**779. K-th Symbol in Grammar: -**

Medium Accepted: 155K Submissions: 350K Acceptance Rate: 44.3%

We build a table of n rows (**1-indexed**). We start by writing 0 in the 1st row. Now in every subsequent row, we look at the previous row and replace each occurrence of 0 with 01, and each occurrence of 1 with 10.

* For example, for n = 3, the 1st row is 0, the 2nd row is 01, and the 3rd row is 0110.

Given two integer n and k, return the kth (**1-indexed**) symbol in the nth row of a table of n rows.

**Example 1:**

**Input:** n = 1, k = 1

**Output:** 0

**Explanation:** row 1: 0

**Example 2:**

**Input:** n = 2, k = 1

**Output:** 0

**Explanation:**

row 1: 0

row 2: 01

**Example 3:**

**Input:** n = 2, k = 2

**Output:** 1

**Explanation:**

row 1: 0

row 2: 01

**Constraints:**

* 1 <= n <= 30
* 1 <= k <= 2n - 1

**Code: -**

class Solution {

public:

    int kthGrammar(int n, int k) {

        int col = pow(2, n-1), mid = col / 2;

        // base case:-

        if(n == 1 and k == 1)   return 0;

        // recursive case:-

        // found in left half

        if(0 <= k and k <= mid)

            return kthGrammar(n-1, k);

        // found in right half

        return !kthGrammar(n-1, k-mid);

    }

};

**T.C: - O(N)**

**S.C: - O(1) without recursive call stack storage**