```
########################
## EXAMPLE: returning a tuple
########################
def
quotient_and_remainder(x, y):
    q = x // y
    r = x % y
    return (q, r)

(quot, rem) =
quotient_and_remainder(5,3)
print(quot)
print(rem)


########################
## EXAMPLE:
iterating over tuples
########################
def get_data(aTuple):
    """

 aTuple, tuple of tuples (int, string)
    Extracts all integers from aTuple and sets
    them
as elements in a new tuple.
    Extracts all unique strings from from aTuple
    and sets
them as elements in a new tuple.
    Returns a tuple of the minimum integer, the
    maximum
integer, and the number of unique strings
    """
    nums = ()    # empty
tuple
    words = ()
    for t in aTuple:
        # concatenating with a singleton tuple

 nums = nums + (t[0],)
        # only add words haven't added before
        if t[1] not in
words:
            words = words + (t[1],)
    min_n = min(nums)
    max_n = max(nums)

unique_words = len(words)
    return (min_n, max_n, unique_words)

test =
((1,"a"),(2, "b"),
        (1,"a"),(7,"b"))
(a, b, c) =
get_data(test)
print("a:",a,"b:",b,"c:",c)

# apply to any data
you want!
tswift = ((2014,"Katy"),
          (2014, "Harry"),

(2012,"Jake"),
          (2010,"Taylor"),

(2008,"Joe"))
(min_year, max_year, num_people) =
get_data(tswift)
print("From", min_year, "to", max_year, \

"Taylor Swift wrote songs about", num_people,
"people!")

########################
```

```
## EXAMPLE: sum of elements in a
list
#########################
def sum_elem_method1(L):
  total = 0
  for i in range(len(L)):

      total += L[i]
  return total

def sum_elem_method2(L):
    total = 0
    for i in L:

        total += i
    return total


print(sum_elem_method1([1,2,3,4]))
print(sum_elem_method2([1,2,3,4]))


######################
###
## EXAMPLE: various list operations
## put print(L) at different locations to see how it
gets mutated
#########################
L1 = [2,1,3]
L2 = [4,5,6]
L3 = L1 +
L2
L1.extend([0,6])

L = [2,1,3,6,3,7,0]
L.remove(2)
L.remove(3)
del(L[1])
print(L.pop())

s =
"I<3 cs"
print(list(s))
print(s.split('<'))
L = ['a', 'b',
'c']
print(''.join(L))
print('_'.join(L))

L=[9,6,0,3]
print(sorted(L))
L.sort()
L.reverse()



#########################
## EXAMPLE: aliasing
#########################
a = 1
b =
a
print(a)
print(b)

warm = ['red', 'yellow', 'orange']
hot =
warm
hot.append('pink')
print(hot)
print(warm)

#########################
## EXAMPLE:
cloning
```

```
#########################
cool = ['blue', 'green', 'grey']
chill =
cool[:]
chill.append('black')
print(chill)
print(cool)

#########################
## EXAMPLE:
sorting with/without mutation
#########################
warm = ['red', 'yellow',
'orange']
sortedwarm = warm.sort()
print(warm)
print(sortedwarm)

cool = ['grey', 'green',
'blue']
sortedcool = sorted(cool)
print(cool)
print(sortedcool)

#########################
##
EXAMPLE: lists of lists of lists...
#########################
warm = ['yellow', 'orange']
hot =
['red']
brightcolors =
[warm]
brightcolors.append(hot)
print(brightcolors)
hot.append('pink')
print(hot)
print(brightc
olors)


################################
## EXAMPLE: mutating a list while iterating over
it
################################
def remove_dups(L1, L2):
    for e in L1:
        if e in
L2:
            L1.remove(e)

def remove_dups_new(L1, L2):
    L1_copy = L1[:]
    for e
in L1_copy:
        if e in L2:
            L1.remove(e)

L1 = [1, 2, 3, 4]
L2 = [1, 2, 5,
6]
remove_dups(L1, L2)
print(L1, L2)

L1 = [1, 2, 3, 4]
L2 = [1, 2, 5, 6]
remove_dups_new(L1,
L2)
print(L1, L2)

################################
## EXERCISE: Test yourself by predicting
what the output is and
##           what gets mutated then check with the Python
Tutor
```

```
##############################
cool = ['blue', 'green']
warm = ['red', 'yellow',
'orange']
print(cool)
print(warm)

colors1 =
[cool]
print(colors1)
colors1.append(warm)
print('colors1 = ', colors1)

colors2 = [['blue',
'green'],
          ['red', 'yellow', 'orange']]
print('colors2 =',
colors2)

warm.remove('red')
print('colors1 = ', colors1)
print('colors2 =', colors2)

for e
in colors1:
    print('e =', e)

for e in colors1:
    if type(e) == list:
        for e1 in
e:
            print(e1)
    else:
        print(e)

flat = cool + warm
print('flat =',
flat)

print(flat.sort())
print('flat =', flat)

new_flat = sorted(flat, reverse =
True)
print('flat =', flat)
print('new_flat =', new_flat)

cool[1] =
'black'
print(cool)
print(colors1)
```