

## Problem Set 2 Exercise #27: Reverse Number

**Reference:** Lecture 4 notes

**Learning objective:** Algorithm design

**Estimated completion time:** 70 minutes

### Problem statement:

[Past year CS1101 sit-in lab question]

You are to write a program **PS2\_Ex27\_ReverseNumber.java** that reads two positive integers *low* and *high*, and finds out how many integers in the range [*low*, *high*] (both inclusive) whose reverse is the same as the original value. You may assume that  $low \leq high$ .

For example, if *low* is 150 and *high* is 202, then there are 6 integers in the range [150, 202] whose reverse is the same as itself. They are: 151, 161, 171, 181, 191, and 202.

You need to write a modular program. Besides the `main()` method, there should be at least another method that computes some result.

You are **not** allowed to use array or string methods for this task.

A tip is given at the end of this page.

### Sample run #1:

```
Enter the range: 150 202
There are 6 reverse numbers.
```

### Sample run #2:

```
Enter the range: 12 21
There are 0 reverse numbers.
```

### Useful tip:

The challenge is how to get the reverse of a given integer. Hardcode different cases (has 1 digit, 2 digits, 3 digits...) is not the proper way of problem solving.

Engage a loop to extract digits round by round, along the way accumulate the reverse number. When the loop ends, reverse number should be created. Of course, it's easier said than done 😊.