Activity 1: Pseudocode for a Capped-capacity Stack

Write pseudocode for the functions `isEmpty()`, `push(obj)`, and `pop()` for a capped-capacity stack. Assume your stack has the following constructor and `size()` functions. Write the big-O runtime on each operation.

```
Stack(): O(     )
    data = array of size 20
    count = 0

function push(obj): O(     )
    # TODO

function size(): O(1)
    return count

function isEmpty(): O(     )
    # TODO

function pop(): O(     )
    # TODO
```

What should happen if the user tries to push to a stack that is at full capacity? What about when someone tries to pop from an empty stack?

Activity 2: Expanding Stack - Analysis of Incremental Strategy

Based on the calculations in lecture of the number of operations per push for 5, 10, and 15 pushes, using an incremental expansion strategy where \( c = 5 \), what would be the average number of operations per push for 20 pushes?