Problem A. Optimal Currency Exchange

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Andrew was very excited to participate in Olympiad of Metropolises. Days flew by quickly, and Andrew is already at the airport, ready to go home. He has n rubles left, and would like to exchange them to euro and dollar bills. Andrew can mix dollar bills and euro bills in whatever way he wants. The price of one dollar is d rubles, and one euro costs e rubles.

Recall that there exist the following dollar bills: 1, 2, 5, 10, 20, 50, 100, and the following euro bills — 5, 10, 20, 50, 100, 200 (note that, in this problem we do **not** consider the 500 euro bill, it is hard to find such bills in the currency exchange points). Andrew can buy any combination of bills, and his goal is to minimize the total number of rubles he has.

Help him — write a program that given integers n, e and d, finds the minimum number of rubles Andrew can get after buying dollar and euro bills.

Input

The first line of the input contains one integer n ($1 \le n \le 10^8$) — the initial sum in rubles Andrew has.

The second line of the input contains one integer d ($30 \le d \le 100$) — the price of one dollar in rubles.

The third line of the input contains integer e ($30 \le e \le 100$) — the price of one euro in rubles.

Output

Output one integer — the minimum number of rubles Andrew can have after buying dollar and euro bills optimally.

Examples

standard input	standard output
100 60 70	40
410 55 70	5
600 60 70	0

Note

In the first sample, we can buy just 1 dollar because there is no 1 euro banknote.

In the second sample, optimal exchange is to buy 5 euro and 1 dollar.

In the third sample, optimal exchange is to buy 10 dollars in one bill.

Scoring

This problem has 25 tests. Each test **including samples** costs 4 points and is scored independently.

Solutions that succeed on test cases where $n \le 100$ will get at least 20 points. Solutions that succeed on test cases where $n \le 1000$ will get at least 40 points.

Problem B. Petya and Construction Set

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

It's Petya's birthday party and his friends have presented him a brand new "Electrician-n" construction set, which they are sure he will enjoy as he always does with weird puzzles they give him.

Construction set "Electrician-n" consists of $2n-1$ wires and $2n$ light bulbs. Each bulb has its own unique index that is an integer from 1 to $2n$, while all wires look the same and are indistinguishable. In order to complete this construction set one has to use each of the wires to connect two distinct bulbs. We define a *chain* in a completed construction set as a sequence of distinct bulbs of length at least two, such that every two consecutive bulbs in this sequence are directly connected by a wire. Completed construction set configuration is said to be correct if a resulting network of bulbs and wires has a tree structure, i.e. any two distinct bulbs are the endpoints of some chain.

Petya was assembling different configurations for several days, and he noticed that sometimes some of the bulbs turn on. After a series of experiments he came up with a conclusion that bulbs indexed $2i$ and $2i-1$ turn on if the chain connecting them consists of exactly d_i wires. Moreover, the following **important** condition holds: the value of d_i is never greater than n.

Petya did his best but was not able to find a configuration that makes all bulbs to turn on, so he seeks your assistance. Please, find out a configuration that makes all bulbs shine. It is guaranteed that such configuration always exists.

Input

The first line of the input contains a single integer n ($1 \le n \le 100\,000$) — the parameter of a construction set that defines the number of bulbs and the number of wires.

Next line contains n integers $d_1, d_2, \ldots, d_n$ ($1 \le d_i \le n$), where d_i stands for the number of wires the chain between bulbs $2i$ and $2i-1$ should consist of.

Output

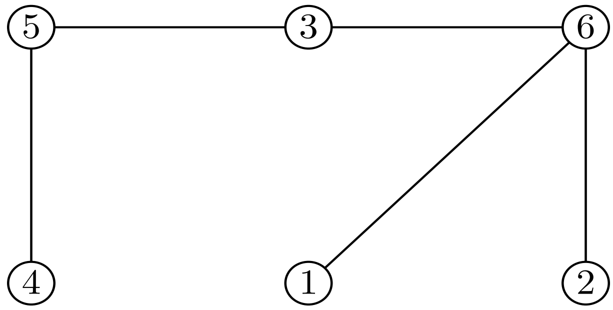
Print $2n-1$ lines. The i-th of them should contain two distinct integers a_i and b_i ($1 \le a_i, b_i \le 2n$, $a_i \ne b_i$) — indices of bulbs connected by a wire.

If there are several possible valid answer you can print any of them.

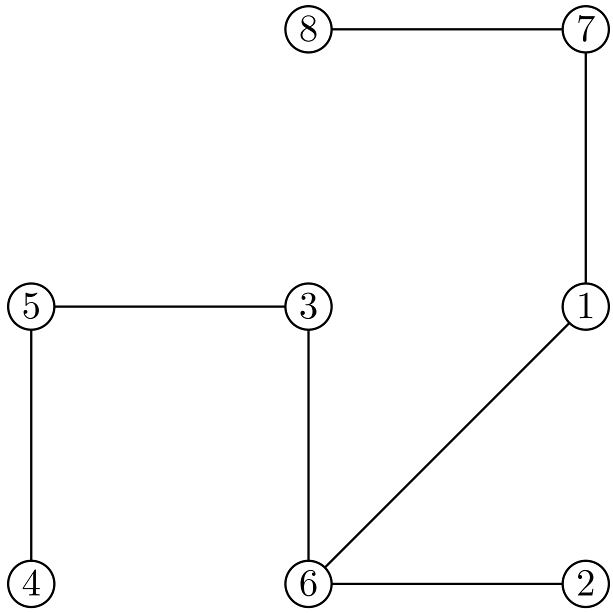
Examples

standard input	standard output
3 2 2 2	1 6 2 6 3 5 3 6 4 5
4 2 2 2 1	1 6 1 7 2 6 3 5 3 6 4 5 7 8

Note



Answer for the first sample test.



Answer for the second sample test.

Scoring

Tests for this problem are divided into five groups. For each of the groups you earn points only if your solution passes all tests in this group. For last group you earn points if your solution passes all tests of all groups.

Group	Points	Additional constaints	Comment
		n	
0	0	–	Samples
1	11	$n \le 10$	$d_i = n$
2	16	$n \le 100$	$d_i \le 2$
3	17	$n \le 1000$	All d_i are distinct
4	20	–	All d_i are equal
5	36	–	

Problem C. Employment

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Two large companies "Cecsi" and "Poca Pola" are fighting against each other for a long time. In order to overcome their competitor, "Poca Pola" started a super secret project, for which it has total n vacancies in all of their offices. After many tests and interviews n candidates were selected and the only thing left was their employment.

Because all candidates have the same skills, it doesn't matter where each of them will work. That is why the company decided to distribute candidates between workplaces so that the total distance between home and workplace over all candidates is minimal.

It is well known that Earth is round, so it can be described as a circle, and all m cities on Earth can be described as points on this circle. All cities are enumerated from 1 to m so that for each i $(1 \le i \le m-1)$ cities with indexes i and $i+1$ are neighbors and cities with indexes 1 and m are neighbors as well. People can move only along the circle. The distance between any two cities equals to minimal number of transitions between neighboring cities you have to perform to get from one city to another. In particular, the distance between the city and itself equals 0.

The "Poca Pola" vacancies are located at offices in cities $a_1, a_2, \ldots, a_n$. The candidates live in cities $b_1, b_2, \ldots, b_n$. It is possible that some vacancies are located in the same cities and some candidates live in the same cities.

The "Poca Pola" managers are too busy with super secret project, so you were asked to help "Poca Pola" to distribute candidates between workplaces, so that the sum of the distance between home and workplace over all candidates is minimum possible.

Input

The first line contains two integers m and n $(1 \le m \le 10^9,\ 1 \le n \le 200\,000)$ — the number of cities on Earth and the number of vacancies.

The second line contains n integers $a_1, a_2, a_3, \ldots, a_n$ $(1 \le a_i \le m)$ — the cities where vacancies are located.

The third line contains n integers $b_1, b_2, b_3, \ldots, b_n$ $(1 \le b_i \le m)$ — the cities where the candidates live.

Output

The first line should contain the minimum total distance between home and workplace over all candidates.

The second line should contain n different integers from 1 to n. The i-th of them should be the index of candidate that should work at i-th workplace.

Examples

standard input	standard output
10 3 1 5 5 10 4 6	3 1 2 3
10 3 1 4 8 8 3 6	4 2 3 1

Note

In the first example, the distance between each candidate and his workplace equals to 1 (from 1 to 10, from 4 to 5 and from 6 to 5).

In the second example:

- The second candidate works at first workplace, the distance between cities 3 and 1 equals to 2.
- The third candidate works at second workplace, the distance between cities 6 and 4 equals to 2.
- The first candidate works at third workplace, the distance between cities 8 and 8 equals to 0.

Scoring

Tests for this problem are divided into six groups. For each of the groups you earn points only if your solution passes all tests in this group and all tests in **required** groups.

Group	Points	Additional constraints		Required groups	Comment
		n	m		
0	0	—	—	—	Sample tests
1	9	$n \leq 10$	—	0	—
2	10	$n \leq 18$	—	0, 1	—
3	14	$n \leq 100$	—	0–2	—
4	19	$n \leq 5000$	—	0–3	—
5	24	—	$m = n$	—	$b_i = i$ for each i
6	24	—	—	0–5	—

Problem D. Feeling Good

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

Recently biologists came to a fascinating conclusion about how to find a chameleon mood. Consider chameleon body to be a rectangular table $n \times m$, each cell of which may be green or blue and may change between these two colors. We will denote as (x, y) ($1 \le x \le n$, $1 \le y \le m$) the cell in row x and column y.

Let us define a chameleon *good mood certificate* to be four cells which are corners of some subrectangle of the table, such that colors in opposite cells among these four are similar, and at the same time not all of the four cell colors are similar. Formally, it is a group of four cells (x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2) for some $1 \le x_1 < x_2 \le n$, $1 \le y_1 < y_2 \le m$, that colors of (x_1, y_1) and (x_2, y_2) coincide and colors of (x_1, y_2) and (x_2, y_1) coincide, but not all of the four cells share the same color. It was found that whenever such four cells are present, chameleon is in good mood, and vice versa: if there are no such four cells, chameleon is in bad mood.

You are asked to help scientists write a program determining the mood of chameleon. Let us consider that initially all cells of chameleon are green. After that chameleon coloring may change several times. On one change, colors of contiguous segment of some table row are replaced with the opposite. Formally, each color change is defined by three integers a, l, r ($1 \le a \le n$, $1 \le l \le r \le m$). On such change colors of all cells (a, b) such that $l \le b \le r$ are replaced with the opposite.

Write a program that reports mood of the chameleon after each change. Additionally, if the chameleon mood is good, program should find out any four numbers x_1, y_1, x_2, y_2 such that four cells (x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2) are the good mood certificate.

Input

The first line of input contains three integers n, m, q ($1 \le n, m \le 2000$, $1 \le q \le 500\,000$), the sizes of the table and the number of changes respectively.

Each of the following q lines contains 3 integers a_i, l_i, r_i ($1 \le a_i \le n$, $1 \le l_i \le r_i \le m$), describing i-th coloring change.

Output

Print q lines. In the i-th line report the chameleon mood after first i color changes for all $1 \le i \le q$.

If chameleon is in bad mood, print the only integer -1.

Otherwise, print four integers x_1, y_1, x_2, y_2 ($1 \le x_1 < x_2 \le n$, $1 \le y_1 < y_2 \le m$) such that four cells (x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2) are the good mood certificate. If there are several ways to choose such four integers, print any valid one.

Examples

standard input	standard output
2 2 6 1 1 1 2 2 2 2 1 1 1 2 2 2 2 2 1 1 1	-1 1 1 2 2 -1 -1 -1 1 1 2 2
4 3 9 2 2 3 4 1 2 2 1 3 3 2 2 3 1 3 1 2 2 4 2 3 1 1 3 3 1 3	-1 2 1 4 3 -1 2 1 3 2 3 2 4 3 1 1 2 2 1 1 2 2 -1 2 1 3 2

Scoring

Tests for this problem are divided into five groups. For each of the groups you earn points only if your solution passes all tests in this group and all tests in **required** groups.

Group	Points	Additional constraints			Required groups	Comment
		n, m	q	Additionally		
0	0	—	—	—	—	Sample tests
1	5	$n, m \leq 10$	$q \leq 100$	—	0	—
2	15	$n, m \leq 100$	$q \leq 1000$	—	0, 1	—
3	15	$n, m \leq 500$	$q \leq 10\,000$	—	0–2	—
4	20	$n, m \leq 2000$	$q \leq 200\,000$	$l_i = r_i$		—
5	45	—	—	—	0–4	—