**Worksheet 7** ___ **Name:** _____

1. Next week, we'll be writing a program to determine prime numbers within a user–entered range.

   (a) What is the smallest prime number? [ ]

   (b) Fill in the table of testcases below with the prime numbers in the range `low` to `high` — use Google:

| *low* | *high* | Expected Results |
|---|---|---|
| 1 | 10 | |
| 50 | 75 | |
| 100 | 110 | |
| 1090 | 1100 | |

   (c) A method, `isPrime()`, which **determines whether a given integer is prime**, is desired.

      i. What should the return type of this method be? [ ]

      ii. How many and what type of arguments should this method have? [ ]

      iii. Give an *algorithm* for determining whether a given integer, $n$, is prime:

   (d) Utilizing the `isPrime()` method from part (c), provide the Java code (`for` loop and body) which **finds and prints all the prime numbers in the positive integer range `low` to `high`**.

2. A `palindromic number` is one which reads the same forwards and backwards, like 12321 or 894498.

   (a) A method, `reverseDigits()`, which takes a positive integer and *returns* the **number** that is its reverse, is desired.

      i. What should the return type of this method be? [ ]

      ii. How many and what type of arguments should this method have? [ ]

      iii. Give an *algorithm* for reversing a given integer, $n$:

(b) A method, `isPalindrome()`, which **determines whether a given integer is a palindrome**, is desired.

i. What should the return type of this method be?

ii. How many and what type of arguments should this method have?

iii. Utilizing `reverseDigits()` from part (a), give an *algorithm* for determining whether a given integer, $n$, is a palindrome:

(c) For each number, continue reversing and adding 2 to the result until you reach a palindrome. Record the palindrome you found and how many steps (reversals) each took.

| Number | Palindrome | Steps Taken |
|---|---|---|
| 103 | | |
| 67 | | |

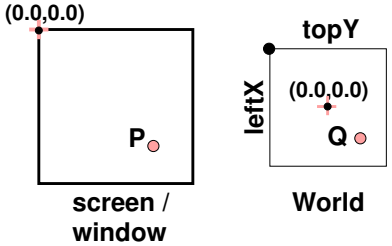
(d) Devise a loop which will continue reversing the integer $n$ and adding 2 to it, until it becomes a palindrome. Make use of both the `reverseDigits()` and `isPalindrome()` methods.

3. Assume that the constants `SCREENSIZE` (width & height of the graphics window, which is square), `BLOCKSIZE` (width and height of blocks to be placed in the window), and `WORLDSIZE`, `WORLDCENTERX` and `WORLDCENTERY` (width & height and center coordinates of a square ***World*** in the Cartesian plane) have been declared and are available.

(a) Given a `row` and `column` (both starting at zero) in a grid of blocks covering the graphics window, give expressions to determine the graphics coordinate of the block in that position:

`block's x =` `block's y =`

(b) We wish to *translate* a point in the window to its *corresponding* point in the cartesian plane. Find formulas to map a point $P$ in the graphics window to its corresponding point $Q$ in the square *World* in the Cartesian plane. Start by devising formulas for `leftX` and `topY` of *World*, then determine the $Q_x$ and $Q_y$ which place point $Q$ in the same relative position within *World* as $P$ is within the window.




`leftX` =

`topY` =

$Q_x =$

$Q_y =$