### **Question 1 [15 Points]**

You are given a stack of integers. Write a function named ***sum\_stack*** that takes a stack st as input and modifies it such that the resulting stack contains the sum of corresponding elements from the **top** and **bottom** of the original stack. Specifically, the last element (top) of the stack is summed with the 0th element (bottom), the second last element is summed with the 1st element, and so on. It is ensured that there will be an even number of elements in the original stack.

Assume the Stack class is already given and provides standard methods: push, pop, peek, and isEmpty.

Constraints:

* You are only allowed to use instances of the provided Stack class, which supports the provided methods
* No other data structures can be used other than Stack.

**[Hint: You can create multiple instances of the Stack class to assist in solving the problem.]**

| **Sample Input:** | **Sample Output:** | **Explanation:** |
| --- | --- | --- |
| **Input Stack:**  **10 15 20 50 25 35** | **Output Stack:**  **70 40 45** | **Pair 1: 35 (Top) + 10 (Bottom) = 45**  **Pair 2: 25 (Second from top) + 15 (Second from bottom) = 40**  **Pair 3: 50 (Third from top) + 20 (Third from bottom) = 70** |