CODEFORCES β
Sponsored by Telegram

Codeforces Round #290 (Div. 2)

A. Fox And Snake

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

Fox Ciel starts to learn programming. The first task is drawing a fox! However, that turns out to be too hard for a beginner, so she decides to draw a snake instead.

A snake is a pattern on a n by m table. Denote c-th cell of r-th row as (r, c). The tail of the snake is located at $(1, 1)$, then it's body extends to $(1, m)$, then goes down 2 rows to $(3, m)$, then goes left to $(3, 1)$ and so on.

Your task is to draw this snake for Fox Ciel: the empty cells should be represented as dot characters ('.') and the snake cells should be filled with number signs ('#').

Consider sample tests in order to understand the snake pattern.

Input

The only line contains two integers: n and m ($3 \le n, m \le 50$).

n is an **odd** number.

Output

Output n lines. Each line should contain a string consisting of m characters. Do not output spaces.

Sample test(s)

input

```
3 3
```

output

```
###
..#
###
```

input

```
3 4
```

output

```
####
...#
####
```

input

```
5 3
```

output

```
###
..#
###
#..
###
```

input

```
9 9
```

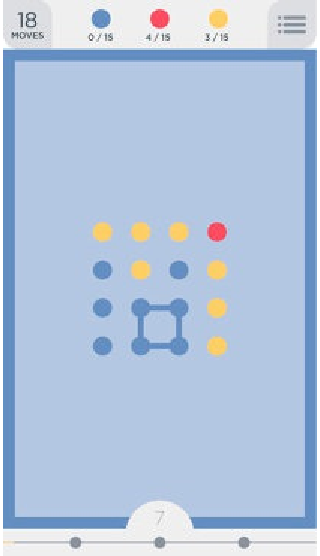
output

```
#########
........#
#########
#........
#########
........#
#########
#........
#########
```

B. Fox And Two Dots

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

Fox Ciel is playing a mobile puzzle game called "Two Dots". The basic levels are played on a board of size $n \times m$ cells, like this:



Each cell contains a dot that has some color. We will use different uppercase Latin characters to express different colors.

The key of this game is to find a cycle that contain dots of same color. Consider 4 blue dots on the picture forming a circle as an example. Formally, we call a sequence of dots $d_1, d_2, \ldots, d_k$ a *cycle* if and only if it meets the following condition:

1. These k dots are different: if $i \neq j$ then d_i is different from d_j.
2. k is at least 4.
3. All dots belong to the same color.
4. For all $1 \le i \le k - 1$: d_i and d_{i+1} are adjacent. Also, d_k and d_1 should also be adjacent. Cells x and y are called adjacent if they share an edge.

Determine if there exists a *cycle* on the field.

Input

The first line contains two integers n and m $(2 \le n, m \le 50)$: the number of rows and columns of the board.

Then n lines follow, each line contains a string consisting of m characters, expressing colors of dots in each line. Each character is an uppercase Latin letter.

Output

Output "Yes" if there exists a *cycle*, and "No" otherwise.

Sample test(s)

input

```
3 4
AAAA
ABCA
AAAA
```

output

```
Yes
```

input

```
3 4
AAAA
ABCA
AADA
```

output

```
No
```

input

```
4 4
YYYR
BYBY
BBBY
BBBY
```

```
output
```

```
Yes
```

```
input
```

```
7 6
AAAAAB
ABBBAB
ABAAAB
ABABBB
ABAAAB
ABBBAB
AAAAAB
```

```
output
```

```
Yes
```

```
input
```

```
2 13
ABCDEFGHIJKLM
NOPQRSTUVWXYZ
```

```
output
```

```
No
```

Note

In first sample test all 'A' form a cycle.

In second sample there is no such cycle.

The third sample is displayed on the picture above ('Y' = Yellow, 'B' = Blue, 'R' = Red).

C. 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 $(1 \leq n \leq 100)$: number of names.

Each of the following n lines contain one string $name_i$ $(1 \leq |name_i| \leq 100)$, 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).

Sample test(s)

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
```

D. 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.

Sample test(s)

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.

E. 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.

Sample test(s)

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.