1. Counting Operations

To find the Big-O runtime of a program, we take two steps:

(1) Count the number of operations, in terms of n.

(2) Remove all multiplicative constants, and keep only the highest-order term.

Here are some examples of step (2):

$5n+3$	$\longrightarrow$	$O(n)$	drop the $+3$, then drop the 5
$100n+2$	$\longrightarrow$	$O(n)$	a big constant is still just a constant
$2n^2+50n+7$	$\longrightarrow$	$O(n^2)$	n^2 is the highest-order term
$\frac{1}{2}n^2$	$\longrightarrow$	$O(n^2)$	the constant can be a fraction
8	$\longrightarrow$	$O(1)$	no n at all

a. Try these two yourself:

$6n^2+300n \rightarrow O(\_\_\_)$ $\quad 2.5n \rightarrow O(\_\_\_)$ $\quad 9n^3+100n^2+7n \rightarrow O(\_\_\_)$ $\quad 12 \rightarrow O(\_\_\_)$

b. Now we will count the operations in a real program. The function below takes a list `lst` of n numbers.

```
def sum_doubles(lst):
    total = 0
    for i in range(len(lst)):
        total = total + lst[i] * 2
    return total
```

How many times does the body of the `for` loop run, in terms of n?

c. Count each kind of operation in the whole function. Your answers should be in terms of n. (Do not forget the assignment on the line `total = 0`.)

total $=$ _____ assignments (=) $+$ _____ multiplies (*) $+$ _____ additions (+)

Now add them up:

total $=$ _____ $\cdot n +$ _____

d. Apply step (2) to your answer from part **c**: drop the multiplicative constant and the lower-order term. What is the Big-O runtime?

$O($ _____ $)$

2. Reading Loops

For each program below, first write how many times the `print` line runs, and then write the Big-O runtime in terms of n where `n = len(lst)`.

a.

```
for i in range(n):
    for j in range(n):
        print(i, j)
```

`print` runs _____ times. Runtime: $O($ _____ $)$

b.

```
for i in range(n):
    print(i)

for j in range(n):
    print(j)
```

`print` runs _____ times. Runtime: $O($ _____ $)$

c.

```
i = 1
while i < n:
    print(i)
    i = i * 2
```

Hint: write out the values that `i` takes when `n = 16`:

1, _____, _____, _____, stop

`print` runs _____ times. Runtime: $O($ _____ $)$