

Hunan University 2023 the 19th Programming Contest



Contest Section



acm International Collegiate
Programming Contest



Hunan University
May 21, 2023



Problem Set

- A Playf and ABC**
- B Color**
- C Construct A Permutation**
- D Sakuyalove and Fast FFT**
- E stone(easy version)**
- F stone(hard version)**
- G XOR Segments**
- H Contest Simulation**
- I Infinite recurring Decimal**
- J Sakuyalove and Latin Square**
- K The Demon Jester**
- L Timetable Queries**

The 2023 19th Hunan University Programming Contest

Problem A. Playf and ABC

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

Playf has a string S with length n , which only contains three types of letters A, B, and C. She wants to find some triples (i, j, k) where $0 \leq i < j < k < n$, such that $\overline{S_i S_j S_k} = ABC$ or $\overline{S_i S_j S_k} = CBA$. But Playf does not want each character to be used repeatedly, so the index of each character appears in at most one triple.

Playf wants to know how many triples she can find at most, please help her.

Input

The first line contains a string S ($1 \leq |S| \leq 3 \times 10^5$).

Output

Print an integer, the maximum number of triples Playf can find.

Examples

standard input	standard output
ABCBBAC	2
ABCBBACCA	2
BACABA	1

Note

For example1, two triples are (0,1,6) and (2,3,5). There might be other set of two triples, but Playf can not find more than 2 triples in this example.

Problem B. Color

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

There are n objects with different colors in a row. We use integers in $[1, n]$ to represent different colors. Now you can do operations like below:

Choose any number of objects with the same color, and change their colors arbitrarily. You must guarantee that after the operation, colors of objects you chose are **non – decreasing**, i. e. that each element is not less than the previous **selected** element.

Calculate the minimum number of operations to make the whole color sequence **non – decreasing**.

For example, for the color sequence $[1, 5, 4, 5, 5, 3, 5]$, choose first three objects with color 5, and change them to 1, 3, 3 ($1 \leq 3 \leq 3$), the sequence will turn into $[1, 1, 4, 3, 3, 3, 5]$. Then choose the objects with color 4 and change it into 2, the whole sequence will become $[1, 1, 2, 3, 3, 3, 5]$ and satisfy our goal.

Input

First line contains an integer n ($1 \leq n \leq 2 \times 10^5$).

Second line contains n integers $c_1, c_2, \dots, c_n$ ($1 \leq c_i \leq n$), c_i represents the color of the i th object.

Output

Print one integer– the minimum number of operations to make the whole color sequence non-decreasing.

Examples

standard input	standard output
5 1 2 3 1 2	2
5 1 5 2 5 1	2
10 5 1 2 4 1 4 4 9 2 6	4

Problem C. Construct A Permutation

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

p is a permutation with length n , and p_i denotes the i -th element of p .

find a permutation of length n that this permutation has the maximum value of

$$||\dots||p_1 - p_2| - p_3|\dots| - p_{n-1}| - p_n|$$

among all possible permutations with length n .

Input

the input contains only one integer n ($1 \leq n \leq 10^6$).

Output

output n integers, denoting the answer. if there are multiple answers, output any.

Examples

standard input	standard output
1	1
2	1 2
6	4 3 5 2 1 6
15	13 2 10 3 1 12 8 4 5 9 7 6 15 11 14

Note

A permutation of length n is an array of length n , and each integer from 1 to n appears exactly once in that array. for example, $[1, 3, 4, 2]$, $[1]$, $[4, 3, 2, 1, 5]$ are permutations while $[1, 1, 4, 5, 1, 4]$, $[2]$, $[1, -2]$ are not.

Problem D. Sakuyalove and Fast FFT

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

Sakuyalove has an array a with $n + 1$ integers $a_0, a_1, \dots, a_n$. She wants to build two arrays b, c such that:

$$b_k = \sum_{i=0}^k C_k^i (-1)^i a_i \mod 1000000007$$

$$c_k = \sum_{i=0}^k C_k^i (-1)^i b_i \mod 1000000007$$

for $k = 0, 1, \dots, n$, where $C_k^i = \frac{k!}{i!(k-i)!}$ for integer $i, k (0 \leq i \leq k)$.

Now Sakuyalove wants to know the answer of $c_0 \oplus c_1 \oplus \dots \oplus c_n$ (the operator " $\oplus$ " means XOR operation). Please help her.

Input

The input contains two lines.

The first line contains one integer $n (1 \leq n \leq 10^6)$, which describes the size of array a is $n + 1$.

The second line contains $n + 1$ integers $a_0, a_1, \dots, a_n (0 \leq a_i < 10^9 + 7)$ separated by space, which describes the element of array a .

Output

The output contains one line with one integer, which is the answer of $c_0 \oplus c_1 \oplus \dots \oplus c_n$.

Examples

standard input	standard output
1 0 1	1
7 25 34 18 26 0 55 37 11	42
1 1000000006 73741826	1073740804

Note

For the first sample input, we have: $n = 1, a_0 = 0, a_1 = 1$. Then we can calculate:

$$b_0 = a_0 \mod 1000000007 = 0,$$

$$b_1 = (a_0 - a_1) \mod 1000000007 = 1000000006,$$

$$c_0 = b_0 \mod 1000000007 = 0,$$

$$c_1 = (b_0 - b_1) \mod 1000000007 = 1.$$

So the answer is $c_0 \oplus c_1 = 0 \oplus 1 = 1$.

Problem E. stone(easy version)

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

This is the the easy version. The only difference bewteen this problem and the other problem is the constraint of k . In this problem, $k = 1$.

Alice and Bob are playing a new stone game.

There are n piles of stones in a row, the i -th pile has a_i stones. They play in turn and Alice goes first. In each turn, the player performs the following operation exactly k times consecutively[1]:

Remove any number of stones(at least one) from the leftmost pile of stones.

When all stones are removed, the player who will perform the next operation loses, and therefore, their opponent wins. Who will win if they both play optimally?

Input

There are T test cases.

For each test case, there are two lines.

The first line has two intergers n, k , indicates the number of piles and the number of operations performed by a player consecutively.

The second line has n intergers, indicates the number of stones of every pile, from left to right.

$1 \leq T \leq 10^5, 1 \leq \sum n \leq 10^5, k = 1, 1 \leq a_i \leq 10^9$

Output

Out put T lines. For each test case, output one line 'Alice' or 'Bob', indicates who will win.

Example

standard input	standard output
3	Alice
1 1	Bob
2	Alice
2 1	
1 2	
3 1	
2 2 2	

Note

[1]consecutively: one after another without an interruption.

In the third case of the example, there are three piles of stones, denoted as $(2, 2, 2)$. One possible state changes as follows:

$(2, 2, 2) \xrightarrow{Alice} (1, 2, 2) \xrightarrow{Bob} (2, 2) \xrightarrow{Alice} (1, 2) \xrightarrow{Bob} (2) \xrightarrow{Alice} ()$

Thus, Bob loses.

Problem F. stone(hard version)

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

This is the the hard version. The only defference bewteen this problem and the other problem is the constraint of k . In this problem, $1 \leq k \leq 10^5$.

Alice and Bob are playing a new stone game.

There are n piles of stones in a row, the i -th pile has a_i stones. They play in turn and Alice goes first. In each turn, the player performs the following operation exactly k times consecutively:

Remove any number of stones(at least one) from the leftmost pile of stones.

When all stones are removed, the player who will perform the next operation loses, and therefore, their opponent wins. Who will win if they both play optimally?

Input

There are T test cases.

For each test case, there are two lines.

The first line has two intergers n, k , indicates the number of piles and the number of operations performed by a player consecutively.

The second line has n intergers, indicates the number of stones of every pile, from left to right.

$1 \leq T \leq 10^5, 1 \leq \sum n \leq 10^5, 1 \leq k \leq 10^5, 1 \leq a_i \leq 10^9$

Output

Out put T lines. For each test case, output one line 'Alice' or 'Bob', indicates who will win.

Example

standard input	standard output
3	Alice
2 1	Alice
2 1	Bob
3 2	
3 1 1	
3 2	
1 1 3	

Note

In the second test case of the example, there are three piles of stones, denoted as $(3, 1, 1)$. Because $k = 2$, Alice operates twice first, then Bob operates twice, then Alice operates twice again and so on. One possible state changes as follows:

$(3, 1, 1) \xrightarrow{Alice} (1, 1) \xrightarrow{Alice} (1) \xrightarrow{Bob} ()$

The next operator is Bob and there are no stones. Thus Bob loses.

Problem G. XOR Segments

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

One Day when Lskkkno1 was struggling with Codeforces, he met a question called ‘AND Segments’, he immediately worked out the question but only to find that he misunderstood the question. :D

Now Lskkkno1 is very annoying, so he wants you to solve the misunderstood question too!

Given three non-negative integers n , k , m , and m restrictions $(l_1, r_1, x_1), (l_2, r_2, x_2), \dots, (l_m, r_m, x_m)$.

Please calculate how many arrays of length n satisfy the following requirements:

- for each $1 \leq i \leq n$, $0 \leq a_i < 2^k$.

- for each $1 \leq i \leq m$, $a_{l_i} \oplus a_{l_i+1} \oplus \dots \oplus a_{r_i} = x_i$. ($\oplus$ is bitwise exclusive or operator)

Two arrays a , b are different if and only if there exists a position i that $a_i \neq b_i$.

For the answer may be very large, you should print the answer modulo 998244353.

Input

The first line contains three integers n , k , m — the length of array a , the elements in array a should less than 2^k , and the number of restrictions.

The next m lines each contains three integers, l_i, r_i, x_i describe a piece of restriction.

It's guaranteed that $n, m \leq 10^6$, $k \leq 31$, $1 \leq l_i \leq r_i \leq n$, $x_i < 2^k$.

Output

Output a single integer — the number of arrays satisfying the requirements modulo 998244353.

Examples

standard input	standard output
2 1 0	4
3 1 1 1 3 0	4
2 0 1 1 2 0	1

Note

In the first example, there are no restrictions, so the arrays are : $[0, 0]$, $[0, 1]$, $[1, 0]$, $[1, 1]$.

In the second example, the four arrays are : $[0, 0, 0]$, $[0, 1, 1]$, $[1, 0, 1]$, $[1, 1, 0]$ which satisfy $a_1 \oplus a_2 \oplus a_3 = 0$.

In the third example, the only array is : $[0, 0]$, because a_i must be less than 2^k .

Problem H. Contest Simulation

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

This is a simulation problem. Briefly speaking, you need to read the submissions of this contest, and output the ranking.

Hunan University is now holding a programming contest. This contest is based on ICPC rules but maybe a little different.

The contest is made of n ($1 \leq n \leq 20$) problems and m ($1 \leq m \leq 2000$) participants, each problem can be solved by any participant if they could.

During the contest, aiming to solve more problems by using less time, the participants could make submissions for any problem they want, one participant can make no more than one submission in one second, the submissions will be described as follows.

A submission contains the following information:

1. user name, denotes who make this submission.
2. time, denotes when do this submission was made.
3. problem number, denotes which problem do this submission for.
4. status, denotes whether the problem is solved by this submission or not, the details of status are as follows:

AC means Accepted, if a submission turns out to be this status, the problem is solved by this participant at this time.

WA means Wrong Answer, **TL** means Time Limit Exceed, **ML** means Memory Limit Exceed, **RE** means Runtime Error, **CE** means Compilation Error, all these status mean that there must be some problems in this submission, thus this problem is not solved by this participant at this time.

The duration of this contest is exactly **5 hours**, in other words, 300 minutes or 18000 seconds, participants can not make submissions before or after the contest.

When the contest ends in 5 hours, a ranking will be made for awards. Participants who solve more problems will be ranked in front of those who solve problems less than them. If two participants solve the same amount of problems, who has the less penalty will be ranked in front of the other participant.

The penalty is calculated in the following formula.

For a specific problem, if this participant never solved this problem during the contest, the penalty of this problem is 0. If this participant first solved it at time t_1 minutes after the beginning of this contest, and before this first submission with status AC, there was f submissions with WA, TL, ML and RE but **except CE**, the penalty of this problem is $t_1 + 20 \times f$.

The penalty of a participant is the sum of their penalty over all problems.

For example, Steve has the following submissions:

Steve 00:12:55 A CE

Steve 00:15:12 A WA

Steve 00:16:30 A AC

Steve 01:34:07 C TL

The penalty of Steve is $16 + 1 \times 20 = 36$. The first CE on problem A does not count for penalty and the same as the last TL on problem C since Steve tried but did not solve this problem.

If two participants have the same amount of problem solved, and the same penalty, whose user name is

lexicographically smaller will be ranked in front of the other participant. That's so unfair right? But just take it. We don't want the rules to be more complex.

And finally, this ranking will be used for awards, you should output this ranking with some dividing lines, denotes who win the GOLD/SILVER/BRONZE medals. you could refer to the samples for the specific format.

The rules of medals awards are as follows.

The priority of the medals is: GOLD>SILVER>BRONZE.

At least 10% of the participants wins GOLD medal.

At least 30% of the participants wins SILVER medal or higher.

At least 60% of the participants wins BRONZE medal or higher.

If it is possible to distribute medals in several ways, the option with the smallest number of gold medals is chosen. If there are several such options, the option with the smallest number of silver medals is chosen. If there are several such options, the option with the smallest number of bronze medals is chosen. Thus, the distribution must be determined.

Now could you please solve this problem?

Input

The first line contains three numbers, n , m , and t ($1 \leq n \leq 20, 1 \leq m \leq 2000, 0 \leq t \leq 2 \times 10^5$), denotes the number of problems, the number of participants, and the number of submissions.

The next m lines, each line contains one string $username_i$ ($1 \leq |username_i| \leq 100$), denotes the user name of the i -th participant. It is guaranteed that this string contains only English letters (lowercase and uppercase), numbers and '_'. No duplicate names incident would happen.

The next t lines, each line contains four strings separated by spaces, $username_i$, $time_i$, $problem_i$, and $status_i$ ($1 \leq |username_i| \leq 100, |time_i| = 8, |problem_i| = 1, |status_i| = 2$), denotes the information of the i -th submission.

it is guaranteed the following restrictions always being satisfied.

$username_i$ must be some one of the m usernames above.

$time_i$ is in the format of $HH : MM : SS$ ($0 \leq HH < 5, 0 \leq MM < 60, 0 \leq SS < 60$), for example, 01:23:45, 02:34:59, which means this submission was submitted on the HH hours MM minutes and SS seconds after the contest begins.

No participant can make submission twice or more at the same second.

$problem_i$ is an uppercase English letter and $A \leq problem_i < A + n$, A means the uppercase letter 'A', and $A + n$ means the n -th letter after 'A'.

$status_i$ is a string contains only two letters, and it must one string in the list "AC", "WA", "TL", "ML", "RE", "CE".

Notice that the input of submissions could be at any order.

Output

The output contains exactly $m + 3$ lines, each line is either a dividing line or information of a specific participant in the ranking at that position.

If the i -th line is the participant information, and there is d ($0 \leq d \leq \min(3, i - 1)$) dividing lines above, this line should be the information of the $i - d$ -th participant in the ranking. The information contains 3 strings separated by spaces, their username, number of problems they solved, and their total penalty.

If the i -th line is the dividing line, just output "GOLD", "SILVER" or "BRONZE" in one line. The participants above this dividing line should be awarded with the corresponding medal or higher, and those participants below should not. For example, if this line is "SILVER", the participants above this line

should be awarded with GOLD or SILVER medals but those below should not.

Examples

standard input	standard output
13 10 16 Alice Bob ChinaBoy Dingzhen_official Eileen_Gu Fulafu Genshin Huawei_Aiguo IKEA Jojo Alice 00:00:00 M WA Alice 00:01:20 M AC Alice 01:20:32 A ML Bob 00:07:19 D AC Bob 00:09:23 E WA Bob 00:20:27 E AC Dingzhen_official 00:23:24 D AC Dingzhen_official 01:34:17 E AC Dingzhen_official 03:09:53 J AC Dingzhen_official 03:27:03 H AC Eileen_Gu 01:31:59 A AC Eileen_Gu 01:31:58 B AC Fulafu 01:31:00 A AC Fulafu 01:31:01 B AC Fulafu 01:31:20 B WA Fulafu 01:31:30 B AC	Dingzhen_official 4 513 GOLD Bob 2 47 Eileen_Gu 2 182 SILVER Fulafu 2 182 Alice 1 21 ChinaBoy 0 0 BRONZE Genshin 0 0 Huawei_Aiguo 0 0 IKEA 0 0 Jojo 0 0
1 2 3 Dingzhen_official Eileen_Gu Dingzhen_official 04:59:58 A CE Dingzhen_official 04:59:59 A AC Eileen_Gu 04:59:00 A AC	Dingzhen_official 1 299 GOLD SILVER Eileen_Gu 1 299 BRONZE
1 1 2 Dingzhen_official Dingzhen_official 03:31:35 A WA Dingzhen_official 04:59:59 A AC	Dingzhen_official 1 319 GOLD SILVER BRONZE

Problem I. Infinite recurring Decimal

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

Consider an infinite recurring decimal $\frac{1}{n}$, Where n is a positive integer, n is not a multiple of 2 and 5. For example:

$$\frac{1}{3} = 0.3333.....$$

$$\frac{1}{7} = 0.1428.....$$

n, m are given, calculate the m -th digit after decimal point of $\frac{1}{n}$.

Input

First line contains two integers n, m ($1 \leq n \leq 10^5, 1 \leq m \leq 10^9$), separated by space.

Output

Output the m -th digit after decimal point of $\frac{1}{n}$.

Examples

standard input	standard output
3 5	3
233 23	5

Problem J. Sakuyalove and Latin Square

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

An " n -Latin Square" is an $n \times n$ square, each cell contains one integer in range $[1, n]$. All integers in a row or a column are different. Formally, an " n -Latin Square" can be seen as a 2-dimensions array a , we have the set equation $\{a_{i,1}, a_{i,2}, \dots, a_{i,n}\} = \{1, 2, \dots, n\}$ for $i = 1, 2, \dots, n$ and $\{a_{1,j}, a_{2,j}, \dots, a_{n,j}\} = \{1, 2, \dots, n\}$ for $j = 1, 2, \dots, n$.

1	3	6	2	5	4
6	2	3	4	1	5
3	4	1	5	6	2
2	1	5	6	4	3
5	6	4	3	2	1
4	5	2	1	3	6

An Example of 6-Latin Square

Sakuyalove wants to build an n -Latin Square. She finished writing the first m rows of the square. Now she asks you to construct the square on the basis of the first m rows. She can make sure the first m rows are correct. Please help her.

Input

The first line of the input contains two integers n, m ($1 \leq m < n \leq 100$) separated by space, which describes the size of the square, and the number of rows Sakuyalove has written.

Next m lines, the i -th line contains n integers $a_{i,1}, a_{i,2}, \dots, a_{i,n}$ ($1 \leq a_{i,j} \leq n$), which describes the integers Sakuyalove has written on the i -th rows.

It is guaranteed that the rows given by Sakuyalove are correct. Formally, we have the set equation $\{a_{i,1}, a_{i,2}, \dots, a_{i,n}\} = \{1, 2, \dots, n\}$ for $i = 1, 2, \dots, m$, and $a_{1,j}, a_{2,j}, \dots, a_{m,j}$ are different integers in range $[1, n]$ for $j = 1, 2, \dots, n$.

Output

The output should contain n lines, each line contains n integers separated by space, which described the " n -Latin Square" you built.

You should notice that the first m lines of the output should be same as the first m rows given by Sakuyalove.

It can be proved that it exists a way to build the n -Latin Square on the basis of the first m rows given by Sakuyalove.

If it has more than one solution, you can print any.

Example

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

Note

The sample input and output are like this:

1	2	5	4	3
3	5	4	1	2
4	3	1	2	5

The Sample Input

1	2	5	4	3
3	5	4	1	2
4	3	1	2	5
2	1	3	5	4
5	4	2	3	1

The Sample Output

Problem K. The Demon Jester

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

Playf has recently found joy in life in LOL: He falls in love with the Demon Jester.

At this time, Playf will fight against Malphite. They fight on a tree consisting of n vertices numbered from 1 to n . Malphite's initial position is vertex 1, and Playf can choose any vertex as his initial position. The length of each edge on the tree is 1, and both Playf and Malphite have a movement speed of 1 per second. (At any moment, they can choose to move in any direction along the edge or stay still, even if they are on an edge at that time.)

Before the game starts, Playf has placed m boxes in advance. The i_{th} box is placed on the vertex p_i , with trigger range a_i and attack power b_i , which means that if Malphite reaches any vertex that satisfies $dis(x, p_i) \leq a_i$ ($dis(u, v)$ represents the shortest distance from u to v), box i will be triggered and deal b_i damage to Malphite. Each box can only be triggered once, and the box will disappear after dealing damage.

Malphite has been carried away by anger, and will chase Playf along the shortest path by maximum speed all the time after the game starts. When Malphite's position coincides with Playf (which means that the distance between Malphite and playf is exactly 0, no matter they meet on the vertex or on the edge), Playf will be killed by angry Malphite, ending his sinful life. Malphite will no longer move after Playf dies.

In the game, Playf has **one** chance to use the skill "Fraud Magic" (He can also choose not to use this skill during the game). Using this skill does not take time, it can help Playf teleport to any vertex on the tree, and this skill can be used at any moment.

Since Malphite is so Sturdy that he will not die during the game.

Now Playf wants to know the maximum amount of damage he can deal to Malphite during the game.

Input

The first line contains two integers n, m ($2 \leq n \leq 200\,000, 0 \leq m \leq 200\,000$) — the number of vertices in the tree, and the number of boxes have been placed.

Each of the next $n - 1$ lines contain a pair of numbers u_i, v_i ($1 \leq u_i, v_i \leq n$) — an edge of the tree.

Each of the next m lines contain three integers p_i, a_i, b_i ($2 \leq p_i \leq n, 0 \leq a_i \leq n, 1 \leq b_i \leq 10^9$) — the i_{th} box's position, trigger range and attack power.

It is guaranteed that for each $1 \leq i \leq m$, $dis(1, p_i) > a_i$.

Output

Print one integer in a line — the maximum amount of damage Playf can deal to Malphite.

Examples

standard input	standard output
4 2 1 2 1 3 3 4 4 0 2 2 0 1	2
4 2 1 2 1 3 3 4 4 1 2 2 0 1	3
2 3 1 2 2 0 1000000000 2 0 1000000000 2 0 1000000000	3000000000

Problem L. Timetable Queries

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

There is a bus stop where exactly one bus arrives every minute.

Thus, a timetable is necessary for the passengers.

There are n minutes in one day, so the timetable is an array a containing n integers. let a_i be the i -th number in array a , denoting that a bus with its number a_i arrives exactly on the i -th minute.

There will be q queries, each query will ask you when is the y -th time does the bus with number x arrive in one day?

Input

The first line contains two integers, n ($1 \leq n \leq 10^5$) and q ($1 \leq q \leq 10^5$), denoting the length of the timetable and the number of queries.

The second line contains n integers, the i -th integer is a_i ($1 \leq a_i \leq 10^9$), denoting the bus with number a_i arrives on the i -th minute.

For the next q lines, each line contains two integers, x ($1 \leq x \leq 10^9$) and y ($1 \leq y \leq 10^5$), denoting there is an query, asking you when is the y -th time does the bus numbered x arrive.

Output

For each query, you should output one line containing exactly one integer, denoting your answer for this query. If the x -th bus arrives less than y times in one day according to the timetable, output -1 .

Examples

standard input	standard output
10 5 1 2 3 4 5 5 4 3 2 1 1 1 1 2 1 3 6 1 5 2	1 10 -1 -1 6
10 5 1 2 1 2 1 2 3 3 3 1 1 4 1 5 3 3 3 4 2 2	10 -1 9 -1 4