## Code:

*#365 Lab2*

*#Cache Angus, 20000629*

**def** findSum(arrayinput, left, current, right):

sm =0

lsm = -10000 *#try and garuntee that the first number is greater than this original number*

l\_index\_pos = 0 *#initalize an index variable*

for i in range(current, left-1, -1): *#start at your midpoint and move to the left*

sm = sm + arrayinput[i] *#add your new value to the sum*

if (sm > lsm): *#if the sum is greater than your old left-side sum, the left-side sum because the new value*

lsm = sm

l\_index\_pos = i *#replace the left index value*

sm = 0

rsm = -1000

right\_index\_pos = 0

for i in range(current+1, right+1):

sm = sm + arrayinput[i]

if (sm > rsm):

rsm = sm

right\_index\_pos = i+1 *#the right index is +1 because of the length of the index*

return lsm + rsm, (l\_index\_pos, right\_index\_pos) *#use a tuple to return the indexes so they can be called with the function*

**def** actualMaxArrayVal(arrayinput, left, right):

current = (left + right) // 2 *#this truncates the value so it is the middle*

*#if there is no max sum then just print the biggest number*

if(left == right) :

return arrayinput[left], (left, right)

*# do the recursion with your new left and rights continuously, then find the sum of these left and rights and current*

return max(actualMaxArrayVal(arrayinput, left, current),

actualMaxArrayVal(arrayinput, current+1, right),

findSum(arrayinput, left, current, right))

*#there seems to be an issue when testing it with replacing values look into that*

inputArray = [1, 19 ,-12, 2, -18]

print("inputArray: ", inputArray)

n = len(inputArray) *#find the length for the first input*

maximal, maximal\_array = actualMaxArrayVal(inputArray, 0, n-1)

print("The maximal value is: ", maximal)

array = []

if (maximal\_array[0] == maximal\_array[1]): *#prints out only the one value if there is one value for the maximal*

array.insert(0, inputArray[maximal\_array[0]])

for i in range(maximal\_array[0], maximal\_array[1]): *#get the range from the tuple and put it into a new subarray*

array.insert(i, inputArray[i])

inputArray[i] = -1000 *#replace the subarray*

*#^the above line is part of the additions for part 2*

print("The maximal subarray is: ", array)

#The below section is the additions for part 2

maximal, maximal\_array = actualMaxArrayVal(inputArray, 0, n-1) *#call again for the new input*

array = []

print("The new maximal value is: ", maximal)

if (maximal\_array[0] == maximal\_array[1]): *#prints out only the one value if there is one value for the maximal*

array.insert(0, inputArray[maximal\_array[0]])

for i in range(maximal\_array[0], maximal\_array[1]): *#get the range from the tuple and put it into a new subarray*

array.insert(i, inputArray[i])

print("The new maximal subarray is: ", array)

## Test Results:

1. inputArray: [1, 19, -12, 2, -18]  
   The maximal value is: 20  
   The maximal subarray is: [1, 19]  
   The new maximal value is: 2  
   The new maximal subarray is: [2]
2. inputArray: [-1, 19, 12, -2]  
   The maximal value is: 31  
   The maximal subarray is: [19, 12]  
   The new maximal value is: -1  
   The new maximal subarray is: [-1]
3. inputArray: [1, 19, 12, 2]  
   The maximal value is: 34  
   The maximal subarray is: [1, 19, 12, 2]  
   The new maximal value is: -1000  
   The new maximal subarray is: [-1000]
4. inputArray: [1, 2, 4, -4, 2, -10, 3, 4]  
   The maximal value is: 7  
   The maximal subarray is: [3, 4]  
   The new maximal value is: 7  
   The new maximal subarray is: [1, 2, 4]
5. inputArray: [-1, -1, -3]  
   The maximal value is: -1  
   The maximal subarray is: [-1]  
   The new maximal value is: -1  
   The new maximal subarray is: [-1]