CMSC 142 Laboratory Exercise 1
Progress Check (Required): February 13, 2025
Final Deadline: February 18, 2025

Given an integer `n`, find the next smallest palindrome integer.
Palindrome = reading it backwards is same as reading it forward (e.g. 101, 212, 999)

Example:

Input: 5	Next Palindrome = 6
Input: 18	Next Palindrome = 22
Input: 934	Next Palindrome = 939

INPUT / OUTPUT

1. You will be reading input from the command line
2. First input is the number of test cases to be solved, say X
3. This will be followed by X integers
4. For each integer, print the next smallest palindrome integer.

EXAMPLE INPUT

```
5
0
18
934
5
757
```

EXPECTED OUTPUT

```
1
22
939
6
767
```

Files for Submission

1. **The implemented program code should be written in python**. The program should match the pseudocode that is provided in item #2.
2. **A report in pdf format**
    a. **Write a pseudocode**
    The pdf file should contain the pseudocode of your chosen implementation. The pseudocode may be expressed in natural language through a list of steps. *Note*: Do not use a specific programming language in writing your pseudocode. For each step, or block of code, specify the cost and the

estimated number of timesteps of execution. Given these, compute the worst case Big-O running time complexity of the program.

b. Analyse the performance of the code given the structure of an input

  i. What is the best-case complexity of your program? What is the structure of the input that results in the best case?

  ii. What is the worst case complexity of the program? What is the structure of the input that results in the worst case?

c. Check the **correctness** and **performance** of your program by testing it on a set of inputs. The following files are provided:

  i. `small_input` – contains 5 inputs. **Note**: the first element/value is the number of items/cases

  ii. `small_solution` – contains the solution to `small_input` (item `#2b i`)

  iii. `full_input` – contains 100,000 inputs. **Note**: the first element/value is the number of items/cases

  iv. full_`solution` – contains the solution to `full_input` (`item #2b iii`)

If your solutions (NEXT PALINDROME) match the given solution, then zero errors are found.

Measure the elapsed running time of your performance when the program is executed using the

1. small input
2. full input

You may check the Time module in python to compute this information.

Deadlines:

1. Progress check will be conducted on February 13, 2025.
2. The Final Deadline is set on February 18, 2025.