Tsuen Wan Government Secondary School Olympiad in Informatics Team

St. Mark's School Programming Team

Ying Wa Girls' School Programming Team

2025/26 HKOI Final Mock II

SCHEDULE

Date	3 / 4 December 2025
Duration	3 hours

PROBLEMS

Problem		*Limit*		*Score*									
		Time	*Memory*	*Subtasks*									*Total*
TMF265	Pikmin +	1s	256MB	11	7	12	13	10	14	15	18		100
TMF266	Mario Bros. +	1s	256MB	8	10	12	16	14	18	22			100
TMF267	Mario Kart +	1s	256MB	9	7	17	10	12	15	16	14		100
TMF268	Splatoon +	1s	256MB	7	16	9	13	8	12	11	10	14	100
												Full Score	400

REMINDERS

- All problems are divided into subtasks. All test cases in a subtask must be passed to get points.
- You may attempt the problems using C++20 or Python 3 (secondary language). It is not guaranteed that the problems are solvable using secondary language.
- Unless otherwise specified, your output will be automatically fixed as follows:
  - Trailing spaces in each line will be removed.
  - An end-of-line character will be added to the end of the output if not present.
- 64-bit integers may be required in some problems, use `long long` in C++ if needed.

TMF265 Pikmin +

Score	Limit	
100	1s	256MB

It's nearly Christmas! As you have dreamed for so long, your parents finally bought you a brand new game console – *NintendOI Switch* +! Upon powering it on, you soon discover there are some *NintendOI Switch + Edition* games preloaded with the system. You can't wait and decide to play them immediately.

The first game is called *Pikmin* +. As a gamer, you know that the series involves some little creatures called Pikmin directed by Captain Olimar to perform various tasks. As a Pikmin is not particularly strong on its own, they always form large groups and work together to finish tasks.

In this game, there are N Pikmin. You soon encounter a task. You, as Olimar, need to select some of them to perform the task. However, you cannot choose any arbitrary Pikmin. As Pikmin are now able to think, each of them will only work if there are currently not enough Pikmin taking part in the task. In specific, the i^{th} Pikmin would only work **if and only if** there are currently less than or equal to T_i Pikmin working (including itself).

In other words, let $W_i = 1$ if the i^{th} Pikmin works, and $W_i = 0$ otherwise. You want to find such a sequence $W_1, W_2, \ldots, W_N$ with the total number of working Pikmin $x = W_1 + W_2 + \cdots + W_N$, such that for all $1 \leq i \leq N$, $x \leq T_i$ if $W_i = 1$ and $x > T_i$ if $W_i = 0$.

You want to know whether it is possible to select some Pikmin to work such that each of their requirements is satisfied, and find the total number of working Pikmin if so.

INPUT

The first line consists of an integer N, representing the number of Pikmin.

The second line consists of N integers, the i^{th} integer T_i represents the threshold of the number of working Pikmin for the i^{th} Pikmin to work.

OUTPUT

If it is impossible to select some Pikmin to work such that each of their requirements is satisfied, output `-1`.

Otherwise, output an integer, the total number of working Pikmin.

SAMPLE TESTS

	Input	Output
1	`5` `3 1 4 1 5`	`3`

The 1^{st}, 3^{rd} and 5^{th} Pikmin are chosen to work.

	Input	Output
2	`8` `2 0 2 5 1 2 0 6`	`-1`

It can be proven that it is impossible to satisfy all requirements.

	Input	Output
3	6 0 0 3 0 3 0	2

This sample satisfies the constraints of Subtask 5.

	Input	Output
4	4 1 4 4 1	2

This sample satisfies the constraints of Subtask 6.

CONSTRAINTS

$1 \le N \le 2 \times 10^5$

$0 \le T_i \le N$

SUBTASKS

	Points	Constraints
1	11	$1 \le N \le 2$
2	7	$T_1 = T_2 = \cdots = T_N$
3	12	$T_i = i$
4	13	$T_i = 0 \text{ or } N$
5	10	$T_i = 0 \text{ or } K$, where K is constant and $1 \le K \le N$
6	14	$T_i = A \text{ or } B$, where A, B are constants and $0 \le A \le B \le N$
7	15	$1 \le N \le 5000$
8	18	No additional constraints

TMF266 Mario Bros. +

Score	Limit	
100	1s	256MB

The second game is called *Mario Bros. +*. As a gamer, you know that the series involves 2D adventures of Mario and his friends, facing enemies on the way and saving the princess at the end.

In this game, our friend *Cappy* from *Super Mario Odyssey* once again joins you in the adventure. Soon, you encounter a challenge on stacking Goomba Tower.

Initially, there are N Goombas lining up in a row. Each Goomba has a distinct number between 1 and N written on it, where the i^{th} Goomba from the left has the number P_i written on it. A valid Goomba Tower consists of some Goombas stacking up on top of each other, where every Goomba must be on top of other Goombas with higher number then it. In other words, for a Goomba Tower of height K having Goombas with number $X_1, X_2, \ldots, X_K$ from top to bottom, X_a must be smaller than X_b for all $1 \leq a < b \leq K$.

At first, we can consider each Goomba to be a Goomba Tower of height 1. To stack Goomba Towers onto each other, in each move, you can use Cappy to capture a Goomba Tower, then jump onto one of its adjacent towers to stack on top of it. The new Goomba Tower formed by stacking up the two must still be valid. For each move, its jumping height is the height of tower being jumped on. In other words, for Goomba Tower u stacking on top of Goomba Tower v, the jumping height is the height of v.

The target of the challenge is to form a Goomba Tower with height N where all Goombas are on the same tower. You want to know if it is possible to form such a tower. If so, you would also want to minimise the maximum jump height among all steps in forming the tower.

INPUT

The first line consists of an integer N, representing the number of Goombas.

The second line consists of N integers, the i^{th} integer P_i represents the number written on the i^{th} Goomba.

OUTPUT

If it is impossible to form a valid Goomba tower with height N, output `IMPOSSIBLE` on the first and only line.

Otherwise, if it is possible to do so, output two lines.

On the first line, output `POSSIBLE`.

On the second line, output an integer, the minimised maximum jump height among all steps in forming the tower.

SCORING

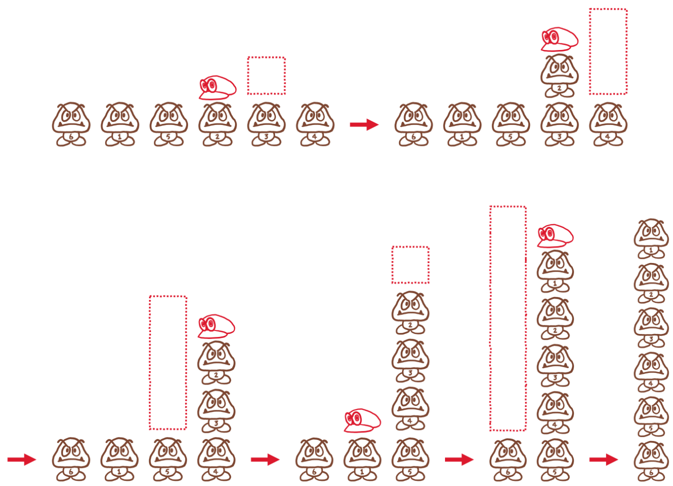
For each test case:

- You score 100% if your program outputs a correct answer.
- Otherwise, you score 50% if the first line of your output is correct.
- Otherwise, you score 0%.

Your score for each subtask is the lowest score among all test cases within the subtask.

SAMPLE TESTS

	Input	Output
1	6 6 1 5 2 3 4	POSSIBLE 4



	Input	Output
2	5 1 4 2 5 3	IMPOSSIBLE
3	5 2 1 3 5 4	POSSIBLE 2

This sample satisfies the constraints of Subtask 3.

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

This sample satisfies the constraints of Subtask 4.

	Input	Output
5	5 1 3 5 4 2	POSSIBLE 4

This sample satisfies the constraints of Subtask 5.

CONSTRAINTS

$1 \le N \le 2 \times 10^5$

P is a permutation of $1, 2, \ldots, N$

SUBTASKS

	Points	Constraints
1	8	$1 \le N \le 3$
2	10	$1 \le N \le 4$
3	12	$\lvert P_i - i \rvert \le 1$
4	16	$P_1 < P_2 < \cdots < P_{k-1} < P_k > P_{k+1} > \cdots > P_{N-1} > P_N$, where k is constant and $1 \le k \le N$
5	14	P_i is odd for $i \le \lceil \frac{N}{2} \rceil$
6	18	$1 \le N \le 2000$
7	22	No additional constraints

TMF267 Mario Kart +

Score	Limit	
100	1s	256MB

The third game is called *Mario Kart +*. As a gamer, you know that the series involves racing in beautifully crafted scenarios with fun *(and unfair)* items.

In this game, there are N players, numbered from 1 to N. You are player 1.

There are multiple races in a cup, players earn points according to their finishing position in each race. At the end, the players are ranked by the total points gained in all races. Formally, a Player p's final rank for the cup $= 1 + \text{number of players with total points greater than Player } p$.

Now, most of the races have been finished, and there is only one race left! Up to this point, Player i already has gained A_i points. In this final race, each player will finish in a distinct position, where the player who finish in the j^{th} place will gain P_j points.

You want to get Rank R for the cup. Therefore, you decide to play the final race strategically. Can you find a sequence of the finishing position $F_1, F_2, \ldots, F_N$, where player k finished in the F_k^{th} place for the final race, such that you can fulfil your wish of getting Rank R for the cup? Each player must have a distinct finishing position between 1 and N. In other words, $F_x \neq F_y$ for $x \neq y$.

INPUT

The first line consists of two integers N and R, representing the number of players and your target rank for the cup respectively.

The second line consists of N integers, the i^{th} integer A_i represents the points Player i have before the final race.

The third line consists of N integers, the j^{th} integer P_j represents the points that the player who finish in the j^{th} place will get for the final race.

OUTPUT

If it is impossible to arrange a sequence of placements in the final race such that you (Player 1) can get Rank R for the cup, output `IMPOSSIBLE` on the first and only line.

Otherwise, if it is possible to do so, output two lines.

On the first line, output `POSSIBLE`.

On the second line, output N integers $F_1, F_2, \ldots, F_N$, separate each by a space, representing a valid arrangement of players' finishing positions in the final race such that you can get Rank R for the cup, where the k^{th} integer F_k should represent the finishing position of Player k.

If there exist multiple valid arrangements, you may output any of them.

SCORING

For each test case:

- You score 100% if your program outputs a correct answer.
- Otherwise, you score 50% if the first line of your output is correct.
- Otherwise, you score 0%.

Your score for each subtask is the lowest score among all test cases within the subtask.

SAMPLE TESTS

	Input	Output

1	`4 2`
	`20 30 10 40`
	`15 11 6 0`
		`POSSIBLE`
		`1 4 2 3`

`1 4 3 2`, `2 4 1 3` and `2 4 3 1` are also valid arrangements.

	Input	Output

2	`4 3`
	`30 20 20 10`
	`25 15 10 5`
		`IMPOSSIBLE`

	Input	Output

3	`6 5`
	`13 10 16 12 14 17`
	`1000000000 1000000000 1000000000 1000000000 0 0`
		`POSSIBLE`
		`5 6 1 3 2 4`

This sample satisfies the constraints of Subtask 5.

	Input	Output

4	`3 2`
	`500 100 300`
	`400 200 200`
		`IMPOSSIBLE`

This sample satisfies the constraints of Subtask 6.

CONSTRAINTS

$1 \le R \le N \le 5000$

$0 \le A_i \le 10^9$

$0 \le P_N \le P_{N-1} \le \cdots \le P_2 \le P_1 \le 10^9$

SUBTASKS

	Points	Constraints
1	9	$1 \le N \le 3$
2	7	$A_i = 0$
3	17	$1 \le N \le 10$
4	10	$R = 1$
5	12	$A_i > 0$ $P_i = 0$ or 10^9
6	15	$P_i = X$ or Y, where X, Y are constants and $0 \le X \le Y \le 10^9$
7	16	$1 \le N \le 200$
8	14	No additional constraints

TMF268 Splatoon +

Score	Limit	
100	1s	256MB

The last game is called *Splatoon +*. As a gamer, you know that the series involves players using various weapons to spray ink to claim territory in the arena.

In this game, the arena is a rectangular grid with N rows and M columns, where the top-left corner is $(1, 1)$ and the bottom-right corner is (N, M). There are K teams, each represented by a colour numbered from 1 to K. As the name suggests, the game involves weapons with the spraying range of a "+" shape. In specific, if a player with team colour f uses the weapon at (r, c), the whole row r and column c will be filled with colour f.

For the whole duration of the game, the weapon is used P times by players from different teams, where the i^{th} use involves colour f_i at (r_i, c_i). The colour filled by a later use of weapon (with greater i) will cover the colour filled by an earlier use of weapon.

After all uses of weapon ends, the calculation of results begins. The rule is rather interesting, it is determined by number of cells with specific colour on each row or column. There are Q queries. For the j^{th} query, a dimension d_j is specify. If $d_j =$ `R`, row x_j is queried. If $d_j =$ `C`, column y_j is queried. You are asked to find the number of cells filled with the target colour t_j on the queried row or column.

To get the game results first-hand, you decide to find the answers of all queries by yourself before the game tells you!

INPUT

The first line consists of three integers N, M and K, representing the number of rows and number of columns of the arena and the number of colours respectively.

The second line consists of two integers P and Q, representing the number of weapon use and the number of queries respectively.

The i^{th} line of the next P lines consists of three integers f_i, r_i and c_i, representing the colour, row and column of the i^{th} use of weapon respectively.

The j^{th} line of the next Q lines first consists of an integer t_j and a character d_j, representing the target colour and the dimension of the j^{th} query respectively. If $d_j =$ `R`, an integer x_j follows, representing the row x_j is queried. If $d_j =$ `C`, an integer y_j follows, representing the column y_j is queried.

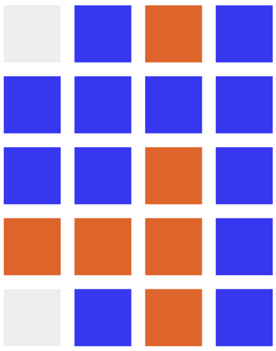
OUTPUT

Output Q lines.

On the j^{th} line, output an integer, the answer of the j^{th} query. If $d_j =$ `R`, output the number of cells on row x_j filled with colour t_j. If $d_j =$ `C`, output the number of cells on column y_j filled with colour t_j.

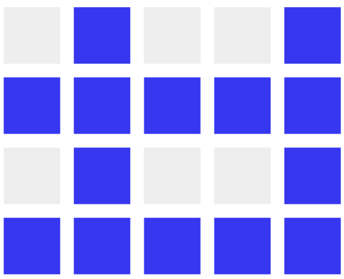
SAMPLE TESTS

	Input	Output
1	5 4 2 3 4 2 3 2 1 4 3 2 2 4 1 R 4 2 C 1 1 R 2 2 R 2	3 2 0 4



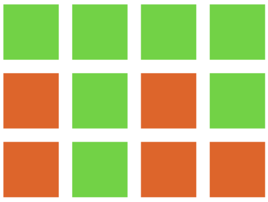
	Input	Output
2	4 5 1 2 3 1 2 2 1 4 5 1 R 2 1 R 3 1 C 5	5 2 4

This sample satisfies the constraints of Subtask 4.



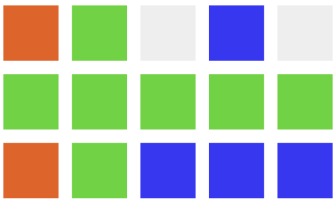
3	3 4 3 4 3 3 1 4 1 3 1 1 1 3 3 1 2 1 R 2 1 C 3 1 R 3	2 2 3

This sample satisfies the constraints of Subtask 6.



4	3 5 3 3 4 2 3 4 1 2 1 3 2 2 1 C 1 2 R 3 1 C 3 3 C 4	2 3 0 1

This sample satisfies the constraints of Subtask 8.



CONSTRAINTS

$1 \le N, M, K, P, Q \le 2 \times 10^5$

$1 \le f_i, t_j \le K$

$1 \le r_i, x_j \le N$

$1 \le c_i, y_j \le M$

$d_j =$ `R` or `C`

SUBTASKS

	Points	Constraints
1	7	$P = 1$
2	16	$1 \le N, M, P, Q \le 2000$
3	9	$d_j =$ `R` $x_j = 1$
4	13	$K = 1$ $r_u \ne r_v$ for $u \ne v$ $c_u \ne c_v$ for $u \ne v$
5	8	$K = 1$
6	12	$t_i = 1$
7	11	$r_u \ne r_v$ for $u \ne v$ $c_u \ne c_v$ for $u \ne v$
8	10	$f_u \ne f_v$ for $u \ne v$
9	14	No additional constraints