Willy and Prime Numbers

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

During the 21st century, despite the existence of countless mathematicians and countless scientists, mathematician Willy was, undoubtedly the most important mind of the century for the advancement of the technologies. His contributions to the society extend to innumerable branches of science: in Mathematics, Physics, Chemistry, Informatics, among other areas.

At one point in his life, Willy realized that he was becoming a sedentary person because he had invested a lot of time in his research and calculations. So, he decided that, at the beginning of each morning, he was going to run at his city. Willy's city can be seen as n metro stations and n - 1 streets connecting them. It is guaranteed that, moving only through the city streets, it is possible to go from any metro station to any other. For each street, due to his ease with calculations, he manages to estimate how long it takes to walk through it.

Initially, Willy walks form his home to a metro station. This initial run does not count in his running time as he is only doing a warm-up. Then, he starts running between the metro stations. Due to his obsession with prime numbers, he will only walk through streets such that the time he takes to cross it is a prime number. If there is no street with this characteristic, Willy will refuse to run and will remain sedentary. However, if there is at least one street with this characteristic, he wants to travel between two stations to maximize the amount of time he spent without visiting the same street twice. When he arrives at the final metro station, he will walk back home and, analogously to the initial part of his training, that time will not be counted in his total running time.

Although he has contributed with numerous programming algorithms, he is too lazy to code. Therefore, he asked for your help in finding the path he must take. Willy will provide you the configuration of his city and the amount of time he would spend walking through each street. Help him to discover the maximum time he can spend running around his city.

Input

The first line contains one integer n ($1 \leq n \leq 10^5$), indicating the number of metro stations in Willy's city.

The following n - 1 lines contains 3 integers U_i, V_i, W_i ($1 \leq U_i, V_i \leq n$, $U_i \neq V_i$, $2 \leq W_i \leq 10^5$), indicating that there's a street between the metro stations U_i and V_i which Willy will spend W_i of time get through it.

Output

The output must contain one integer: the maximum time he can spend running around his city. If there's no street that Willy can run, print 0.

Examples

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

Note

In the first sample, the optimal path for Willy is to start at station 4 and end at station 2. The time of this path is $2 + 3 + 5 + 2 = 12$ and all times are primes.

In the second sample, there are no streets with a prime time. So, the answer is 0.

Boats

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

Because of several wars and crises that occurred in the world in the 20th century, the city of Nlogonia has always been a city that attracted many immigrants. Its inhabitants were peaceful and no crisis hit them. Boats from all over the world brought immigrants to the city. For this, the city had to plan new projects for moving boats in its bay. The project used at that time has some rules:

- Each port may contain a maximum of one vessel;
- There cannot be two boats not moored simultaneously in the bay because there is a high risk of them colliding;
- There are some risk points in the bay that boats cannot pass through.

In a simplified way, the Nlogonia bay can be seen as an $n \times m$ board where each cell has a value of 0 or 1. Let $(i,\, j)$ be the cell in the i-th row and in the j-th column and $v_{i,j}$ the value of that cell. $v_{i,j}$ is equal to 1 if the cell $(i,\, j)$ is a risk point that boats cannot pass through, otherwise, the boats can travel through that cell. When a boat is in a cell, spending 1 unit of time, it can go to every cell that is not a risk point and share a side with that cell.

During the period in question, there were x boats wanting to dock in the y ports of Nlogonia. It is guaranteed that there is at least one port for each vessel. All boats appear in the bay on line 1 and all ports are located on line n. The boats are numbered from 1 to x and enter the bay in ascending order of their numbers and, as soon as the i-th vessel $(1 \leq i < x)$ docked in a port, the $(i+1)$-th boat will enter the city bay at the same time, until all boats are in a port. Boat i $(1 \leq i \leq x)$ will enter the bay in cell $(1,\, a_i)$ and port j $(1 \leq j \leq y)$ is located in the cell $(n,\, b_j)$. Your mission is to find out what is the minimum amount of time needed for all boats to dock at any port or to say that it is impossible if the risk points do not allow ships to reach ports.

Input

The first line contains 4 integers n, m, x, y $(2 \leq n, m \leq 300,\ 1 \leq x \leq y \leq m)$, indicating the number of rows and columns of the board, the number of boats and the number of ports.

The following n lines contains m integers $v_{i,j}$ $(v_{i,j} \in \{0, 1\})$, indicating the value of cell $(i,\, j)$. It is guaranteed that for every i such that $1 \leq i \leq m$, $v_{1,i} = v_{n,i} = 0$. That is, there are no risk points in the first and in the last line of the board.

The next x lines contains one integer a_i $(1 \leq a_i \leq m)$, indicating the column that the i-th boat will appear in the bay. It's guaranteed that for all i, j such that $1 \leq i, j \leq x$ ans $i \neq j$, $a_i \neq a_j$.

The following y lines contains one integer b_j $(1 \leq b_j \leq m)$, indicating the column that the j-th port is located. It's guaranteed that for all i, j such that $1 \leq i, j \leq y$ and $i \neq j$, $b_i \neq b_j$.

Output

The output must have one integer: the minimum amount of time needed for all boats to dock at any port. If it's impossible, print -1.

Examples

standard input	standard output
5 5 2 3 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 3 1 4 5	13
5 5 2 2 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 2 4 1 4	13
4 4 1 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 3 1	-1

Note

One possible answer for the first sample is the first boat moor in the last port, spending 8 units of time and the second boat moor in the second port, spending more 5 units of time, totaliting 13 units of time.

Trees and Decrees

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

Frederick is a gentleman who loves to care for his trees. He has n trees arranged in a row in his garden. He was very happy when he realized that they had all reached the mark of 10^{18} meters in height!

However, due to the colossal height of their trees, these trees can fall and cause great damage. Therefore, the city hall had to intervene. For m days, the city will send a decree to Frederick. During the i-th day, the prefecture decree will have 3 numbers: L_i, R_i and X_i. This means that, on the i-th day, he will have to cut a few meters of some trees located in the range from L_i to R_i, inclusive. The total number of meters taken from the trees on the i-th day has to be exactly X_i. When Frederick cuts a tree, he cuts an entire number of meters.

As he values his trees so much, he gave you a mission. Given all m decrees of the city hall, what is the maximum number of meters that he will have to remove from any of the trees if he cuts them optimally, that is, in order to minimize the maximum amount that he had to remove from a tree after the m days.

Input

The first line contains two integers n, m ($1 \leq n, m \leq 10^5$), indicating the number of trees in Frederick's garden and the number of days that the city hall will send a decree to Frederick.

The following m lines contain 3 integers L_i, R_i, X_i ($1 \leq L_i \leq R_i \leq n$, $1 \leq X_i \leq 10^6$), indicating the decree of the city hall.

Output

The output must contains one integer: maximum number of meters that Frederick will have to remove from any of the trees if he cuts them optimally.

Examples

standard input	standard output
3 2 1 2 2 2 3 2	2
4 3 2 4 1 2 3 2 1 2 5	3

Note

In the first sample, a possible optimal sequence of cuts would be remove 1 meter of the first and the second tree on the first day and remove 1 meter of the second and the third tree on the second day. In the end, he cuts 1 meter of the first tree, 2 meters of the second tree and 1 meter of the third tree. So the maximum number of meters that he removed of a tree was 2 meters.

The 100

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

During a nuclear war on Earth, 12 space stations were sent to Earth's orbit with a few people in each. For a greater chance of survival, these 12 stations decided to unify into just one. This new station became known as the Ark.

After a while, they decided to send 100 people who lived in the Ark to the earth's surface to find out if the Earth still had reasonable conditions to be inhabited again. This group of people was under the command of a woman named Clarke.

Arriving at the Earth's surface, from some search missions, they found that there were groups of surviving people at n points on the surface (just in that question, suppose the Earth can be seen as a plane). However, Clarke doesn't trust this search teams so much. With some parts of the ship, Clarke managed to create a drone that could take pictures of rectangular areas. The sides of the rectangular photo must be parallel to the axis. With the drone, she could take some pictures and check the veracity of the information about the survivors. She can verify that a group of survivors really exists at a given point if, and only if, it is visible in at least one photo. However, due to the failure of some parts, her drone presented a problem: when it takes a photo of a rectangular area, only the borders of that photo are visible. Its interior has several blots and stains, making it impossible to see its interior area. For her drone to take a picture, it uses up a unit of energy. During its movement, no unit of energy is consumed. As the available resources are limited, Clarke wants to spend the least amount of energy possible to check if there are really groups of surviving earthlings at the N points provided by the search teams. Help Clarke finding out what is the least amount of energy.

Input

The first line contains one integer n ($1 \le n \le 2000$), indicating the number of possible groups of survivors.

The following n lines contain 2 integers x_i, y_i ($10^{-9} \le x_i, y_i \le 10^9$), indicating the coordinate (x_i, y_i) of a possible group of surviving earthlings. It's possible that there are several groups in the same coordinate.

Output

The output must contain one number: the minimum amount of energy that Clarke needs to spend to check all the n points.

Examples

standard input	standard output
5 2 3 4 1 4 5 8 3 4 3	2
8 1 -1 2 -2 2 2 -1 1 -1 -1 -2 2 1 1 -2 -2	2

Note

In the first sample, Clarke can take a picture with the following rectangular areas: rectangle with bottom left coordinate (2, 1) and top right coordinate (8, 5), covering all the points, except the fifth point and other rectangle with bottom left coordinate (4, 0) and top right coordinate (10, 3), covering the second, the fourth and the fifth point. With these pictures, all the points were visible in at least one photo.

Programming Tournament

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

The mayor of the city of Byteland holds an annual programming tournment. In 2020, the competition has n participants. To slightly change the dynamics of the tournament, the mayor decided to change the rules of this edition of the tournament. These rules mainly affected the mode of combat between the participants: each pair of participant will face each other only once. Each match will be carried out in a certain programming language. When two participants face each other using a certain language, the person with the greatest skill in that language wins the match. There will be no draws in these matches because each participant has a unique ability in each language.

Before the tournament started, the mayor asked his assistant to write down the results of each battle, the programming language in which it was fought, and to count the number of languages used in the tournament. However, at the end of the event, the mayor realized that his assistant had noted only the results of each match. Therefore, the mayor hired you to find out what is the smallest number of programming languages used in the tournament so that it is possible to get to the results given, considering the players with a certain skill in each language and also demanded that you give a possible configuration of the tournament, that is, for each match, a number representing the programming language used in the tournament.

Input

The first line contains one integer n ($1 \leq n \leq 2000$), indicating the number of participants in the tournament.

The following n lines contain n integers $a_{i,j}$ ($a_{i,j} \in \{0, 1\}$), indicating the result of the combat of the participants i and j. $a_{i,j}$ will be equal to 1 if and only participant i won the combat. It's guaranteed that $a_{i,j}$ = 1 - $a_{j,i}$ for all i and j such that $1 \leq i, j \leq n$ and $i \neq j$. It's guaranteed too that $a_{i,i} = 0$ for all i such that $1 \leq i \leq n$.

Output

The first line must contain the smallest number of programming languages used in the tournament. The languages are numbered from 1 to the smallest number of programming languages.

After that, the output must have n - 1 more lines. The i-th line must have n - i integers. The integer j-th integer in the i-th line must be the number of the programming language used in the combat between participant i and participant $i + j$.

Examples

standard input	standard output
3 0 1 0 0 0 1 1 0 0	2 1 2 1
3 0 1 1 0 0 1 0 0 0	1 1 1 1

Milky Way

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

Jonn is an alien who lives in a planet far, far away from Earth. His planet can be seen as a $n \times m$ grid, where each cell is the house of an alien. Jonn's planet format is really peculiar: it is a toroid, in addition to rows i and $i+1$ being adjacents for $1 \leq i < n$ and columns j and $j+1$ being adjacents for $1 \leq j < m$, rows 1 and n are adjacent and columns 1 and m are adjacent.

Jonn is the only milk deliverer from this planet. Therefore, every morning, he leaves his house, visits all other houses and come back home. Jonn's house is in the cell (x, y). Even though Jonn likes his profession, he wants to come back home the fastest way possible because he really likes watching TV series. Consequently he never visits the same house more than once in the same morning. When Jonn is in a house, he can only visits adjacent houses. Two houses are considered adjacents if they are in the same row and in adjacent columns or in the same column and in adjacent rows.

Help Jonn to build a valid path for his delivery, that is, a path that begins in his house, goes through all the other houses in the planet and returns to his house and the adjacents houses in this path are adjacents in the planet.

Input

The only line of the input contains four integers n, m, x, y ($2 \leq n, m \leq 2000$, $1 \leq x \leq n$, $1 \leq y \leq m$), indicating the number of rows and columns of the grid and the coordinates of the Jonn's house.

Output

Your output must have $n \cdot m + 1$ lines, each one containing the house that Jonn must visit in that order. The first and the last house must be the Jonn's house. If there's more than one answer, any one will be accepted. (Be careful! The output to this problem is huge.)

Examples

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

Programming Tournament again

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

The mayor of Byteland, given the disaster the edition of 2020 was because of his assistent's mistakes, decided to return the original rules of the programming tournament. According to these rules, initially, n participants are arranged in a line, this order being decided by the mayor, and the fighting must be carried out as follows: in the first round, the first participant will battle against the second participant, the third against the fourth, and so on. In the second round, the winner of the first battle of the first round plays against the winner of the second battle of the first round, the winner of the third against the winner of the fourth, and so on. This pattern is repeated until only one player remains.

With this style of battle, participants are eager to know when they are going to battle against an opponent and keep asking these questions to the mayor. Sometimes, the mayor also decides the order of two participants in the current configuration. The mayor hired you again to help him to deal with these questions and these changes.

Input

The first line contains 2 integers n, q ($1 \le n, q \le 10^5$), indicating the number of tournament participants and the number of queries. It's guaranteed that n is a power of two.

The next line contains n integers a_1, a_2, $\ldots$, a_n indicating the initial disposition of the participants. It's guaranteed that the sequence a_1, a_2, $\ldots$, a_n is a permutation of the n participants.

Each of the following q lines describes the questions and the changes and has the following format:

- 1 x y - Swap the position of the participants x and y ($1 \le x, y \le n$, $x \ne y$);
- 2 x y - Ask the round that participants x and y will battle each other if they win all the battles until this round ($1 \le x, y \le n$, $x \ne y$).

Output

For each query of type 2, output the round that participants x and y will battle each other if they win all the battles until this round.

Examples

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

Max and Min

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

A sequence a of n positive integers is given. A division of this sequence is a partition of this sequence in exactly k disjoint contiguous blocks of that sequence. The value of a division is the sum of the costs of the k intervals in this division. The cost of an interval is the difference between the greatest and the smallest values within this interval. Your task is to find the greatest possible value of a division of this sequence.

Input

The first line contains two integers n, k ($1 \le n \le 20000$, $1 \le k \le 100$, $k \le n$), indicating the size of the sequence a and the number of intervals that it needs to be divided.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$, for all $1 \le i \le n$), indicating the numbers of the sequence a.

Output

The output must have one integer: the greatest possible value of a division of this sequence.

Examples

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

Hospital

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

The Seattle Grace hospital is a hospital with an intense flow of pacients everyday. That's why it were installed n elevators in this hospital. Coincidentally, it also has n floors. Despite the number of elevators, the Doctor Bailey is still worried with the locomotion speed between different floors, given the huge size of the building. So she called her intern Cristina to try to solve this problem.

Initially, the i-th elevator is the floor h_i. Bailey wants Cristina to move some elevators to different floors in a way that, in each floor, there is exactly one elevator. Although this hospital is modern, its elevators are old and have two buttons: a button to go up one floor and another one to go down one floor. Cristina asked your help to determine the minimum number of times she has to press one button to complete her given task.

Consider that, for her mobility between floors, Cristina uses the hospital stairs, thus not influencing the configuration of the elevators during her changes of floors and / or elevators.

Input

The first line contain one integer n ($1 \le n \le 10^5$), indicating the numbers of elevators in the hospital.

The second line contains n integers $h_1, h_2, \ldots, h_n$ ($1 \le h_i \le n$, for all $1 \le i \le n$), indicating the inicial floor of each elevator.

Output

The output must contain just one number: the minimum number of times Cristina has to press one button to complete her given task.

Examples

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

Note

In the first sample, one possible answer is move the first elevator to the first floor, move the second elevator to the third floor and move the fourth elevator to the second floor. The total number of moves will be 2 + 1 + 0 + 1 = 4.

In the second sample, one possible answer for this configuration is move the second elevator to the second floor and the third elevator to he third floor. The total number of moves will be 0 + 1 + 2 = 3.

Light Experiments

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

One of Sir Isaac Maliki's studies in the 17th century was on light. In his experiment, he had a strip of n colored segments. Each of these segments initially had a color c_i. After some tests, he noticed something very interesting: when he focused on this band a light of color c, all the segments that had c color reflected that color and the segments that did not have that color turned black. After this discovery, he started to play with that fact. He did q operations in this range: one of the operations was to change the color of one of the segments of the strip ans the other operation was to illuminate it with a light with some color and count how many contiguous intervals of it were not black and the other operation was to change the color of one of its segments. He wrote down all of these operations on a paper, along with his observations.

As early as in the 21st century, some university researchers found this Maliki's paper, with his notes. However, there was a part of the paper that was worn out and no one was able to decipher it: the answers to the second type of operation. The task of discovering these answers has been passed on to that university's computing institute and you are, luckily, part of that institute. Help them by writing a program that simulates the Maliki experiment, assuming the fact that what he discovered is true and supporting the queries and changes made.

Input

The first line contains two integers n, q ($1 \leq n, q \leq 10^6$), indicating the size of the strip and the number of operations made by Maliki.

The second line contains n integers c_i ($1 \leq c_i \leq n$), indicating the initial color of the i-th segment of the strip.

Each of the following q lines describes the questions and the changes and has the following format:

- 1 x c - Change the color of the x-th segment of the strip to color c ($1 \leq x, c \leq n$). It may be that the segment x already has color c, so the configuration of the strip will not be changed.
- 2 c - Count how many contiguous intervals of the current strip are not black if we illuminate it with a light with color c ($1 \leq c \leq n$);

Output

For each query of type 2, output the number of contiguous intervals of the current strip are not black if we illuminate it with a light with color c.

Example

standard input	standard output
5 5 1 1 3 2 1 2 1 2 3 1 3 1 2 1 2 3	2 1 2 0

Can you answer these queries X?

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

n arrays with size 2^i $(1 \le i \le 16)$ and q queries are given. There are 4 type of queries:

- Type 1: x, y, v - Add v in the element of index y from the array x;
- Type 2: x - The answer to this query is the sum of the maximum sum interval in the array x;
- Type 3: x, y - Glue blocks x and y, both with the same size. The array x will be in the left half and the array y will be in the right half of the new array. The arrays x and y will be **destroyed** after this query;
- Type 4: x - Break the array x in the middle, creating two new arrays and **destroying** the array x.

In a query of type 3, the new array receives the index of the smallest positive integer that has not been used yet. Likewise, in a query of type 4, when you split an array in half, the left half array will receive the smallest unused positive integer and the right half array will receive the second unused smallest integer.

Input

The first line contains 2 integers n, q $(1 \le n, q \le 10^5)$, indicating the number of arrays and the number of queries.

Each of the following n lines starts with an integer s_i $(1 \le s_i \le 2^{16})$ indicating that the size of the i-th array is s_i, followed by s_i integers. The j-th integer $v_{i,j}$ $(-10^6 \le v_{i,j} \le 10^6)$, indicating j-th number of the i-th array. It's guaranteed that $\sum_{i=1}^{N} s_i \le 10^5$.

The following q lines describe a query and has the following format:

- 1 x y v - Add v $(-10^6 \le v \le 10^6)$ in the element of index y $(1 \le y \le s_x)$ from the array x;
- 2 x - The answer to this query is the sum of the maximum sum interval in the array x;
- 3 x y - Glue blocks x and y $(x \ne y)$, both with the same size. The array x will be in the left half and the array y will be in the right half of the new array. The arrays x and y will be **destroyed** after this query;
- 4 x - Break the array x in the middle, creating two new arrays and **destroying** the array x. In this query, it's guaranteed that $s_x \ge 2$.

In all queries, it's guaranteed that the array used in this query exist in this moment.

Output

For each query of type 2, output the sum of the maximum sum interval in the array x.

Example

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