

## Java

```
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode() {}
 *     ListNode(int val) { this.val = val; }
 *     ListNode(int val, ListNode next) { this.val = val; this.next = next; }
 * }
 */
class Solution {
    public int[][] spiralMatrix(int m, int n, ListNode head) {

    }
}
```

---

## JavaScript

```
/**
 * Definition for singly-linked list.
 * function ListNode(val, next) {
 *     this.val = (val===undefined ? 0 : val)
 *     this.next = (next===undefined ? null : next)
 * }
 */
/**
 * @param {number} m
 * @param {number} n
 * @param {ListNode} head
 */
```

```

    * @return {number[][]}
    */
var spiralMatrix = function(m, n, head) {

};

```

---

## TypeScript

```

/**
 * Definition for singly-linked list.
 * class ListNode {
 *     val: number
 *     next: ListNode | null
 *     constructor(val?: number, next?: ListNode | null) {
 *         this.val = (val===undefined ? 0 : val)
 *         this.next = (next===undefined ? null : next)
 *     }
 * }
 */

function spiralMatrix(m: number, n: number, head: ListNode | null): number[][] {

};

```

---

## C++

```

/**
 * Definition for singly-linked list.
 * struct ListNode {

```

```

*     int val;
*     ListNode *next;
*     ListNode() : val(0), next(nullptr) {}
*     ListNode(int x) : val(x), next(nullptr) {}
*     ListNode(int x, ListNode *next) : val(x), next(next) {}
* };
*/
class Solution {
public:
    vector<vector<int>> spiralMatrix(int m, int n, ListNode* head) {

    }
};
-----

```

## C#

```

/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     public int val;
 *     public ListNode next;
 *     public ListNode(int val=0, ListNode next=null) {
 *         this.val = val;
 *         this.next = next;
 *     }
 * }
 */
public class Solution {
    public int[][] SpiralMatrix(int m, int n, ListNode head) {

    }
}

```

```
}
```

---

## Kotlin

```
/**
 * Example:
 * var li = ListNode(5)
 * var v = li.`val`
 * Definition for singly-linked list.
 * class ListNode(var `val`: Int) {
 *     var next: ListNode? = null
 * }
 */
class Solution {
    fun spiralMatrix(m: Int, n: Int, head: ListNode?): Array<IntArray> {

    }
}
```

---

## Go

```
/**
 * Definition for singly-linked list.
 * type ListNode struct {
 *     Val int
 *     Next *ListNode
 * }
 */
func spiralMatrix(m int, n int, head *ListNode) [][]int {
```

}

-----