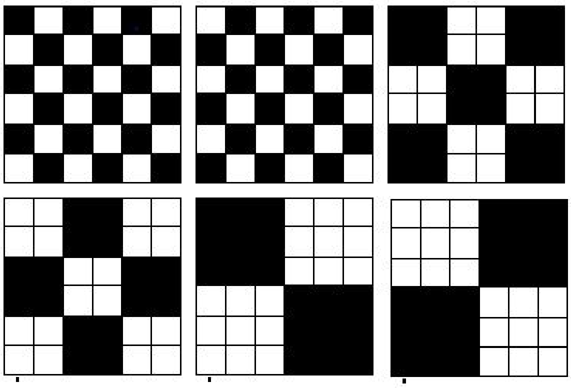
Problem A. Chessboard

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

Tima has $N \times N$ sized checkered board, where K sub-rectangles are painted black and the rest of board in white. The sub-rectangle of the board is a rectangular area with sides parallel to the sides of the board, and whose corners are in integer coordinates. Lines are numbered from top to bottom, columns are numbered from left to right, both from 1 to N.

We call the board *chess* if it can be divided into several identical squares (with side greater than or equal to 1 and **strictly** less than N), and inside of each of these squares all cells are of the same color, and two adjacent squares are of different colors. Two squares are called adjacent if they have a common side. Below we show all possible *chess* boards for $N = 6$:



At one repainting, Tima may change the color of only one cell. If the cell was white, then the cell will be black after repainting, and vice versa. What is the minimal number of repaintings Tima needs to get *chess* board?

Input

The first line of input contains two integers N, K ($2 \le N \le 10^5$, $0 \le K \le min(N^2, 10^5)$) — the length of the board and the number of black sub-rectangles. In the following K lines there are four integers $x1_i, y1_i, x2_i, y2_i$ ($1 \le x1_i, y1_i, x2_i, y2_i \le N$, $x1_i \le x2_i$, $y1_i \le y2_i$) — the indices of the upper left and the bottom right corners of the i'th black sub-rectangle, it is guaranteed that no two sub-rectangles intersect.

Output

Output a single integer — the minimal number of repaints to get a *chess* board.

Scoring

This task contains six sub-tasks:

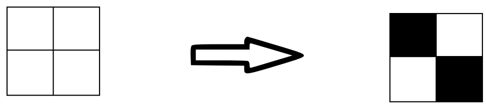
1. $2 \le N \le 100$, $K = 0$. Scored 8 points.
2. N — prime number and area of each sub-rectangle is equal to 1. Scored 8 points.
3. $2 \le N \le 100, 0 \le K \le min(N^2, 1000)$. Area of each sub-rectangle is equal to 1. Scored 15 points.
4. $2 \le N \le 1000, 0 \le K \le min(N^2, 10^5)$. Area of each sub-rectangle is equal to 1. Scored 16 points.
5. $2 \le N \le 10^5, 0 \le K \le min(N^2, 10^5)$. Area of each sub-rectangle is equal to 1. Scored 23 points.
6. $2 \le N \le 10^5, 0 \le K \le min(N^2, 10^5)$. Scored 30 points.

Examples

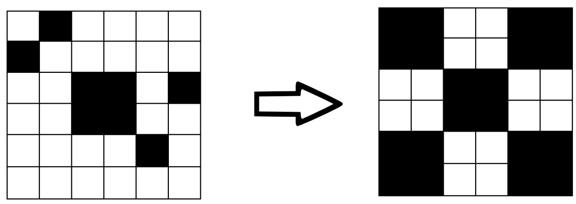
standard input	standard output
2 0	2
6 8 3 3 3 3 1 2 1 2 3 4 3 4 5 5 5 5 4 3 4 3 4 4 4 4 2 1 2 1 3 6 3 6	14
4 1 4 1 4 4	8

Note

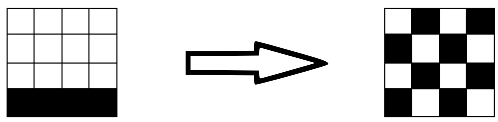
1)



2)



3)



Problem B. Evacuation plan

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

According to the forecast of seismologists a strong earthquake is expected in Bitland. There are n cities in Bitland, numbered from 1 to n. Some of them are connected by two way roads. We consider route as a sequence of cities, where each consequent two cities are connected by some road. Length of the route is the sum of lengths of all roads that are involved in the route. Minimal route between pair of cities (a, b) is defined by the route with minimal length that starts in city a and ends in city b.

The country's government considers radiation leakage from nuclear power plants (NPP) as the main problem — in this case, evacuation of the population will be required. Each NPP is located in one of the cities and their total number is equal to k, each city has no more than one NPP. The government wants to draw up an evacuation plan that will work during the earthquake.

The route of evacuation between cities must be chosen, so that it lays as far as possible from all cities with NPP. The *dangerousness* of the route is estimated by calculating the minimal distance between cities on that route and any city with NPP. More formally, let $(a_1, a_2, ..., a_s)$ be cities on the route and let $(g_1, g_2, ..., g_k)$ be the cities with NPP, then *dangerousness* of the route equals to the minimum among all values of $dist(a_i, g_j)$, where $dist(a, b)$ is equal to the length of the minimal route between a and b.

Given Q pairs of cities (s_i, t_i) for each of which you have to come up with evacuation plan with maximal *dangerousness*.

Input

The first line contains two integers n and m separated by space $(2 \le n \le 10^5, 1 \le m \le 5 \cdot 10^5)$ — the number of cities and the number of roads in Bitland. Then in m lines there are descriptions of roads, one per line. Each road is given by three numbers a_i, b_i, w_i ($1 \le a_i, b_i \le n$, $1 \le w_i \le 1000$, $a_i \ne b_i$) — pair of connected cities and length of the road. The next line contains one integer k $(1 \le k \le n)$ — number of cities with NPP. On the next line k integers g_i ($1 \le g_i \le n$, for $1 \le i \le k$) are given — the numbers of cities in which NPP are located. The next line contains one integer Q $(1 \le Q \le 10^5)$ — the number of pairs of cities of evacuation plan. Then on each of next Q lines i-th pair of cities (s_i, t_i) ($1 \le s_i, t_i \le n$, $s_i \ne t_i$) are given.

It is guaranteed, that no road connects the city with itself, between any pair of cities there are no more than one road and it is possible to reach any city from any city.

Output

Output Q lines.

On i-th line output single integer — maximal *dangerousness* for the pair of cities (s_i, t_i).

Scoring

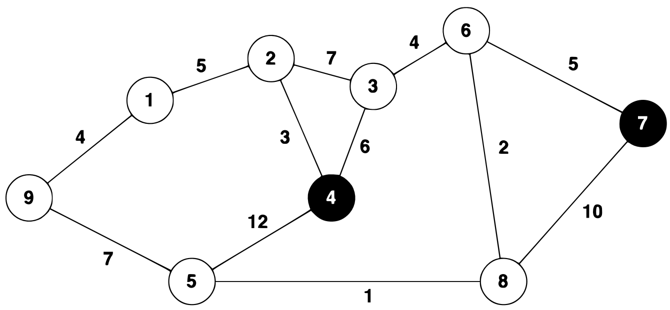
This task contains five sub-tasks:

1. $n \le 10^3$, $1 \le m \le 10^3$, $Q \le 10^3$. Between each of Q pairs (s_i, t_i) there exists a direct road, Scored 10 points.
2. $n \le 10^5$, $Q \le 10^5$. Between each of Q pairs (s_i, t_i) there exists a direct road. Scored 13 points.
3. $n \le 15$, $1 \le m \le 200$, $1 \le Q \le 200$. Scored 12 points.
4. $Q = 1$. Scored 19 points.
5. Constraints from problem statement above. Scored 46 points.

Example

standard input	standard output
9 12 1 9 4 1 2 5 2 3 7 2 4 3 4 3 6 3 6 4 8 7 10 6 7 5 5 8 1 9 5 7 5 4 12 6 8 2 2 4 7 5 1 6 5 3 4 8 5 8 1 5	5 5 0 7 8

Note



Problem C. Gift

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

Alan received from Bekzhan unusual gift on birthday. The gift was locked under a mathematical lock.

Lock contains N numbers, positions numbered from 1 to N. Initially, all of them are equal to zero. In one operation Alan can choose some integer X $(1 \leq X)$ and K **different** positions in the lock $1 \leq i_1, i_2, ..., i_K \leq N$, then add X to all values at positions $i_1, i_2, ..., i_K$. Bekzhan also reported the sequence of numbers which opens the lock — $A_1, A_2, ..., A_N$. Order of numbers is important.

Alan cannot handle this problem. Help him to unlock or determine that solution does not exist.

Note that you do not need to minimize number of operations. But Alan does not want to select more than 3000000 $(3 \cdot 10^6)$ positions in total, i.e. if M equals to the number of operations in a solution and $M \cdot K \leq 3 \cdot 10^6$, then the solution is considered as correct, otherwise not.

Input

The first line of input contains two positive integer numbers N and K $(2 \leq K \leq N \leq 10^6$, $N \cdot K \leq 2 \cdot 10^6)$ — length of the sequence in lock and number of positions, which can be chosen on each operation. The second line of input contains N positive integer numbers A_1, A_2, ... , A_N $(1 \leq A_i$, for all $1 \leq i \leq N$, $\sum_{i=1}^{N} A_i \leq 10^{18})$ separated with single space — sequence of numbers which opens the lock.

Output

If solution does not exist, output "-1" (without quotes). Otherwise, in the first line output M — the number of operations. In j'th line of the next M lines print X_j, then K **different** numbers $i_{j,1}, i_{j,2}, ..., i_{j,K}$ — the added value and positions, to which the value is added on j'th operation.

Scoring

This problem consists of five subtasks, in each subtask tests satisfy constraints in statement:

1. $\sum_{i=1}^{N} A_i \leq 10$, $K = 2$. Score 7 points.
2. $\sum_{i=1}^{N} A_i \leq 10^5$, $K = 2$. Score 11 points.
3. $\sum_{i=1}^{N} A_i \leq 10^5$. Score 12 points.
4. $A_1 = A_2 = ... = A_N$. Score 19 points.
5. Constraints are from problem statement. Score 51 points.

Example

standard input	standard output
4 2 2 3 3 2	3 2 3 1 1 3 2 2 2 4