AIO. 2025

AUSTRALIAN INFORMATICS OLYMPIAD

AMT. AUSTRALIAN MATHS TRUST

DATE

Thursday 28 August

DURATION

3 hours

- There are six problems. All problems are of equal value (100 marks).
- The problems are sorted by approximate difficulty, but you may attempt them in any order.
- The contest runs for 3 hours.
- You may submit once per minute per problem. You may make an unlimited number of submissions.
- Submit solutions as you write them. Do not wait until the last minute to submit them.

Contest website: https://aio.edu.au/

STOP!
Do not open the paper until you are instructed to do so.

PROBLEM 1

Soccer Match

Time and memory limits: 1 second, 256 MB

Your favourite soccer team (codenamed Team 1) hasn't been doing too well lately. To cheer yourself up, you are watching a replay of their latest match against Team 2. You can stop watching the game at any point but, to end on a high, you only want to do so when your team is in the lead.

During the game, the two teams scored a combined total of N goals. You are given the sequence of N goals in the order that they were scored and your task is to determine whether there was any point during the game where Team 1 had scored more goals than Team 2. If so, you can stop watching the game while your favourite team is in the lead!

Subtasks and constraints

Your program will be graded using many secret tests. Every test follows some rules:

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

The secret tests are divided into subtasks. Your program must correctly solve **every test** within a subtask to earn the marks for that subtask:

- For Subtask 1 (30 marks), $N = 1$.
- For Subtask 2 (50 marks), $N \leq 1000$.
- For Subtask 3 (20 marks), no special rules apply.

Input

Your solution must read input and print output. We recommend using the solution templates (which you can find on the competition website) to help you with input and output.

The input follows a specific format:

- The 1st line contains the integer N.
- The 2nd line contains N integers, describing the goals in the order that they were scored. A `1` represents a goal for Team 1 (your favourite team) and a `2` represents a goal for Team 2.

Output

Your program must output `YES` if there was any point during the game where Team 1 had more goals than Team 2, and `NO` otherwise.

Sample input 1

```
1
1
```

Sample output 1

```
YES
```

Sample input 2

```
4
2 1 2 1
```

Sample output 2

```
NO
```

Sample input 3

```
7
2 2 1 1 1 2 2
```

Sample output 3

```
YES
```

Explanation

- In the 1st sample case, Team 1 scores the only goal of the game and goes into the lead. Therefore, the answer is `YES`.
- In the 2nd sample case, Team 1 is only ever tied with or behind Team 2 and so the answer is `NO`.
- In the 3rd sample case, Team 1 is in the lead after the 5th goal is scored and so the answer is `YES`.

PROBLEM 2

Buried Treasure

Time and memory limits: 1 second, 256 MB

For generations, stories have been told of a hidden treasure buried beneath the shore of Bitwise Beach. You know that the treasure is hidden at one of L distinct locations, numbered 1 through L.

After speaking with the locals of Bitwise Beach, you've received N clues. The ith clue consists of two integers A_i and B_i (with $A_i \leq B_i$), indicating that the treasure lies somewhere between locations A_i and B_i (inclusive). For example, if $A_i = 3$ and $B_i = 6$, then the treasure must be at location 3, 4, 5 or 6.

To save time and beat the other treasure hunters, you want to dig at as few locations as possible. So you ask yourself, "How many locations are consistent with all N clues?". It is possible that the answer is 0, meaning that the locals have tricked you!

Subtasks and constraints

Your program will be graded using many secret tests. Every test follows some rules:

- $2 \leq N \leq 200\,000$.
- $1 \leq L \leq 1\,000\,000$.
- $1 \leq A_i \leq B_i \leq L$ for all i.

The secret tests are divided into subtasks. Your program must correctly solve **every test** within a subtask to earn the marks for that subtask:

- For Subtask 1 (40 marks), $N = 2$ and $L \leq 1000$.
- For Subtask 2 (40 marks), $N \leq 1000$ and $L \leq 1000$.
- For Subtask 3 (20 marks), no special rules apply.

Input

Your solution must read input and print output. We recommend using the solution templates (which you can find on the competition website) to help you with input and output.

The input follows a specific format:

- The 1st line contains the integers N and L.
- The next N lines describe the clues. The ith of these contains the two integers A_i and B_i.

Output

Your program must output the number of locations that are consistent with all N clues.

Sample input 1

```
2 10
3 6
5 8
```

Sample output 1

```
2
```

Sample input 2

```
2 6
1 6
2 5
```

Sample output 2

```
4
```

Sample input 3

```
2 8
7 8
1 3
```

Sample output 3

```
0
```

Sample input 4

```
5 20
3 13
7 15
6 14
2 12
1 20
```

Sample output 4

```
6
```

Explanation

- In the 1st sample case, there are $N = 2$ clues and Bitwise Beach has $L = 10$ locations. The 1st clue tells you that the treasure is buried at location 3, 4, 5 or 6. The 2nd clue tells you that the treasure is buried at location 5, 6, 7 or 8. Only locations 5 and 6 are consistent with both clues and so the answer is 2.
- In the 2nd sample case, there are $N = 2$ clues and Bitwise Beach has $L = 6$ locations. The 1st clue tells you that the treasure is buried at location 1, 2, 3, 4, 5 or 6. The 2nd clue tells you that the treasure is buried at location 2, 3, 4 or 5. Only locations 2, 3, 4 and 5 are consistent with both clues and so the answer is 4.
- In the 3rd sample case, there are $N = 2$ clues and Bitwise Beach has $L = 8$ locations. The 1st clue tells you that the treasure is buried at location 7 or 8. The 2nd clue tells you that the treasure is buried at location 1, 2 or 3. No locations are consistent with both clues and so the answer is 0.
- In the 4th sample case, there are $N = 5$ clues and Bitwise Beach has $L = 20$ locations. There are 6 locations that are consistent with all clues: 7, 8, 9, 10, 11 and 12.

PROBLEM 3

Off the Track

Time and memory limits: 1 second, 256 MB

Your school has an L metre long running track and there are N students standing at different locations on the track. The ith student is standing P_i metres from the start of the track, where P_i is an integer between 1 and $L-1$ (inclusive).

You are playing a game with these students. You will yell one instruction each second, which will be either "forwards" or "backwards":

- If you yell "forwards", every student will step 1 metre towards the **end** of the track. Specifically, a student at location p will move to location $p+1$.
- If you yell "backwards", every student will step 1 metre towards the **start** of the track. Specifically, a student at location p will move to location $p-1$.

If a student reaches the start of the track (location 0) or the end of the track (location L), they are *off the track* and **stop participating in the game**. The game ends when all N students are off the track. Two example games are illustrated at the end of the problem statement.

If you strategically choose when to yell "forwards" and "backwards", what is the fewest number of seconds needed to end the game?

Subtasks and constraints

Your program will be graded using many secret tests. Every test follows some rules:

- $2 \le N \le 200\,000$.
- $N < L \le 10\,000\,000$.
- $1 \le P_i \le L-1$ for all i.
- $P_i < P_{i+1}$ for all i. That is, the initial locations are all different and in ascending order.

The secret tests are divided into subtasks. Your program must correctly solve **every test** within a subtask to earn the marks for that subtask:

- For Subtask 1 (20 marks), the best solution involves yelling the same instruction over and over.[1]
- For Subtask 2 (40 marks), $N = 2$.
- For Subtask 3 (40 marks), no special rules apply.

Input

Your solution must read input and print output. We recommend using the solution templates (which you can find on the competition website) to help you with input and output.

The input follows a specific format:

- The 1st line contains the integers N and L.
- The 2nd line contains N integers describing the initial locations of the students. The ith of these is P_i.

Output

Your solution must print a single integer: the fewest number of seconds needed to end the game.

[1]Sample input 1 satisfies the rules of Subtask 1 but sample inputs 2 and 3 do not.

Sample input 1

```
2 5
1 3
```

Sample output 1

```
3
```

Sample input 2

```
2 7
2 6
```

Sample output 2

```
4
```

Sample input 3

```
4 15
1 3 11 14
```

Sample output 3

```
10
```

Explanation

- In the 1st sample case, there are $N = 2$ students at locations 1 and 3 on the $L = 5$ metre long track. You can end the game in 3 seconds by yelling "backwards" 3 times. It is impossible to end the game in fewer than 3 seconds, and so the answer is 3.
- In the 2nd sample case, there are $N = 2$ students at locations 2 and 6 on the $L = 7$ metre long track. You can end the game in 4 seconds by yelling "forwards" 1 time and then "backwards" 3 times. It is impossible to end the game in fewer than 4 seconds, and so the answer is 4.
- In the 3rd sample case, there are $N = 4$ students at locations 1, 3, 11 and 14 on the $L = 15$ metre long track. You can end the game in 10 seconds by yelling "backwards" 3 times and then "forwards" 7 times.

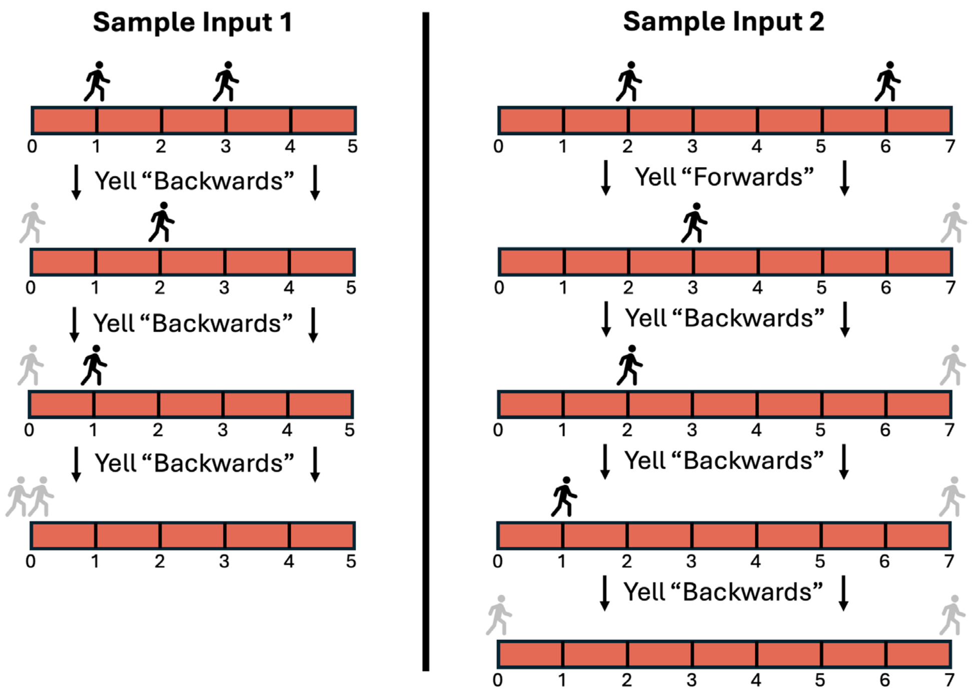

Figure 1: The solutions to sample inputs 1 and 2. The diagrams should be read from top to bottom.

PROBLEM 4

ORAC

Time and memory limits: 1 second, 256 MB

There are N informatics problems on the ORAC training site and you want to solve all of them! You have analysed all the problems, and the ith problem has a *difficulty score* of D_i. You can only solve a problem if your current informatics *skill level* is at least D_i. Unfortunately, your current skill level is 0.

Until you've solved all N problems, you will structure every day as follows:

- Each morning, you either
  - *Train*, increasing your skill level by 1, or
  - *Relax*, decreasing your skill level by 1 (your skill level can become negative).
- Each afternoon, you may solve **one** problem of your choosing, if your skill level is greater than or equal to the difficulty score of the problem.

You don't mind how many days it takes to finish all N problems, but you do enjoy relaxing. Specifically, you want to **minimise the number of mornings that you spend training**.

What is the minimum number of mornings you must spend training to solve all N problems, if you carefully plan your training, relaxing, and problem-solving?

Subtasks and constraints

Your program will be graded using many secret tests. Every test follows some rules:

- $1 \leq N \leq 200\,000$.
- $1 \leq D_i \leq 200\,000$ for all i.
- $D_i \leq D_{i+1}$ for all i. That is, the problem difficulties are in ascending order. However, you are allowed to solve the problems in any order.

The secret tests are divided into subtasks. Your program must correctly solve **every test** within a subtask to earn the marks for that subtask:

- For Subtask 1 (10 marks), $D_i = 1$ for all i and $N \leq 1000$.
- For Subtask 2 (20 marks), $D_i \leq 2$ for all i and $N \leq 1000$.
- For Subtask 3 (40 marks), $D_i \leq 1000$ for all i and $N \leq 1000$.
- For Subtask 4 (30 marks), no special rules apply.

Input

Your solution must read input and print output. We recommend using the solution templates (which you can find on the competition website) to help you with input and output.

The input follows a specific format:

- The 1st line contains the integer N.
- The 2nd line contains N integers describing the difficulty level of the problems. The ith of these is D_i.

Output

Your solution must print a single integer: the minimum number of mornings you must spend training to solve all N problems.

Sample input 1

```
6
1 1 1 1 1 1
```

Sample output 1

```
4
```

Sample input 2

```
4
1 2 2 2
```

Sample output 2

```
3
```

Sample input 3

```
10
3 3 3 3 4 4 4 5 5 7
```

Sample output 3

```
8
```

Explanation

In the 1st sample case, the minimum number of days you must train is 4. One way to structure your days is shown below.

Day	Skill before	Morning	Skill after	Afternoon
Day 1	0	Train	1	Solve 1st task ($D_1 = 1$)
Day 2	1	Train	2	Solve 2nd task ($D_2 = 1$)
Day 3	2	Relax	1	Solve 3rd task ($D_3 = 1$)
Day 4	1	Train	2	Solve 4th task ($D_4 = 1$)
Day 5	2	Relax	1	Solve 5th task ($D_5 = 1$)
Day 6	1	Train	2	Solve 6th task ($D_6 = 1$)

In the 2nd sample case, the minimum number of days you must train is 3. One way to structure your days is shown below.

Day	Skill before	Morning	Skill after	Afternoon
Day 1	0	Train	1	Do nothing
Day 2	1	Train	2	Solve 2nd task ($D_2 = 2$)
Day 3	2	Train	3	Solve 3rd task ($D_3 = 2$)
Day 4	3	Relax	2	Solve 4th task ($D_4 = 2$)
Day 5	2	Relax	1	Solve 1st task ($D_1 = 1$)

In the 3rd sample case, the minimum number of days you must train is 8.

PROBLEM 5

Pairing Cards

Time and memory limits: 1 second, 256 MB

You have found an unusual deck of N cards, where N is an even integer. Each card has an integer value written on it, and the ith card has the value A_i.

Your task is to rearrange the N cards into $\frac{N}{2}$ pairs, such that each card belongs to exactly one pair. However, you are only allowed to pair two cards with values A_i and A_j if **at least one** of the following conditions holds:

- The absolute difference of their values is equal to D. That is, $|A_i - A_j| = D$.[1] Or,
- The sum of their values is equal to S. That is, $A_i + A_j = S$.

You are given the integers D and S as part of the input.

Determine whether it is possible to divide the cards into $\frac{N}{2}$ valid pairs according to the rules above.

Subtasks and constraints

Your program will be graded using many secret tests. Every test follows some rules:

- $2 \leq N \leq 200\,000$ and N is even.
- $0 \leq D, S \leq 1\,000\,000$.
- $1 \leq A_i \leq 1\,000\,000$ for all i.
- $A_i \leq A_{i+1}$ for all i. That is, the values are in ascending order.

The secret tests are divided into subtasks. Your program must correctly solve **every test** within a subtask to earn the marks for that subtask:

- For Subtask 1 (20 marks), $S = 0$ and $N \leq 1000$.
- For Subtask 2 (20 marks), $D = 0$ and $N \leq 1000$.
- For Subtask 3 (35 marks), $N \leq 1000$.
- For Subtask 4 (25 marks), no special rules apply.

Input

Your solution must read input and print output. We recommend using the solution templates (which you can find on the competition website) to help you with input and output.

The input follows a specific format:

- The 1st line contains the integers N, D and S.
- The 2nd line contains N integers describing the values on the cards. The ith of these is A_i.

Output

Your program must output `YES` if it is possible to divide the cards into $\frac{N}{2}$ valid pairs, and `NO` otherwise.

[1] The notation $|x|$ denotes the absolute value of x. The absolute value of a number is equivalent to its distance from 0. For example, $|2| = |-2| = 2$. The absolute difference between two numbers is equivalent to the distance between them. For example, the absolute difference between 3 and 7 is $|3 - 7| = |-4| = 4$.

Sample input 1

```
6 2 0
1 1 2 3 3 4
```

Sample output 1

```
YES
```

Sample input 2

```
6 2 0
1 1 3 3 5 5
```

Sample output 2

```
NO
```

Sample input 3

```
8 0 8
1 2 3 4 4 5 6 7
```

Sample output 3

```
YES
```

Sample input 4

```
4 7 8
1 3 5 8
```

Sample output 4

```
YES
```

Explanation

- In the 1st sample case, $D = 2$ and $S = 0$, meaning that each pair must either have an absolute difference of 2 or a sum of 0. You can pair the 1st and 4th cards (values 1 and 3), the 2nd and 5th cards (values 1 and 3), and the 3rd and 6th cards (values 2 and 4). All the pairs have an absolute difference of $D = 2$.
- In the 2nd sample case, it is impossible to pair the cards such that each pair either has an absolute difference of $D = 2$ or a sum of $S = 0$.
- In the 3rd sample case, you can pair the 1st and 8th cards (values 1 and 7), the 4th and 5th cards (values 4 and 4), the 2nd and 7th cards (values 2 and 6), and the 3rd and 6th cards (values 3 and 5). All the pairs have a sum of $S = 8$.
- In the 4th sample case, you can pair the 1st and 4th cards (values 1 and 8), and the 2nd and 3rd cards (values 3 and 5). The 1st pair has a difference of $D = 7$ and the 2nd pair has a sum of $S = 8$.

PROBLEM 6

Robot Writing

Time and memory limits: 1 second, 256 MB

You are programming a robot that walks along a row of N tiles. Each tile has an integer value written on it, and the ith tile has the value T_i.

The robot can start on any tile. It will then perform M steps, where each step involves performing the following 3 actions in order:

1. The robot records the value of the tile it is currently standing on.
2. The robot moves exactly one tile either left or right (if the robot is on the 1st tile, it must move right, and if it is on the last tile, it must move left).
3. After moving, if the value on the new tile is less than the value on the tile it just moved from, the robot swaps these two tiles.

The **output** of the robot is the sequence of M numbers that it records, in order. An example is shown below.

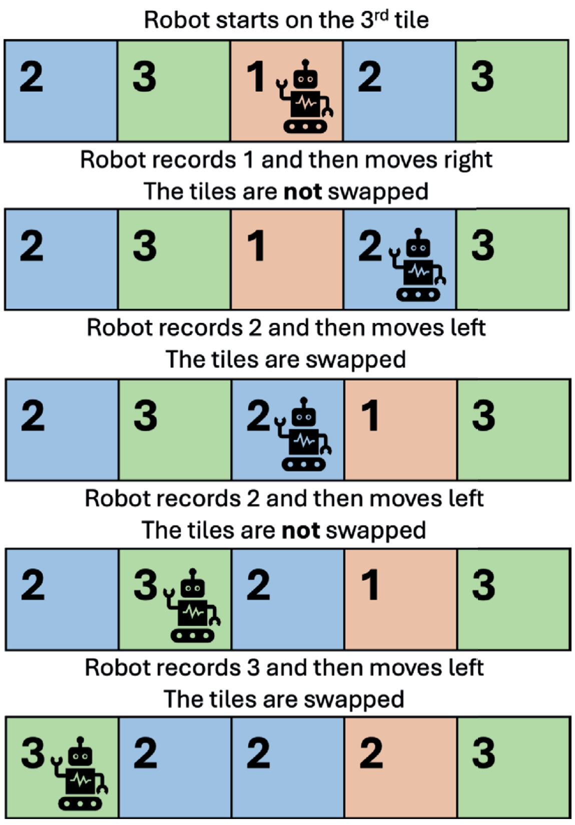
Robot starts on the 3rd tile
2 3 1 2 3
Robot records 1 and then moves right
The tiles are not swapped
2 3 1 2 3
Robot records 2 and then moves left
The tiles are swapped
2 3 2 1 3
Robot records 2 and then moves left
The tiles are not swapped
2 3 2 1 3
Robot records 3 and then moves left
The tiles are swapped
3 2 2 2 3

Figure 1: An example with $N = 5$ tiles. The tiles initially have $T_1 = 2$, $T_2 = 3$, $T_3 = 1$, $T_4 = 2$ and $T_5 = 3$. The robot starts on the 3rd tile and has $M = 4$ steps where it moves right, then left, then left, and then left. The output of the robot is $1, 2, 2, 3$.

You have a target sequence S of M integers. You would like to choose where the robot starts and whether it moves left or right for each step, so that its output is your target sequence. Is it possible to do this?

Subtasks and constraints

Your program will be graded using many secret tests. Every test follows some rules:

- $2 \leq N \leq 200\,000$.
- $1 \leq M \leq 200\,000$.
- $1 \leq T_i \leq 200\,000$ for all i.
- $1 \leq S_i \leq 200\,000$ for all i.

The secret tests are divided into subtasks. Your program must correctly solve **every test** within a subtask to earn the marks for that subtask:

- For Subtask 1 (20 marks), $N, M \leq 1000$ and $T_i = i$ for all i.
- For Subtask 2 (20 marks), $N, M \leq 1000$ and the values of T are distinct. That is, $T_i \neq T_j$ for all $i \neq j$.
- For Subtask 3 (20 marks), the values of T are distinct. That is, $T_i \neq T_j$ for all $i \neq j$.
- For Subtask 4 (20 marks), $N, M \leq 1000$.
- For Subtask 5 (20 marks), no special rules apply.

Input

Your solution must read input and print output. We recommend using the solution templates (which you can find on the competition website) to help you with input and output.

The input follows a specific format:

- The 1st line contains the integer N.
- The 2nd line contains N integers, describing the tiles. The ith of these is T_i.
- The 3rd line contains the integer M.
- The 4th line contains M integers, describing the target sequence. The ith of these is S_i.

Output

Your program must output `YES` if it is possible for the robot to output the target sequence, and `NO` otherwise.

Sample input 1

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

Sample output 1

```
YES
```

Sample input 2

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

Sample output 2

```
YES
```

Sample input 3

```
4
1 2 3 4
2
1 1
```

Sample output 3

```
NO
```

Sample input 4

```
4
1 2 3 4
3
2 3 2
```

Sample output 4

```
NO
```

Explanation

- The 1st sample case corresponds to the example at the start of the statement.
- In the 2nd sample case, if the robot starts at the 2nd tile and moves left, then right, then right, then right, then left, and then right, it will output the target sequence.
- In the 3rd and 4th sample cases, it is impossible for the robot to output the target sequences.