

Java

```
/**
 * // This is the BinaryMatrix's API interface.
 * // You should not implement it, or speculate about its implementation
 * interface BinaryMatrix {
 *     public int get(int row, int col) {}
 *     public List<Integer> dimensions {}
 * };
 */

class Solution {
    public int leftMostColumnWithOne(BinaryMatrix binaryMatrix) {

    }
}
```

JavaScript

```
/**
 * // This is the BinaryMatrix's API interface.
 * // You should not implement it, or speculate about its implementation
 * function BinaryMatrix() {
 *     @param {integer} row, col
 *     @return {integer}
 *     this.get = function(row, col) {
 *         ...
 *     };
 *
 *     @return {[integer, integer]}
 *     this.dimensions = function() {
```

```

*
*      ...
*    };
* };
*/

/**
 * @param {BinaryMatrix} binaryMatrix
 * @return {number}
 */
var leftMostColumnWithOne = function(binaryMatrix) {

};
-----

```

C++

```

/**
 * // This is the BinaryMatrix's API interface.
 * // You should not implement it, or speculate about its implementation
 * class BinaryMatrix {
 * public:
 *     int get(int row, int col);
 *     vector<int> dimensions();
 * };
 */

class Solution {
public:
    int leftMostColumnWithOne(BinaryMatrix &binaryMatrix) {

    }
};

```

C#

```
/**
 * // This is BinaryMatrix's API interface.
 * // You should not implement it, or speculate about its implementation
 * class BinaryMatrix {
 *     public int Get(int row, int col) {}
 *     public IList<int> Dimensions() {}
 * }
 */

class Solution {
    public int LeftMostColumnWithOne(BinaryMatrix binaryMatrix) {

    }
}
```
