

# Problem 1545: Find Kth Bit in Nth Binary String

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 70.14%

**Paid Only:** No

**Tags:** String, Recursion, Simulation

## Problem Description

Given two positive integers `n` and `k`, the binary string `Sn` is formed as follows:

\* `S1 = "0"` \* `Si = Si - 1 + "1" + reverse(invert(Si - 1))` for `i > 1`

Where `+` denotes the concatenation operation, `reverse(x)` returns the reversed string `x`, and `invert(x)` inverts all the bits in `x` ('0' changes to '1' and '1' changes to '0').

For example, the first four strings in the above sequence are:

\* `S1 = "0"` \* `S2 = "0\*\*1\*\* 1"` \* `S3 = "011\*\*1\*\* 001"` \* `S4 = "0111001\*\*1\*\* 0110001"`

Return \_the\_ `kth` \_bit\_ \_in\_ `Sn`. It is guaranteed that `k` is valid for the given `n`.

**Example 1:**

**Input:** n = 3, k = 1 **Output:** "0" **Explanation:** S3 is "\*\*\*\_0\_\*\* 111001". The 1st bit is "0".

**Example 2:**

**Input:** n = 4, k = 11 **Output:** "1" **Explanation:** S4 is "0111001101\*\* \_1\_\*\* 0001". The 11th bit is "1".

**Constraints:**

`* `1 <= n <= 20` * `1 <= k <= 2n - 1``

## Code Snippets

### C++:

```
class Solution {  
public:  
    char findKthBit(int n, int k) {  
  
    }  
};
```

### Java:

```
class Solution {  
public char findKthBit(int n, int k) {  
  
}  
}
```

### Python3:

```
class Solution:  
    def findKthBit(self, n: int, k: int) -> str:
```