Exercise

```python
def ave_positives(my_list):
    """Returns: average (float) of the positive values in my_list
my_list: a list of numbers with at least one positive value"
```

- Be goal oriented → can work backwards
- Name a variable for any value that you need but don't have yet
- Break down a problem!
  - ... break into parts
  - ... solve simpler version first
- Remember loop/accumulation pattern

Exercise (Example Solution)

```python
def ave_positives(my_list):
    """Returns: average (float) of the positive values in my_list
my_list: a list of numbers with at least one positive value"
```

```python
result = 0
count = 0
for x in my_list:
    if x > 0:
        result = result + x
        count = count + 1
ave = result / count
return ave
```

For-Loop Mistake #1 (Q)

Modifying the loop variable (here: x).

```python
def add_one(the_list):
    """Adds 1 to every element in the list
Precondition: the_list is a list of all numbers
(either floats or ints)"
    for x in the_list:
        x = x+1
```

```text
a = [5, 4, 7]
add_one(a)
print(a)
```

- A: [5, 4, 7]
- B: [5, 4, 7, 5, 4, 7]
- C: [6, 5, 8]
- D: Error
- E: I don’t know

For-Loop Mistake #1 (A)

Modifying the loop variable (here: x).

```python
def add_one(the_list):
    """Adds 1 to every element in the list
Precondition: the_list is a list of all numbers
(either floats or ints)"
    for x in the_list:
        x = x+1
```

```text
grades = [5, 4, 7]
add_one(grades)
```

- A: [5, 4, 7]  CORRECT
- B: [5, 4, 7, 5, 4, 7]
- C: [6, 5, 8]
- D: Error
- E: I don’t know

Modifying the Loop Variable (1)

```python
def add_one(the_list):
    """Adds 1 to every elt
Pre: the_list is all numb."
    for x in the_list:
        x = x+1
```

```text
grades = [5, 4, 7]
add_one(grades)
```

Call Frame

Global Space

Heap Space

Modifying the Loop Variable (2)

```python
def add_one(the_list):
    """Adds 1 to every elt
Pre: the_list is all numb."
    for x in the_list:
        x = x+1
```

```text
grades = [5, 4, 7]
add_one(grades)
```

Call Frame
Modifying the Loop Variable (3)

```python
def add_one(the_list):
    """Adds 1 to everyelt
    Pre: the_list is all numb."""
    for x in the_list:
        x = x + 1
grades = [5, 4, 7]
add_one(grades)
```

Modifying the Loop Variable (4)

```python
def add_one(the_list):
    """Adds 1 to everyelt
    Pre: the_list is all numb."""
    for x in the_list:
        x = x + 1
grades = [5, 4, 7]
add_one(grades)
```

Modifying the Loop Variable (5)

```python
def add_one(the_list):
    """Adds 1 to everyelt
    Pre: the_list is all numb."""
    for x in the_list:
        x = x + 1
grades = [5, 4, 7]
add_one(grades)
```

Modifying the Loop Variable (6)

```python
def add_one(the_list):
    """Adds 1 to everyelt
    Pre: the_list is all numb."""
    for x in the_list:
        x = x + 1
grades = [5, 4, 7]
add_one(grades)
```

Modifying the Loop Variable (7)

```python
def add_one(the_list):
    """Adds 1 to everyelt
    Pre: the_list is all numb."""
    for x in the_list:
        x = x + 1
grades = [5, 4, 7]
add_one(grades)
```

Modifying the Loop Variable (8)

```python
def add_one(the_list):
    """Adds 1 to everyelt
    Pre: the_list is all numb."""
    for x in the_list:
        x = x + 1
grades = [5, 4, 7]
add_one(grades)
```
Modifying the Loop Variable (9)

```python
def add_one(the_list):
    """Adds 1 to every elt
    Pre: the_list is all numb."
    for x in the_list:
        x = x+1
grades = [5, 4, 7]
add_one(grades)
```

For-Loop Mistake #2 (Q)

Modifying the loop sequence as you walk through it.

What gets printed?

```python
b = [1, 2, 3]
for a in b:
    b.append(a)
p
``` 

A: never prints b  
B: [1, 2, 3, 1, 2, 3]  
C: [1, 2, 3]  
D: I do not know

For-Loop Mistake #2 (A)

Modifying the loop sequence as you walk through it.

What gets printed?

```python
b = [1, 2, 3]
for a in b:  
    b.append(a)  
p
``` 

A: never prints b  
CORRECT*  
B: [1, 2, 3, 1, 2, 3]  
C: [1, 2, 3]  
D: I do not know

* Runs out of memory eventually, then probably throws an error.