Croatian Olympiad in Informatics

April 27th 2025

Tasks

Task	Time Limit	Memory Limit	Score
Automatizacija	6 seconds	512 MiB	100
Bolivija	2 seconds	512 MiB	100
Korupcija	1 second	512 MiB	100
Lirili Larila	2 seconds	512 MiB	100
Total			400

Task Automatizacija

The lifelong ambition of successful entrepreneur Elena Mošus is to replace all human labor with artificial intelligence. To accelerate this process, she decided to get involved in Croatian legislation. Her initiative reached the president, who appointed her as the head of the newly formed Ministry of Automation of Logical Notions and Analytical Reasoning (abbreviated as MALNAR). Their first task is to automate the following game.

The game is played between two players. Each player receives a set of K distinct numbers between 1 and N. Players have access only to the numbers from their own set, and the goal of the game is to determine the size of the intersection of the two sets. Players cannot communicate directly — they may only communicate through a shared board by placing tokens on it. The rules of the game are as follows:

- The board consists of N empty fields where tokens can be placed.
- Players take turns placing tokens on any available field. Once a token is placed on a field, that field becomes occupied and cannot be used again.
- The first player places blue tokens, and the second player places red tokens.
- Both players have full visibility of the entire board at all times.
- On their turn, instead of placing a token, a player may choose to end the game by declaring the size of the intersection of the two sets. If there are no free fields remaining, the player must end the game.

During the game, players may communicate only through their moves on the board; however, before the game begins, they are allowed to agree on a strategy.

MALNAR has decided that the players in this game should be replaced by an automated artificial intelligence system using an infallible strategy, capable of making a move immediately after reading the current board state. Help MALNAR by designing a strategy for both players that guarantees that at least one player, at some point, ends the game and correctly declares the size of the intersection.

To enable the automated system to make moves quickly, the strategy must specify, for every possible board state, which move to make. This means that the system will not have access to the sequence of previous moves leading to the current state — it must decide solely based on the present board configuration.

Input

The first line contains a positive integer P ($P = 1$ or $P = 2$) indicating whether you are controlling the first or the second player.

The second line contains two positive integers N and K as described above.

The third line contains K distinct positive integers between 1 and N, representing the player's set.

The fourth line contains a positive integer T — the number of board states for which a move must be determined. In the test data, T will always equal the total number of possible board states, meaning your strategy must specify a move for every possible state. A board state is considered *possible* if it can arise during gameplay. However, in sample cases, T may be smaller.

Each of the next T lines describes one possible board state. A board state is given as a sequence of N characters "`P`", "`C`", or "`.`", where "`P`" denotes a blue token, "`C`" denotes a red token, and "`.`" denotes an empty field.

Output

For each of the T given board states, output one line of the form "`+` m" or "`!` m", for some integer m.

An output of the form "`+` m" represents placing a token on the m-th field of the board. For the output to be valid, it must satisfy $1 \le m \le N$, and the chosen field must be unoccupied.

An output of the form "`! m`" represents declaring that the size of the intersection of the two sets is m and ending the game. For the output to be valid, it must satisfy $0 \le m \le N$.

Scoring

Your solution will be evaluated in two stages. First, it will be tested with $P = 1$, and then with $P = 2$. The values of N and K will be identical in both tests. Assuming your program produces valid outputs for both stages, a simulator developed by the organizers will simulate the game according to your printed strategy. If the simulation concludes with the correct intersection size being declared, your solution will be considered correct. The total execution time will be the sum of the times for both stages.

In all subtasks, it holds that $2 \le N \le 16$ and $1 \le K \le N$.

Subtask	Points	Constraints
1	11	The sets will consist of K consecutive numbers.
2	7	N is even and all numbers in the sets are between 1 and $\frac{N}{2}$.
3	16	$N \le 4$
4	13	$N = 14$ and $K = 2$
5	12	All numbers in the sets are between 1 and $N - 1$.
6	41	No additional constraints.

Sample Cases

input

```
1
4 2
2 3
3
....
P.C.
PCCP
```

output

```
+ 1
+ 4
! 1
```

input

```
2
4 2
1 3
2
P...
P.CP
```

output

```
+ 3
+ 2
```

Explanation of the Sample Cases:

This represents only one possible strategy. The corresponding course of the game is given below.

Board State	Move	Note
`....`	`+ 1`	The first player places a token on the first field.
`P...`	`+ 3`	The second player places a token on the third field.
`P.C.`	`+ 4`	The first player places a token on the fourth field.
`P.CP`	`+ 2`	The second player places a token on the second field.
`PCCP`	`! 1`	The first player ends the game and declares the intersection size to be 1. Correct!

Task Bolivija

Bolivia, a beautiful South American country rich in culture and history, is filled with natural wonders, including parts of the Amazon rainforest and the Andes mountain range. More importantly for our contestants, it will be the host of the next International Olympiad in Informatics!

As part of promoting the competition, the organizers have been tasked with photographing the mountain range and creating an album of the most breathtaking images. The mountain range is represented as an array v of N non-negative integers, where v_i denotes the height of the i-th mountain. It is guaranteed that N is odd and that the tallest mountain — located at position $\frac{N+1}{2}$ — is the extinct volcano Nevado Sajama.

The organizers have very specific conditions for the photographs. First, they choose two non-negative integers A and B such that $A < B$ and B is less than or equal to the height of Nevado Sajama. They then adjust the camera so that the width of the photograph captures all N mountains, but only the vertical range between A and B. Additionally, the organizers are satisfied with a photograph only if it is symmetric with respect to the vertical axis passing through the central mountain.

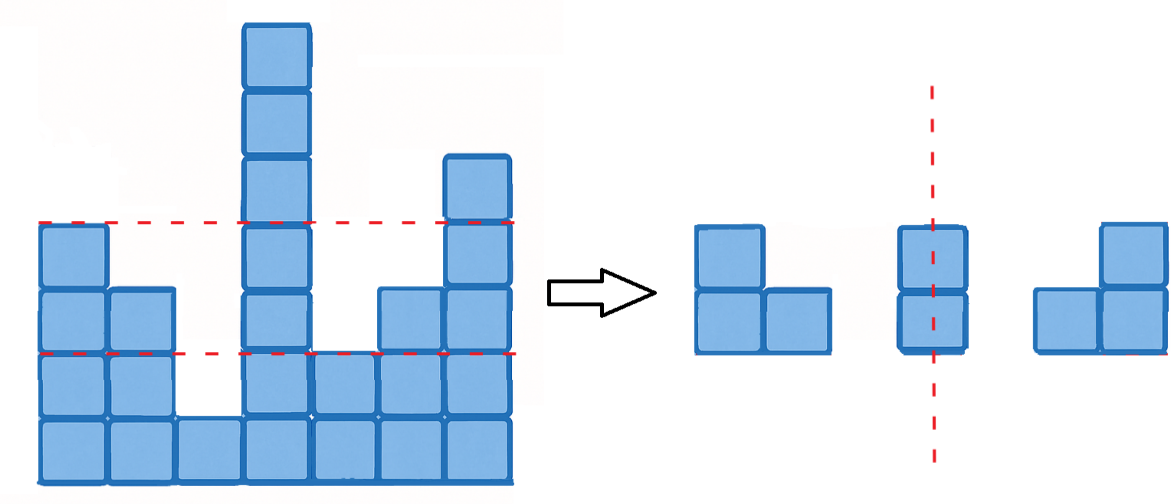

Illustration: Example of a valid photo selection corresponding to the second sample case

The organizers now want to know how many different photographs they can take — that is, how many pairs (A, B) satisfy the given conditions. While thinking too long about the answer, intense tectonic activity caused the heights of some mountains to change. A total of Q height changes occur, and your task is to help the organizers determine the number of valid photographs after each change. Importantly, none of the changes affects the height of the central mountain, and it always remains the tallest mountain.

Input

The first line contains two positive integers N and Q, representing the number of mountains and the number of height changes.

The second line contains an array v of N non-negative integers, the heights of the mountains in order. It is guaranteed that N is odd and that the central mountain is the tallest.

Each of the next Q lines contains two non-negative integers x_i and h_i $(1 \leq x_i \leq N)$, indicating that the height of the mountain at position x_i changes to h_i. It is guaranteed that $x_i \neq \frac{N+1}{2}$ and that the new height is less than or equal to the height of the central mountain.

Output

Print $Q+1$ lines. In the i-th line, print the number of valid photographs after the first $i-1$ height changes.

Scoring

In all subtasks, it holds that $3 \le N \le 200\,000$ and $0 \le Q \le 200\,000$.

For all $i = 1, \ldots, N$, it holds that $v_i \le 654\,200$ (the height of Nevado Sajama in centimeters).

Subtask	Points	Constraints
1	9	$Q = 0$, $N \le 300$, and $v_i \le 300$ for all $i = 1, \ldots, N$
2	23	$Q = 0$
3	31	Each change modifies a mountain's height by at most 1.
4	37	No additional constraints.

Sample Cases

input

```
5 5
1 5 8 7 3
1 8
4 1
2 0
4 0
5 8
```

output

```
5
6
1
3
6
36
```

input

```
7 0
4 3 1 7 2 3 5
```

output

```
7
```

input

```
7 10
1 6 7 10 5 4 3
2 7
2 8
2 9
2 9
2 10
6 5
6 6
6 7
6 8
6 9
```

output

```
8
8
5
3
3
2
4
4
4
5
7
```

Explanation of the Second Sample Case:

The valid choices for (A, B) are: $(0,1), (2,3), (2,4), (3,4), (5,6), (5,7), (6,7)$. There are a total of seven pairs.

The figure above corresponds to the choice $A = 2$ and $B = 4$.

Task Korupcija

... Corruption for all, not just for some. I offer corruption, a corrupt order, work, and growth. Whatever these other bosses offer you, I offer double. I even propose an eighth case: To whom? How much? ...

Little Mirko was fascinated by the speech of the man on television. He was convinced he understood the message: he had to corrupt the bits of his binary numbers.

Mirko considers the numbers $0, 1, \dots, 2^N - 1$ (viewed as binary numbers with N binary digits). Driven by his desire for corruption, Mirko will choose two numbers X and Y ($0 \leq X, Y < 2^N$) that differ in exactly one bit. He will then overwrite that bit with a "`?`" symbol in both numbers X and Y, thus achieving corruption: the numbers X and Y can no longer be distinguished. Mirko will repeat this process with the remaining numbers until he obtains exactly 2^{N-1} pairs of numbers that cannot be distinguished. In other words, each number between 0 and $2^N - 1$ belongs to exactly one pair, and two numbers can form a pair if and only if they differ in exactly one bit.

For an extra challenge, Mirko decides he wants exactly a_i pairs where the overwritten "`?`" symbol is at the i-th bit position, for each $i = 0, 1, \dots, N-1$. Here, bits are numbered from least significant to most significant, so the i-th bit corresponds to the value 2^i. Help Mirko by choosing the pairs to satisfy the desired conditions, or determine that it is impossible to do so.

Input

The first line contains a positive integer N as described above.

The second line contains N non-negative integers a_i, for $i = 0, \dots, N-1$, where a_i represents the required number of pairs that differ at the i-th bit position. The sum of all a_i is exactly 2^{N-1}.

Output

If it is impossible to form pairs satisfying the required conditions, output a single line containing `-1`.

Otherwise, output 2^{N-1} lines. Each line should contain two space-separated integers X and Y, representing a selected pair. You may output the pairs in any order.

If multiple solutions exist, output any.

Scoring

In all subtasks, it holds that $1 \leq N \leq 20$.

In every subtask, 20% of the points are awarded for simply determining whether it is possible to satisfy the conditions. For these points, if you output anything other than `-1`, you may print any pairing (even if it does not fully satisfy the required condition).

Subtask	Points	Constraints
1	15	$N \leq 4$
2	15	$N \geq 2$ and $a_i = 0$ for all $i > 2$
3	20	$N \leq 6$
4	50	No additional constraints.

Sample Cases

input

```
2
2 0
```

output

```
0 1
2 3
```

input

```
2
1 1
```

output

```
-1
```

input

```
3
2 0 2
```

output

```
0 1
2 6
3 7
4 5
```

Task Lirili Larila

Socrates: *Tell me, Plato, do you agree with me on this: the strongest fighters are those who can fly, like Bombardiro Crocodillo or Bombombini Gusini?*

Plato: *That is simply not the case. Land fighters, such as Brr Brr Patapim and Tung Tung Tung Sahur, have achieved their success despite their inability to fly.*

Socrates: *I believe the only way to find the truth is to let the fighters fight, and determine the outcome based on that.*

Plato: *Bravo, Socrates, I agree that this is the way to reach the truth.*

The decisive battle will take place on a connected graph with N vertices and M edges. Lirili Larila, a half-elephant half-cactus creature, owns the graph and insists that it is of her favorite type: a cactus graph. For the purposes of this problem, a *cactus graph* is defined as a simple connected graph in which each vertex belongs to at most one cycle.

The battle unfolds as follows: Initially, all flying fighters are placed at one designated starting vertex, and all land fighters are placed at a different designated starting vertex. As the battle progresses, the fighters spread their influence across the graph, attempting to conquer as many vertices as possible. Ultimately, a vertex is conquered either by the flying fighters or the land fighters, depending on whether it is closer to the starting vertex of the flying fighters or that of the land fighters. Vertices that are equidistant from both starting vertices remain unconquered, as they pose a significant challenge to both sides.

Lirili Larila wishes to control the outcome of the battle. She has already predetermined two positive integers A and B, representing the number of vertices to be conquered by the flying and land fighters, respectively. Help this lovable cactus-elephant choose starting vertices for both types of fighters so that, at the end of the battle, the number of conquered vertices matches the values A and B.

Additionally, you must find such a choice for T different scenarios.

Input

The first line contains a positive integer T, the number of different scenarios.

Each scenario is described as follows:

The first line contains four positive integers N, M, A, and B, representing the number of vertices and edges in the cactus graph, and the number of vertices to be conquered by the flying and land fighters, respectively.

Each of the next M lines contains two integers a and b ($1 \leq a, b \leq N$, $a \neq b$), representing an edge of the graph.

The given graph is guaranteed to be a cactus graph — that is, a simple connected graph in which each vertex belongs to at most one cycle.

The test data will guarantee that it is always possible to find a valid choice of starting vertices.

Output

Print T lines, one for each scenario.

In the i-th line, output two space-separated positive integers, representing the chosen starting vertices for the flying and land fighters in the i-th scenario. If multiple solutions exist, you may output any.

Scoring

In all subtasks, it holds that $2 \leq N \leq 200\,000$ and $2 \leq A + B \leq N$. Additionally, the sum of all N over all scenarios is at most $200\,000$.

The constraints listed below apply individually to each of the T given scenarios.

Subtask	Points	Constraints
1	6	The sum of all N is ≤ 300.
2	8	The given graph is a tree and the sum of all N is ≤ 5000.
3	25	The given graph is a tree.
4	13	The given graph has exactly one cycle and the sum of all N is ≤ 5000.
5	17	The given graph has exactly one cycle, and it is guaranteed that a solution exists where both starting vertices are within that cycle.
6	8	The given graph has exactly one cycle.
7	11	The sum of all N is ≤ 5000.
8	12	No additional constraints.

Sample Cases

input

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

output

```
4 3
```

input

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

output

```
1 6
```

input

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

output

```
4 2
```

Explanation of the First Sample Case:
The flying fighters conquer vertices 4, 5, and 6, while the land fighters conquer vertex 3. Vertices 1 and 2 remain unconquered.

Explanation of the Second Sample Case:
The flying fighters conquer vertices 1, 2, and 4, while the land fighters conquer vertices 5 and 6. Vertex 3 remains unconquered.

Explanation of the Third Sample Case:
The flying fighters conquer vertices 4, 5, and 6, while the land fighters conquer vertices 1, 2, and 3. There are no unconquered vertices.