CS2010 (HT) Assignment 2: Largest Palindrome

Problem:
A palindromic number reads the same both ways. The largest palindromic number made from the product of two 2-digit numbers is 9009 = 91 × 99.
Develop an algorithm, which will be able to calculate the largest palindromic number, which is a product of two n-digit numbers. In order to achieve important optimisations, you can assume that n is odd.

Download the attached skeleton classes.

1. Implement reverse method.
   This method takes in an integer and returns its reverse representation,
   e.g. reverse(12345) = 54321.
   You shouldn't convert the number into string and use String APIs.

2. Implement the `isPalindrome` method.

3. Look at the naive implementation of `largestPalindromeThreeDigitNumberProduct` method.
   This method calculates the largest palindrome number, which is a product of 3 digit numbers.

4. Once you understand the `largestPalindromeThreeDigitNumberProduct` method, think of how you could generalise this approach so that it works for products of different number sizes

   a. `largestPalindromeNDigitProduct(3)`
      should return largest palindrome which is a product of two, 3 digit numbers

   b. `largestPalindromeNDigitProduct(5)`
      should return largest palindrome which is a product of two, 5 digit numbers

5. Optimise the method so that it runs in reasonable amount of time (~1s-2s) even for products of 7 digit numbers

6. Test your code by providing additional tests in the NumberPalindromeTest class.

Your implementation should be an efficient solution of the problem as discussed in the lectures. The algorithm should execute on the Web-CAT server in less than 5 seconds.