A. Guilty — to the kitchen!

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

It's a very unfortunate day for Volodya today. He got bad mark in algebra and was therefore forced to do some work in the kitchen, namely to cook borscht (traditional Russian soup). This should also improve his algebra skills.

According to the borscht recipe it consists of n ingredients that have to be mixed in proportion $(a_1 : a_2 : \ldots : a_n)$ litres (thus, there should be $a_1 \cdot x, \ldots, a_n \cdot x$ litres of corresponding ingredients mixed for some non-negative x). In the kitchen Volodya found out that he has $b_1, \ldots, b_n$ litres of these ingredients at his disposal correspondingly. In order to correct his algebra mistakes he ought to cook as much soup as possible in a V litres volume pan (which means the amount of soup cooked can be between 0 and V litres). What is the volume of borscht Volodya will cook ultimately?

Input

The first line of the input contains two space-separated integers n and V ($1 \le n \le 20, 1 \le V \le 10000$). The next line contains n space-separated integers a_i ($1 \le a_i \le 100$). Finally, the last line contains n space-separated integers b_i ($0 \le b_i \le 100$).

Output

Your program should output just one real number — the volume of soup that Volodya will cook. Your answer must have a relative or absolute error less than 10^{-4}.

Examples

input

```
1 100
1
40
```

output

```
40.0
```

input

```
2 100
1 1
25 30
```

output

```
50.0
```

input

```
2 100
1 1
60 60
```

output

```
100.0
```

B. Game of chess unfinished

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

Once Volodya was at the museum and saw a regular chessboard as a museum piece. And there were only four chess pieces on it: two white rooks, a white king and a black king. "Aha, blacks certainly didn't win!", — Volodya said and was right for sure. And your task is to say whether whites had won or not.

Pieces on the chessboard are guaranteed to represent a correct position (every piece occupies one cell, no two pieces occupy the same cell and kings cannot take each other). Thus, your task is only to decide whether whites mate blacks. We would remind you that it means that the black king can be taken by one of the opponent's pieces at the moment and also it cannot move to an unbeaten position. A rook moves vertically or horizontally by any number of *free* cells (assuming there are no other pieces on its path), a king — to the adjacent cells (either by corner or by side). Certainly, pieces cannot leave the board. The black king might be able to take opponent's rooks at his turn (see sample 3).

Input

The input contains 4 space-separated piece positions: positions of the two rooks, the white king and the black king. Each position on 8×8 chessboard is denoted by two symbols — (`'a'` - `'h'`) and (`'1'` - `'8'`) — which stand for horizontal and vertical coordinates of the cell occupied by the piece. It is guaranteed, that no two pieces occupy the same cell, and kings cannot take each other.

Output

Output should contain one word: `"CHECKMATE"` if whites mate blacks, and `"OTHER"` otherwise.

Examples

input

```
a6 b4 c8 a8
```

output

```
CHECKMATE
```

input

```
a6 c4 b6 b8
```

output

```
OTHER
```

input

```
a2 b1 a3 a1
```

output

```
OTHER
```

C. Safe cracking

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

Right now you are to solve a very, very simple problem — to crack the safe. Four positive integers stand one by one on a circle protecting the safe. You know that to unlock this striking safe you have to make all four numbers equal to one. Operations are as follows: you may choose two adjacent numbers and increase both by one; you may choose two adjacent even numbers and divide both by two. Nothing else. Crack the safe!

Input

The single line of the input contains four space-separated integer positive numbers not greater than 10^9 each — four numbers on the circle in consecutive order.

Output

The output should contain "-1" (quotes for clarity) if the safe is secure, that is it's impossible to crack it. Otherwise, output should contain the sequence of operations (one operations per line) leading to unlocking the safe. You don't have to minimize the number of operations, but it should not exceed 1000. To make things clear, assume numbers stand on positions 1 through 4. Each operation is encoded by two symbols. If the following operation is dividing then first symbol is '/'; otherwise it's '+' (addition). The second symbol is the position of the first number in pair in consecutive order. (see samples for clarification).

If there are several solutions, output any of them.

Examples

input
```
1 1 1 1
```
output
```
```

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

input
```
3 3 1 1
```
output
```
+1
/1
/1
```

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

D. Strange town

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

Volodya has recently visited a very odd town. There are N tourist attractions in the town and every two of them are connected by a bidirectional road. Each road has some travel price (natural number) assigned to it and all prices are distinct. But the most striking thing about this town is that each city sightseeing tour has the same total price! That is, if we choose any city sightseeing tour — a cycle which visits every attraction exactly once — the sum of the costs of the tour roads is independent of the tour. Volodya is curious if you can find such price system with all road prices not greater than 1000.

Input

Input contains just one natural number ($3 \le N \le 20$) — the number of town attractions.

Output

Output should contain N rows containing N positive integer numbers each — the adjacency matrix of the prices graph (thus, j-th number in i-th row should be equal to the price of the road between the j-th and the i-th attraction). Diagonal numbers should be equal to zero. All numbers should not be greater than 1000. All prices should be positive and pairwise distinct. If there are several solutions, output any of them.

Examples

input

```
3
```

output

```
0 3 4
3 0 5
4 5 0
```

E. Baldman and the military

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Baldman is a warp master. He possesses a unique ability — creating wormholes! Given two positions in space, Baldman can make a wormhole which makes it possible to move between them in both directions. Unfortunately, such operation isn't free for Baldman: each created wormhole makes him lose plenty of hair from his head.

Because of such extraordinary abilities, Baldman has caught the military's attention. He has been charged with a special task. But first things first.

The military base consists of several underground objects, some of which are connected with bidirectional tunnels. There necessarily exists a path through the tunnel system between each pair of objects. Additionally, exactly two objects are connected with surface. For the purposes of security, a patrol inspects the tunnel system every day: he enters one of the objects which are connected with surface, walks the base passing each tunnel **at least** once and leaves through one of the objects connected with surface. He can enter and leave either through the same object, or through different objects. The military management noticed that the patrol visits some of the tunnels multiple times and decided to optimize the process. Now they are faced with a problem: a system of wormholes needs to be made to allow of a patrolling which passes each tunnel **exactly** once. At the same time a patrol is allowed to pass each wormhole any number of times.

This is where Baldman comes to operation: he is the one to plan and build the system of the wormholes. Unfortunately for him, because of strict confidentiality the military can't tell him the arrangement of tunnels. Instead, they insist that his system of portals solves the problem for any arrangement of tunnels which satisfies the given condition. Nevertheless, Baldman has some information: he knows which pairs of objects he can potentially connect and how much it would cost him (in hair). Moreover, tomorrow he will be told which objects (exactly two) are connected with surface. Of course, our hero decided not to waste any time and calculate the minimal cost of getting the job done for some pairs of objects (which he finds likely to be the ones connected with surface). Help Baldman!

Input

First line of the input contains a single natural number n ($2 \le n \le 100000$) — the number of objects on the military base. The second line — one number m ($1 \le m \le 200000$) — the number of the wormholes Baldman can make. The following m lines describe the wormholes: each line contains three integer numbers a, b, c ($1 \le a, b \le n, 1 \le c \le 100000$) — the numbers of objects which can be connected and the number of hair Baldman has to spend to make this wormhole.

The next line contains one natural number q ($1 \le q \le 100000$) — the number of queries. Finally, the last q lines contain a description of one query each — a pair of numbers of different objects a_i, b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$). There could be more than one wormhole between a pair of objects.

Output

Your program should output q lines, one for each query. The i-th line should contain a single integer number — the answer for i-th query: the minimum cost (in hair) of a system of wormholes allowing the optimal patrol for any system of tunnels (satisfying the given conditions) if a_i and b_i are the two objects connected with surface, or "-1" if such system of wormholes cannot be made.

Examples

input

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

output

```
0
```

input

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

output

```
-1
3
```