Problem 2 : Flatten Nested List Iterator

1 Problem statement

Given a nested list of integers, implement an iterator to flatten it. Each element is either an integer or a list—whose elements may also be integers or other lists.

You are given two classes: `NestedInteger` and `NestedIterator`.
About the `NestedInteger` class:

- Member function `isInteger()` returns true if this `NestedInteger` holds a single integer, rather than a nested list.
- Member function `getInteger()` returns the single integer that this `NestedInteger` holds if it holds a single integer. The result is undefined if this `NestedInteger` holds a nested list.
- Member function `getList()` returns the nested list that this `NestedInteger` holds if it holds a nested list. The result is undefined if this `NestedInteger` holds a single integer

About the `NestedIterator` class:

- The constructor initializes with vector whose elements are `NestedInteger` objects.
- Member function `next()` returns the next element.
- Member function `hasNext()` returns true if there exists a next element.
- A `NestedIterator` object will be instantiated and operated as follows:

```
    NestedIterator i(nestedList);
    while (i.hasNext()) cout << i.next();
```

You task is complete the `NestedIterator` class. You are NOT allowed to modify the code outside the `NestedIterator` class.

1.1 Example

Given a list `[[1,1],2,[1,1]]`, calling `next` repeatedly until `hasNext` returns false, the order of elements returned by `next` should be: `[1,1,2,1,1]`.

1.2 Input

1.2.1 Format

The input contains a line of string representing a nested list of integers. A list is enclosed in square brackets, with elements separated by a space.

1.2.2 Sample

```
[ [ [ 1 [ 2 ] ] 3 ] 4 ]
```

1.3 Output

1.3.1 Format

The output contains a line of string representing the elements (separated by a space) of the input list, with the same appearance order.

1.3.2 Sample

```
1 2 3 4
```