Date : __ / __ / ____

6 June.

1: Dragon Tower The Spiral Climb

Approach:

No. of ways to reach floor N = (N-1) + (N-2)

Base cases:

if (N == 0) → 1 way.

if (N == 1) → 1 way.

Code:

```
static int countWays(int n) {
    if(n==0 || n==1)
        return 1;
    int[] dp = new int[n+1];
    dp[0]=1;
    dp[1]=1;
    for (int i=2; i<=n; i++) {
    dp[i] = dp[i-1] + dp[i-2];
    }
    return dp[n];
    }
}
```

Dry Run:

N	Ways
0	1
1	1
2	2 (1+1, 2)
3	3 (1+1+1, 1+2, 2+1)
4	5 (1+1+1+1, 1+1+2, 1+2+1, 2+1+1, 2+2)

Time complexity: O(N)

Space complexity: O(N)

Total ways to reach floor 4: 5.

Date :

2. Magical Dice Reach the treasure:

Approach:

We can use recursive DP (top-down).

ways(N) = ways(n-1) + ways(n-2) + ... ways(n-6).

Base case:

if N==0, 1 way.

if n<0, 0 ways

Code:

```
int static countWays(int n) {
    int[] dp = new int[n+1];
    dp[0] = 1;
    for(int i=1; i<=n; i++) {
        for(int dice = 1; dice <= 6; dice++) {
            if(i-dice >= 0) {
                dp[i] += dp[i-dice]; }
            }
        }
        return dp[n];
    }
}
```

Dry Run. n = 3

1+1+1

1+2.

2+1

3.

Total = 4.

Time Complexity: O(N)

Space Complexity: O(N).