B. Little Pony and Harmony Chest

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

Princess Twilight went to Celestia and Luna's old castle to research the chest from the Elements of Harmony.

A sequence of positive integers b_i is harmony if and only if for every two elements of the sequence their greatest common divisor equals 1. According to an ancient book, the key of the chest is a harmony sequence b_i which minimizes the following expression:

$$\sum_{i=1}^{n} |a_i - b_i|.$$

You are given sequence a_i, help Princess Twilight to find the key.

Input

The first line contains an integer n ($1 \le n \le 100$) — the number of elements of the sequences a and b. The next line contains n integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 30$).

Output

Output the key — sequence b_i that minimizes the sum described above. If there are multiple optimal sequences, you can output any of them.

Sample test(s)

input
5 1 1 1 1 1
output
1 1 1 1 1

input
5 1 6 4 2 8
output
1 5 3 1 8

SPOJ Problem Set (classical)

9964. Fibonacci vs Polynomial

Problem code: PIBO

Define a sequence *Pib*(n) as following

- *Pib*(0) = 1
- *Pib*(1) = 1
- otherwise, *Pib*(n) = *Pib*(n-1) + *Pib*(n-2) + **P**(n)

Here **P** is a polynomial.

Given **n** and **P**, find *Pib*(n) modulo 1,111,111,111.

Input

First line of input contains two integer **n** and **d** ($0 \le \mathbf{n} \le 10^9$, $0 \le \mathbf{d} \le 100$), **d** is the degree of polynomial.

The second line contains **d**+1 integers $\mathbf{c}_0, \mathbf{c}_1 \ldots \mathbf{c}_d$, represent the coefficient of the polynomial(Thus **P**(x) can be written as $\Sigma \mathbf{c}_i x^i$). $0 \le \mathbf{c}_i \le 100$ and $\mathbf{c}_d \ne 0$ unless d = 0.

Output

A single integer represents the answer.

Example

Input:
```
10 0
0
```

Output:
```
89
```

Input:
```
10 0
1
```

Output:
```
177
```

Input:
```
100 1
1 1
```

Output:
```
343742333
```

show comments

© Spoj.com. All Rights Reserved. Spoj uses Sphere Engine™ © by Sphere Research Labs.

E. Little Pony and Lord Tirek

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

Lord Tirek is a centaur and the main antagonist in the season four finale episodes in the series "My Little Pony: Friendship Is Magic". In "Twilight's Kingdom" (Part 1), Tirek escapes from Tartarus and drains magic from ponies to grow stronger.

The core skill of Tirek is called Absorb Mana. It takes all mana from a magic creature and gives them to the caster.

Now to simplify the problem, assume you have n ponies (numbered from 1 to n). Each pony has three attributes:

- s_i : amount of mana that the pony has at time 0;
- m_i : maximum mana that the pony can have;
- r_i : mana regeneration per unit time.

Lord Tirek will do m instructions, each of them can be described with three integers: t_i, l_i, r_i. The instruction means that at time t_i, Tirek will use Absorb Mana on ponies with numbers from l_i to r_i (both borders inclusive). We'll give you all the m instructions in order, count how much mana Tirek absorbs for each instruction.

Input

The first line contains an integer n ($1 \le n \le 10^5$) — the number of ponies. Each of the next n lines contains three integers s_i, m_i, r_i ($0 \le s_i \le m_i \le 10^5$; $0 \le r_i \le 10^5$), describing a pony.

The next line contains an integer m ($1 \le m \le 10^5$) — the number of instructions. Each of the next m lines contains three integers t_i, l_i, r_i ($0 \le t_i \le 10^9$; $1 \le l_i \le r_i \le n$), describing an instruction of Lord Tirek. The instructions are given in strictly increasing order of t_i (all t_i are distinct).

Output

For each instruction, output a single line which contains a single integer, the total mana absorbed in this instruction.

Sample test(s)

input
5 0 10 1 0 12 1 0 20 1 0 12 1 0 10 1 2 5 1 5 19 1 5

output
25 58

Note

Every pony starts with zero mana. For the first instruction, each pony has 5 mana, so you get 25 mana in total and each pony has 0 mana after the first instruction.

For the second instruction, pony 3 has 14 mana and other ponies have mana equal to their m_i.

B. Little Pony and Sort by Shift

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day, Twilight Sparkle is interested in how to sort a sequence of integers $a_1, a_2, ..., a_n$ in non-decreasing order. Being a young unicorn, the only operation she can perform is a unit shift. That is, she can move the last element of the sequence to its beginning:

$$a_1, a_2, ..., a_n \rightarrow a_n, a_1, a_2, ..., a_{n-1}.$$

Help Twilight Sparkle to calculate: what is the minimum number of operations that she needs to sort the sequence?

Input

The first line contains an integer n ($2 \le n \le 10^5$). The second line contains n integer numbers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^5$).

Output

If it's impossible to sort the sequence output `-1`. Otherwise output the minimum number of operations Twilight Sparkle needs to sort it.

Sample test(s)

input
2 2 1
output
1

input
3 1 3 2
output
-1

input
2 1 2
output
0

C. Little Pony and Summer Sun Celebration

Twilight Sparkle learnt that the evil Nightmare Moon would return during the upcoming Summer Sun Celebration after one thousand years of imprisonment on the moon. She tried to warn her mentor Princess Celestia, but the princess ignored her and sent her to Ponyville to check on the preparations for the celebration.

Twilight Sparkle wanted to track the path of Nightmare Moon. Unfortunately, she didn't know the exact path. What she knew is the parity of the number of times that each place Nightmare Moon visited. Can you help Twilight Sparkle to restore any path that is consistent with this information?

Ponyville can be represented as an undirected graph (vertices are places, edges are roads between places) without self-loops and multi-edges. The path can start and end at any place (also it can be empty). Each place can be visited multiple times. The path must not visit more than $4n$ places.

Input

The first line contains two integers n and m ($2 \le n \le 10^5$; $0 \le m \le 10^5$) — the number of places and the number of roads in Ponyville. Each of the following m lines contains two integers u_i, v_i ($1 \le u_i, v_i \le n$; $u_i \neq v_i$), these integers describe a road between places u_i and v_i.

The next line contains n integers: $x_1, x_2, ..., x_n$ ($0 \le x_i \le 1$) — the parity of the number of times that each place must be visited. If $x_i = 0$, then the i-th place must be visited even number of times, else it must be visited odd number of times.

Output

Output the number of visited places k in the first line ($0 \le k \le 4n$). Then output k integers — the numbers of places in the order of path. If $x_i = 0$, then the i-th place must appear in the path even number of times, else i-th place must appear in the path odd number of times. Note, that given road system has no self-loops, therefore any two neighbouring places in the path must be distinct.

If there is no required path, output `-1`. If there multiple possible paths, you can output any of them.

Sample test(s)

input
3 2 1 2 2 3 1 1 1
output
3 1 2 3

input
5 7 1 2 1 3 1 4 1 5 3 4 3 5 4 5 0 1 0 1 0
output
10 2 1 3 4 5 4 5 4 3 1

input
2 0 0 0
output
0

A. Little Pony and Crystal Mine

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Twilight Sparkle once got a crystal from the Crystal Mine. A crystal of size n (n is odd; $n > 1$) is an $n \times n$ matrix with a diamond inscribed into it.

You are given an odd integer n. You need to draw a crystal of size n. The diamond cells of the matrix should be represented by character "D". All other cells of the matrix should be represented by character "*". Look at the examples to understand what you need to draw.

Input
The only line contains an integer n ($3 \le n \le 101$; n is odd).

Output
Output a crystal of size n.

Sample test(s)

input
3

output
D DDD *D*

input
5

output
D *DDD* DDDDD *DDD* **D**

input
7

output
D **DDD** *DDDDD* DDDDDDD *DDDDD* **DDD** ***D***

SPOJ Problem Set (classical)

13419. Modular Fibonacci Period

Problem code: PISANO

Perhaps the first thing one notices when the Fibonacci sequence is reduced mod M is that it seems periodic.

For example :
F (mod 4) = **0 1 1 2 3 1** 0 1 1 2 3 ...
F (mod 5) = **0 1 1 2 3 0 3 3 1 4 0 4 4 3 2 0 2 2 4 1** 0 1 1 2 3 ...

We define K(M) the period of the Fibonacci sequence reduced mod M if it is periodic.
We just saw that K(4) = 6 and K(5) = 20.

Input

The input begins with the number T of test cases in a single line.
In each of the next T lines there are one integer M.

Output

For each test case, on a single line, print K(M), or "Not periodic." without quotes if need.

Example

Input:
```
3
4
5
6
```

Output:
```
6
20
24
```

Constraints

```
1 < T < 10^4
1 < M < 10^12
```

show comments

© Spoj.com. All Rights Reserved. Spoj uses Sphere Engine™ © by Sphere Research Labs.

A. Little Pony and Expected Maximum

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Twilight Sparkle was playing Ludo with her friends Rainbow Dash, Apple Jack and Flutter Shy. But she kept losing. Having returned to the castle, Twilight Sparkle became interested in the dice that were used in the game.

The dice has m faces: the first face of the dice contains a dot, the second one contains two dots, and so on, the m-th face contains m dots. Twilight Sparkle is sure that when the dice is tossed, each face appears with probability . Also she knows that each toss is independent from others. Help her to calculate the expected maximum number of dots she could get after tossing the dice n times.

Input

A single line contains two integers m and n ($1 \leq m, n \leq 10^5$).

Output

Output a single real number corresponding to the expected maximum. The answer will be considered correct if its relative or absolute error doesn't exceed 10^{-4}.

Sample test(s)

input
6 1
output
3.500000000000

input
6 3
output
4.958333333333

input
2 2
output
1.750000000000

Note

Consider the third test example. If you've made two tosses:

1. You can get 1 in the first toss, and 2 in the second. Maximum equals to 2.
2. You can get 1 in the first toss, and 1 in the second. Maximum equals to 1.
3. You can get 2 in the first toss, and 1 in the second. Maximum equals to 2.
4. You can get 2 in the first toss, and 2 in the second. Maximum equals to 2.

The probability of each outcome is 0.25, that is expectation equals to:

You can read about expectation using the following link: `http://en.wikipedia.org/wiki/Expected_value`

E. Xenia and Tree

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Xenia the programmer has a tree consisting of n nodes. We will consider the tree nodes indexed from 1 to n. We will also consider the first node to be initially painted red, and the other nodes — to be painted blue.

The *distance* between two tree nodes v and u is the number of edges in the shortest path between v and u.

Xenia needs to learn how to quickly execute queries of two types:

1. paint a specified blue node in red;
2. calculate which red node is the closest to the given one and print the shortest distance to the closest red node.

Your task is to write a program which will execute the described queries.

Input

The first line contains two integers n and m ($2 \le n \le 10^5$, $1 \le m \le 10^5$) — the number of nodes in the tree and the number of queries. Next $n - 1$ lines contain the tree edges, the i-th line contains a pair of integers a_i, b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$) — an edge of the tree.

Next m lines contain queries. Each query is specified as a pair of integers t_i, v_i ($1 \le t_i \le 2$, $1 \le v_i \le n$). If $t_i = 1$, then as a reply to the query we need to paint a blue node v_i in red. If $t_i = 2$, then we should reply to the query by printing the shortest distance from some red node to node v_i.

It is guaranteed that the given graph is a tree and that all queries are correct.

Output

For each second type query print the reply in a single line.

Sample test(s)

input
5 4 1 2 2 3 2 4 4 5 2 1 2 5 1 2 2 5

output
0 3 2

B. Xenia and Spies

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

Xenia the vigorous detective faced n ($n \geq 2$) foreign spies lined up in a row. We'll consider the spies numbered from 1 to n from left to right.

Spy s has an important note. He has to pass the note to spy f. Xenia interrogates the spies in several steps. During one step the spy keeping the important note can pass the note to one of his neighbours in the row. In other words, if this spy's number is x, he can pass the note to another spy, either $x - 1$ or $x + 1$ (if $x = 1$ or $x = n$, then the spy has only one neighbour). Also during a step the spy can keep a note and not pass it to anyone.

But nothing is that easy. During m steps Xenia watches some spies attentively. Specifically, during step t_i (steps are numbered from 1) Xenia watches spies numbers l_i, $l_i + 1$, $l_i + 2$, ..., r_i ($1 \leq l_i \leq r_i \leq n$). Of course, if during some step a spy is watched, he can't do anything: neither give the note nor take it from some other spy. Otherwise, Xenia reveals the spies' cunning plot. Nevertheless, if the spy at the current step keeps the note, Xenia sees nothing suspicious even if she watches him.

You've got s and f. Also, you have the steps during which Xenia watches spies and which spies she is going to watch during each step. Find the best way the spies should act in order to pass the note from spy s to spy f as quickly as possible (in the minimum number of steps).

Input

The first line contains four integers n, m, s and f ($1 \leq n, m \leq 10^5$; $1 \leq s, f \leq n$; $s \neq f$; $n \geq 2$). Each of the following m lines contains three integers t_i, l_i, r_i ($1 \leq t_i \leq 10^9$, $1 \leq l_i \leq r_i \leq n$). It is guaranteed that $t_1 < t_2 < t_3 < ... < t_m$.

Output

Print k characters in a line: the i-th character in the line must represent the spies' actions on step i. If on step i the spy with the note must pass the note to the spy with a lesser number, the i-th character should equal "L". If on step i the spy with the note must pass it to the spy with a larger number, the i-th character must equal "R". If the spy must keep the note at the i-th step, the i-th character must equal "X".

As a result of applying the printed sequence of actions spy s must pass the note to spy f. The number of printed characters k must be as small as possible. Xenia must not catch the spies passing the note.

If there are miltiple optimal solutions, you can print any of them. It is guaranteed that the answer exists.

Sample test(s)

input
3 5 1 3 1 1 2 2 2 3 3 3 3 4 1 1 10 1 3
output
XXRR