Exercise

```
def ave_positives(my_list):
  """Returns: avg (float) of positive values in my_list
     my_list: a list of #s with at least 1 positive value
  """
```

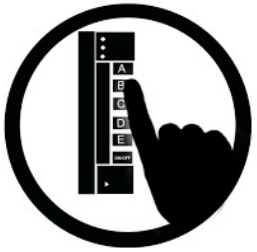
- Be goal oriented → ***can work backwards***
- ***Name a variable*** for any value that you need but don't have yet
- Break down a problem!
  - … ***break into parts***
  - … ***solve simpler version first***
- Remember loop/accumulation pattern

Exercise (Example Solution)

```
def avg_positives(my_list):
    """Returns: avg (float) of positive values in my_list
     my_list: a list of #s with at least 1 positive value
    """
    result= 0
    count= 0
    for x in my_list:
        if x > 0:
            result= result + x
            count= count + 1
    avg= result/count
    return avg
```

- 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

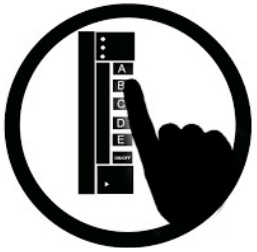
What gets printed? (Q)




```
t= 0
for k in range(5, 1, -1):
    t = t + 1
print(t)
```

A: 0
B: 2
C: 3
D: 4
E: 5

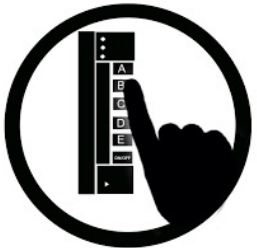
What gets printed? (A)



```
t= 0
for k in range(5, 1, -1):
    t = t + 1
print(t)
```

A: 0
B: 2
C: 3
D: 4 CORRECT
E: 5

For-Loop Mistake #1 (Q)



Modifying the loop variable (here: x).

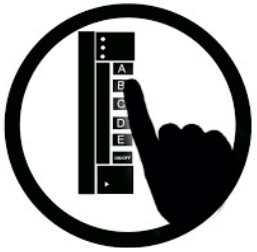
```
def add_one(the_list):
    """Adds 1 to every element in the list
    Precondition: the_list is a list of all numbers
    (either floats or ints)"""
    for x in the_list:
        x = x+1

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

What gets printed?

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).

```
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
```

Actually it does not do this!

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

What gets printed?

A: [5, 4, 7] CORRECT

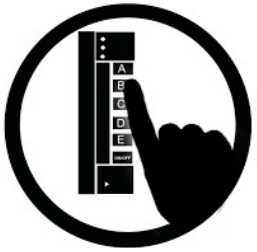
B: [5, 4, 7, 5, 4, 7]

C: [6, 5, 8]

D: Error

E: I don't know

For-Loop Mistake #2 (Q)



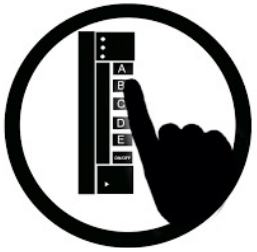
Modifying the loop sequence as you walk through it.

What gets printed?

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

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?

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

INFINITE LOOP!

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.