

Landmarks (landmarks)

Luxembourg City is cut in two by the valley of the Alzette. Lea and Tom are walking along either side of the valley, visiting different landmarks. Along Lea's path there are N landmarks, and along Tom's path there are M landmarks.

The landmarks are one of three types:

1. an old fortification,
2. a statue, or
3. a fountain.

The landmarks along Lea's path are A_0, A_1, \dots, A_{N-1} , and the landmarks along Tom's path are B_0, B_1, \dots, B_{M-1} , with each value being 1, 2 or 3.

They both want to discover the city together, but want to make sure at all times they are at the same type of landmark.

Consider a tuple of four integers (X_1, X_2, Y_1, Y_2) , describing the two stretches they plan to walk, satisfying:

- $0 \leq X_1 \leq X_2 < N$ – representing Lea's stretch starting at X_1 and ending at X_2 ,
- $0 \leq Y_1 \leq Y_2 < M$ – representing Tom's stretch starting at Y_1 and ending at Y_2 ,
- $A_{X_1} = B_{Y_1}$ – they start at the same type of landmark, and
- $A_{X_2} = B_{Y_2}$ – they finish at the same type of landmark.

They both begin at the start of their stretches: Lea starts at $X = X_1$, and Tom at $Y = Y_1$.

Each step, Lea chooses $x \in \{-1, 0, 1\}$, and Tom chooses $y \in \{-1, 0, 1\}$, corresponding to going to the previous landmark, the next one, or staying where they are. They then update their position to $X + x$ and $Y + y$ subject to the condition that after the step, they satisfy:

- $X_1 \leq X \leq X_2$ – Lea stays in her stretch,
- $Y_1 \leq Y \leq Y_2$ – Tom stays in his stretch, and
- $A_X = B_Y$ – they are at the same type of landmark.

The tuple (X_1, X_2, Y_1, Y_2) is said to be good if it is possible to apply zero or more steps until $X = X_2$ and $Y = Y_2$.

You are given Q queries. In each query, you are given a tuple of four integers (X_1, X_2, Y_1, Y_2) , and your task is to determine if the tuple is good.

Implementation

You will have to submit a single `.cpp` source file.



Among this task's attachments you will find a template `landmarks.cpp` with a sample implementation.

You will have to implement the following functions:

C++

```
void init(vector<int> A, vector<int> B)
```

- A is an array of N integers.

- B is an array of M integers.
- This procedure is called exactly once, before any call to `query`.

C++	<code>bool query(int X1, int X2, int Y1, int Y2)</code>
-----	---

- X_1, X_2, Y_1 and Y_2 are integers satisfying the following conditions:
 - $0 \leq X_1 \leq X_2 < N$,
 - $0 \leq Y_1 \leq Y_2 < M$,
 - $A_{X_1} = B_{Y_1}$, and
 - $A_{X_2} = B_{Y_2}$.
- If the tuple (X_1, X_2, Y_1, Y_2) is good, the function must return `true`. Otherwise, the function must return `false`.
- This procedure is called exactly Q times.

Sample Grader

Among this task's attachments you will find a simplified version of the grader used during evaluation, which you can use to test your solutions locally. The sample grader reads data from `stdin`, calls the function `init` once, calls the function `query` Q times, and writes back on `stdout` using the following format.

The input is made up of $Q + 3$ lines, containing:

- Line 1: the integers N, M and Q .
- Line 2: N integers, A_0, A_1, \dots, A_{N-1} .
- Line 3: M integers, B_0, B_1, \dots, B_{M-1} .
- Line $4 + i$ ($0 \leq i < Q$): the integers X_1, X_2, Y_1 and Y_2 .

The output is made up of Q lines, where the i -th line contains the value returned by the i -th call to the function `query` (`true` or `false`).

Constraints

- $1 \leq N, M, Q \leq 100\,000$.
- $1 \leq A_i, B_i \leq 3$ for all i .

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 0 [0 points]**: Sample test cases.
- **Subtask 1 [10 points]**: $N, M, Q \leq 5000$ and $A_0, A_1, \dots, A_{N-1}, B_0, B_1, \dots, B_{M-1} \leq 2$. (there are only two types of landmarks)
- **Subtask 2 [10 points]**: $A_0, A_1, \dots, A_{N-1}, B_0, B_1, \dots, B_{M-1} \leq 2$. (there are only two types of landmarks)
- **Subtask 3 [10 points]**: $N, M, Q \leq 5000$ and $A_i = (i \bmod 3) + 1$ for all i .
- **Subtask 4 [15 points]**: $A_i = (i \bmod 3) + 1$ for all i .
- **Subtask 5 [10 points]**: $N, M, Q \leq 300$.
- **Subtask 6 [25 points]**: $N, M, Q \leq 5000$.
- **Subtask 7 [20 points]**: No additional constraints.

Examples

stdin	stdout
5 7 4	true
1 2 3 2 1	false
1 1 2 3 2 3 1	true
0 2 0 5	false
2 4 3 6	
1 1 4 4	
1 1 2 4	

Explanation

$N = 5$, $M = 7$, $Q = 4$, with $A = [1, 2, 3, 2, 1]$ and $B = [1, 1, 2, 3, 2, 3, 1]$.

Query 1: $(X_1, X_2, Y_1, Y_2) = (0, 2, 0, 5)$ is good, since the following walk is possible:

step	1	2	3	4	5	6
X	0	0	1	2	1	2
Y	0	1	2	3	4	5
type of landmark	1	1	2	3	2	3

Query 2: $(X_1, X_2, Y_1, Y_2) = (2, 4, 3, 6)$ is not good. No walk satisfying the conditions is possible.

Query 3: $(X_1, X_2, Y_1, Y_2) = (1, 1, 4, 4)$ is good.

Query 4: $(X_1, X_2, Y_1, Y_2) = (1, 1, 2, 4)$ is not good.