Assignment: Dynamic Programming

1. **Solve Dynamic Programming Problem and Compare with Naïve approach**
   You are playing a puzzle. A random number N will be given, you have blocks of length 1 unit and 2 units. You need to arrange the blocks back to back such that you get a total length of N units. In how many distinct ways can you arrange the blocks for given N.
   a. Write a description of approach to solve it using Dynamic Programming paradigm
   b. Implement a function **blockpuzzle_dp(N)** that solves this problem using Dynamic Programming using either top-down or bottom-up approach
   c. Write pseudocode for the brute force approach
   d. Compare the time complexity of both the approaches
   e. Write the recurrence formula for the problem
   Name your file **BlockPuzzle.py**

   Example 1:

   Input: N=2, Result: 2

   Explanation: There are two ways. (1+1 , 2)

   Example 2:

   Input: N=3, Result: 3

   Explanation: There are three ways (1+1+1, 1+2, 2+1)

2. **Solve a problem using top-down and bottom-up approaches of Dynamic Programming technique**
   Two players A & B are playing a game. The rules of the game are:
   At the start one number N will be given. The player who starts would have to pick a number i such that 0<i<N, the condition is that N%i == 0. The second player would pick a number j from N-i, satisfying the condition 0<j<(N-i). And the game goes on until there is no more possibility of making any selection. Each player would play in turns, and A always starts the game. Assume both players play optimally.
   Given a number N return if A would win the game or not.

   Example 1:

   Input: N=2, Result: True

   Explanation: A choses 1, and B has no more numbers to chose

   Example 2:

   Input: N=3, Result: False

   Explanation: A choses 1, B choses 1, and A has no more numbers to chose

   a. Implement a solution to this problem using Top-down Approach of Dynamic Programming, name your function **game_topdown(N)**
   b. Implement a solution to this problem using Bottom-up Approach of Dynamic Programming, name your function **game_bottomup(N)**

c. Explain your approach to solve this problem. How is your top-down approach different from the bottom-up approach?
d. What is the time complexity and Space complexity using Top-down Approach
e. What is the time complexity and Space complexity using Bottom-up Approach
f. Write the subproblem and recurrence formula for your approach

Name your file **Game.py**

Debriefing (required!): ----------

Report:

1. Approximately how many hours did you spend on this assignment?
2. Would you rate it as easy, moderate, or difficult?
3. How deeply do you feel you understand the material it covers (0%–100%)?
4. Any other comments?

Note: 'Debriefing' section is intended to help us calibrate the assignments.