Question 1 [15 Points]

A stack of integers is given to you. Now you have to rearrange the given stack in such a way so that on top there will be an ODD integer and the EVEN integers will come after an ODD integer. You have to maintain the relative sequence of the even numbers and the odd numbers.
There can be **TWO** scenarios in the given Stack:
1. There will be the same number of odd and even integers in the given stack if the stack length is EVEN [odd.count = even.count]   **OR**
2. The total number of the odd integers will be one extra from the total number of even integers if the stack length is ODD [odd.count+1 = even.count]
You need to solve the above problem using **Stack class**. You cannot use other methods than pop(), peek(), push(), isEmpty() methods of Stack. Assume the Stack class is already given and provides standard methods: push, pop, peek, and isEmpty.

Constraints:

- 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.]

<table>
<tr><td>Sample Input</td><td>Sample Output</td><td>Explanation</td></tr>
<tr><td>Stack:
| 11 |
| 22 |
| 24 |
| 35 |
| 41 |</td><td>Stack:
| 11 | ← top
| 22 |
| 35 |
| 24 |
| 41 |</td><td>There are 3 odd integers(11,35,41) and 2 even integers(22,24). On the top 11 is kept as it is an even integer. Then comes the odd integer then again comes the odd integer. The arrival sequence is maintained for even and odd integers.</td></tr>
<tr><td>Stack:
| 8 |
|10 |
| 7 |
| 5 |
|12 |
| 3 |</td><td>Stack:
| 7 | ← top
| 8 |
| 5 |
|10 |
| 3 |
|12 |</td><td></td></tr>
</table>