Problem F. Safe Road

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	256 megabytes

Pankrat is playing the game "Pancraft" — a copy of a popular game where everything is made of cubic blocks. He has already built a house and dug a mine. Between the house and the mine, a straight road is laid out, consisting of n blocks.

While Pankrat walks along this road, monsters may attack him. Pankrat doesn't like this, and he decided to secure his path to the mine. Pankrat knows that monsters are afraid of light, so he wants to install torches along the road. Each torch illuminates k blocks to the left and k blocks to the right (including the block on which it is installed). That is, a torch installed on block i illuminates all blocks with numbers from $\max(1, i-k)$ to $\min(n, i+k)$.

However, in this version of the game, the player cannot create and place torches independently. Instead, this can be done by an NPC — a wandering trader. The wandering trader takes a_i emeralds for installing a torch on the block with index i.

Pankrat wants to illuminate the entire road so that monsters don't attack him. Help him find a way to do this with minimal emerald costs. Assume that Pankrat has an infinite number of emeralds, and the trader has an infinite number of torches.

Input

The first line contains two integers n and k — respectively, the number of blocks on the road and the illumination radius of one torch ($1 \le n \le 5 \cdot 10^6$, $0 \le k \le 5 \cdot 10^6$). The second line contains n integers $a_1, a_2, \ldots, a_n$, where a_i — the cost of installing a torch on the block with index i ($0 \le a_i \le 10^9$).

Output

Output one integer — the minimum emeralds needed for illuminating the entire road.

Scoring

Points for each subtask are awarded only if all tests for this subtask and required subtasks are successfully passed.

№	Additional constraints	Points	Subtasks	Feedback
1	$1 \le n \le 20$, $0 \le k \le 20$ $0 \le a_i \le 10^9$	15	—	first error
2	$1 \le n \le 1000$, $0 \le k \le 1000$ $0 \le a_i \le 10^9$	50	—	first error
3	$1 \le n \le 10^5$, $0 \le k \le 10^5$ $0 \le a_i \le 10^9$	85	—	first error
4	No additional constraints	100	1, 2, 3	first error

Examples

standard input	standard output
7 1 0 3 1 2 4 2 3	3
10 3 2 3 5 10 23 4 3 2 1 4	4

Note

If k equals zero, then torches illuminate only the block on which they stand.

Problem G. Chord Filling

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

For his birthday, Artem's friends gave him a circle with n points marked on it, where all distances between adjacent points are equal. The points are numbered from 1 to n clockwise. There are already k chords drawn on the circle, with the property that no two chords intersect. Two chords intersect if they have at least one common point (including chord endpoints).

Before proudly hanging his circle on the wall, Artem needs to add the maximum possible number of chords to it such that the above property is still satisfied. Since Artem is already very tired from programming, he wants you to do this for him. If there are multiple ways to add the maximum number of chords, you can do it in any of them.

Input

The first line contains two integers n and k ($1 < n \le 10^5$, $0 \le k \le \lfloor \frac{n}{2} \rfloor$).

The next k lines describe the already drawn chords: on the i-th line are written two numbers a_i and b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$) — the numbers of the points connected by the i-th chord.

It is guaranteed that no two chords intersect.

Output

On the first line, output one number m — the maximum number of chords that can be added to the circle.

On the next m lines, output m chords in the same format as the input data.

Scoring

Points for each subtask are awarded only if the solution passes all tests of that subtask and all required subtasks. Some subtasks may also require that all tests in the statement are completed. For such subtasks, the letter S is additionally specified in the required subtasks section.

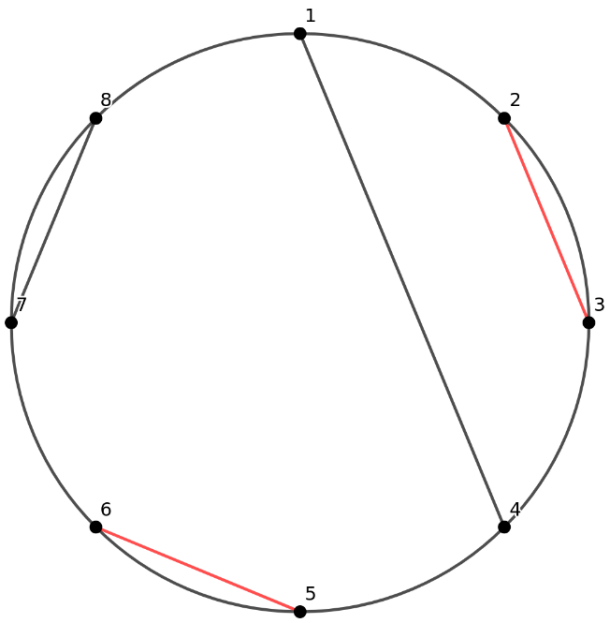
№	Additional constraints	Points	Required subtasks	Feedback policy
1	$k = 0$	5	—	first error
2	$k = 1$	10	S, 1	first error
3	$k = 2$	20	S, 1	first error
4	No additional constraints	65	S, 1	first error

Examples

standard input	standard output
8 2 1 4 7 8	2 2 3 5 6
19 3 7 3 15 8 9 11	5 4 5 12 13 1 2 16 17 18 19
10 3 1 3 5 7 8 10	0

Note

Illustration for example 1:



The chords that Artem will add to the circle are marked in red.

Problem H. Chase

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Soon the International Tuymaada Olympiad 2026 will take place, and Alesha is one of the problem developers. While developing another game problem, he couldn't come up with a solution, so he asked his friend Timur for help. Timur, in turn, managed to solve this problem, but decided to borrow it for another international school olympiad "Adaamyut". Alesha must stop him before it's too late! The friends decided to model this game and determine the owner of the problem in fair competition.

According to the game rules, there are n cities and m flights, each of them allows flying from city u_i to city v_i for w_i rubles. The flights are unidirectional, meaning flight i cannot be used to fly from city v_i to city u_i. On the same day, each player makes at most one flight. On the first day, Alesha is in city a, and Timur is in city t_1. The j-th day ($1 \le j < k$) can be described by the following algorithm:

1. Alesha either does nothing, or, if he is in city u_i, he can use flight i to fly to city v_i, paying w_i rubles for the flight.
2. Timur flies from city t_j to city t_{j+1} (it is guaranteed that there exists such i that $u_i = t_j$ and $v_i = t_{j+1}$). Note that Alesha knows Timur's route in advance.
3. If Alesha and Timur are in the same city, then Alesha catches Timur and the chase ends (Alesha becomes the owner of the problem).

If Alesha cannot catch Timur within $k - 1$ days, then Timur escapes and becomes the owner of the problem.

Obviously, there may be several ways to catch Timur, so Alesha asks you to find the minimum cost of catching Timur.

Input

The first line contains four integers n, m, a, and k — the number of cities and flights, Alesha's starting city, and the number of cities on Timur's route respectively ($2 \le n \le 150$; $1 \le m \le 300$; $1 \le a \le n$; $1 \le k \le 10^5$).

The second line contains k integers $t_1, t_2, \ldots, t_k$ — the cities on Timur's route ($1 \le t_i \le n$; $t_1 \ne a$).

The next m lines contain three integers each: u_i, v_i, and w_i — a flight from city u_i to city v_i that costs w_i rubles ($1 \le u_i, v_i \le n$; $u_i \ne v_i$; there are no i and j ($i \ne j$) such that $u_i = u_j$ and $v_i = v_j$; $0 \le w_i \le 10^6$).

Output

Output the minimum cost of catching Timur, or -1 if Timur cannot be caught.

Scoring

Points for each subtask are awarded only if the solution passes all tests of that subtask and all required subtasks. Some subtasks may also require that all tests in the statement are completed. For such subtasks, the letter S is additionally specified in the required subtasks section.

№	Additional constraints	Points	Required subtasks	Feedback policy
1	$m = n - 1$, $v_i = u_i + 1$	10	—	first error
2	$m = (n-1) \cdot 2$; for each i there exists j such that $u_i = v_j, v_i = u_j$; from any vertex one can reach any other vertex	15	1	first error
3	$n \leq 9$	15	—	first error
4	No additional constraints	60	S, 1, 2	first error

Examples

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

Note

In the first example, we can notice that if Alesha cannot reach city 6 within at most 2 days, then he loses in any case. Thus, the optimal path for Alesha is $1 \to 2 \to 6$, which costs 5 rubles.

The second example is similar to the first, except that now the day limit is 3. Therefore, in this example, we can choose a longer but cheaper path $1 \to 2 \to 3 \to 7$, which costs 4 rubles.

In the third example, Alesha cannot catch Timur.

Problem I. Profit

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Dima is a novice entrepreneur dreaming of organizing the best tourist route through Yakutia. He has already compiled a map including N cities connected by bidirectional roads. The map represents a tree: between each pair of cities there exists a unique path.

Each road is characterized by two integers:

- A_i — potential income that can be obtained if this road is included in the route;
- B_i — time required to travel this road.

Some cities have airports, and the route **must** start and end in such cities.

Dima wants to choose a continuous path $(i_1, i_2, \ldots, i_K)$ along roads, starting and ending in cities with airports, to maximize the route profitability, defined as:

$$\frac{A_{i_1} + A_{i_2} + \cdots + A_{i_K}}{B_{i_1} + B_{i_2} + \cdots + B_{i_K}}.$$

Your task is to find the road with the maximum possible profitability of a route satisfying the above conditions.

Input

The first line contains two integers N and M — the number of cities and cities with airports ($2 \le M \le N \le 10^5$).

The second line contains M numbers $T_1, T_2, \ldots, T_M$ — numbers of cities with airports ($1 \le T_i \le N$). It is guaranteed that all T_i are pairwise distinct.

The next $N - 1$ lines describe roads. Each line contains 4 integers u_i, v_i, A_i, B_i, meaning there is a road between cities u_i and v_i with income A_i and travel time B_i ($1 \le u_i, v_i \le N$, $1 \le A_i, B_i \le 10^9$). It is guaranteed that the given roads form a tree.

Output

In the first line output one number K ($2 \le K \le N$) — the number of vertices in the found path.

In the second line output K numbers i_j — vertex numbers in the path. Vertices $i_1, i_2, \ldots, i_K$ must form a simple path of length at least 1. If there are multiple possible answers — output any.

Scoring

Points for each subtask are awarded only if the solution passes all the tests of that subtask and all required subtasks. Some subtasks may also require that all tests in the statement are completed. For such subtasks, the letter S is additionally specified in the required subtasks section.

№	Additional constraints	Points	Required subtasks	Feedback policy
1	$u_i = i$, $v_i = i + 1$, $N \le 500$	5	—	first error
2	$u_i = i$, $v_i = i + 1$, $N \le 5000$	5	1	first error
3	$u_i = i$, $v_i = i + 1$	10	1, 2	first error
4	$M = 2$	5	—	first error
5	$M = N$	5	—	first error
6	$N \le 500$	10	S, 1	first error
7	$N \le 5000$	20	S, 1, 2, 6	first error
8	$N \le 50000$	30	S, 1, 2, 6, 7	first error
9	No additional constraints	10	S, 1, 2, 3, 4, 5, 6, 7, 8	first error

Example

standard input	standard output
6 4 2 3 5 6 2 3 5 3 3 4 1 5 1 5 12 1 6 4 13 4 5 4 9 4	3 5 4 6

Problem J. Alien Prison Break

Input file:	`standard input`
Output file:	`standard output`
Time limit:	9 seconds
Memory limit:	256 megabytes

Talban is a poor student at North-Eastern Federal University who has just been abducted by aliens. Now he needs to secretly escape from his place of confinement before they conduct inhumane experiments on him. In his cell there is a ventilation hatch leading to a room with an escape pod. This room can be considered an infinite two-dimensional plane on which there are many towers.

The aliens' security system is very strange — before building this prison, they had an array of n vectors (Vx_i, Vy_i). First, the aliens chose the coordinates of the first tower (Sx, Sy) and some subarray $[L, R]$ of this array. Then for each number i from L to R, all existing towers at point (x, y) "cloned" themselves to coordinates $(x + Vx_i, y + Vy_i)$. That is, the number of towers doubled after each operation. These towers work as follows — if a person passes strictly through any segment formed by 2 towers, or gets strictly inside any triangle formed by 3 towers, then they are instantly incinerated on the spot. For better understanding of the conditions, see explanations of the examples.

Unfortunately, Talban doesn't really know neither the coordinates of the original tower, nor which subarray the aliens chose for cloning. Moreover, he doesn't even know the coordinates of the ventilation and escape pod! But fortunately, Dayaana, Talban's clairvoyant friend, was also abducted and placed in a cell next to him. She predicted q possible scenarios — the i-th scenario consists of 8 numbers: L_i, R_i, Sx_i, Sy_i, Ax_i, Ay_i, Bx_i, By_i. It means that the aliens chose coordinates of the first tower (Sx_i, Sy_i) and subarray $[L_i, R_i]$ for their cloning, and also that the ventilation hatch and escape pod are located at points (Ax_i, Ay_i) and (Bx_i, By_i) respectively.

Since Talban and Dayaana have very little time left, they need to calculate for each of the q possible events the minimum distance they must travel from ventilation to the escape pod to avoid being incinerated along the way, or say that escape is impossible in this case.

Input

The first line contains three integers n, q and G — the number of vectors in the aliens' array, the number of Dayaana's event scenarios and the subtask number respectively ($1 \le n \le 10^5$, $1 \le q \le 10^5$, $0 \le G \le 9$, where $G = 0$ means tests from the statement).

Each of the next n lines contains two integers Vx_i and Vy_i — coordinates of vector i ($|Vx_i|, |Vy_i| \le 10^4$). It is guaranteed that all vectors are non-zero.

Then each of the next q lines contains eight integers L_i, R_i, Sx_i, Sy_i, Ax_i, Ay_i, Bx_i, By_i — cloning segment boundaries, coordinates of the first tower, coordinates of ventilation and coordinates of escape pod respectively ($1 \le L_i \le R_i \le n$, $|Sx_i|, |Sx_i| \le 10^6$, $|Ax_i|, |Ay_i|, |Bx_i|, |By_i| \le 10^9$).

Output

For each of the q event scenarios, output a single real number — the minimum possible distance that Talban and Dayaana must travel from ventilation to escape pod, or -1 if they will be incinerated in any case. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-4}.

Scoring

Points for each subtask are awarded only if the solution passes all the tests of that subtask and all required subtasks. Some subtasks may also require that all tests in the statement are completed. For such subtasks, the letter S is additionally specified in the required subtasks section.

№	Additional constraints	Points	Required subtasks	Feedback policy
1	$Vx_i, Vy_i \le 0$ for all $1 \le i \le n$; $Sx_i, Sy_i \le 0$ and $Ax_i, Ay_i, Bx_i, By_i \ge 0$ for all $1 \le i \le q$	7	—	first failed test
2	$L_i = R_i$ for all $1 \le i \le q$	12	—	first failed test
3	$R_i - L_i \le 1$ for all $1 \le i \le q$	12	2	first failed test
4	$L_i = 1, R_i = n, Ax_i = Bx_i$ and $Ay_i = By_i$ for all $1 \le i \le q$	13	—	first failed test
5	$L_i = 1, R_i = n$ for all $1 \le i \le q$	13	4	first failed test
6	$Ax_i = Bx_i$ and $Ay_i = By_i$ for all $1 \le i \le q$	13	4	first failed test
7	$n, q \le 1000$	10	S	first failed test
8	$n, q \le 5 \cdot 10^4$	10	S, 7	first failed test
9	No additional constraints	10	S, $1 - 8$	first failed test

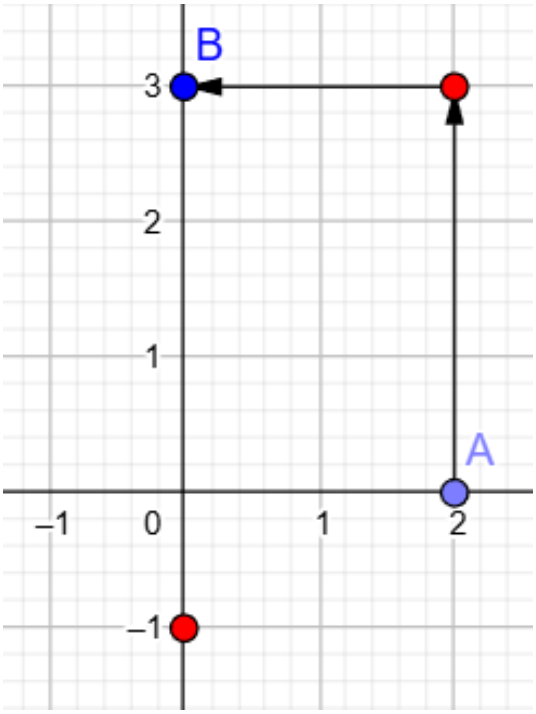
Examples

standard input	standard output
3 5 0 2 4 1 -1 0 2 1 1 0 -1 2 0 0 3 1 3 0 0 2 0 2 8 3 3 3 1 3 2 3 2 1 3 0 0 3 1 1 2 1 3 0 0 1 2 1 2	5.0000000 8.3245553 0.0000000 -1 -1
4 3 0 -2 -2 4 4 -2 0 -2 1 1 2 3 -2 5 0 7 -2 2 2 1 -1 6 4 0 -2 1 4 0 0 -5 1 -4 4	2.8284271 8.4852814 3.1622777

Note

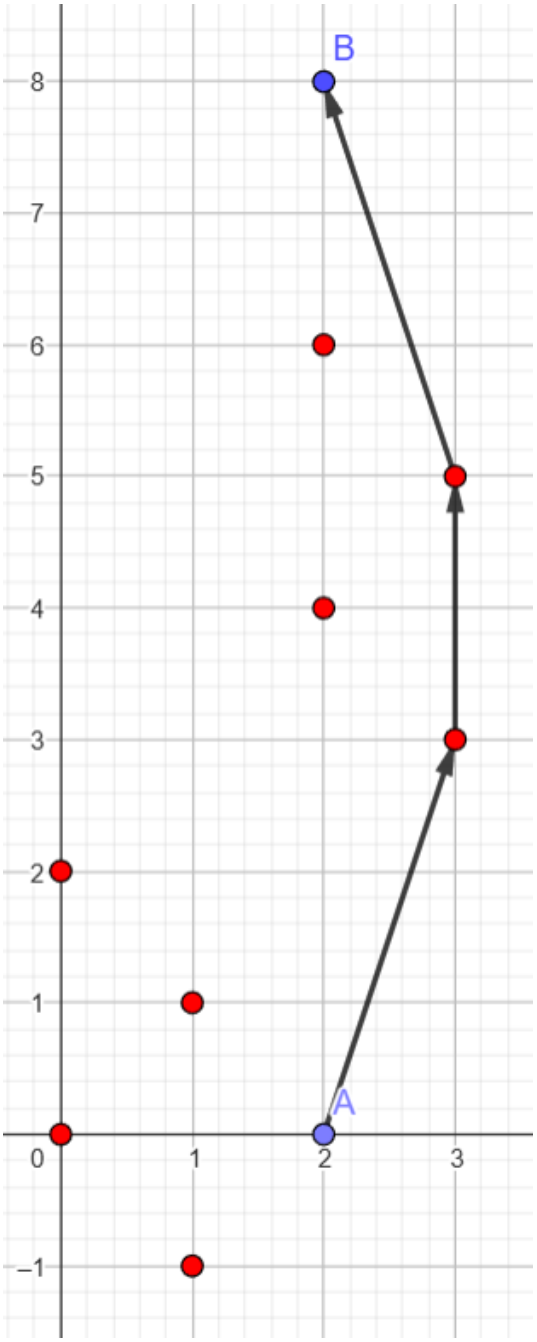
It is considered that you pass strictly through a segment if you enter inside the segment from one half-plane and exit from the other, while touching any point lying strictly inside the segment. The line formed by this segment does not belong to any half-plane.

In the first query of the first example, the optimal path looks as follows:



Red points denote towers. Since we cannot pass strictly through a segment, it must be bypassed. Note that Talban and Dayaana can safely pass through a point containing a tower, since no incineration condition is met.

In the second query of the first example, the optimal path looks as follows:


B
A
8
7
6
5
4
3
2
1
0
−1
0
1
2
3