1

Just Long Neckties

Have you ever heard of Just Odd Inventions, Ltd.? This company is known for their "just odd inventions." We call it JOI, Ltd. in this problem.

JOI, Ltd. has invented its newest product "Just Long Neckties". There are $N + 1$ types of neckties, numbered 1 to $N + 1$. The length of the i-th necktie ($1 \le i \le N + 1$) is A_i.

The company gathered their employees to hold a try-on party. N employees participate in the party, and the j-th employee ($1 \le j \le N$) initially wears a necktie of length B_j.

The try-on party is held following this procedure:

1. CEO of JOI, Ltd. chooses a necktie, which is not used at the party.
2. Then, each employee chooses one of the remaining neckties to try on. No two employees choose the same necktie.
3. Finally, each employee takes off the necktie which (s)he initially wears and puts the selected necktie on.

If an employee initially wearing a necktie of length b tries a necktie of length a, (s)he feels **strangeness** of $\max\{a - b, 0\}$. The **oddity** of the try-on party is defined as the maximum strangeness among the employees.

We also define C_k as the minimum oddity of the try-on party if CEO of JOI, Ltd. chooses the k-th necktie.

Write a program which, given the lengths of the neckties used at the party and the neckties each employee initially wears, calculates the values of $C_1, C_2, \ldots, C_{N+1}$.

Input

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

N
$A_1 \ldots A_{N+1}$
$B_1 \ldots B_N$

Output

Write one line to the standard output. The output should contain the values of $C_1, C_2, \ldots, C_{N+1}$, separated by a space.

Constraints

- $1 \le N \le 200\,000$.
- $1 \le A_i \le 1\,000\,000\,000$ ($1 \le i \le N + 1$).
- $1 \le B_j \le 1\,000\,000\,000$ ($1 \le j \le N$).

Subtasks

1. (1 point) $N \leq 10$.
2. (8 points) $N \leq 2000$.
3. (91 points) No additional constraints.

Sample Input and Output

Sample Input 1	Sample Output 1
3 4 3 7 6 2 6 4	2 2 1 1

Here is an example of a try-on party:

- CEO of JOI, Ltd. chooses the 4th necktie. This necktie is not used in the party.
- The employee 1 chooses the 1st necktie, the employee 2 chooses the 2nd necktie, the employee 3 chooses the 3rd necktie.
- Each employee tries the necktie they choose on.

In this case, strangeness of each employee is 2, 0, 3 in order. Therefore, the oddity of the party is 3.

It is possible to decrease the oddity to 1 if the employees choose different neckties. One of the example is:

- CEO of JOI, Ltd. chooses the 4th necktie. This necktie is not used in the party.
- The employee 1 chooses the 2nd necktie, the employee 2 chooses the 3rd necktie, the employee 3 chooses the 1st necktie.
- Each employee tries the necktie they choose on.

In this case, strangeness of each employee is 1, 1, 0 in order. Therefore, the oddity of the party is 1.

This is the minimum possible oddity when CEO of JOI, Ltd. chooses the 4th necktie, so $C_4 = 1$.

Sample Input 2	Sample Output 2
5 4 7 9 10 11 12 3 5 7 9 11	4 4 3 2 2 2

2

JJOOII 2

Bitaro received a string S of length N for his birthday present. String S consists of three kinds of characters, `J`, `O` and `I`.

For each positive integer K, we will call the string which consists of K `J`'s, K `O`'s, and K `I`'s in this order **JOI-string of level** K. For example, `JJOOII` is a JOI-string of level 2.

Bitaro likes a JOI-string of level K, so he is going to make a JOI-string of level K from string S by using the following three operations any number of times in arbitrary order:

Operation 1 Bitaro deletes the first character of S.

Operation 2 Bitaro deletes the last character of S.

Operation 3 Bitaro deletes a character of S which is neither the first nor the last.

Because using Operation 3 is time-consuming, Bitaro wants to make a JOI-string of level K with as small number of Operation 3 as possible.

Write a program which, given a string S of length N and a positive integer K, prints the smallest number of Operation 3 required to make a JOI-string of level K from S. If it is impossible to make a JOI-string of level K with the operations, print -1 instead.

Input

Read the following data from the standard input. N and K are integers. S is a string.

N K
S

Output

Write one line to the standard output. The output should contain the smallest number of Operation 3 required to make a JOI-string of level K from S. If it is impossible to make a JOI-string of level K, print -1 instead.

Constraints

- $3 \leq N \leq 200\,000$.
- $1 \leq K \leq \frac{N}{3}$.
- S is a string of length N which consists of `J`, `O` and `I`.

Subtasks

1. (1 point) $N \leq 21$.
2. (12 points) $N \leq 3\,000$.
3. (87 points) No additional constraints.

Sample Input and Output

Sample Input 1	Sample Output 1
10 2 OJIJOIOIIJ	2

You can make a JOI-string of level K from string S by the following operations:

1. You use Operation 1 and S becomes `JIJOIOIIJ`.
2. You use Operation 2 and S becomes `JIJOIOII`.
3. You use Operation 3 to remove the second character and S becomes `JJOIOII`.
4. You use Operation 3 to remove the fourth character and S becomes `JJOOII`.

It is impossible to make a JOI-string of level K with using Operation 3 less than twice, so you should print 2.

Sample Input 2	Sample Output 2
9 3 JJJOOOIII	0

You need not use an operation.

Sample Input 3	Sample Input 3
9 1 IIIOOOJJJ	-1

In this sample, it is impossible to make a JOI-string of level 1 from string S.

3 Collecting Stamps 3

Republic of IOI, where JOI-kun lives, is famous for a large lake. Today, a stamp rally event takes place around the lake.

There are N types of stamps situated around the lake. These stamps are numbered from 1 to N, in clockwise order. The perimeter of the lake is L, and the i-th stamp ($1 \leq i \leq N$) is located at X_i meters clockwise from the starting point of the stamp rally.

Each participant stands at the starting point of the stamp rally. After the rally starts, each participant can move around the lake, both clockwise and counter-clockwise. Each participant can collect the i-th stamp ($1 \leq i \leq N$) only if they arrive at where the i-th stamp is located within T_i seconds (inclusive) since the rally starts.

JOI-kun is a participant of the stamp rally. He takes 1 second to move 1 meter. You can ignore all the other times which he takes.

Write a program that, given the number of types of stamps, the perimeter of the lake, where each stamp is located, and the time until which JOI-kun can collect each stamp, calculates the maximum number of types of stamps he can collect in total.

Input

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

N L
$X_1 \ldots X_N$
$T_1 \ldots T_N$

Output

Write the answer in one line to the standard output.

Constraints

- $1 \leq N \leq 200$.
- $2 \leq L \leq 1\,000\,000\,000$.
- $1 \leq X_i < L$ ($1 \leq i \leq N$).
- $X_i < X_{i+1}$ ($1 \leq i \leq N - 1$).
- $0 \leq T_i \leq 1\,000\,000\,000$ ($1 \leq i \leq N$).

Subtasks

1. (5 points) $N \leq 12$, $L \leq 200$, $T_i \leq 200$ $(1 \leq i \leq N)$.
2. (10 points) $N \leq 15$.
3. (10 points) $L \leq 200$, $T_i \leq 200$ $(1 \leq i \leq N)$.
4. (75 points) No additional constraints.

Sample Input and Output

Sample Input 1	Sample Output 1
6 25 3 4 7 17 21 23 11 7 17 10 8 10	4

JOI-kun can collect 4 types of stamps as described below:

1. Walk 2 meters counter-clockwise. He can collect the 6th stamp as it is 2 minutes since the rally starts.
2. Walk 2 meters counter-clockwise. He can collect the 5th stamp as it is 4 minutes since the rally starts.
3. Walk 7 meters clockwise. He can collect the 1st stamp as it is 11 minutes since the rally starts.
4. Walk 1 meter clockwise. He cannot collect the 2nd stamp as it is 12 minutes since the rally starts.
5. Walk 3 meters clockwise. He can collect the 3rd stamp as it is 15 minutes since the rally starts.

It is impossible for JOI-kun to collect 5 or more stamps, so the answer is 4.

Sample Input 2	Sample Output 2
5 20 4 5 8 13 17 18 23 15 7 10	5

JOI-kun can collect all the stamps by walking around the lake counter-clockwise.

Sample Input 3	Sample Output 3
4 19 3 7 12 14 2 0 5 4	0

Unfortunately, JOI-kun cannot collect any stamps, no matter how he moves.

Sample Input 4	Sample Output 4
10 87 9 23 33 38 42 44 45 62 67 78 15 91 7 27 31 53 12 91 89 46	5

4

Olympic Bus

There are N cities in JOI Kingdom, numbered from 1 to N. There are M bus lines connecting cities, numbered from 1 to M. The i-th bus line ($1 \leq i \leq M$) runs from the city U_i to the city V_i, and its fare is C_i yen. On the i-th bus line ($1 \leq i \leq M$), a passenger cannot get on the bus in a city other than the city U_i. Also, a passenger cannot get off the bus in a city other than the city V_i. There may be more than one bus lines from a city to another city.

The Olympic Games will be held in JOI Kingdom soon. President K is the Minister of Transport of JOI Kingdom. President K will choose *at most one* bus line, and invert its direction without changing its fare just before the Olympic Games. Namely, if he chooses the i-th bus line ($1 \leq i \leq M$), it will not run from the city U_i to the city V_i during the Olympic Games; instead, it will run from the city V_i to the city U_i. The cost to invert the direction is D_i yen, and it will be paid by President K. In order to avoid confusion, it is not allowed to invert the direction during the Olympic Games.

Since President K is the Minister of Transport, during the Olympic Games, he will make a round trip between the city 1 and the city N using the bus lines. By choosing (or not choosing) a bus line to be inverted appropriately, he wants to minimize the sum of the cost of the round trip and the cost to invert the chosen bus line.

Write a program which, given the number of cities and information of the bus lines, calculates the minimum sum of the cost of the round trip and the cost to invert the chosen bus line. If it is not possible to make a round trip between the city 1 and the city N by choosing a bus line to be inverted, output -1 instead.

Input

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

N M
U_1 V_1 C_1 D_1
$\vdots$
U_M V_M C_M D_M

Output

Write the minimum sum of the cost of the round trip and the cost to invert the chosen bus line to the standard output. If it is not possible to make a round trip between the city 1 and the city N, write -1 instead.

Constraints

- $2 \leq N \leq 200$.
- $1 \leq M \leq 50\,000$.
- $1 \leq U_i \leq N$ ($1 \leq i \leq M$).
- $1 \leq V_i \leq N$ ($1 \leq i \leq M$).

- $U_i \neq V_i$ $(1 \leq i \leq M)$.
- $0 \leq C_i \leq 1\,000\,000$ $(1 \leq i \leq M)$.
- $0 \leq D_i \leq 1\,000\,000\,000$ $(1 \leq i \leq M)$.

Subtasks

1. (5 points) $M \leq 1000$.
2. (11 points) M is an even integer, $U_{2i-1} = U_{2i},\ V_{2i-1} = V_{2i},\ C_{2i-1} = C_{2i}\ (1 \leq i \leq \frac{M}{2})$.
3. (21 points) $C_i = 0$ $(1 \leq i \leq M)$.
4. (63 points) No additional constraints.

Sample Input and Output

Sample Input 1	Sample Output 1
4 5 1 2 4 4 1 3 2 1 4 3 1 2 4 1 6 1 2 4 2 5	10

Assume that President K will invert the direction of the 2nd bus line; its cost is 1 yen. Then, the minimum cost to travel from the city 1 to the city 4 will be 6 yen, and the minimum cost to travel from the city 4 to the city 1 will be 3 yen. Thus, the sum of the cost of the round trip between the city 1 and the city 4 and the cost to invert the chosen bus line will be 10 yen.

Since the sum of the cost of the round trip between the city 1 and the city 4 and the cost to invert the chosen bus line cannot be cheaper than 10 yen, output 10.

Sample Input 2	Sample Output 2
4 10 1 2 4 4 1 2 4 4 1 3 2 1 1 3 2 1 4 3 1 2 4 3 1 2 4 1 6 1 4 1 6 1 2 4 2 5 2 4 2 5	10

This sample input satisfies the constraints of Subtask 2.

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

This sample input satisfies the constraints of Subtask 3.

Sample Input 4	Sample Output 4
4 5 1 2 4 4 1 3 2 4 4 3 1 5 4 1 6 1 2 4 2 5	12

It is not necessary to invert the direction of a bus line.

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

In this sample input, there are 2 bus lines from the city 4 to the city 3.

5 Fire

There are N districts in JOI Village, numbered from 1 to N. These districts are located in a line. Now, a fire occurs in each district. At time 0, the strength of the fire in the i-th district ($1 \leq i \leq N$) is S_i ($S_i > 0$).

At time 0, the wind blows from the 1st district to the N-th district. For every pair of neighboring districts, if the fire in the upwind district is stronger than the fire in the downwind district at time t ($0 \leq t$), the strength of the fire in the downwind district at time $t + 1$ will be the strength of the fire in the upwind district at time t. Otherwise, the strength of the fire in the downwind district at time $t + 1$ and time t are the same. Namely, if the strength of the fire in the i-th district ($1 \leq i \leq N$) at time t ($0 \leq t$) is denoted by $S_i(t)$, we have $S_i(t) = \max\{S_{i-1}(t-1), S_i(t-1)\}$ for every t ($1 \leq t$). Here, for any t ($0 \leq t$), we put $S_0(t) = 0$. For any i ($1 \leq i \leq N$), we put $S_i(0) = S_i$.

You are a firefighter. You have Q plans to extinguish the fire. You are planning to do only one of the Q plans. In the j-th plan ($1 \leq j \leq Q$), you will use fire extinguishing agent for the k-th district for every k with $L_j \leq k \leq R_j$, and extinguish the fire in these districts. If the strength of the fire in a district is s, you need s liters of fire extinguishing agent to extinguish the fire in that district. Therefore, the amount of fire extinguishing agent needed for the j-th plan is $S_{L_j}(T_j) + S_{L_j+1}(T_j) + \cdots + S_{R_j}(T_j)$ liters.

In order to examine the plan to be done, you want to know the amount of fire extinguishing agent needed for each plan.

Write a program which, given the strength of the fire at time 0 and information of fire extinguishing plans, calculates the amount of fire extinguishing agent needed for each plan.

Input

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

N Q
$S_1 \ldots S_N$
T_1 L_1 R_1
$\vdots$
T_Q L_Q R_Q

Output

Write Q lines to the standard output. In the j-th line ($1 \leq j \leq Q$), output the amount of fire extinguishing agent needed for the j-th plan.

Constraints

- $1 \leq N \leq 200\,000$.
- $1 \leq Q \leq 200\,000$.

- $1 \le S_i \le 1\,000\,000\,000$ $(1 \le i \le N)$.
- $1 \le T_j \le N$ $(1 \le j \le Q)$.
- $1 \le L_j \le R_j \le N$ $(1 \le j \le Q)$.

Subtasks

1. (1 point) $N \le 200$, $Q \le 200$.
2. (6 points) $T_1 = T_2 = \cdots = T_Q$.
3. (7 points) $L_j = R_j$ $(1 \le j \le Q)$.
4. (6 points) $S_i \le 2$ $(1 \le i \le N)$.
5. (80 points) No additional constraints.

Sample Input and Output

Sample Input 1	Sample Output 1
5 5 9 3 2 6 5 1 1 3 2 1 5 3 2 5 4 3 3 5 3 5	21 39 33 9 27

- At time 0, the strength of the fire in each district is $9, 3, 2, 6, 5$ from the 1st district.
- At time 1, the strength of the fire in each district is $9, 9, 3, 6, 6$ from the 1st district. The amount of fire extinguishing agent needed for the 1st plan is $9 + 9 + 3 = 21$ liters.
- At time 2, the strength of the fire in each district is $9, 9, 9, 6, 6$ from the 1st district. The amount of fire extinguishing agent needed for the 2nd plan is $9 + 9 + 9 + 6 + 6 = 39$ liters.
- At time 3, the strength of the fire in each district is $9, 9, 9, 9, 6$ from the 1st district. The amount of fire extinguishing agent needed for the 3rd plan is $9 + 9 + 9 + 6 = 33$ liters.
- At time 4, the strength of the fire in each district is $9, 9, 9, 9, 9$ from the 1st district. The amount of fire extinguishing agent needed for the 4th plan is 9 liters.
- At time 5, the strength of the fire in each district is $9, 9, 9, 9, 9$ from the 1st district. The amount of fire extinguishing agent needed for the 5th plan is $9 + 9 + 9 = 27$ liters.

Sample Input 1 satisfies the constraints of Subtask 1 and Subtask 5.

Sample Input 2	Sample Output 2
10 10	28
3 1 4 1 5 9 2 6 5 3	21
1 1 6	34
2 8 10	4
4 2 7	64
8 3 3	43
6 1 10	55
3 2 8	9
5 1 9	27
7 4 5	9
9 7 9	
10 10 10	

Sample Input 2 satisfies the constraints of Subtask 1 and Subtask 5.

Sample Input 3	Sample Output 3
10 10	9
3 1 4 1 5 9 2 6 5 3	9
1 6 6	3
2 8 8	4
4 2 2	3
8 3 3	4
6 1 1	5
3 4 4	9
5 5 5	9
7 10 10	9
9 8 8	
10 7 7	

Sample Input 3 satisfies the constraints of Subtask 1, Subtask 3, and Subtask 5.

Sample Input 4	Sample Output 4
10 10	28
3 1 4 1 5 9 2 6 5 3	27
7 1 6	34
7 8 10	4
7 2 7	64
7 3 3	43
7 1 10	55
7 2 8	9
7 1 9	27
7 4 5	9
7 7 9	
7 10 10	

Sample Input 4 satisfies the constraints of Subtask 1, Subtask 2, and Subtask 5.

Sample Input 5	Sample Output 5

`20 20`
`2 1 2 2 1 1 1 1 2 2 2 1 2 1 1 2 1 2 1 1`
`1 1 14`
`2 3 18`
`4 10 15`
`8 2 17`
`9 20 20`
`4 8 19`
`7 2 20`
`11 1 5`
`13 2 8`
`20 1 20`
`2 12 15`
`7 1 14`
`12 7 18`
`14 2 17`
`9 19 20`
`12 12 12`
`6 2 15`
`11 2 15`
`19 12 17`
`4 1 20`
	`25`
	`30`
	`12`
	`32`
	`2`
	`24`
	`38`
	`10`
	`14`
	`40`
	`8`
	`28`
	`24`
	`32`
	`4`
	`2`
	`28`
	`28`
	`12`
	`40`

Sample Input 5 satisfies the constraints of Subtask 1, Subtask 4, and Subtask 5.