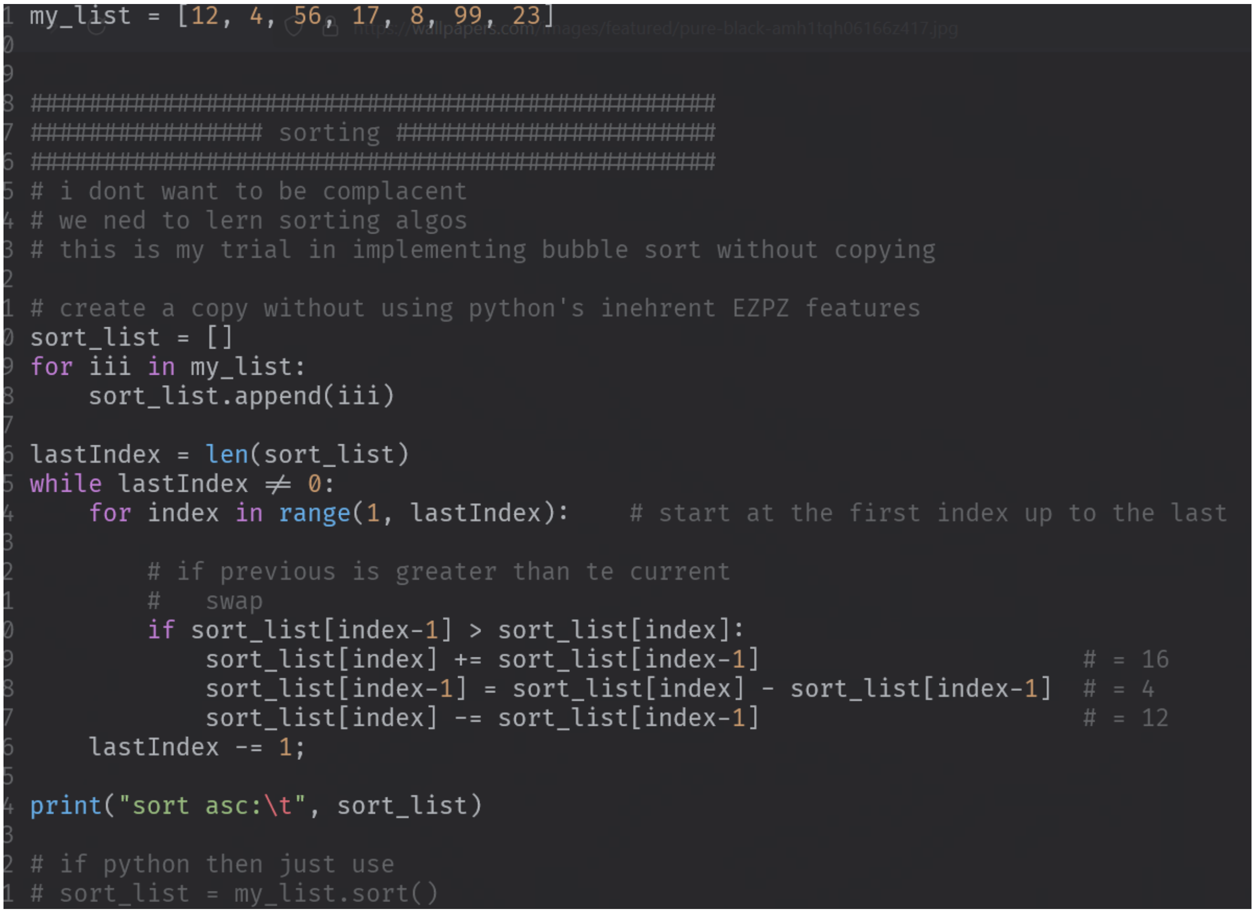
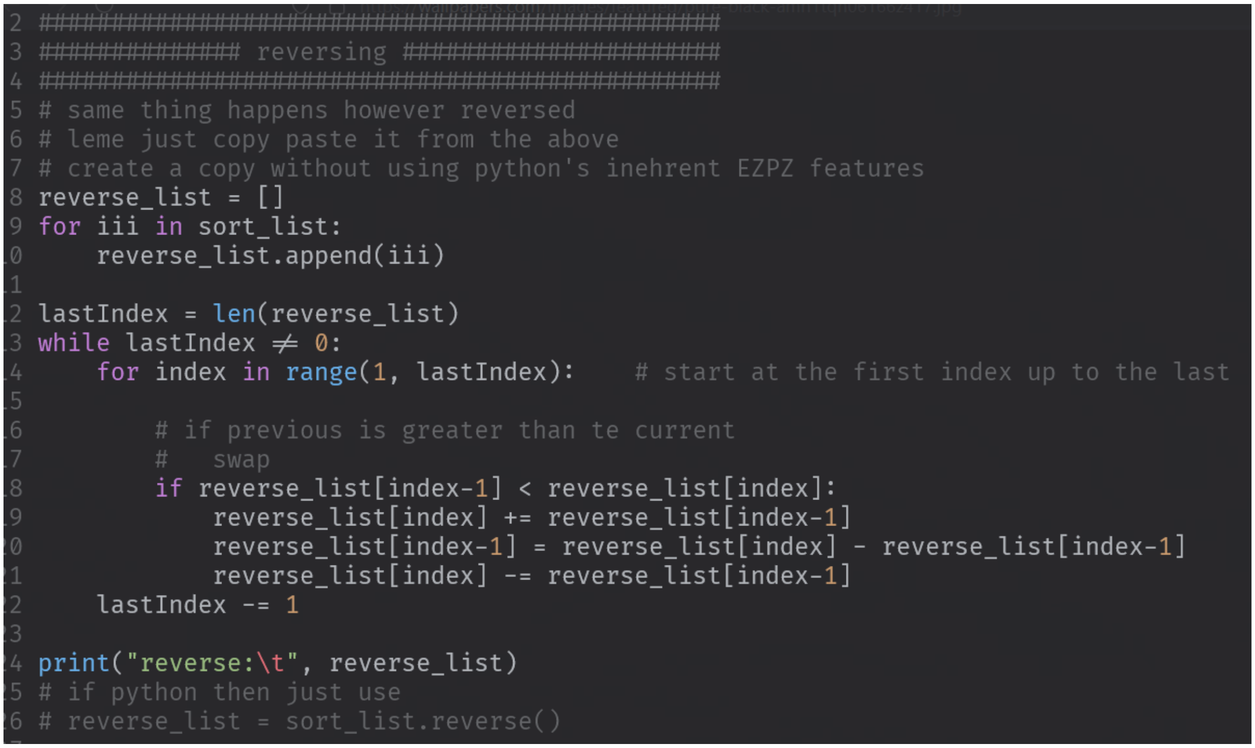
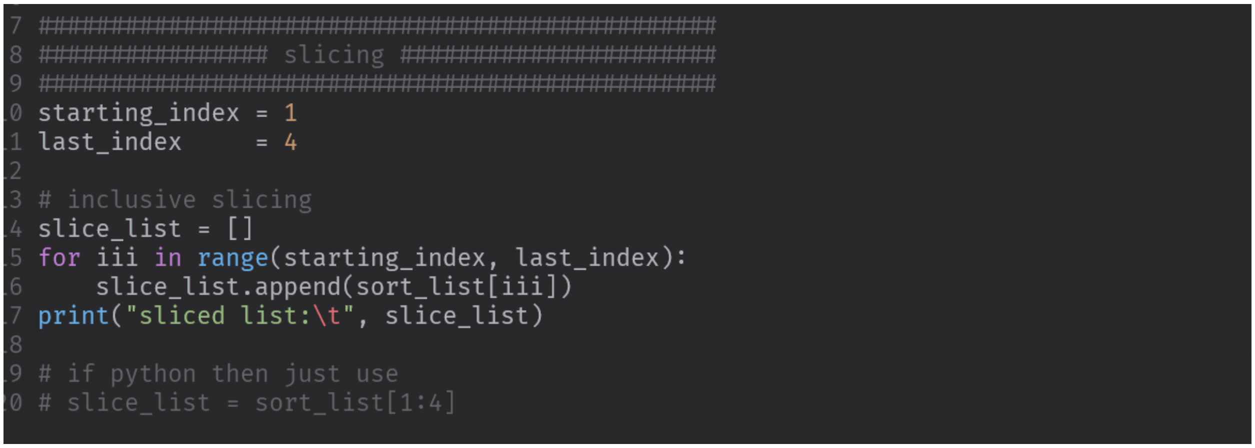
Activity 3:



```python
my_list = [12, 4, 56, 17, 8, 99, 23]


###############################################
################ sorting ######################
###############################################
# i dont want to be complacent
# we ned to lern sorting algos
# this is my trial in implementing bubble sort without copying

# create a copy without using python's inehrent EZPZ features
sort_list = []
for iii in my_list:
    sort_list.append(iii)

lastIndex = len(sort_list)
while lastIndex != 0:
    for index in range(1, lastIndex):    # start at the first index up to the last

        # if previous is greater than te current
        #   swap
        if sort_list[index-1] > sort_list[index]:
            sort_list[index] += sort_list[index-1]                      # = 16
            sort_list[index-1] = sort_list[index] - sort_list[index-1]  # = 4
            sort_list[index] -= sort_list[index-1]                      # = 12
    lastIndex -= 1;

print("sort asc:\t", sort_list)

# if python then just use
# sort_list = my_list.sort()
```

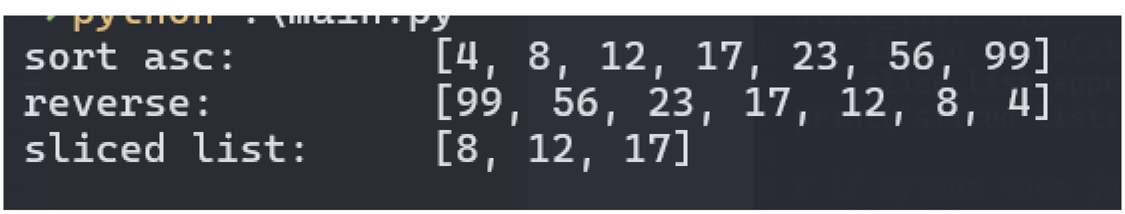


```
############################################
############## reversing ###################
############################################
# same thing happens however reversed
# leme just copy paste it from the above
# create a copy without using python's inehrent EZPZ features
reverse_list = []
for iii in sort_list:
    reverse_list.append(iii)

lastIndex = len(reverse_list)
while lastIndex ≠ 0:
    for index in range(1, lastIndex):    # start at the first index up to the last

        # if previous is greater than te current
        #   swap
        if reverse_list[index-1] < reverse_list[index]:
            reverse_list[index] += reverse_list[index-1]
            reverse_list[index-1] = reverse_list[index] - reverse_list[index-1]
            reverse_list[index] -= reverse_list[index-1]
    lastIndex -= 1

print("reverse:\t", reverse_list)
# if python then just use
# reverse_list = sort_list.reverse()
```



```
############################################
################ slicing ###################
############################################
starting_index = 1
last_index     = 4

# inclusive slicing
slice_list = []
for iii in range(starting_index, last_index):
    slice_list.append(sort_list[iii])
print("sliced list:\t", slice_list)

# if python then just use
# slice_list = sort_list[1:4]
```

OUTPUT:

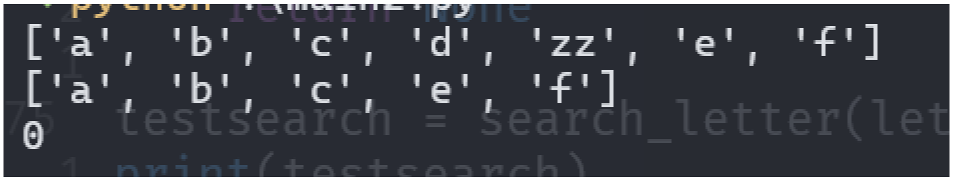


```
python .\main.py
sort asc:         [4, 8, 12, 17, 23, 56, 99]
reverse:          [99, 56, 23, 17, 12, 8, 4]
sliced list:      [8, 12, 17]
```

Activity 4

```
letters = ['a', 'b', 'c', 'd', 'e', 'f']


def insert_letter(listlist, letter=None, index=0):
    # i dont wnat to use the insert built-in function
    # concept:
    # 1.    divide the array (1 copy left half and 1 copy to the right half)
    # 2.    then insert whatatevber to the left of half at the last elemnet
    # 3.    return the (join the left half and right half)

    # create a copy of the left half and right half
    # hehe tinatamad na me
    leftHalf = listlist[0:index]
    rightHalf = listlist[index:]    # 1

    leftHalf.append(letter)         # 2

    leftHalf.extend(rightHalf)      # 3
    return leftHalf


testinsert = insert_letter(letters, "zz", 4)
print(testinsert)

def remove_letter(listlist, letter):
    # concept
    # have an another list
    # just loop through the listlist and dont include letter to the another
list
    tempCache = []
    for iii in listlist:
        if iii == letter:
            continue
        tempCache.append(iii)
    return tempCache


testremove = remove_letter(letters, 'd')
print(testremove)


def search_letter(listlist, letter = None):
    for iii in range(len(listlist)):
        if letter == listlist[iii]:
            return iii
    print("not found")
    return None

testsearch = search_letter(letters, 'a')
print(testsearch)
```

OUTPUT:



```
['a', 'b', 'c', 'd', 'zz', 'e', 'f']
['a', 'b', 'c', 'e', 'f']
0
```