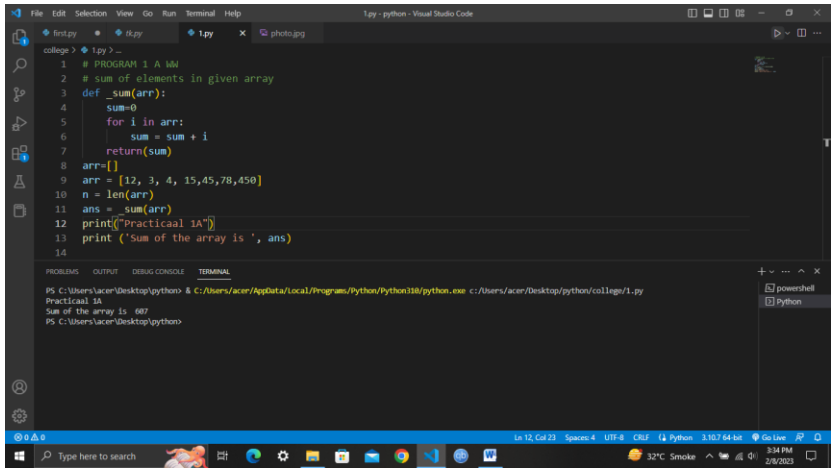
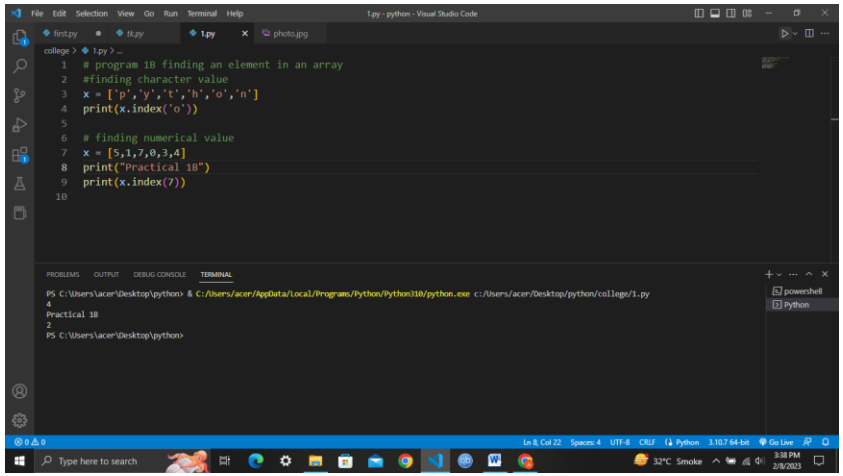
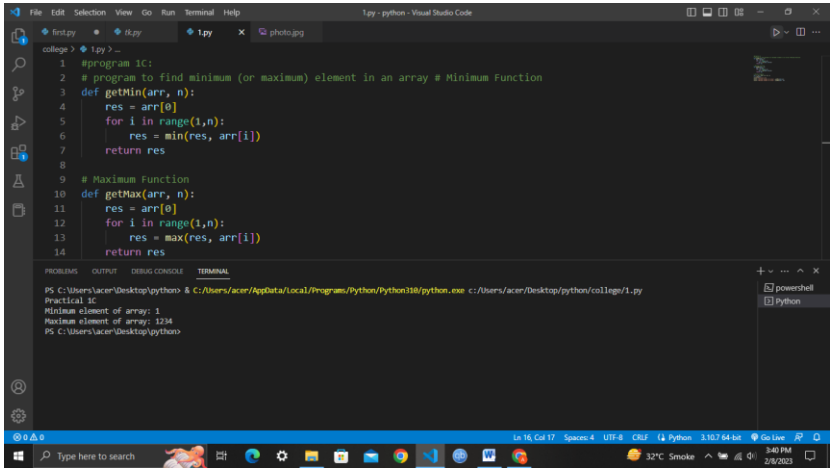
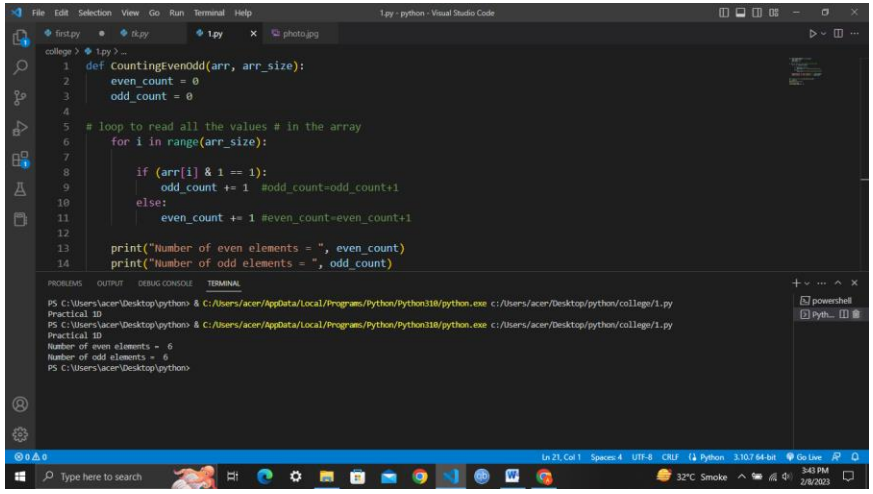
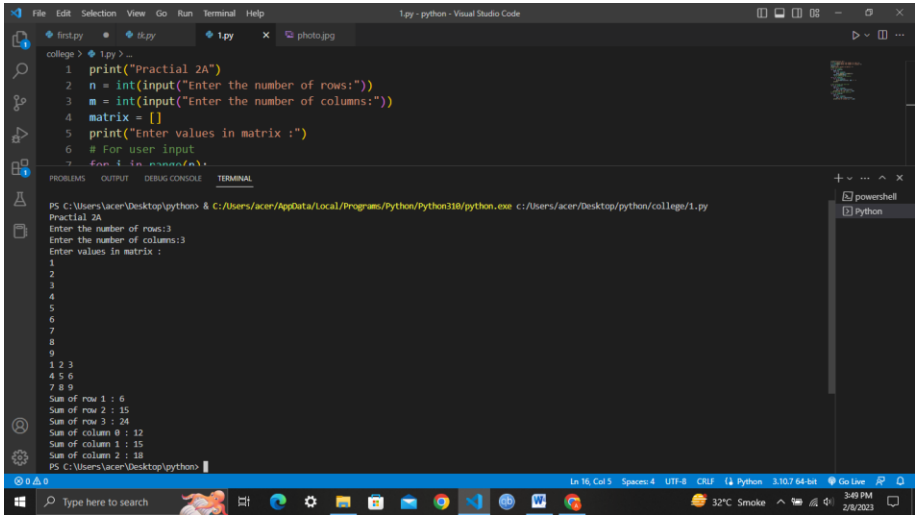

1.py - python - Visual Studio Code
# PROGRAM 1 A WW
# sum of elements in given array
def _sum(arr):
sum=0
for i in arr:
sum = sum + i
return(sum)
arr=[]
arr = [12, 3, 4, 15,45,78,450]
n = len(arr)
ans = _sum(arr)
print("Practicaal 1A")
print ('Sum of the array is ', ans)
PS C:\Users\acer\Desktop\python> & C:/Users/acer/AppData/Local/Programs/Python/Python310/python.exe c:/Users/acer/Desktop/python/college/1.py
Practicaal 1A
Sum of the array is 607
PS C:\Users\acer\Desktop\python>


1.py - python - Visual Studio Code
# program 1B finding an element in an array
#finding character value
x = ['p','y','t','h','o','n']
print(x.index('o'))
# finding numerical value
x = [5,1,7,0,3,4]
print("Practical 1B")
print(x.index(7))
PS C:\Users\acer\Desktop\python> & C:/Users/acer/AppData/Local/Programs/Python/Python310/python.exe c:/Users/acer/Desktop/python/college/1.py
4
Practical 1B
2
PS C:\Users\acer\Desktop\python>

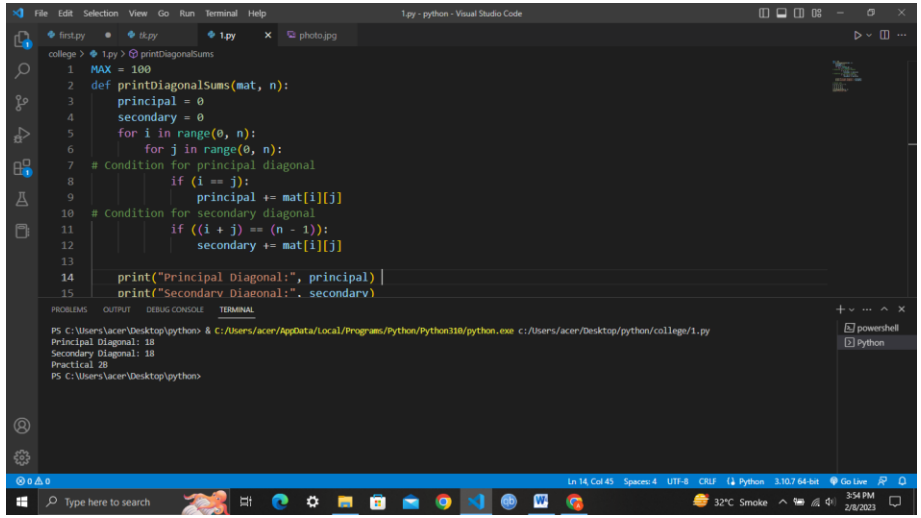

1.py - python - Visual Studio Code
#program 1C:
# program to find minimum (or maximum) element in an array # Minimum Function
def getMin(arr, n):
res = arr[0]
for i in range(1,n):
res = min(res, arr[i])
return res
# Maximum Function
def getMax(arr, n):
res = arr[0]
for i in range(1,n):
res = max(res, arr[i])
return res
PS C:\Users\acer\Desktop\python> & C:/Users/acer/AppData/Local/Programs/Python/Python310/python.exe c:/Users/acer/Desktop/python/college/1.py
Practical 1C
Minimum element of array: 1
Maximum element of array: 1234
PS C:\Users\acer\Desktop\python>


1.py - python - Visual Studio Code
def CountingEvenOdd(arr, arr_size):
even_count = 0
odd_count = 0
# loop to read all the values # in the array
for i in range(arr_size):
if (arr[i] & 1 == 1):
odd_count += 1 #odd_count=odd_count+1
else:
even_count += 1 #even_count=even_count+1
print("Number of even elements = ", even_count)
print("Number of odd elements = ", odd_count)
PS C:\Users\acer\Desktop\python> & C:/Users/acer/AppData/Local/Programs/Python/Python310/python.exe c:/Users/acer/Desktop/python/college/1.py
Practical 1D
PS C:\Users\acer\Desktop\python> & C:/Users/acer/AppData/Local/Programs/Python/Python310/python.exe c:/Users/acer/Desktop/python/college/1.py
Practical 1D
Number of even elements = 6
Number of odd elements = 6
PS C:\Users\acer\Desktop\python>



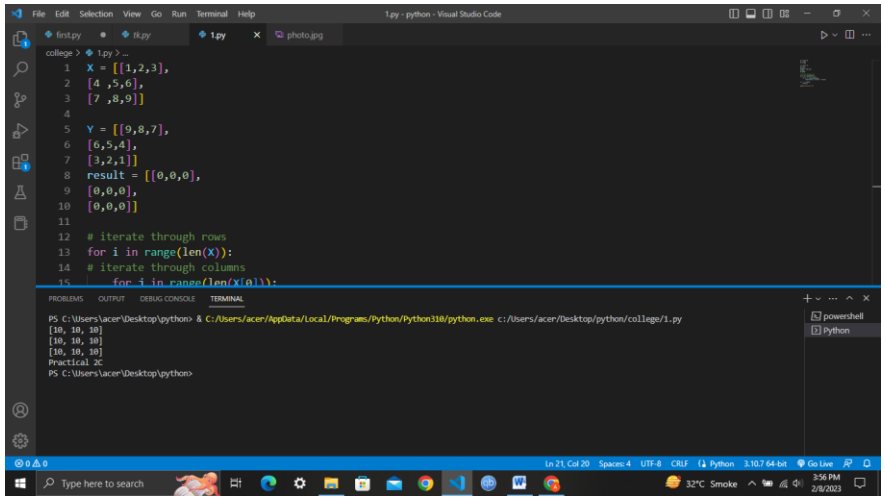
```
print("Practial 2A")
n = int(input("Enter the number of rows:"))
m = int(input("Enter the number of columns:"))
matrix = []
print("Enter values in matrix :")
# For user input
```

```
PS C:\Users\acer\Desktop\python> & C:/Users/acer/AppData/Local/Programs/Python/Python310/python.exe c:/Users/acer/Desktop/python/college/1.py
Practial 2A
Enter the number of rows:3
Enter the number of columns:3
Enter values in matrix :
1
2
3
4
5
6
7
8
9
1 2 3
4 5 6
7 8 9
Sum of row 1 : 6
Sum of row 2 : 15
Sum of row 3 : 24
Sum of column 0 : 12
Sum of column 1 : 15
Sum of column 2 : 18
PS C:\Users\acer\Desktop\python>
```



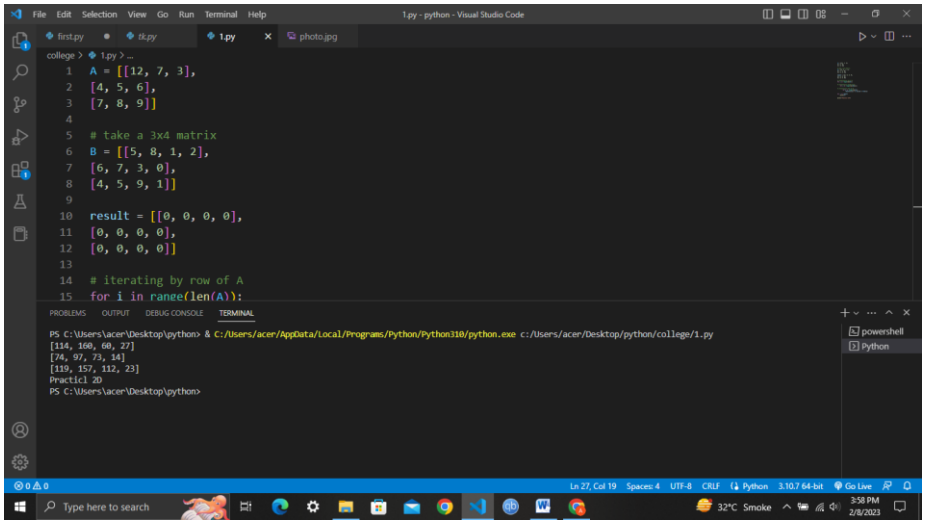
```
MAX = 100
def printDiagonalSums(mat, n):
    principal = 0
    secondary = 0
    for i in range(0, n):
        for j in range(0, n):
# Condition for principal diagonal
            if (i == j):
                principal += mat[i][j]
# Condition for secondary diagonal
            if ((i + j) == (n - 1)):
                secondary += mat[i][j]

    print("Principal Diagonal:", principal)
    print("Secondary Diagonal:", secondary)
```

```
PS C:\Users\acer\Desktop\python> & C:/Users/acer/AppData/Local/Programs/Python/Python310/python.exe c:/Users/acer/Desktop/python/college/1.py
Principal Diagonal: 18
Secondary Diagonal: 18
Practical 2B
PS C:\Users\acer\Desktop\python>
```



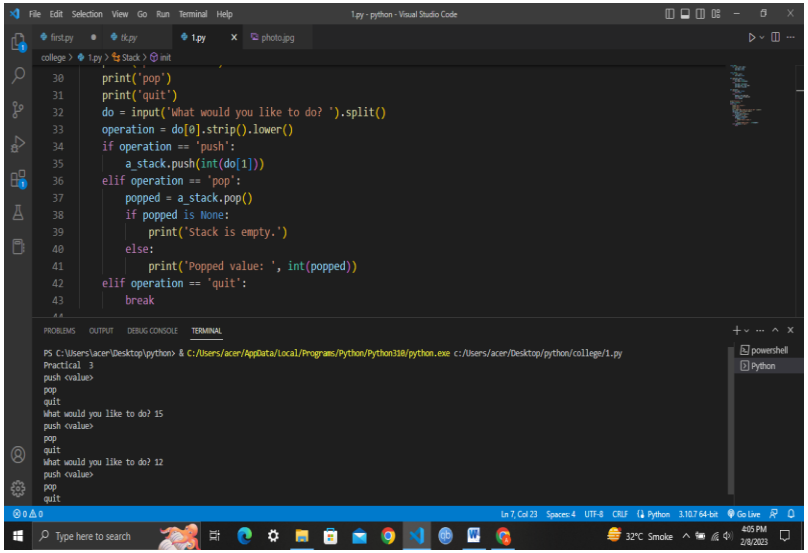
```
X = [[1,2,3],
[4 ,5,6],
[7 ,8,9]]

Y = [[9,8,7],
[6,5,4],
[3,2,1]]
result = [[0,0,0],
[0,0,0],
[0,0,0]]

# iterate through rows
for i in range(len(X)):
# iterate through columns
    for j in range(len(X[0])):
```

```
PS C:\Users\acer\Desktop\python> & C:/Users/acer/AppData/Local/Programs/Python/Python310/python.exe c:/Users/acer/Desktop/python/college/1.py
[10, 10, 10]
[10, 10, 10]
[10, 10, 10]
Practical 2C
PS C:\Users\acer\Desktop\python>
```



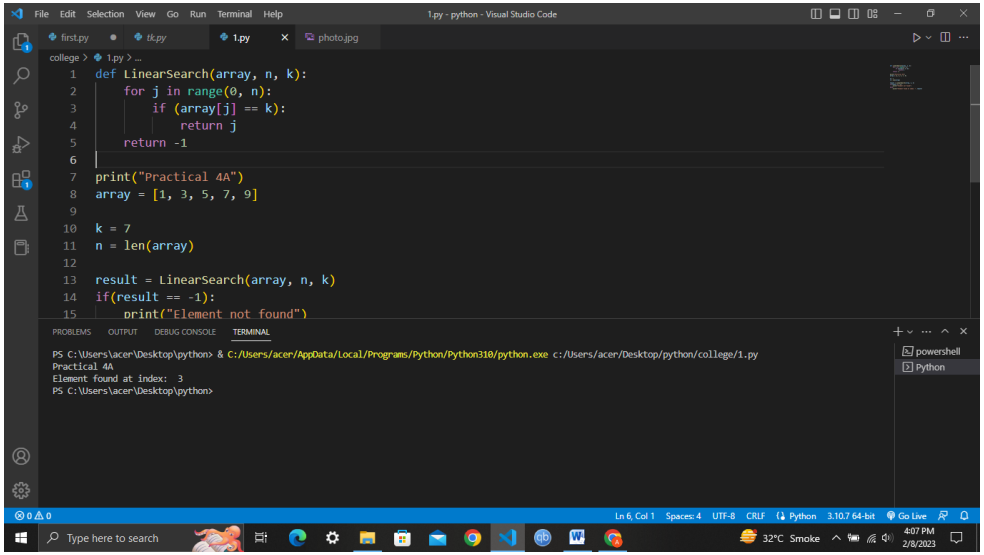
```
A = [[12, 7, 3],
[4, 5, 6],
[7, 8, 9]]

# take a 3x4 matrix
B = [[5, 8, 1, 2],
[6, 7, 3, 0],
[4, 5, 9, 1]]

result = [[0, 0, 0, 0],
[0, 0, 0, 0],
[0, 0, 0, 0]]

# iterating by row of A
for i in range(len(A)):
```

```
PS C:\Users\acer\Desktop\python> & C:/Users/acer/AppData/Local/Programs/Python/Python310/python.exe c:/Users/acer/Desktop/python/college/1.py
[114, 160, 60, 27]
[74, 97, 73, 14]
[119, 157, 112, 23]
Practicl 2D
PS C:\Users\acer\Desktop\python>
```



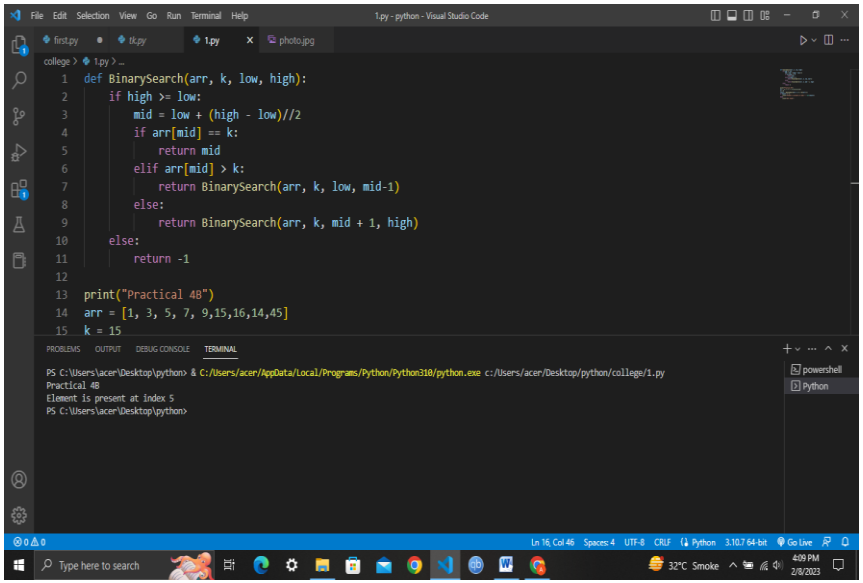
```
    print('pop')
    print('quit')
    do = input('What would you like to do? ').split()
    operation = do[0].strip().lower()
    if operation == 'push':
        a_stack.push(int(do[1]))
    elif operation == 'pop':
        popped = a_stack.pop()
        if popped is None:
            print('Stack is empty.')
        else:
            print('Popped value: ', int(popped))
    elif operation == 'quit':
        break
```

```
PS C:\Users\acer\Desktop\python> & C:/Users/acer/AppData/Local/Programs/Python/Python310/python.exe c:/Users/acer/Desktop/python/college/1.py
Practical 3
push <value>
pop
quit
What would you like to do? 15
push <value>
pop
quit
What would you like to do? 12
push <value>
pop
quit
```



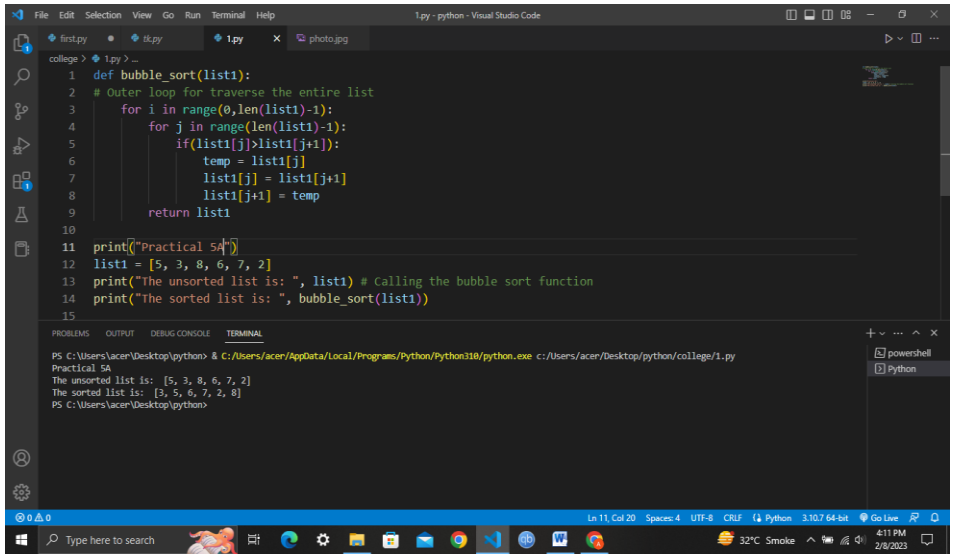
```
def LinearSearch(array, n, k):
    for j in range(0, n):
        if (array[j] == k):
            return j
    return -1

print("Practical 4A")
array = [1, 3, 5, 7, 9]

k = 7
n = len(array)

result = LinearSearch(array, n, k)
if(result == -1):
    print("Element not found")
```

```
PS C:\Users\acer\Desktop\python> & C:/Users/acer/AppData/Local/Programs/Python/Python310/python.exe c:/Users/acer/Desktop/python/college/1.py
Practical 4A
Element found at index:  3
PS C:\Users\acer\Desktop\python>
```



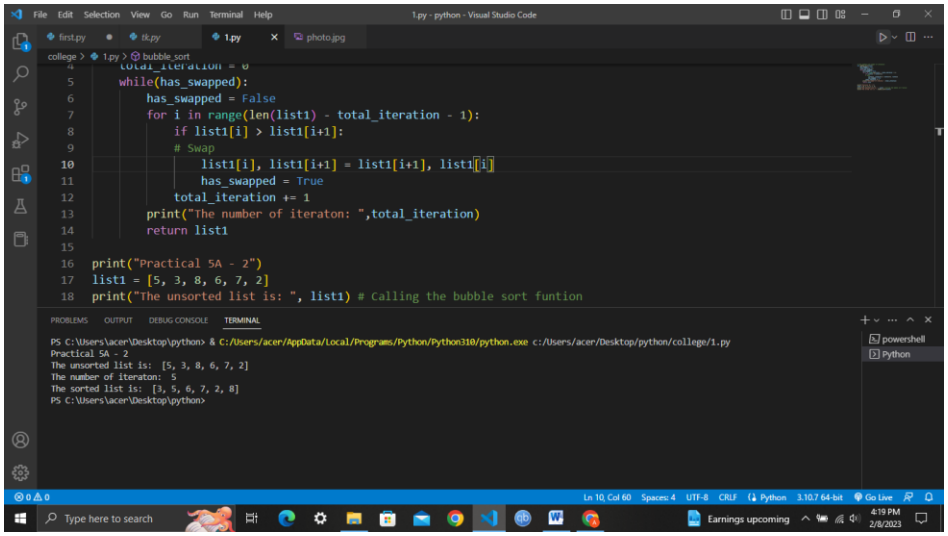
```
def BinarySearch(arr, k, low, high):
    if high >= low:
        mid = low + (high - low)//2
        if arr[mid] == k:
            return mid
        elif arr[mid] > k:
            return BinarySearch(arr, k, low, mid-1)
        else:
            return BinarySearch(arr, k, mid + 1, high)
    else:
        return -1

print("Practical 4B")
arr = [1, 3, 5, 7, 9,15,16,14,45]
k = 15
```

```
PS C:\Users\acer\Desktop\python> & C:/Users/acer/AppData/Local/Programs/Python/Python310/python.exe c:/Users/acer/Desktop/python/college/1.py
Practical 4B
Element is present at index 5
PS C:\Users\acer\Desktop\python>
```



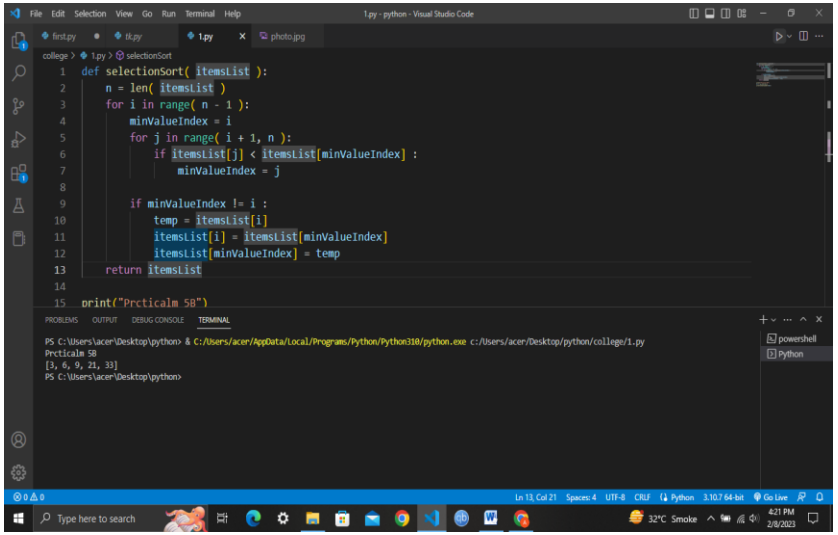
```
def bubble_sort(list1):
# Outer loop for traverse the entire list
    for i in range(0,len(list1)-1):
        for j in range(len(list1)-1):
            if(list1[j]>list1[j+1]):
                temp = list1[j]
                list1[j] = list1[j+1]
                list1[j+1] = temp
        return list1

print("Practical 5A")
list1 = [5, 3, 8, 6, 7, 2]
print("The unsorted list is: ", list1) # Calling the bubble sort function
print("The sorted list is: ", bubble_sort(list1))
```

```
PS C:\Users\acer\Desktop\python> & C:/Users/acer/AppData/Local/Programs/Python/Python310/python.exe c:/Users/acer/Desktop/python/college/1.py
Practical 5A
The unsorted list is:  [5, 3, 8, 6, 7, 2]
The sorted list is:  [3, 5, 6, 7, 2, 8]
PS C:\Users\acer\Desktop\python>
```



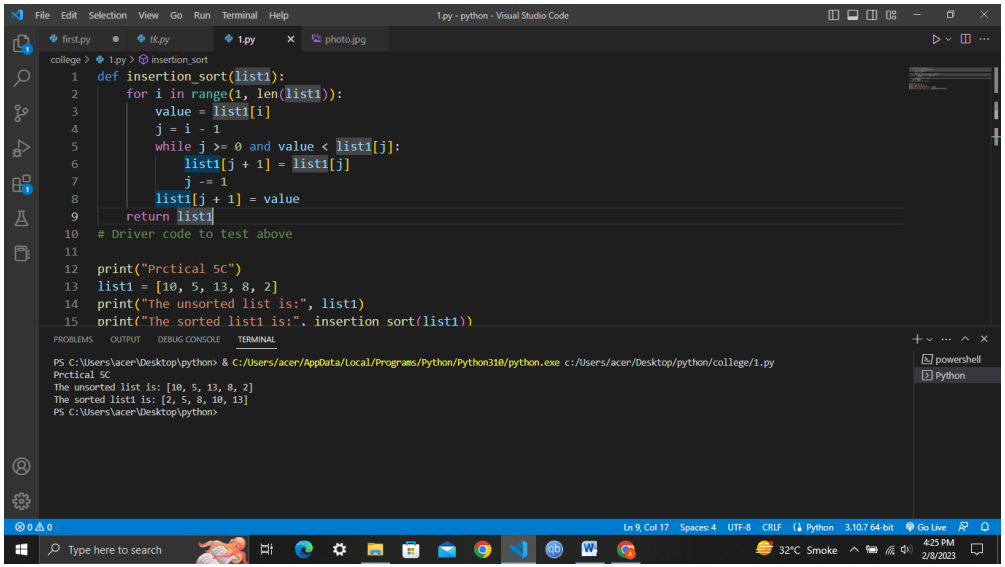
```
    while(has_swapped):
        has_swapped = False
        for i in range(len(list1) - total_iteration - 1):
            if list1[i] > list1[i+1]:
            # Swap
                list1[i], list1[i+1] = list1[i+1], list1[i]
                has_swapped = True
        total_iteration += 1
        print("The number of iteraton: ",total_iteration)
        return list1

print("Practical 5A - 2")
list1 = [5, 3, 8, 6, 7, 2]
print("The unsorted list is: ", list1) # Calling the bubble sort funtion
```

```
PS C:\Users\acer\Desktop\python> & C:/Users/acer/AppData/Local/Programs/Python/Python310/python.exe c:/Users/acer/Desktop/python/college/1.py
Practical 5A - 2
The unsorted list is:  [5, 3, 8, 6, 7, 2]
The number of iteraton:  5
The sorted list is:  [3, 5, 6, 7, 2, 8]
PS C:\Users\acer\Desktop\python>
```



```
def selectionSort( itemsList ):
    n = len( itemsList )
    for i in range( n - 1 ):
        minValueIndex = i
        for j in range( i + 1, n ):
            if itemsList[j] < itemsList[minValueIndex] :
                minValueIndex = j

        if minValueIndex != i :
            temp = itemsList[i]
            itemsList[i] = itemsList[minValueIndex]
            itemsList[minValueIndex] = temp
    return itemsList

print("Prcticalm 5B")
```

```
PS C:\Users\acer\Desktop\python> & C:/Users/acer/AppData/Local/Programs/Python/Python310/python.exe c:/Users/acer/Desktop/python/college/1.py
Prcticalm 5B
[3, 6, 9, 21, 33]
PS C:\Users\acer\Desktop\python>
```



```
def insertion_sort(list1):
    for i in range(1, len(list1)):
        value = list1[i]
        j = i - 1
        while j >= 0 and value < list1[j]:
            list1[j + 1] = list1[j]
            j -= 1
        list1[j + 1] = value
    return list1
# Driver code to test above

print("Prctical 5C")
list1 = [10, 5, 13, 8, 2]
print("The unsorted list is:", list1)
print("The sorted list1 is:", insertion_sort(list1))
```

```
PS C:\Users\acer\Desktop\python> & C:/Users/acer/AppData/Local/Programs/Python/Python310/python.exe c:/Users/acer/Desktop/python/college/1.py
Prctical 5C
The unsorted list is: [10, 5, 13, 8, 2]
The sorted list1 is: [2, 5, 8, 10, 13]
PS C:\Users\acer\Desktop\python>
```