Problem H
Hidden Permutation

When a fixed shuffle is applied to a deck of cards repeatedly, the deck will eventually return to its original order. The number of times the shuffle must be applied before this happens is called the *period* of the deck for that shuffle. In the present task, we consider the inverse question: rather than starting with a shuffle and finding its period, you are told, for each possible period, how many decks of cards have that period. Your task is to construct any shuffle consistent with this information.

Formally, there is a hidden permutation $f = (f_1, f_2, \ldots, f_N)$, where N is a given positive integer. A permutation is a list of numbers where each number from 1 to N occurs exactly once. Let $x = x_1 x_2 \ldots x_N$ be a binary string. We say that $f(x)$ (f applied to x) is the new binary string $x_{f_1} x_{f_2} \ldots x_{f_N}$, and the *period* of x is the smallest positive integer p such that

$$\underbrace{f(f(\cdots f(x)\cdots))}_{p \text{ times}} = x.$$

It can be proven that for any permutation f and any binary string x of the same length, the period of x exists. In other words, if you apply f to x enough times, you eventually get back x.

You are given the number N, and for each possible period p you are given the number of binary strings (out of all 2^N possible binary strings) that have period p. Your task is to find any permutation f that corresponds to the given information, or conclude that there is no such permutation.

Input

The input consists of:

- One line with integers N, K ($2 \leq N \leq 100$, $1 \leq K \leq 1000$), the length of the permutation and the number of possible periods.
- One line with K integers $p_1, p_2, \ldots, p_K$ ($1 \leq p_i \leq 10^9$). These are all the possible periods that binary strings of length N can have, with respect to the hidden permutation. The numbers p_i are all distinct, and are sorted in increasing order.
- One line with K integers $m_1, m_2, \ldots, m_K$ ($1 \leq m_i \leq 2^{100}$). This means that there are m_i binary strings with period p_i.

Note that the numbers m_i can be quite large!

Output

If there is no permutation f that corresponds to the given information, print `impossible`. Otherwise, print one line with N integers $f_1, f_2, \ldots, f_N$, the permutation that you chose. Recall that the permutation should contain every integer from 1 to N exactly once. If there are multiple solutions, you can print any one of them.

Sample Explanation

In the second sample, the input almost corresponds to the permutation $(2, 3, 4, \ldots, 91, 1)$. The only error is that the last digit in the last number of the input should be 0 instead of 1.

In the third sample, all 4 binary strings have period 1, which means that the identity permutation $(1, 2)$ is a valid answer.

Sample Input 1

```
7 6
1 2 3 4 6 12
4 4 12 24 12 72
```

Sample Output 1

```
2 3 1 5 6 7 4
```

Sample Input 2

```
91 4
1 7 13 91
2 126 8190 24758800785707605497982401 31
```

Sample Output 2

```
impossible
```

Sample Input 3

```
2 1
1
4
```

Sample Output 3

```
1 2
```