Examples:

if a is	return	reason
{9, 0, 2, -5, 7}	2	The square pairs are <2, 7> and <7, 9>. Note that <-5, 9> and <0, 9> are not square pairs, even though they sum to perfect squares, because both members of a square pair have to be greater than 0. Also <7,2> and <9,7> are not square pairs because the first number has to be less than the second number.
{9}	0	The array must have at least 2 elements

2. A **prime number** is an integer that is divisible only by 1 and itself. A **porcupine number** is a prime number whose last digit is 9 and the next prime number that follows it also ends with the digit 9. For example 139 is a porcupine number because:

a. it is prime

b. it ends in a 9

c. The next prime number after it is 149 which also ends in 9. Note that 140, 141, 142, 143, 144, 145, 146, 147 and 148 are **not** prime so 149 is the next prime number after 139.

Write a method named *findPorcupineNumber* which takes an integer argument *n* and returns the first porcupine number **that is greater than *n***. So findPorcupineNumber(0) would return 139 (because 139 happens to be the first porcupine number) and so would findPorcupineNumber(138). But findPorcupineNumber(139) would return 409 which is the second porcupine number.

The function signature is

int findPorcupineNumber(int n)

You may assume that a porcupine number greater than *n* exists.

You may assume that a function *isPrime* exists that returns 1 if its argument is prime, otherwise it returns 0. E.G., isPrime(7) returns 1 and isPrime(8) returns 0.

Hint: Use modulo base 10 arithmetic to get last digit of a number.

3. Consider the following algorithm

Start with a positive number n

if n is even then divide by 2