

What Time Is It Mr. Fox?

Problem ID: whattimeisitmrfox
Time limit: 8 seconds

The game *What Time Is It Mr. Fox?* is a popular children's game. One child is the fox, the rest are the sheep. The game is on a field that is much longer than it is wide, so we will just assume it is played on a line that is L micrometers long. Position 0 is the left side and position L is the right side.

The sheep are initially located at position 0 on the line and the fox is initially located at position $L/2$. The game proceeds as follows:

- The sheep call out "What time is it Mr. Fox?".
- The fox announces the hour, either X o'clock where $X \in \{1, 2, \dots, 11\}$ or *lunch time!*.
- If the fox announced X o'clock, then all sheep take X steps to the right and the game proceeds with another call.
- But if the fox announced *lunch time!*, the sheep run to the side of the field they are closest to. If they are exactly halfway across the field, they run to the right side.
- The fox chooses at most one of the sheep to chase. If they catch the sheep before the sheep reaches the side it is running to, that sheep is out for the rest of the game.
- All sheep who make it to the right side of the field are *safe* for the rest of the game and no longer play.
- All sheep who make it to the left side of the field are still in the game.
- After all sheep are done running, the fox returns to the center of the field and the game resumes but only with the sheep remaining on the left side.

Each sheep i has a different step size $s_{i,X}$ for each possible value of $X \in \{1, 2, \dots, 11\}$. So when the fox announces it is X o'clock then sheep i moves exactly $X \cdot s_{i,X}$ micrometers toward the right side of the field. If this brings them to the right side, they stop at position L and will be declared safe next time the fox calls *lunch time!*. These step distances might not be increasing for a sheep, for example a sheep may take two large steps at 2 o'clock but three tiny steps at 3 o'clock.

All sheep run at the same speed and the fox runs at exactly twice the speed of the sheep. So if the fox chooses to chase a sheep whose distance is D micrometers to the side they are running to, the fox will catch the sheep if $L < 4D$ and otherwise the sheep will reach the side without being caught.

Every *lunch time!*, the fox uses the following to decide which sheep to chase:

- If the fox can catch a sheep running to the right side, the fox will choose the one with the smallest index i .
- Otherwise, if the fox can catch a sheep running to the left side, the fox will choose the one with the smallest index i .
- Otherwise, the fox chooses to not chase any sheep so all sheep will successfully reach their side.

You will be given the sequence of calls made by the fox during the game. Your job is to determine which sheep reach the right side and which sheep is caught after each *lunch time!* call.

Input

The first line of input contains three integers N ($1 \leq N \leq 200\,000$), M ($1 \leq M \leq 200\,000$), and L ($2 \leq L \leq 10^8$) indicating the number of sheep, the number of instructions, and the length of the field. The sheep are numbered 1 to N , inclusive.

Each of the next N lines contains exactly 11 integers. The X -th value on the i -th line is $s_{i,X}$ ($1 \leq s_{i,X} \leq 10^8$), i.e. the step size sheep i takes forward whenever the fox calls X o'clock.

Finally, M lines follow where each line contains an integer from 1 to 12. The j -th such number indicates what the fox calls: either X o'clock if the number X is from 1 to 11 or *lunch time!* if $X = 12$.

You are also guaranteed that L is an even number, that there are at most 20 *lunch time!* calls, and that the last call is *lunch time!*

Output

For each *lunch time!* call, output two lines. The first line should contain a list of the numbers of the sheep that reached the right side after this call. These numbers should be given in ascending order. If no sheep reached the right side this call, output *None* instead. The second line should contain the number of the sheep that the fox caught this call. If the fox did not catch a sheep this call, output *None* instead.

Sample Input 1

```
2 6 16
1 1 1 1 1 1 1 1 1 1 1
1 2 1 2 1 2 1 2 1 2 1
4
1
12
6
6
12
```

Sample Output 1

```
None
2
1
None
```

Sample Input 2

```
2 4 10
1 1 1 1 1 1 1 1 1 1 1
1 2 1 2 1 2 1 2 1 2 1
2
4
10
12
```

Sample Output 2

```
1 2
None
```

Sample Input 3

```
3 5 20
1 1 1 1 1 1 1 1 1 1 1
1 2 1 2 1 2 1 2 1 2 1
3 3 3 3 3 3 3 3 3 3 3
10
12
10
10
12
```

Sample Output 3

```
2 3
1
None
None
```