1

Growing Vegetables is Fun 4

Bitaro likes gardening. He is now growing plants called Biba-herbs in the garden. There are N Biba-herbs in the garden, planted in a line from the west to the east. The Biba-herbs are numbered from 1 to N from the west to the east. Now, the height of the Biba-herb i ($1 \leq i \leq N$) is A_i.

Due to the breed improvement, if Bitaro waters a Biba-herb once, then its height increases by 1. Since he wants to decorate the garden, he will water the Biba-herbs several times so that the following condition is satisfied.

- After Bitaro finishes watering, let B_i be the height of the Biba-herb i. Then, there exists an integer k ($1 \leq k \leq N$) such that $B_j < B_{j+1}$ holds for every $1 \leq j \leq k-1$, and $B_j > B_{j+1}$ holds for every $k \leq j \leq N-1$.

However, Bitaro is not good at watering. When he waters Biba-herbs, he can only water consecutive Biba-herbs in an interval. In other words, he chooses integers L and R ($1 \leq L \leq R \leq N$) and waters the Biba-herbs $L, L+1, \ldots, R$.

Bitaro wants to minimize the number of times of watering.

Write a program which, given the number of Biba-herbs and their current heights, calculates the minimum number of times of watering so that the above condition is satisfied.

Input

Read the following data from the standard input. Given values are all integers.

N
$A_1 \cdots A_N$

Output

Write one line to the standard output. The output should contain the minimum number of times of watering.

Constraints

- $2 \leq N \leq 200\,000$.

- $1 \le A_i \le 1\,000\,000\,000$ $(1 \le i \le N)$.

Subtasks

1. (40 points) $N \le 2\,000$.
2. (60 points) No additional constraints.

Sample Input and Output

Sample Input 1	Sample Output 1
`5` `3 2 2 3 1`	`3`

The condition is satisfied if Bitaro waters the Biba-herbs three times as follows.

- Let $L = 2$ and $R = 5$. Then Bitaro waters the Biba-herbs $2, 3, 4, 5$. The heights of the Biba-herbs become $3, 3, 3, 4, 2$ from the west.
- Let $L = 2$ and $R = 3$. Then Bitaro waters the Biba-herbs $2, 3$. The heights of the Biba-herbs become $3, 4, 4, 4, 2$ from the west.
- Let $L = 3$ and $R = 3$. Then Bitaro waters the Biba-herb 3. The heights of the Biba-herbs become $3, 4, 5, 4, 2$ from the west.

It is impossible to satisfy the condition if Bitaro waters the Biba-herbs less than three times. Hence the minimum number of times of watering is 3.

Sample Input 2	Sample Output 2
`5` `9 7 5 3 1`	`0`

Since the condition is already satisfied, the minimum number of times of watering is 0.

Sample Input 3	Sample Output 3
`2` `2021 2021`	`1`

The condition is satisfied if Bitaro waters the Biba-herb 1 by choosing $L = 1$ and $R = 1$, or he waters the Biba-herb 2 by choosing $L = 2$ and $R = 2$.

Sample Input 4	Sample Output 4
8 12 2 34 85 4 91 29 85	93

2

Snowball

The JOI plain is a wide plain spreading from west to east. We can consider the JOI plain as a number line. A spot on the JOI plain is denoted by a coordinate. The positive direction of the number line corresponds to the east direction. Now winter comes in the JOI plain. There are N snowballs on it, numbered from 1 to N from the west to the east. In the beginning, the coordinate of the snowball i ($1 \leq i \leq N$) is an integer X_i.

Strong wind blows in the JOI plain in winter. You have observation data of wind for Q days. On the j-th day ($1 \leq j \leq Q$), the wind is described by an integer W_j. If W_j is negative, then the wind blows to the west direction. If W_j is not negative, then the wind blows to the east direction. The strength of the wind is $|W_j|$.

When wind blows, a snowball is moved to the same direction as the wind, and its length of move is equal to the strength of the wind. In other words, if there is a snowball in the coordinate x in the beginning of the j-th day ($1 \leq j \leq Q$), then the snowball is moved from the coordinate x to the coordinate $x + W_j$. At the end of the j-th day, the coordinate of the snowball becomes $x + W_j$. Note that, in each day, the snowballs are moved at the same time, and their speeds are the same.

Initially, the JOI plain is covered with snow. If a snowball is moved on an interval covered with snow, then the snowball accumulates the snow, the weight of the snowball is increased, and the snow on the interval disappears. In other words, for an integer a, assume that the interval from a to $a + 1$ is covered with snow. If a snowball is moved on this interval, then the weight of the snowball is increased by 1, and the snow on the interval from a to $a + 1$ disappears. However, if a snowball is moved on an interval without snow, the weight of the snowball remains the same.

Initially, the weight of every snowball is 0. It did not snow on the Q days of observation data.

You want to know the weight of each snowball at the end of the Q-th day.

Write a program which, given the initial position of each snowball and observation data of wind for Q days, calculates the weight of each snowball at the end of the Q-th day.

Input

Read the following data from the standard input. Given values are all integers.

$N\ Q$
$X_1 \cdots X_N$
W_1
$\vdots$
W_Q

Output

Write N lines to the standard output. The i-th line ($1 \le i \le N$) should contain the weight of the snowball i at the end of the Q-th day.

Constraints

- $1 \le N \le 200\,000$.
- $1 \le Q \le 200\,000$.
- $|X_i| \le 1\,000\,000\,000\,000$ ($= 10^{12}$) ($1 \le i \le N$).
- $X_i < X_{i+1}$ ($1 \le i \le N-1$).
- $|W_j| \le 1\,000\,000\,000\,000$ ($= 10^{12}$) ($1 \le j \le Q$).

Subtasks

1. (33 points) $N \le 2\,000$, $Q \le 2\,000$.
2. (67 points) No additional constraints.

Sample Input and Output

Sample Input 1	Sample Output 1
4 3	5
-2 3 5 8	4
2	2
-4	6
7	

In this sample input, the weight of each snowball varies as follows.

- Initially, the coordinates of the snowballs are $-2, 3, 5, 8$ from the west to the east. The weights of the snowballs are $0, 0, 0, 0$, respectively.
- On the first day, wind blows to the east direction, and its strength is 2. At the end of the first day, the coordinates of the snowballs are $0, 5, 7, 10$. The weights of the snowballs are $2, 2, 2, 2$, respectively.
- On the second day, wind blows to the west direction, and its strength is 4. At the end of the second day,

the coordinates of the snowballs are $-4, 1, 3, 6$. The weights of the snowballs are $4, 4, 2, 3$, respectively.

- On the third day, wind blows to the east direction, and its strength is 7. At the end of the third day, the coordinates of the snowballs are $3, 8, 10, 13$. The weights of the snowballs are $5, 4, 2, 6$, respectively.

Hence output $5, 4, 2, 6$, which are the weights of the snowballs at the end of the third day.

Sample Input 2	Sample Output 2
1 4 1000000000000 1000000000000 -1000000000000 -1000000000000 -1000000000000	3000000000000

Sample Input 3	Sample Output 3
10 10 -56 -43 -39 -31 -22 -5 0 12 18 22 -3 0 5 -4 -2 10 -13 -1 9 6	14 8 7 9 11 10 9 8 5 10

3 Group Photo

At the end of a training camp, a group photo is taken with the N participants. The participants are numbered from 1 to N, in order of height. The height of the participant h is h ($1 \leq h \leq N$).

The participants stand on the stairs for the photo. There are N steps in the stairs. The steps are numbered from 1 to N, from a lower place to a higher place.

The step $i + 1$ is higher than the step i by 2 ($1 \leq i \leq N - 1$). Since the steps of the stairs are narrow, only one participant will stand on each step. A group photo will be taken when the participants are lined up behind each other.

A group photo will be taken soon. One participant stands on each step. Now, the participant H_i stands on the step i ($1 \leq i \leq N$).

However, since the difference of the heights of the participants are too large, if a photo is taken with the current order of the participants, some participants might be hidden behind other participants. Hence, you want to change the order of the participants so that at least the head of every participant shows on the photo. In other words, the following condition should be satisfied.

- Let a_i be the height of the participant on the step i ($1 \leq i \leq N$). Then, the inequality $a_i < a_{i+1} + 2$ is satisfied for every i ($1 \leq i \leq N - 1$).

You can only swap two consecutive participants. In other words, by an operation, you choose a step i ($1 \leq i \leq N - 1$) arbitrarily, and you swap the participant on the step i and the participant on the step $i + 1$.

You want to minimize the number of operations so that the above condition is satisfied.

Write a program which, given the order of the participants, calculates the minimum number of operations.

Input

Read the following data from the standard input. Given values are all integers.

N
$H_1 \cdots H_N$

Output

Write one line to the standard output. The output should contain the minimum number of operations.

Constraints

- $3 \le N \le 5\,000$.
- $1 \le H_i \le N$ $(1 \le i \le N)$.
- $H_i \ne H_j$ $(1 \le i < j \le N)$.

Subtasks

1. (5 points) $N \le 9$.
2. (7 points) $N \le 20$.
3. (32 points) $N \le 200$.
4. (20 points) $N \le 800$.
5. (36 points) No additional constraints.

Sample Input and Output

Sample Input 1	Sample Output 1
5 3 5 2 4 1	3

The condition is satisfied if you perform three operations as follows.

- First, you swap the participants on the steps 2 and 3. The heights of the participants become $3, 2, 5, 4, 1$ from a lower place to a higher place.
- Second, you swap the participants on the steps 4 and 5. The heights of the participants become $3, 2, 5, 1, 4$ from a lower place to a higher place.
- Finally, you swap the participants on the steps 3 and 4. The heights of the participants become $3, 2, 1, 5, 4$ from a lower place to a higher place. Then the condition is satisfied.

Since the condition is not satisfied if you perform operations less than three times, output 3.

Sample Input 2	Sample Output 2
5 3 2 1 5 4	0

The condition is already satisfied. You do not need to do any operation.

Sample Input 3	Sample Output 3
9 6 1 3 4 9 5 7 8 2	9

4

Robot

There are N crossings in the IOI town, numbered from 1 to N. There are M roads, numbered from 1 to M. Each road connects two different crossings in both directions. The road i ($1 \leq i \leq M$) connects the crossing A_i and the crossing B_i. No two different roads connect the same pair of crossings. Each of the roads has a color, which is described as an integer between 1 and M, inclusive. Currently, the color of the road i is C_i. More than one road may have the same color.

The JOI Co., Ltd. developed a robot moving around the crossings of the IOI town. Whenever you tell a color to the robot, the robot will find the road with that color, and then the robot will pass through it and moves to the adjacent crossing. However, if there are more than one roads with the told color connected to the current crossing of the robot, it cannot decide which road it should pass through, and will halt.

The robot is currently in the crossing 1. Your task is to move the robot to the crossing N by telling colors to it. However, it is not always true that the robot can be moved to the crossing N. You may change the colors of some of the roads **in advance** so that the robot can be moved to the crossing N. It costs P_i yen to change the color of the road i ($1 \leq i \leq M$) to any color between 1 and M, inclusive.

Write a program which, given the information of the crossings and the roads, calculates the minimum total cost. However, if it is impossible to move the robot to the crossing N even if you change the colors of the roads, output `-1` instead.

Input

Read the following data from the standard input. Given values are all integers.

$N\ M$
$A_1\ B_1\ C_1\ P_1$
$\vdots$
$A_M\ B_M\ C_M\ P_M$

Output

Write one line to the standard output. The output should contain the minimum total cost. However, if it is impossible to move the robot to the crossing N even if you change the colors of the roads, output `-1` instead.

Constraints

- $2 \leq N \leq 100\,000$.
- $1 \leq M \leq 200\,000$.
- $1 \leq A_i < B_i \leq N$ $(1 \leq i \leq M)$.
- $(A_i, B_i) \neq (A_j, B_j)$ $(1 \leq i < j \leq M)$.
- $1 \leq C_i \leq M$ $(1 \leq i \leq M)$.
- $1 \leq P_i \leq 1\,000\,000\,000$ $(1 \leq i \leq M)$.

Subtasks

1. (34 points) $N \leq 1\,000, \quad M \leq 2\,000$.
2. (24 points) $P_i = 1$ $(1 \leq i \leq M)$.
3. (42 points) No additional constraints.

Sample Input and Output

Sample Input 1	Sample Output 1
4 6 1 4 4 4 3 4 1 3 1 3 4 4 2 4 3 1 2 3 3 2 1 2 4 2	3

You can change the color of the road 4 from color 3 to color 4 at the cost of 1 yen. You can change the color of the road 6 from color 4 to color 2 at the cost of 2 yen. The total cost is 3 yen.

After that, you tell the color 2 to the robot, then it moves from the crossing 1 to the crossing 2. And, you tell the color 4 to the robot, then it moves to the crossing 4.

It is impossible to pay less than 3 yen so that the robot can be moved to the crossing 4. Hence output 3.

Sample Input 2	Sample Output 2

`5 2`
`1 4 1 2`
`3 5 1 4`
	`-1`

It is impossible to move the robot to the crossing 5 even if you change the colors of the roads. Hence output `-1`.

Sample Input 3	Sample Output 3

`5 7`
`2 3 7 1`
`1 4 5 1`
`4 5 3 1`
`3 4 7 1`
`2 4 3 1`
`3 5 6 1`
`1 2 5 1`
	`1`

This sample input satisfies the constraints of the subtask 2.

Sample Input 4	Sample Output 4
13 21 7 10 4 4 3 6 4 7 8 10 4 5 3 9 2 5 1 4 4 5 2 6 4 2 3 11 2 2 3 8 16 2 8 11 16 1 6 10 4 14 6 8 16 6 9 12 16 5 5 13 4 6 1 12 4 7 2 4 4 18 2 9 4 10 2 12 4 6 10 13 4 28 5 7 2 5 5 11 2 16 7 13 4 20	7

5

Dungeon 3

There is a dungeon with $N + 1$ floors. There are M players in the dungeon. The floors are numbered from 1 to $N + 1$, starting from the entrance. The players are numbered from 1 to M.

A player uses energy to move from a floor to the next floor. The amount of energy a player uses is A_i if he moves from the floor i ($1 \leq i \leq N$) to the floor $i + 1$. As this is a one-way dungeon, the only possible moves between floors are from the floor i to the floor $i + 1$ for some i ($1 \leq i \leq N$).

In each of the floors from the floor 1 to the floor N, inclusive, there is a fountain of recovery. At the fountain of recovery in the floor i ($1 \leq i \leq N$), a player can increase his energy by 1 paying B_i coins. A player can use a fountain multiple times as long as he has needed coins. However, each player has a maximum value of his energy, and his energy cannot exceed that value even if he uses a fountain of recovery.

Now the player j ($1 \leq j \leq M$) is in the floor S_j. His current energy is 0. His maximum value of energy is U_j. He wants to move to the floor T_j. His energy cannot be smaller than 0 along the way. How many coins does he need?

Write a program which, given the information of the dungeon and the players, determines whether it is possible for each player to move to the destination so that his energy does not become smaller than 0 along the way. If it is possible to move, the program should calculate the minimum number of coins he needs.

Input

Read the following data from the standard input. Given values are all integers.

N M
$A_1 \cdots A_N$
$B_1 \cdots B_N$
S_1 T_1 U_1
$\vdots$
S_M T_M U_M

Output

Write M lines to the standard output. The j-th line ($1 \leq j \leq M$) should contain the minimum number of coins the player j needs to move to the floor T_j. If it is impossible for the player j to move to the floor T_j, output -1 instead.

Constraints

- $1 \le N \le 200\,000$.
- $1 \le M \le 200\,000$.
- $1 \le A_i \le 200\,000$ $(1 \le i \le N)$.
- $1 \le B_i \le 200\,000$ $(1 \le i \le N)$.
- $1 \le S_j < T_j \le N + 1$ $(1 \le j \le M)$.
- $1 \le U_j \le 100\,000\,000$ $(1 \le j \le M)$.

Subtasks

1. (11 points) $N \le 3\,000$, $M \le 3\,000$.
2. (14 points) $U_1 = U_2 = \cdots = U_M$.
3. (31 points) $T_j = N + 1$ $(1 \le j \le M)$.
4. (44 points) No additional constraints.

Sample Input and Output

Sample Input 1	Sample Output 1

`5 4`
`3 4 1 1 4`
`2 5 1 2 1`
`1 6 3`
`1 6 4`
`3 5 1`
`2 5 9`
	`-1`
	`29`
	`3`
	`22`

Since the maximum value of energy of the player 1 is 3, the player 1 cannot move from the floor 2 to the floor 3. Hence the first line of output is -1.

On the other hand, the maximum value of energy of the player 2 is 4. The player 2 can move to the floor 6 by the following way.

- In the floor 1, he pays 8 coins, and his energy becomes 4. Then he moves to the floor 2, and his energy becomes 1.
- In the floor 2, he pays 15 coins, and his energy becomes 4. Then he moves to the floor 3, and his energy becomes 0.
- In the floor 3, he pays 4 coins, and his energy becomes 4. Then he moves to the floor 4, and his energy becomes 3.
- In the floor 4, he does not pay coins. Then he moves to the floor 5, and his energy becomes 2.
- In the floor 5, he pays 2 coins, and his energy becomes 4. Then he moves to the floor 6, and his energy becomes 0.

In total, the player 2 pays 29 coins. Since it is impossible for the player 2 to move to the floor 6 by paying less than 29 coins, the second line of output is 29.

Sample Input 2	Sample Output 2
10 10 1 8 9 8 1 5 7 10 6 6 10 10 2 8 10 3 9 8 3 7 2 11 28 5 11 28 7 11 28 1 11 18 3 11 18 8 11 18 4 11 11 6 11 11 10 11 11 9 11 5	208 112 179 248 158 116 234 162 42 -1

This sample input satisfies the constraints of the subtask 3.

Sample Input 3	Sample Output 3
20 20	151
2 3 2 11 4 6 9 15 17 14 8 17 3 12 20 4 19 8 4 5	591
19 3 18 2 13 7 5 19 10 1 12 8 1 15 20 1 13 2 18 6	4
12 15 67	284
7 15 18	339
16 17 14	517
9 21 97	35
1 19 43	581
3 18 31	254
16 20 70	58
7 20 28	-1
1 16 61	178
3 5 69	519
9 10 15	-1
2 13 134	-1
11 19 23	-1
16 20 14	219
5 21 16	-1
15 20 11	-1
7 11 54	214
7 16 16	
13 17 10	
3 15 135	