CS 313E (Mitra) Climbing Stairs Problem

Given a staircase with N steps in total to reach the top where you can climb either 1, 2, or 3 steps at a time, your goal is to find out the number of distinct ways you can reach the top. For example, in a staircase with 3 stairs, you can reach the top the following ways:

1. 1 step + 1 step + 1 step
2. 1 step + 2 steps
3. 2 steps + 1 step
4. 3 steps

This gives you a total of 4 ways to reach the top.

If we consider a random step K, this can be reached from the $(K-1)^{th}$ $(K-2)^{th}$, and $(K-3)^{th}$ steps by climbing 1, 2, or 3 steps respectively. Because of this property, the number of ways to reach the K^{th} step is simply the sum of the number of ways to reach the 3 steps before it. Therefore, this problem can be solved using dynamic programming if we let DP[i] be the number of ways to reach the ith step by using the following recurrence:

```
DP[i] = DP[i-1] + DP[i-2] + DP[i-3]
```

if i-1, i-2, or i-3 are negative simply let DP[negative] = 0

Using this formula, complete the table below to find the number of ways to reach the top of a staircase with 10 steps.

Stair, i	DP[i]
0	1
1	1
2	2
3	4
4	
5	
6	
7	
8	
9	
10	