Daily Coding Problem #101

Problem

This problem was asked by Alibaba.

Given an even number (greater than 2), return two prime numbers whose sum will be equal to the given number.

A solution will always exist. See Goldbach's conjecture.

Example:

```
Input: 4
Output: 2 + 2 = 4
```

If there are more than one solution possible, return the lexicographically smaller solution.

If [a, b] is one solution with a <= b, and [c, d] is another solution with c <= d, then

```
[a, b] < [c, d]
```

If a < c OR a==c AND b < d.

Solution

We can search for the smallest solution by iterating through possible values of the first potential prime. We check whether the first and second numbers are prime, and if so, return them.

```
private static boolean isPrime(int n) {
```

```
    for (int i = 2; i < Math.sqrt(n); i++)
        if (n % i == 0)
            return false;
    return true;
}

public ArrayList<Integer> primesum(int a) {
    for (int i = 2; i <= a / 2; i++) {
        if (isPrime(i) && isPrime(a - i)) {
            ArrayList<Integer> output = new ArrayList<>();
            output.add(i);
            output.add(a - i);
            return output;
        }
    }

    return null;
}
```

```
def primesum(self, n):
    for i in xrange(2, n):
        if self.is_prime(i) and self.is_prime(n - i):
            return i, n - i

def is_prime(self, n):
    if n < 2:
        return False

    for i in xrange(2, int(n**0.5) + 1):
        if n % i == 0:
            return False

    return True
```