he keeps in his home some valuable objects to show to friends and relatives. Jose knows that is a big risk to keep items of great value stored at home, so he came up with a security system to confuse anyone trying to find something valuable in his home.

At Joses house are **C** safes, all with numerical combinations. The valuable piece is stored in one of these safes. Jose sets a six-digit numeric password for one of the safes. The security system (common to the safes) sums the digits of the password and displays the result on the display box that received the combination. Jose, then know that to find the safe where the part is stored, the person should go to the cashier which number is the remainder from dividing the net profit shown on the display of the safe amount of current for the safe house (**C**). The system records for each of the **C** - 1 coffers remaining two numbers, A and B, which are coefficients of a linear equation of the form **Y** = **AX** + **B**.

The number **X** is inserted as a combination and its value should be the number printed on the display above the vault. The remainder of the division of the value of **Y** by the amount of safe house indicates the number of the next safe to be "visited".

Starting from the first vault, the piece should be safe if the ith the diagram is followed properly. So anyone who wants to find the piece must know the first safe, their combination and go to the ith safe, where is the piece.

Consider any safe only opens if it is the **i**th in the sequence of trials, that is, if you get the number produced by the **i**th a-safe. Therefore, if the safe of matches in which the piece is stored be visited before the ith attempt, he will not open and the person will not know that the piece is there.

As the coefficients of the safes are constantly changed by the security system, Joseph did not want to have to follow the circuit every time you want to open the safe in which the piece is saved (it does not know what the value of **Y** that opens the safe that kept the piece). His work is therefore to write a program for the security module that, given a safe home, their combination, the safe amount of intermediate circuit and the coefficients **A** and **B** of each of the boxes (known by the system), discover the safe in which the part is stored and the combination that opens.

Input

The input consists of several test cases. Each test case consists of several lines as described below. The first line of a test case consists of three integers separated by single spaces: ***C***, ***P*** and ***I***, representing the number of vaults, the number of the first safe vaults and the amount of the intermediate circuit (not counting the first and last) , respectively. Consider: $3 \le C \le 10$, $0 < I \le C - 2$. Each of the following **C** lines contains two integers separated by a single space representing the coefficients ***A*** and ***B*** of a safe. Consider the lines represent the coffers of coefficients ordered from *0* to ***C***-*1*. Consider also that $0 \le A, B \le 100$. The last line of the test case containing a string of six digits (each digit can take values from *0* to *9*) which represents the combination of the safe **P**.

The last line of the input file contains three zeros separated by single spaces.

Output

For each test case your program should print one line containing the number of safe in which the piece is stored and open the final combination separated by a single space.

Example

Input:

```
6 3 2
10 25
4 45
10 99
8 7
0 0
1 81
908710
10 5 8
100 100
100 100
100 100
100 100
100 100
100 100
100 100
100 100
100 100
100 100
999999
0 0 0
```

Output:

```
1 625
0 550101010101010100
```