

Mobile numeric keypad

Given the mobile numeric keypad. You can only press buttons that are up, left, right, or down to the current button. You are not allowed to press bottom row corner buttons (i.e. * and #). Given a number **N**, the task is to find out the number of possible numbers of the given length.

Example 1:

```
Input: 1
Output: 10
Explanation: Number of possible numbers
would be 10 (0, 1, 2, 3, ..., 9)
```

Example 2:

```
Input: N = 2
Output: 36
Explanation: Possible numbers: 00, 08, 11,
12, 14, 22, 21, 23, 25 and so on.
If we start with 0, valid numbers
will be 00, 08 (count: 2)
If we start with 1, valid numbers
will be 11, 12, 14 (count: 3)
If we start with 2, valid numbers
will be 22, 21, 23, 25 (count: 4)
If we start with 3, valid numbers
will be 33, 32, 36 (count: 3)
If we start with 4, valid numbers
will be 44, 41, 45, 47 (count: 4)
If we start with 5, valid numbers
will be 55, 54, 52, 56, 58 (count: 5)
and so on..
```

Your Task:

You don't need to read input or print anything. Complete the function `getCount()` which takes **N** as input parameter and returns the integer value

Expected Time Complexity: $O(N)$

Expected Auxiliary Space: $O(N)$

Constraints:

$1 \leq N \leq 25$

```
class Solution:
    def getCount(self, N):
        # code here
        dialTable = {
            0:[0,8],
            1:[1,2,4],
            2:[1,2,3,5],
            3:[2,3,6],
            4:[1,4,5,7],
            5:[2,4,5,6,8],
            6:[3,5,6,9],
            7:[4,7,8],
            8:[5,7,8,9,0],
            9:[6,8,9]
        }
        dp = [[0]*(10) for _ in range(N+1)]
        for i in range(1,N+1):
            for j in range(10):
                if i==1:
                    dp[i][j] = 1
                else:
                    candidates = dialTable[j]
                    dp[i][j]=sum(dp[i-1][k] for k in candidates)
        return sum(dp[N])
```