Logical Java Problem Statements

1. Without Using Loops or Recursion, Print 1 to 100

> Write a program that prints numbers from 1 to 100 **without using** any loop (`for`, `while`, `do-while`) or recursion.

Hint: Use streams, or object constructors, or thread hacks.

---

2. Find the Second Highest Number Without Sorting or Duplicates

> Given an integer array, find the **second highest unique number** without using `Collections.sort()` or streams.

```
Input: [3, 1, 4, 4, 5, 5, 2]
Output: 4
```

---

3. Find Missing Number in an Arithmetic Progression

> Given an arithmetic progression with one missing number (unsorted), find the missing number.

```
Input: [2, 6, 10, 14, 18] → 4 is missing
Output: 4
```

---

4. Number Is Reversible?

> A number is reversible if **reversing it and adding** gives a number where **all digits are odd**.

```
Input: 36 → 36 + 63 = 99 → valid
Output: true
```

Write a method to check this.

---

5. Find Repeating Pattern in a String

Given a string, return the smallest repeating unit if possible.

```
Input: "abcabcabc"
Output: "abc"
```

---

6. Reverse Only Digits in a Mixed String

Reverse the digits in a string without touching other characters.

```
Input: "a1b2c3d4"
Output: "a4b3c2d1"
```

---

7. Rearrange Characters So That No Adjacent Are the Same

Given a string, rearrange characters so no two adjacent characters are the same.

```
Input: "aaabbc"
Possible Output: "ababac"
```

Hint: Use PriorityQueue or Greedy logic.

---

8. Valid Sudoku Checker

Given a 9x9 board, write logic to check whether it's a valid Sudoku board (rows, cols, boxes).

Trick: No easy built-in validator, you must track all 3 constraints logically.

---

9. Validate Email Without Regex

Write your own logic to validate a basic email without using regex.

```
Input: "test@gmail.com" → true
Input: "invalid@com" → false
```

---

10. Find Subarray with Given Sum (Positive Integers Only)

Find a subarray in an array whose elements sum to a given number.

```
Input: [1, 2, 3, 7, 5], sum = 12
Output: [2, 3, 7]
```
