Codeforces Beta Round #25 (Div. 2 Only)

A. IQ test

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

Bob is preparing to pass IQ test. The most frequent task in this test is to find out which one of the given n numbers differs from the others. Bob observed that one number usually differs from the others in evenness. Help Bob — to check his answers, he needs a program that among the given n numbers finds one that is different in evenness.

Input

The first line contains integer n $(3 \le n \le 100)$ — amount of numbers in the task. The second line contains n space-separated natural numbers, not exceeding 100. It is guaranteed, that exactly one of these numbers differs from the others in evenness.

Output

Output index of number that differs from the others in evenness. Numbers are numbered from 1 in the input order.

Sample test(s)

input
5 2 4 7 8 10
output
3

input
4 1 2 1 1
output
2

B. Phone numbers

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

Phone number in Berland is a sequence of n digits. Often, to make it easier to memorize the number, it is divided into groups of two or three digits. For example, the phone number `1198733` is easier to remember as `11-987-33`. Your task is to find for a given phone number any of its divisions into groups of two or three digits.

Input

The first line contains integer n ($2 \le n \le 100$) — amount of digits in the phone number. The second line contains n digits — the phone number to divide into groups.

Output

Output any of divisions of the given phone number into groups of two or three digits. Separate groups by single character `-`. If the answer is not unique, output any.

Sample test(s)

input
6 549871
output
54-98-71

input
7 1198733
output
11-987-33

C. Roads in Berland

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

There are n cities numbered from 1 to n in Berland. Some of them are connected by two-way roads. Each road has its own length — an integer number from 1 to 1000. It is known that from each city it is possible to get to any other city by existing roads. Also for each pair of cities it is known the shortest distance between them. Berland Government plans to build k new roads. For each of the planned road it is known its length, and what cities it will connect. To control the correctness of the construction of new roads, after the opening of another road Berland government wants to check the sum of the shortest distances between all pairs of cities. Help them — for a given matrix of shortest distances on the old roads and plans of all new roads, find out how the sum of the shortest distances between all pairs of cities changes after construction of each road.

Input

The first line contains integer n $(2 \le n \le 300)$ — amount of cities in Berland. Then there follow n lines with n integer numbers each — the matrix of shortest distances. j-th integer in the i-th row — $d_{i,j}$, the shortest distance between cities i and j. It is guaranteed that $d_{i,i} = 0$, $d_{i,j} = d_{j,i}$, and a given matrix is a matrix of shortest distances for some set of two-way roads with integer lengths from 1 to 1000, such that from each city it is possible to get to any other city using these roads.

Next line contains integer k $(1 \le k \le 300)$ — amount of planned roads. Following k lines contain the description of the planned roads. Each road is described by three space-separated integers a_i, b_i, c_i $(1 \le a_i, b_i \le n, a_i \ne b_i, 1 \le c_i \le 1000)$ — a_i and b_i — pair of cities, which the road connects, c_i — the length of the road. It can be several roads between a pair of cities, but no road connects the city with itself.

Output

Output k space-separated integers q_i $(1 \le i \le k)$. q_i should be equal to the sum of shortest distances between all pairs of cities after the construction of roads with indexes from 1 to i. Roads are numbered from 1 in the input order. Each pair of cities should be taken into account in the sum exactly once, i. e. we count unordered pairs.

Sample test(s)

input

```
2
0 5
5 0
1
1 2 3
```

output

```
3
```

input

```
3
0 4 5
4 0 9
5 9 0
2
2 3 8
1 2 1
```

output

```
17 12
```

D. Roads not only in Berland

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

Berland Government decided to improve relations with neighboring countries. First of all, it was decided to build new roads so that from each city of Berland and neighboring countries it became possible to reach all the others. There are n cities in Berland and neighboring countries in total and exactly $n - 1$ two-way roads. Because of the recent financial crisis, the Berland Government is strongly pressed for money, so to build a new road it has to close some of the existing ones. Every day it is possible to close one existing road and immediately build a new one. Your task is to determine how many days would be needed to rebuild roads so that from each city it became possible to reach all the others, and to draw a plan of closure of old roads and building of new ones.

Input

The first line contains integer n ($2 \le n \le 1000$) — amount of cities in Berland and neighboring countries. Next $n - 1$ lines contain the description of roads. Each road is described by two space-separated integers a_i, b_i ($1 \le a_i, b_i \le n, a_i \ne b_i$) — pair of cities, which the road connects. It can't be more than one road between a pair of cities. No road connects the city with itself.

Output

Output the answer, number t — what is the least amount of days needed to rebuild roads so that from each city it became possible to reach all the others. Then output t lines — the plan of closure of old roads and building of new ones. Each line should describe one day in the format `i j u v` — it means that road between cities `i` and `j` became closed and a new road between cities `u` and `v` is built. Cities are numbered from 1. If the answer is not unique, output any.

Sample test(s)

input
2 1 2

output
0

input
7 1 2 2 3 3 1 4 5 5 6 6 7

output
1 3 1 3 7

E. Test

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

Sometimes it is hard to prepare tests for programming problems. Now Bob is preparing tests to new problem about strings — input data to his problem is one string. Bob has 3 wrong solutions to this problem. The first gives the wrong answer if the input data contains the substring s_1, the second enters an infinite loop if the input data contains the substring s_2, and the third requires too much memory if the input data contains the substring s_3. Bob wants these solutions to fail single test. What is the minimal length of test, which couldn't be passed by all three Bob's solutions?

Input

There are exactly 3 lines in the input data. The i-th line contains string s_i. All the strings are non-empty, consists of lowercase Latin letters, the length of each string doesn't exceed 10^5.

Output

Output one number — what is minimal length of the string, containing s_1, s_2 and s_3 as substrings.

Sample test(s)

input
ab bc cd
output
4

input
abacaba abaaba x
output
11