



Problems

A. Photo Day	1
B. Stones on a Tree	3
C. Rain in Hue	5
D. Escape the Maze	7
E. Snowbound Supplies	9
F. Zone	11
G. Pascal's Pyramid	13
H. Souvenirs at Dong Ba Market	15
I. Same Level	17
J. Dominant Champion	19
K. Drone	21
L. Quadrilateral Classification	23
M. Aegis	25
N. Lanterns on the Perfume River	27

CONTEST SYSTEM INFORMATION

- Online Judge URL: <https://icpc.vnoi.info>
- Judging System: VNOJ

ACCOUNT CREDENTIALS

- Your **Username** and **Password** will be distributed directly by the Invigilator prior to the start of the contest.
- Please log in immediately upon receiving your credential slip to verify your account and network connection.



Blank page



Problem A: Photo Day

Time limit: 1s; Memory limit: 256 MB

It is photo day at a primary school. The n students stand in a queue, where n is even; the i -th student from the front has height h_i and all heights are distinct.

Everyone in the queue must appear in exactly one photo. For a photo, the photographer calls any $2k$ people still in the queue ($k \geq 1$), not necessarily adjacent. In queue order, the first k of them form one row and the other k form another. She chooses which row stands in front; the other row stands right behind it. People inside a row may stand in any order, so to be sure that nobody is hidden, the shortest person in the back row must be taller than the tallest person in the front row.

In the figure below, photo (a) is possible: the shortest person at the back (128) is taller than the tallest in front (124). The rows in (b) cannot form a photo: whichever row stands at the back, someone may be hidden.

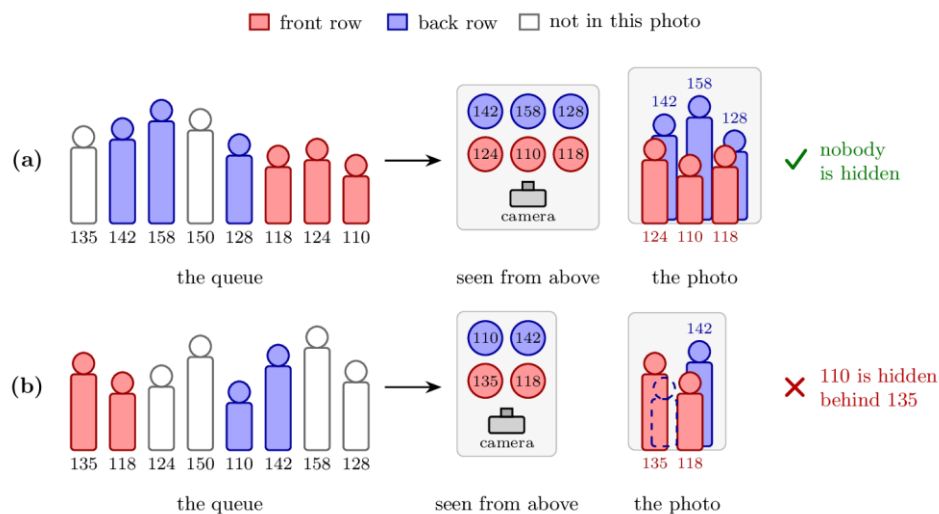


Figure 1: Valid photo (a) vs invalid photo (b).

Before the photos, the teacher may let two neighbours in the queue swap places, any number of times. A swap takes a seconds and a photo takes b seconds.

Find the minimum total time to photograph everybody.

Input

The first line contains three integers n , a , and b ($2 \leq n \leq 2 \times 10^5$, n is even, $1 \leq a, b \leq 10^6$) — the number of students, the time of one swap, and the time of one photo, in seconds.

The second line contains n distinct integers $h_1, h_2, \dots, h_n$ ($1 \leq h_i \leq 10^9$) — the heights of the students from the front of the queue to the end.

Output

Print one integer — the minimum total time in seconds.



Sample

Sample 1

Sample Input	Sample Output
4 1 10 120 130 110 140	11

Sample 2

Sample Input	Sample Output
4 20 10 120 130 110 140	20

Note / Explanation

In the first sample, the students with heights 130 and 110 swap places, and the queue becomes 120, 110, 130, 140. Then one photo takes everybody: 120, 110 in front and 130, 140 behind. The total time is $1 + 10 = 11$.

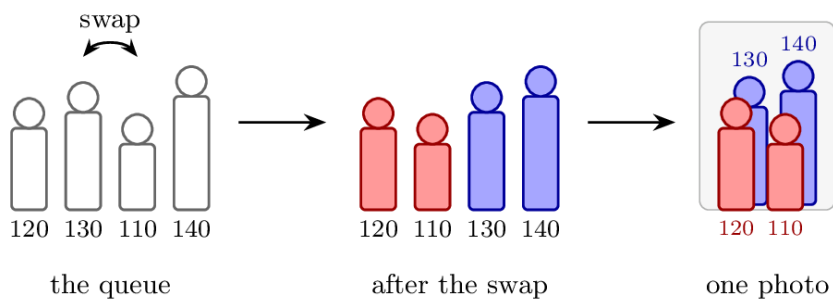


Figure 2: Process of swapping and taking one photo for Sample 1.

In the second sample, there are no swaps and two photos: first 120 in front of 140, then 110 in front of 130. The total time is $10 + 10 = 20$.

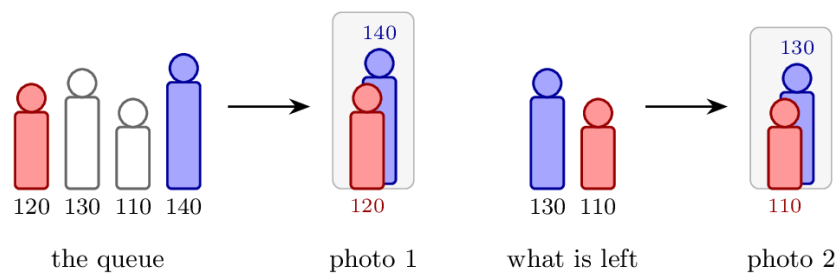


Figure 3: Process of taking two separate photos without swaps for Sample 2.



Problem B: Stones on a Tree

Time limit: 1.0s; Memory limit: 256 MB

You are given a tree with n vertices. You have to put some stones on every vertex: a_i stones on vertex i , where $0 \leq a_i \leq 10^9$. The sequence $(a_1, a_2, \dots, a_n)$ is called a placement.

For a fixed placement, Alice and Bob play the following game with one token:

- First, Alice chooses a starting vertex and puts the token on it. This is not a move.
- Then the players take turns making moves, Alice moves first.
- Let v be the vertex with the token. In one move, the player does exactly one of the following:
 - Take: remove one or more stones from vertex v , the token stays on v .
 - Jump: move the token to a vertex u adjacent to v such that the token has never been on u before (the starting vertex counts as visited), and u currently has exactly as many stones as v . No stones are removed during a jump.
- The player who cannot make a move loses.

A placement is good if Bob wins no matter which starting vertex Alice chooses, assuming both players play optimally.

Count the number of good placements. Since the answer may be large, print it modulo 998,244,353.

Input

- The first line contains a single integer n ($1 \leq n \leq 5000$) — the number of vertices.
- Each of the next $n - 1$ lines contains two integers u and v ($1 \leq u, v \leq n, u \neq v$) — an edge of the tree.
- It is guaranteed that the given edges form a tree.

Output

Print a single integer — the number of good placements modulo 998,244,353.

Sample

Sample 1

Sample Input	Sample Output
3 1 2 2 3	2

Sample 2

Sample Input	Sample Output
7 1 2 1 3	8



1 4	
2 5	
2 6	
3 7	

Sample 3

Sample Input	Sample Output
8 1 2 1 3 2 4 1 5 2 6 3 7 4 8	16

Note

In the first test, the tree is the path 1 - 2 - 3, and the good placements are (0, 1, 0) and (1, 0, 1). Let us check that (0, 1, 0) is good:

- Alice starts at vertex 1. Vertex 1 has no stones, so Alice cannot take. Vertex 2 has 1 stone, not 0, so Alice cannot jump. Alice loses immediately.
- Alice starts at vertex 3. This is the same as vertex 1.
- Alice starts at vertex 2. Alice cannot jump, because vertex 2 has 1 stone and its neighbours have 0. So Alice takes the stone, and now vertex 2 has 0 stones. Bob jumps to vertex 1, since both vertices have 0 stones. Alice cannot take (vertex 1 is empty) and cannot jump (vertex 2 is already visited), so Alice loses.

The placement (0, 0, 0) is not good: Alice starts at vertex 2 and immediately jumps to vertex 1, since both vertices have 0 stones. Bob cannot take and cannot jump back to vertex 2, so Bob loses.



Problem C: Rain in Hue

Time limit: 1s; Memory limit: 256 MB

The rainy season in Hue lasts for months, so the festival committee reads the gauge at the observatory every single day.

The observatory recorded the rainfall of n consecutive days. On day i it measured a_i millimetres.

The committee calls day i a postponed day when the rainfall of that day is at least m millimetres. A day that measured exactly m millimetres is therefore a postponed day as well.

Report how many postponed days there were, and the total rainfall measured on those days alone.

Input

The first line contains two integers n and m ($1 \leq n \leq 1000$, $0 \leq m \leq 1000$) — the number of days and the threshold.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 1000$) — the rainfall of each day in order.

Output

Print two integers separated by one space: the number of postponed days, and the total rainfall of the postponed days. If there is no postponed day, print 0 0.

Sample

Sample Input	Sample Output
5 30 12 30 7 45 29	2 75
3 100 99 1 0	0 0

Explanation of the Samples

- In the first sample, day 2 measured 30 millimetres and day 4 measured 45 millimetres. Both are at least 30, so there are two postponed days and their total rainfall is $30 + 45 = 75$. Day 2 measured exactly 30 millimetres and still counts.
- In the second sample, no day reached 100 millimetres, so there is no postponed day and both numbers are zero.



Blank page



Problem D: Escape the Maze

Time limit: 1s; Memory limit: 256 MB

You have been kidnapped and abandoned inside a mysterious maze. While desperately searching for a way out, you accidentally find a small piece of paper containing some strange writings. Since part of the paper has been torn away, you cannot determine the original string of characters. However, the paper still contains four integers a_1 , a_2 , a_3 , and a_4 . You realize that they represent the number of occurrences of the four characters L, R, U, and D in a mysterious string:

- a_1 — the number of occurrences of the character L.
- a_2 — the number of occurrences of the character R.
- a_3 — the number of occurrences of the character U.
- a_4 — the number of occurrences of the character D.

There is also a very large integer k ($1 \leq k \leq 10^9$) written on the paper.

You suspect that this must be a clue left behind by someone who was trapped in the maze before you. After carefully studying the strange message, you finally manage to decode it.

You realize that a path through the maze is represented by a string consisting only of the four characters L, R, U, and D (L — move left; R — move right; U — move up; D — move down). The escape route must use exactly a_1 steps L, a_2 steps R, a_3 steps U, and a_4 steps D. The clue does not directly tell you the exact order of the moves. Instead, the escape route is the k -th string in lexicographical order among all distinct strings that can be formed by rearranging these moves. For lexicographical ordering, the characters are ordered as: **L < R < U < D**.

Your task is to determine the escape route. If there are fewer than k distinct possible paths, output -1.

Input

The first line contains four integers a_1, a_2, a_3, a_4 ($0 \leq a_1, a_2, a_3, a_4 \leq 10^4$, $1 \leq a_1 + a_2 + a_3 + a_4 \leq 10^4$) and an integer k ($1 \leq k \leq 10^9$).

Output

Print the escape route — the k -th distinct string in lexicographical order formed using exactly a_1 characters L, a_2 characters R, a_3 characters U, and a_4 characters D.

If there are fewer than k distinct strings that can be formed, print -1.

Sample

Sample Input	Sample Output
1 1 1 1 3	LURD
1 2 1 1 50	DLRUR
1 2 1 1 61	-1
1 0 1 1 5	DLU



Blank page



Problem E: Snowbound Supplies

Time limit: 1s; Memory limit: 256 MB

A winter storm has covered the roads between a group of mountain villages with snow. A relief crew needs to collect supplies stored in the villages. To travel between villages, the crew uses a snowplough to clear the roads.

There are n villages, numbered from 1 to n , connected by $n - 1$ two-way roads. There is exactly one path between any two villages. Village i has a_i crates of supplies in storage, and each road has a known snow depth.

A snowplough with clearing capacity h can travel along a road if and only if its snow depth is at most h . The crew may visit any number of villages and use each road any number of times. It can collect the supplies from each village it visits, including its starting village, but each crate can be collected only once.

Your task is to answer q queries. Each query specifies a starting village v and a required number of crates k . Determine the smallest nonnegative clearing capacity that allows the crew to collect at least k crates, or determine that this is impossible.

The queries are independent: collecting supplies in one query does not affect the supplies available in any other query.

Input

The first line contains two integers n and q ($1 \leq n, q \leq 200,000$) — the number of villages and the number of queries.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^9$), where a_i is the number of crates stored in village i .

Each of the next $n - 1$ lines contains three integers u, v , and w ($1 \leq u, v \leq n, u \neq v, 1 \leq w \leq 10^9$), describing a two-way road between villages u and v with snow depth w .

It is guaranteed that the roads connect all villages and that there is exactly one path between any two villages.

Each of the next q lines contains two integers v and k ($1 \leq v \leq n, 1 \leq k \leq 10^{15}$), describing a query.

Output

For each query, output one line containing the minimum clearing capacity needed to collect at least k crates when starting from village v . If this is impossible, output -1.

Sample

Sample 1

Sample Input	Sample Output
6 7	0
2 0 4 1 3 5	3
1 2 3	6
2 3 7	7
2 4 3	7
4 5 6	3
5 6 7	-1
1 2	



1 3 1 6 1 7 3 5 2 1 4 16	
---	--

Sample 2

Sample Input	Sample Output
4 3 0 2 2 2 1 2 3 1 3 3 1 4 3 1 5 2 2 2 7	3 0 -1

Note / Explanation

First sample:

- In the first query, village 1 already has two crates, so the answer is 0.
- In the second query, starting from village 1, the crew can reach villages 1, 2, and 4 with a clearing capacity of 3. These villages store three crates in total.
- In the last query, sixteen crates are required, but only fifteen are available in total, so the answer is -1.

Second sample:

- In the first query, the crew must visit all three villages with supplies, returning through village 1 between visits. A clearing capacity of 3 is sufficient.



Problem F: Zone

Time limit: 1s; Memory limit: 256 MB

A research team is planning to build a large observation zone on a coordinate plane. The zone must be a square whose sides are parallel to the coordinate axes.

Several objects have already been placed on the map:

- There are n observation points, each represented by a single point.
- There are m buildings, each represented by a rectangle whose sides are parallel to the coordinate axes.
- There are q protected areas, each represented by a circle.

The observation zone must completely contain every object on the map. In other words, every point, every rectangle, and every circle must lie entirely inside the square.

Your task is to determine the minimum possible area of the observation zone.

Input

The first line of the input contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases. t test cases follow, each presented as below:

- The first line contains three integers n , m , and q ($1 \leq n, m, q \leq 10^5$), representing the number of observation points, rectangles, and circles.
- The next n lines each contain two integers x, y ($0 \leq x, y \leq 10^8$), describing an observation point at coordinates (x, y) .
- The next m lines each contain four integers x_1, y_1, x_2, y_2 ($0 \leq x_1, y_1, x_2, y_2 \leq 10^8$, $x_1 < x_2$, $y_1 < y_2$), describing a rectangular building whose bottom-left corner is (x_1, y_1) and top-right corner is (x_2, y_2) .
- The next q lines each contain three integers x, y, r ($0 \leq x, y, r \leq 10^8$), describing a circular protected area with center (x, y) and radius r .

It is guaranteed that the sum of n , the sum of m , and the sum of q over all test cases do not exceed 10^5 .

Output

For each test case, print a single integer — the minimum possible area of a square, with sides parallel to the coordinate axes, that completely contains all the given objects.

Sample

Sample Input	Sample Output
<pre> 3 2 2 2 8 4 5 7 1 1 5 3 3 0 8 5 3 5 2 7 2 1 1 1 1 5 5 4 4 6 6 5 5 1 1 1 1 3 6 1 2 5 6 3 4 1 </pre>	<pre> 49 4 16 </pre>

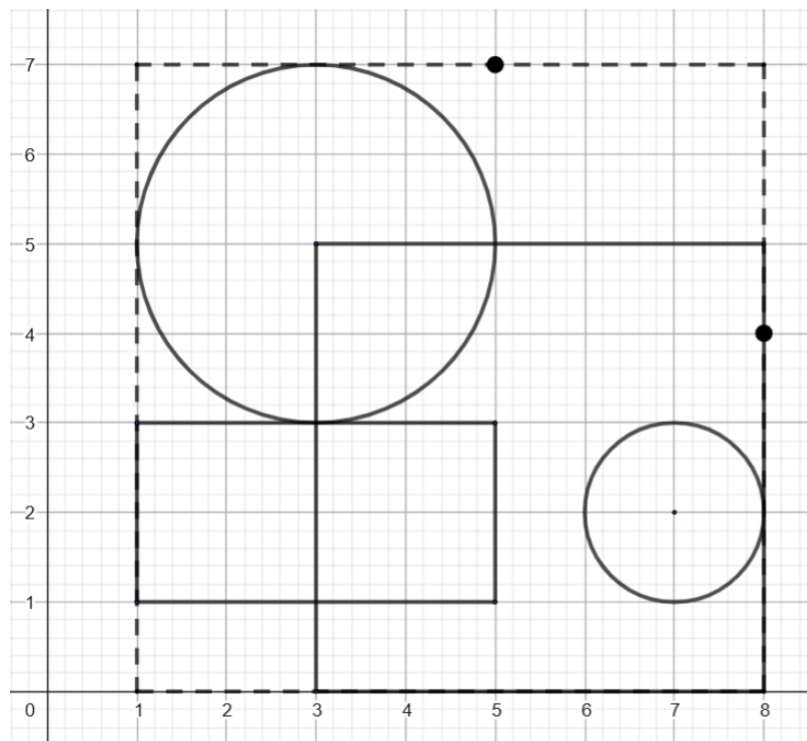


Figure 1: Illustration for the first test case

Sample Explanation

Figure 1 illustrates the first test case. The observation zone with the minimum possible area is indicated by dashed lines.



Problem G: Pascal's Pyramid

Time limit: 1s; Memory limit: 256 MB

A construction consists of n floors and uses n types of cubes, numbered from 1 to n .

A type- k cube has side length k , so the volume of each type- k cube is k^3 .

The number of cubes required to build each floor follows Pascal's Triangle from top to bottom:

- The first floor consists of 1 type-1 cube.
- The second floor consists of 2 cubes: 1 type-1 cube and 1 type-2 cube.
- The third floor consists of 4 cubes: 1 type-1 cube, 2 type-2 cubes, and 1 type-3 cube.
- The fourth floor consists of 8 cubes: 1 type-1 cube, 3 type-2 cubes, 3 type-3 cubes, and 1 type-4 cube.
- The construction continues following the same rule.

The numbers of cubes on the first four floors are arranged as follows:

```
1
1 1
1 2 1
1 3 3 1
```

Here, the k -th position on the n -th floor corresponds to type- k cubes.

Determine the total volume of all cubes required to build the n -th floor.

Input

A single line containing a positive integer n ($1 \leq n \leq 10^{18}$).

Output

Print the total volume of all cubes on the n -th floor, modulo $10^9 + 7$.

Sample

Sample 1

Sample Input	Sample Output
3	44

Sample 2

Sample Input	Sample Output
4	170

Note / Explanation

In the first sample, the third floor consists of: 1 type-1 cube, 2 type-2 cubes, and 1 type-3 cube.

Therefore, the total volume is:

$$1^3 + 2 \times 2^3 + 3^3 = 1 + 16 + 27 = 44.$$

In the second sample, the fourth floor consists of: 1 type-1 cube, 3 type-2 cubes, 3 type-3 cubes, and 1 type-4 cube.

Therefore, the total volume is:

$$1^3 + 3 \times 2^3 + 3 \times 3^3 + 4^3 = 1 + 24 + 81 + 64 = 170.$$



Blank page



Problem H: Souvenirs at Dong Ba Market

Time limit: 1s; Memory limit: 256 MB

A stall at Dong Ba Market lays out n souvenirs in a row. Souvenir i carries a name s_i and a price p_i . Two souvenirs may carry the same name and still have different prices, because the stall keeper prices every piece by hand.

A tourist wants to bring home one souvenir of every distinct name on the stall, and for each name always picks a souvenir of that name with the lowest price.

Report how many distinct names the stall has, and the total amount the tourist pays.

Input

The first line contains one integer n ($1 \leq n \leq 500,000$) — the number of souvenirs.

Each of the next n lines contains a name s_i and an integer p_i ($1 \leq p_i \leq 10^{18}$) separated by one space. Every name is a non-empty string of at most 20 lowercase English letters.

Output

Print two integers separated by one space: the number of distinct names on the stall, and the total amount the tourist pays.

Sample

Sample Input	Sample Output
4 non 50 tranh 120 non 35 tranh 120	2 155
3 nem 9 nem 9 nem 9	1 9

Explanation of the Samples

- In the first sample the stall has two distinct names. The cheapest souvenir named non costs 35 and the cheapest one named tranh costs 120, so the tourist pays $35 + 120 = 155$. Note that the cheapest non is not the first one in the row.
- In the second sample every souvenir carries the same name, so the tourist buys exactly one of them and pays 9.



Blank page



Problem I: Same Level

Time limit: 1s; Memory limit: 256 MB

A guild in a role-playing game consists of N characters, numbered from 1 to N . The i -th character is currently at level a_i .

There are $N - 1$ direct connections between characters, each connecting two characters, such that between any two characters there is exactly one path (the characters and connections form a tree).

Two directly connected characters can complete a pair quest together. Each pair quest increases the level of both participating characters by exactly 1. A character's level never decreases, and pair quests can only be performed between directly connected characters.

The guild leader wants all characters in the guild to reach the same level. Determine the minimum number of pair quests required, or report if it is impossible.

Constraints

- $1 \leq T \leq 10^4$.
- $1 \leq N \leq 2 \times 10^5$; the sum of N over all test cases does not exceed 2×10^5 .
- $0 \leq a_i \leq 10^8$.
- $1 \leq u, v \leq N$, $u \neq v$; the connections form a tree.

Input

The first line contains an integer T — the number of test cases. T test cases follow, each presented as below:

- The first line contains an integer N .
- The second line contains N integers $a_1, a_2, \dots, a_N$.
- The next $N - 1$ lines, each contains two integers u and v indicating a connection between character u and character v .

Output

For each test case, print on a single line the minimum number of pair quests required, or -1 if it is impossible to make all characters in the guild reach the same level.

Sample

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



1 2	
4	
0 3 3 2	
1 2	
1 3	
1 4	
4	
2 0 1 3	
1 2	
2 3	
3 4	

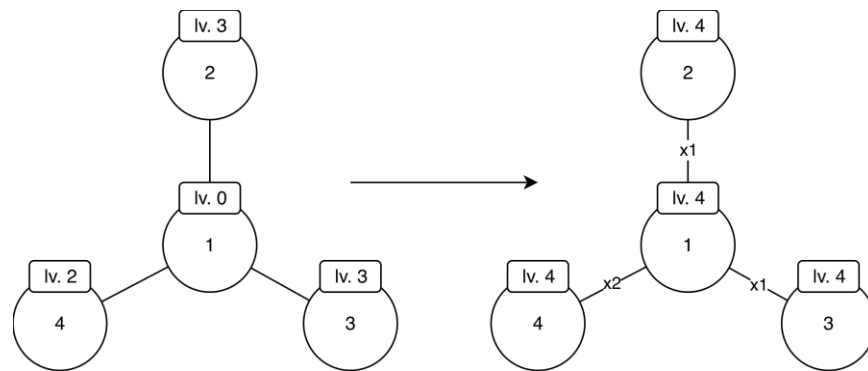


Figure 1: Illustration for test case 3.

Sample Explanation

- Test case 1: Character 1 and character 2 complete one quest, character 2 and character 3 complete one quest; all three characters reach level 2.
- Test case 2: The two characters always level up together, so the level difference of 2 can never be eliminated.
- Test case 3: Character 1 completes one quest with character 2, one quest with character 3, and two quests with character 4; all four characters reach level 4.
- Test case 4: Pair 1-2 completes one quest, pair 2-3 completes two quests; all four characters reach level 3.

Figure 1 illustrates test case 3. The number on each connection represents the number of pair quests completed by the two characters at its endpoints.



Problem J: Dominant Champion

Time limit: 1s; Memory limit: 256 MB

The English Premier League has 20 clubs. A win is worth 3 points, a draw is worth 1 point, and a loss is worth 0 points.

#	Club	Pts	Last 5
1	Arsenal	85	✓✓✓✓✓
2	Man City	78	—✓✓✓—
3	Man Utd	71	✓✓✓✓✓
4	Aston Villa	65	×××—✓
5	Liverpool	60	✓×—×—
6	Bournemouth	57	—✓✓—
7	Sunderland	54	×—×✓✓
8	Brighton	53	✓××✓×
9	Brentford	53	×××—
10	Chelsea	52	××—✓×
11	Fulham	52	✓××—✓
12	Newcastle	49	××—✓×
13	Everton	49	×—×××
14	Leeds	47	—✓—✓×
15	Palace	45	×—×—×
16	Nottm Forest	44	✓✓—×—
17	Spurs	41	✓✓—×✓
18	West Ham	39	✓×××✓
19	Burnley	22	××—×—
20	Wolves	20	×—×—

Figure 1: Example league table showing only points and recent form.

For this problem, the standings are taken immediately after round 35. Therefore, every club has exactly 3 matches left to play before the end of the season.

The club with the highest current score is called the leader. The input is guaranteed to have exactly one leader.

The leader is called a dominant champion if, regardless of what happens in the remaining three matches, it is guaranteed to finish with strictly more points than every other club.

Only points are considered in this problem. Goal difference and all other tie-breaking rules are ignored. A tie for first place is not considered a dominant championship.

For the purpose of this problem, each other club may gain at most 9 additional points from its three remaining matches. The detailed remaining fixture list is not given and does not need to be considered.

Determine whether the current leader is already guaranteed to be the dominant champion.



Input

The input contains exactly 20 lines. Each line contains a club name and its current number of points after round 35.

Club names are written as a single token. Spaces in a club name are replaced by underscores.

The 20 club names are: Arsenal, Man_City, Man_Utd, Aston_Villa, Liverpool, Bournemouth, Sunderland, Brighton, Brentford, Chelsea, Fulham, Newcastle, Everton, Leeds, Palace, Nottm_Forest, Spurs, West_Ham, Burnley, and Wolves.

For every club, $0 \leq \text{points} \leq 105$.

All 20 club names appear exactly once.

There is exactly one club with the maximum current number of points.

Output

Print YES if the current leader is guaranteed to be the dominant champion. Otherwise, print NO.

Sample

Sample 1

Sample Input	Sample Output
Arsenal 82 Man_City 69 Man_Utd 66 Aston_Villa 61 Liverpool 60 Bournemouth 55 Sunderland 52 Brighton 50 Brentford 49 Chelsea 48 Fulham 47 Newcastle 46 Everton 44 Leeds 43 Palace 41 Nottm_Forest 40 Spurs 39 West_Ham 35 Burnley 21 Wolves 19	YES

Explanation:

Arsenal has 82 points. Every other club can gain at most 9 more points. Man_City, the closest challenger, can reach at most $69 + 9 = 78$ points. Therefore Arsenal is guaranteed to finish strictly ahead.

Sample 2

Sample Input	Sample Output
Arsenal 74 Man_City 67 Man_Utd 63 Aston_Villa 59 Liverpool 57 Bournemouth 53 Sunderland 51 Brighton 49 Brentford 48 Chelsea 47 Fulham 45 Newcastle 44 Everton 42 Leeds 41 Palace 40 Nottm_Forest 38 Spurs 37 West_Ham 34 Burnley 20 Wolves 18	NO

Explanation:

Man_City can still reach $67 + 9 = 76$ points, which is greater than Arsenal's current 74 points. Therefore the leader is not yet guaranteed to be the dominant champion.



Problem K: Drone

Time limit: 1s; Memory limit: 256 MB

In the futuristic city of Da Nang, the aerial transit network is modeled as a directed acyclic graph (DAG) consisting of N transfer stations and M one-way flight routes. To ensure security, the operations center must deploy patrol drone teams. Each drone departs from any station with an in-degree of 0, traverses valid flight routes, and finishes its journey at any station with an out-degree of 0.

The system enforces strict safety regulations:

- For each flight route e , there must be at least L_e and at most R_e drones flying along it.
- For each transfer station v , the total number of drones visiting the station (i.e., the sum of drones arriving at and departing from the station) must lie within the range $[A_v, B_v]$.
- Constructing and operating drones is extremely expensive.

Your task is to calculate the minimum number of drones required to form a flight route system that satisfies all of the above regulations.

Input

- The first line contains two positive integers N and M ($1 \leq N \leq 1000$, $1 \leq M \leq 5000$). Stations are numbered from 1 to N .
- The next N lines, where the v -th line contains two integers A_v, B_v ($0 \leq A_v \leq B_v \leq 10^5$) describing the constraints of station v .
- The next M lines, each containing four integers u, v, L, R ($1 \leq u, v \leq N$; $0 \leq L \leq R \leq 10^5$) describing a one-way flight route from u to v with the constraint that the number of traversing drones must lie within the range $[L, R]$. The graph is guaranteed to be acyclic.

Output

- A single line containing an integer representing the minimum number of drones required.
- If no valid configuration exists, output -1.

Sample

Sample Input	Sample Output
<pre> 4 4 1 5 1 2 1 2 0 5 1 2 1 2 1 3 0 5 2 4 1 2 3 4 0 5 </pre>	<pre> 2 </pre>



Explanation

The graph has 4 vertices. Edges $1 \rightarrow 2$ and $2 \rightarrow 4$ require at least 1 drone, while vertex 3 requires 1 to 2 drones visiting it, so at least one drone must pass through vertex 3 (via edge $1 \rightarrow 3$ then $3 \rightarrow 4$).

We deploy 2 drones: one takes the route $1 \rightarrow 2 \rightarrow 4$ and the other takes the route $1 \rightarrow 3 \rightarrow 4$. Then, edge $1 \rightarrow 2$ has 1 drone and edge $2 \rightarrow 4$ has 1 drone (satisfying $[1, 2]$); edges $1 \rightarrow 3$ and $3 \rightarrow 4$ have 1 drone (satisfying $[0, 5]$); vertex 2 and vertex 3 each have 1 drone visiting (satisfying $[1, 2]$); vertex 1 has 2 drones, and vertex 4 has 2 drones.

Because at least 1 drone must traverse edge $1 \rightarrow 2$ and at least 1 drone must visit vertex 3 (two separate paths starting from vertex 1), fewer than 2 drones cannot be used. The minimum total number of drones is 2.



Problem L: Quadrilateral Classification

Time limit: 1s; Memory limit: 256 MB

Tuan has just learned about different types of quadrilaterals and wants to test his ability to recognize them.

Given N points on the Oxy coordinate plane. The points are numbered from 1 to N . Two or more initial points may share the same coordinates.

Tuan chooses a set of 4 points with distinct indices (i, j, k, l) such that $1 \leq i < j < k < l \leq N$. A set of 4 points is considered a valid quadrilateral if it simultaneously satisfies two conditions:

- The four selected points have pairwise distinct coordinates.
- No three selected points are collinear.

Each set of 4 distinct indices is counted as a separate choice, even if some points share the same coordinates.

For each valid set of 4 points, determine which types of quadrilaterals they form:

- **Square (S):** Has 4 equal sides and 1 right angle.
- **Rectangle (R):** Has 4 right angles.
- **Rhombus (H):** Has 4 equal sides.
- **Trapezoid (T):** Has at least 1 pair of parallel opposite sides.
- **Parallelogram (P):** Has 2 pairs of parallel opposite sides.

Input

The first line contains an integer N ($4 \leq N \leq 1000$).

The next N lines, the i -th line contains two integers x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$).

Output

Print 5 integers on a single line separated by spaces, representing the counts of S, R, H, T, and P, respectively.

Sample

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



Blank page



Problem M: Aegis

Time limit: 1.0s; Memory limit: 256 MB

In the Aegis space defense perimeter, a system of N outpost stations is arranged in a closed ring, numbered from 1 to N in clockwise order.

Between any two adjacent stations on the ring, there is a bidirectional power cable:

- The i -th cable connects station i and station $i + 1$ (for $1 \leq i \leq N - 1$); disconnecting it saves an amount of energy of B_i .
- The N -th cable connects station N and station 1; disconnecting it saves an amount of energy of B_N .

The military high command establishes M emergency strategic links. The j -th link requires that stations u_j and v_j ($u_j \neq v_j$) must maintain communication connectivity with each other along the remaining active cable network.

To assist coordination efforts, the command proposes Q independent maintenance plans. In the k -th plan, the command designates a strategic sector consisting of cables with consecutive indices from L_k to R_k ($1 \leq L_k \leq R_k \leq N$):

- All cables outside the sector $[L_k, R_k]$ must remain operational.
- Within the sector $[L_k, R_k]$, the command wants to disconnect a subset of cables such that the total energy saved is maximized, while ensuring that all M strategic links across the entire perimeter remain uninterrupted.

Your task is to determine the maximum total energy saving achievable in the sector for each maintenance plan.

Input

- The first line contains three integers N , M , and Q ($3 \leq N \leq 2 \times 10^5$, $1 \leq M \leq 2 \times 10^5$, $1 \leq Q \leq 2 \times 10^5$) — the number of outpost stations, the number of strategic links, and the number of maintenance plans.
- The second line contains N positive integers $B_1, B_2, \dots, B_N$ ($1 \leq B_i \leq 10^9$) — the energy saved when disconnecting each corresponding cable.
- The next M lines, the j -th line contains two integers u_j and v_j ($1 \leq u_j, v_j \leq N$, $u_j \neq v_j$) — the two stations of the j -th strategic link.
- The next Q lines, the k -th line contains two integers L_k and R_k ($1 \leq L_k \leq R_k \leq N$) — the sector range of the k -th maintenance plan.

Output

Print Q lines, each containing a single integer representing the maximum total energy saved for the corresponding plan.



Sample

Sample Input	Sample Output
5 2 2 10 20 15 30 40 1 3 2 4 1 5 2 4	70 30

Explanation

The perimeter has 5 cables with energy savings respectively: $B = [10, 20, 15, 30, 40]$. There are 2 strategic links: (1, 3) and (2, 4).

- Plan 1 ($L = 1, R = 5$): The considered sector is the entire ring $[1, 5]$. We can simultaneously disconnect the two cables e_4 and e_5 while both pairs (1, 3) and (2, 4) still maintain communication. The total energy saved is $B_4 + B_5 = 30 + 40 = 70$.
- Plan 2 ($L = 2, R = 4$): The considered sector consists of cables $\{e_2, e_3, e_4\}$. The cables outside the sector (e_1, e_5) must remain operational. In this sector, we cannot disconnect any two cables simultaneously because it would disrupt at least one strategic link. Therefore, we can only disconnect the single most profitable cable among the three: choosing e_4 yields a saving of $B_4 = 30$.



Problem N: Lanterns on the Perfume River

Time limit: 1s; Memory limit: 256 MB

During the Hue Festival, a row of n slots is prepared along the bank of the Perfume River. The slots are numbered 1 through n from left to right. Each slot either holds one lantern or is empty, described by a string s of length n : the i -th character of s is 1 if slot i holds a lantern, and 0 if slot i is empty.

The current pushes a lantern by exactly two slots at a time. One move takes a lantern from slot i to slot j , where $j = i - 2$ or $j = i + 2$. The move is allowed only when slot j belongs to the row, that is $1 \leq j \leq n$, and when slot j and the slot strictly between i and j are both empty.

All lanterns look the same, so an arrangement is determined only by which slots are occupied. Any number of moves may be performed, in any order.

The organisers inspect q parts of the row. For a part described by two integers l and r , consider the slots $l, l + 1, \dots, r$ together with the lanterns that those slots hold at the beginning, and treat them as a row of their own: during the inspection of this part, lanterns may move only between slots l and r , and the rest of the river is ignored.

For each part, determine how many different arrangements of that part can be obtained. The arrangement that the part starts with is counted as well. Since the answer may be large, print it modulo 998,244,353.

Input

The first line contains two integers n and q ($1 \leq n \leq 2 \times 10^5$, $1 \leq q \leq 2 \times 10^5$) — the number of slots and the number of parts to inspect.

The second line contains the string s of length n , consisting of the characters 0 and 1.

Each of the next q lines contains two integers l and r ($1 \leq l \leq r \leq n$), describing one part of the row.

Output

Print q lines. The i -th of them must contain the number of different arrangements that can be obtained in the i -th part of the row, taken modulo 998,244,353.

Sample

Sample Input	Sample Output
7 6 1001010 1 7 2 7 2 4 4 6 3 3 2 5	4 3 2 1 1 2



Explanation of the Sample

- In the second part, the slots 2 through 7 start as 001010, and exactly three arrangements can be obtained: 001010, 100010 and 101000.
- In the third part, the slots 2 through 4 start as 001, and the lantern may slide two slots to the left, giving 001 and 100. That lantern sits in slot 4 of the river but in the third slot of this part, and it is the position inside the part that decides where it can go.
- In the fourth part, the slots 4 through 6 start as 101 and no move is possible, because each lantern is blocked by the other.
- In the fifth part, slot 3 holds no lantern, so the empty arrangement is the only one.