Problem 251: Flatten 2D Vector

Problem Information

Difficulty: Medium
Acceptance Rate: 50.36%
Paid Only: Yes
Tags: Array, Two Pointers, Design, Iterator

Problem Description

Design an iterator to flatten a 2D vector. It should support the `next` and `hasNext` operations.

Implement the `Vector2D` class:

* `Vector2D(int[][] vec)` initializes the object with the 2D vector `vec`. * `next()` returns the next element from the 2D vector and moves the pointer one step forward. You may assume that all the calls to `next` are valid. * `hasNext()` returns `true` if there are still some elements in the vector, and `false` otherwise.

Example 1:

Input ["Vector2D", "next", "next", "next", "hasNext", "hasNext", "next", "hasNext"] [[[[1, 2], [3], [4]]], [], [], [], [], [], [], []] **Output** [null, 1, 2, 3, true, true, 4, false] **Explanation** Vector2D vector2D = new Vector2D([[1, 2], [3], [4]]); vector2D.next(); // return 1 vector2D.next(); // return 2 vector2D.next(); // return 3 vector2D.hasNext(); // return True vector2D.hasNext(); // return True vector2D.next(); // return 4 vector2D.hasNext(); // return False

Constraints:

* `0 <= vec.length <= 200` * `0 <= vec[i].length <= 500` * `-500 <= vec[i][j] <= 500` * At most `105` calls will be made to `next` and `hasNext`.

Follow up: As an added challenge, try to code it using only [iterators in C++](http://www.cplusplus.com/reference/iterator/iterator/) or [iterators in Java](http://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html).

Code Snippets

C++:

```
class Vector2D {
public:
Vector2D(vector<vector<int>>& vec) {

}

int next() {

}

bool hasNext() {

}
};

/**
* Your Vector2D object will be instantiated and called as such:
* Vector2D* obj = new Vector2D(vec);
* int param_1 = obj->next();
* bool param_2 = obj->hasNext();
*/
```

Java:

```
class Vector2D {

public Vector2D(int[][] vec) {

}

public int next() {

}

public boolean hasNext() {

}
}

/**
* Your Vector2D object will be instantiated and called as such:
* Vector2D obj = new Vector2D(vec);
* int param_1 = obj.next();
* boolean param_2 = obj.hasNext();
*/
```

Python3:

```
class Vector2D:

def __init__(self, vec: List[List[int]]):



def next(self) -> int:



def hasNext(self) -> bool:




# Your Vector2D object will be instantiated and called as such:
# obj = Vector2D(vec)
# param_1 = obj.next()
# param_2 = obj.hasNext()
```