1 Add a method, `peek()`, to `Stack` that returns the most recently inserted element on the stack, without removing it. (1)

2 Add a method, `isEmpty()`, to `Stack` that returns `true` if there are no elements in the stack and `false` otherwise. (1)

3 Suppose that a client performs an intermixed sequence of `push()` and `pop()` operations on a stack of integers. The push operations put the integers 0 through 9, in order, on the stack; the pop operations print out the return value. Which of the following sequence(s) could *not* occur? (2)

a. 4321098765
b. 4687532901
c. 2567489310
d. 4321056789
e. 1234569870
f. 0465381729
g. 1479865302
h. 2143658790

4 Suppose `stack` is an object representing an implemented stack of integers. What does the following code fragment print when `N` is 50? Explain what your output represents (you might want to try additional values for `N` to verify your answer). (2)

```
while (N > 0) {
  stack.push(N % 2);
  N = N / 2;
}
while (!stack.isEmpty()) {
  System.out.print(stack.pop())
}
System.out.println()
```

5 Write the method `Parentheses()` that takes a string of different parentheses and uses a stack to determine whether its parenthesis are properly balanced. For example, your method should return `true` for "`[()]{[()()]()}`" and `false` for "`[(])`". (4)

Hint: Use the `substring()` method of the `String` class.

6 Suppose that we have a sequence of intermixed `push()` and `pop()` operations on a stack of integers. The push operations put the N integers, 0 through $N-1$, in order on the stack; the pop operations print out the return value. Write the method, `ValidPermutation()`, that will return `true` if a given string representing a space-separated sequence of these integers represents valid output for this stack and `false` otherwise. (6)

Hint #1: Your method should also take `N` as a parameter.
Hint #2: This method should be able to answer Question #3 for you.
Hint #3: The `split()` method of the `String` class may be useful.