6th June's

Problem 1: Dragon Towers

Approach: climb either :
- 1 floor → normal stairs
- 2 " → enchanted boots

we can use DP formula :

ways(n) = ways(n-1) + ways(n-2)

pseudo code:

```
if (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=4 ; dp[0] = 1
dp[1] = 1
dp[2] = 2 // (1+1), (2)
dp[3] = 3 // (1+1+1), (1+2), (2+1)
dp[4] = 5 // (1+2+1), (2+1+1), (1+1+2), (2+2)

output: 5

Problem 2 :- Magical Dice

Approach :- • Let dp[i] be number of ways
• to reach tile i

dp[i] = dp[i-1] + dp[i-2] + dp[i-3] + dp[i-4] +
dp[i-5] + dp[i-6]

• we will use recursive approach.

pseudo code :-

```
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: dp[0] = 1

dp[1] = dp[0] = 1 → (1)

dp[2] = dp[1] + dp[0] = 1+1 = 2

dp[3] = dp[2] + dp[1] + dp[0]

= 2+1+1 = 4

Output = 4