A. Fox And Names

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Fox Ciel is going to publish a paper on FOCS (Foxes Operated Computer Systems, pronounce: "Fox"). She heard a rumor: the authors list on the paper is always sorted in the *lexicographical* order.

After checking some examples, she found out that sometimes it wasn't true. On some papers authors' names weren't sorted in *lexicographical* order in normal sense. But it was always true that after some modification of the order of letters in alphabet, the order of authors becomes *lexicographical*!

She wants to know, if there exists an order of letters in Latin alphabet such that the names on the paper she is submitting are following in the *lexicographical* order. If so, you should find out any such order.

Lexicographical order is defined in following way. When we compare s and t, first we find the leftmost position with differing characters: $s_i \neq t_i$. If there is no such position (i. e. s is a prefix of t or vice versa) the shortest string is less. Otherwise, we compare characters s_i and t_i according to their order in alphabet.

Input

The first line contains an integer n ($1 \le n \le 100$): number of names.

Each of the following n lines contain one string $name_i$ ($1 \le |name_i| \le 100$), the i-th name. Each name contains only lowercase Latin letters. All names are different.

Output

If there exists such order of letters that the given names are sorted lexicographically, output any such order as a permutation of characters 'a'–'z' (i. e. first output the first letter of the modified alphabet, then the second, and so on).

Otherwise output a single word "Impossible" (without quotes).

Examples

input

```
3
rivest
shamir
adleman
```

output

```
bcdefghijklmnopqrsatuvwxyz
```

input

```
10
tourist
petr
wjmzbmr
yeputons
vepifanov
scottwu
oooooooooooooooo
subscriber
rowdark
tankengineer
```

output

```
Impossible
```

input

```
10
petr
egor
endagorion
feferivan
ilovetanyaromanova
kostka
dmitriyh
```

```
maratsnowbear
bredorjaguarturnik
cgyforever
```

output

```
aghjlnopefikdmbcqrstuvwxyz
```

input

```
7
car
care
careful
carefully
becarefuldontforgetsomething
otherwiseyouwillbehacked
goodluck
```

output

```
acbdefhijklmnogpqrstuvwxyz
```

B. Fox And Jumping

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Fox Ciel is playing a game. In this game there is an infinite long tape with cells indexed by integers (positive, negative and zero). At the beginning she is standing at the cell 0.

There are also n cards, each card has 2 attributes: length l_i and cost c_i. If she pays c_i dollars then she can apply i-th card. After applying i-th card she becomes able to make jumps of length l_i, i. e. from cell x to cell $(x - l_i)$ or cell $(x + l_i)$.

She wants to be able to jump to any cell on the tape (possibly, visiting some intermediate cells). For achieving this goal, she wants to buy some cards, paying as little money as possible.

If this is possible, calculate the minimal cost.

Input

The first line contains an integer n ($1 \le n \le 300$), number of cards.

The second line contains n numbers l_i ($1 \le l_i \le 10^9$), the jump lengths of cards.

The third line contains n numbers c_i ($1 \le c_i \le 10^5$), the costs of cards.

Output

If it is impossible to buy some cards and become able to jump to any cell, output -1. Otherwise output the minimal cost of buying such set of cards.

Examples

input

```
3
100 99 9900
1 1 1
```

output

```
2
```

input

```
5
10 20 30 40 50
1 1 1 1 1
```

output

```
-1
```

input

```
7
15015 10010 6006 4290 2730 2310 1
1 1 1 1 1 1 10
```

output

```
6
```

input

```
8
4264 4921 6321 6984 2316 8432 6120 1026
4264 4921 6321 6984 2316 8432 6120 1026
```

output

```
7237
```

Note

In first sample test, buying one card is not enough: for example, if you buy a card with length 100, you can't jump to any cell whose index is not a multiple of 100. The best way is to buy first and second card, that will make you be able to jump to any cell.

In the second sample test, even if you buy all cards, you can't jump to any cell whose index is not a multiple of 10, so you should output -1.

C. Fox And Dinner

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Fox Ciel is participating in a party in Prime Kingdom. There are n foxes there (include Fox Ciel). The i-th fox is a_i years old.

They will have dinner around some round tables. You want to distribute foxes such that:

1. Each fox is sitting at some table.
2. Each table has at least 3 foxes sitting around it.
3. The sum of ages of any two adjacent foxes around each table should be a prime number.

If k foxes f_1, f_2, ..., f_k are sitting around table in clockwise order, then for $1 \le i \le k - 1$: f_i and f_{i+1} are adjacent, and f_1 and f_k are also adjacent.

If it is possible to distribute the foxes in the desired manner, find out a way to do that.

Input

The first line contains single integer n ($3 \le n \le 200$): the number of foxes in this party.

The second line contains n integers a_i ($2 \le a_i \le 10^4$).

Output

If it is impossible to do this, output "Impossible".

Otherwise, in the first line output an integer m ($1 \le m \le \frac{n}{3}$): the number of tables.

Then output m lines, each line should start with an integer k -=– the number of foxes around that table, and then k numbers — indices of fox sitting around that table in clockwise order.

If there are several possible arrangements, output any of them.

Examples

input

```
4
3 4 8 9
```

output

```
1
4 1 2 4 3
```

input

```
5
2 2 2 2 2
```

output

```
Impossible
```

input

```
12
2 3 4 5 6 7 8 9 10 11 12 13
```

output

```
1
12 1 2 3 6 5 12 9 8 7 10 11 4
```

input

```
24
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
```

output

```
3
6 1 2 3 6 5 4
10 7 8 9 12 15 14 13 16 11 10
8 17 18 23 22 19 20 21 24
```

Note

In example 1, they can sit around one table, their ages are: 3-8-9-4, adjacent sums are: 11, 17, 13 and 7, all those integers are primes.

In example 2, it is not possible: the sum of $2+2 = 4$ is not a prime number.

D. Fox And Travelling

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Fox Ciel is going to travel to New Foxland during this summer.

New Foxland has n attractions that are linked by m undirected roads. Two attractions are called adjacent if they are linked by a road. Fox Ciel has k days to visit this city and each day she will visit exactly one attraction.

There is one important rule in New Foxland: you can't visit an attraction if it has more than one adjacent attraction that you haven't visited yet.

At the beginning Fox Ciel haven't visited any attraction. During her travelling she may move aribtrarly between attraction. After visiting attraction a, she may travel to **any** attraction b satisfying conditions above that hasn't been visited yet, even if it is not reachable from a by using the roads (Ciel uses boat for travelling between attractions, so it is possible).

She wants to know how many different travelling plans she can make. Calculate this number modulo $10^9 + 9$ for every k from 0 to n since she hasn't decided for how many days she is visiting New Foxland.

Input

First line contains two integers: n, m ($1 \le n \le 100$, $0 \le m \le \frac{n(n-1)}{2}$), the number of attractions and number of undirected roads.

Then next m lines each contain two integers a_i and b_i ($1 \le a_i, b_i \le n$ and $a_i \ne b_i$), describing a road. There is no more than one road connecting each pair of attractions.

Output

Output $n + 1$ integer: the number of possible travelling plans modulo $10^9 + 9$ for all k from 0 to n.

Examples

input
```
3 2
1 2
2 3
```
output
```
1
2
4
4
```

input
```
4 4
1 2
2 3
3 4
4 1
```
output
```
1
0
0
0
0
```

input
```
12 11
2 3
4 7
4 5
5 6
4 6
6 12
5 12
5 8
8 9
10 8
11 9
```
output
```
1
6
31
135
```

```
483
1380
3060
5040
5040
0
0
0
0
```

input

```
13 0
```

output

```
1
13
156
1716
17160
154440
1235520
8648640
51891840
259459200
37836791
113510373
227020746
227020746
```

Note

In the first sample test for $k = 3$ there are 4 travelling plans: $\{1, 2, 3\}, \{1, 3, 2\}, \{3, 1, 2\}, \{3, 2, 1\}$.

In the second sample test Ciel can't visit any attraction in the first day, so for $k > 0$ the answer is 0.

In the third sample test Foxlands look like this:

E. Fox And Polygon

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Fox Ciel just designed a puzzle game called "Polygon"! It is played using triangulations of a regular n-edge polygon. The goal is to transform one *triangulation* to another by some tricky rules.

Triangulation of an n-edge poylgon is a set of $n - 3$ diagonals satisfying the condition that no two diagonals share a common internal point.

For example, the initial state of the game may look like (a) in above figure. And your goal may look like (c). In each step you can choose a diagonal inside the polygon (but not the one of edges of the polygon) and *flip* this diagonal.

Suppose you are going to *flip* a diagonal $a – b$. There always exist two triangles sharing $a – b$ as a side, let's denote them as $a – b – c$ and $a – b – d$. As a result of this operation, the diagonal $a – b$ is replaced by a diagonal $c – d$. It can be easily proven that after *flip* operation resulting set of diagonals is still a *triangulation* of the polygon.

So in order to solve above case, you may first *flip* diagonal $6 – 3$, it will be replaced by diagonal $2 – 4$. Then you *flip* diagonal $6 – 4$ and get figure (c) as result.

Ciel just proved that for any starting and destination triangulations this game has a solution. She wants you to solve it in no more than $20\,000$ steps for any puzzle satisfying $n \le 1000$.

Input

The first line contain an integer n ($4 \le n \le 1000$), number of edges of the regular polygon.

Then follows two groups of $(n - 3)$ lines describing the original triangulation and goal triangulation.

Description of each triangulation consists of $(n - 3)$ lines. Each line contains 2 integers a_i and b_i ($1 \le a_i, b_i \le n$), describing a diagonal $a_i – b_i$.

It is guaranteed that both original and goal triangulations are correct (i. e. no two diagonals share a common internal point in both of these triangulations).

Output

First, output an integer k ($0 \le k \le 20,000$): number of steps.

Then output k lines, each containing 2 integers a_i and b_i: the endpoints of a diagonal you are going to flip at step i. You may output a_i and b_i in any order.

If there are several possible solutions, output any of them.

Examples

input

```
4
1 3
2 4
```

output

```
1
1 3
```

input

```
6
2 6
3 6
4 6
6 2
5 2
4 2
```

output

```
2
6 3
6 4
```

input

```
8
7 1
2 7
7 3
```

```
6 3
4 6
6 1
6 2
6 3
6 4
6 8
```

output

```
3
7 3
7 2
7 1
```

Note

Sample test 2 is discussed above and shown on the picture.