

Worksheet – 3.1

Student Name: Vivek Kumar

UID: 21BCS8129

Branch: BE-CSE (LEET)

Section/Group: 809/A

Semester: 4th

Date of Performance: 20/04/2022

Subject Name: Programming in Python Lab

Subject Code: 20CSP-259

1. Aim/Overview of the practical:

- I. Python program to implement linear search.
- II. Python program to implement bubble sort.
- III. Python program to implement binary search without recursion.
- IV. Python program to implement selection sort.

2. Task to be done/ Which logistics used:

- I. Write a Python program to implement linear search.
- II. Write a Python program to implement bubble sort.
- III. Write a Python program to implement binary search without recursion.
- IV. Write a Python program to implement selection sort.

3. Steps for experiment/practical/Code:

- I. Write a Python program to implement linear search.

Source Code:

```
def linear_Search(list1, n, key):
    for i in range(0, n):
        if (list1[i] == key):
            return i
    return -1

list1 = []
n = int(input('Enter the Size of the List: '))
for i in range(0,n):
    ele=int(input())
    list1.append(ele)
key = int(input('Enter the Key to be Searched: '))
res = linear_Search(list1, n, key)
if(res == -1):
```

```
print("Element { } not found in the list".format(key))  
else:  
    print("Element { } found at index position { }:".format(key,res))
```

II. Write a Python program to implement bubble sort.**Source Code:**

```
def bubble_sort(list1,n):  
    for i in range(0,n-1):  
        for j in range(n-1):  
            if(list1[j]>list1[j+1]):  
                temp = list1[j]  
                list1[j] = list1[j+1]  
                list1[j+1] = temp  
    return list1  
  
list1 = []  
n = int(input('Enter the Size of the List: '))  
for i in range(0, n):  
    ele = int(input())  
    list1.append(ele)  
print("The Given Unsorted list is: ", list1)  
print("The Sorted list is: ", bubble_sort(list1,n))
```

III. Write a Python program to implement binary search without recursion.**Source Code:**

```
def binary_search(list1, key):  
    low = 0  
    high = len(list1) - 1  
    mid = 0  
    while low <= high:  
        mid = (high + low) // 2  
        if list1[mid] < key:  
            low = mid + 1  
        elif list1[mid] > key:  
            high = mid - 1
```

```
        else:
            return mid
    return -1

list1 = []
n = int(input('Enter the Size of the List: '))
for i in range(0, n):
    ele = int(input())
    list1.append(ele)
key = int(input('Enter the Key to be Searched: '))
res = binary_search(list1, key)
if(res == -1):
    print("Element { } not found in the list".format(key))
else:
    print("Element { } found at index position { }:".format(key, res))
```

IV. Write a Python program to implement selection sort.

Source Code:

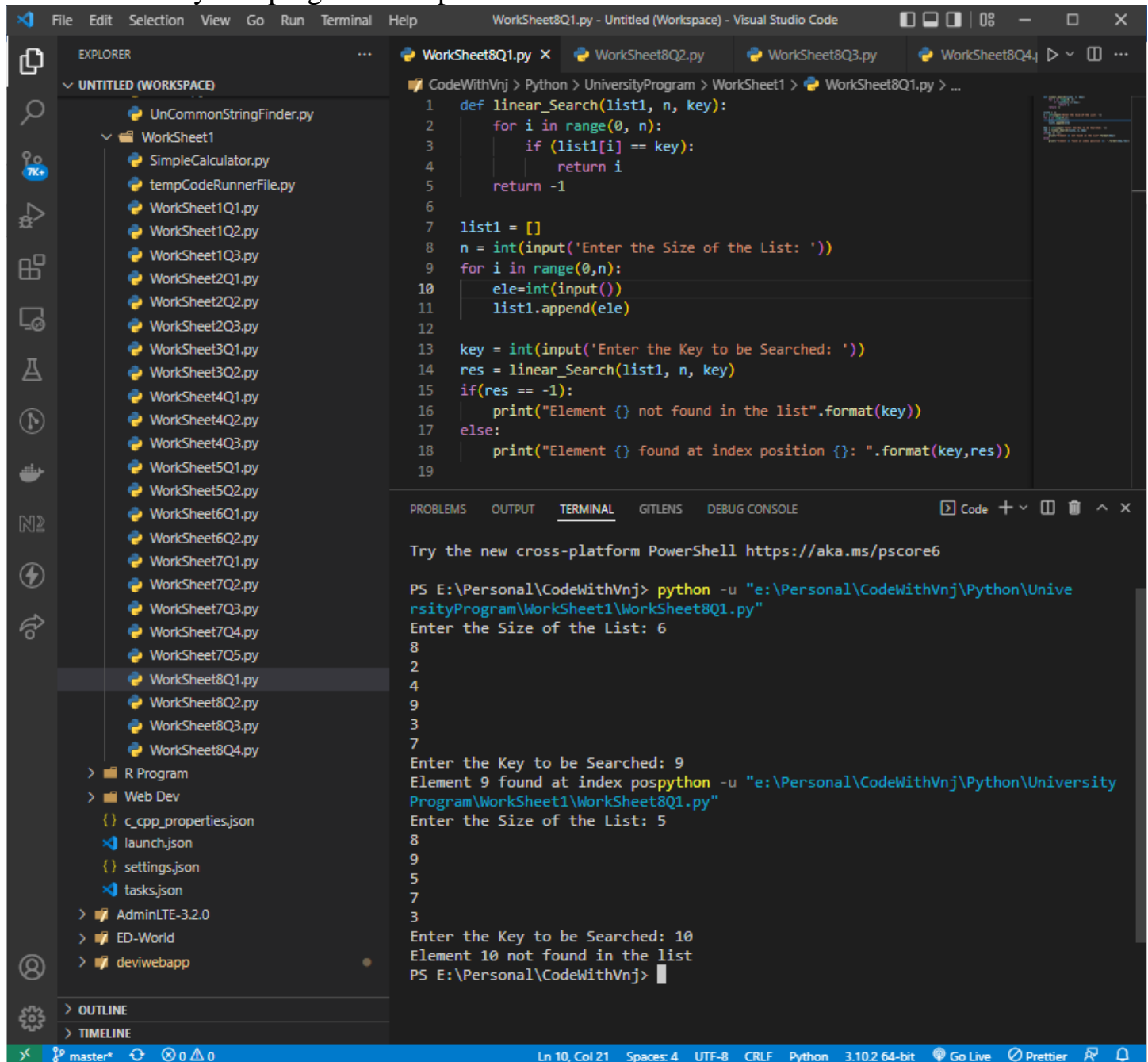
```
def selectionSort(array):
    n = len(array)
    for i in range(n):
        minimum = i
        for j in range(i+1, n):
            if (array[j] < array[minimum]):
                minimum = j
        temp = array[i]
        array[i] = array[minimum]
        array[minimum] = temp
    return array

list1 = []
n = int(input('Enter the Size of the List: '))
for i in range(0, n):
    ele = int(input())
```

```
list1.append(ele)
print("The Given Unsorted list is: ", list1)
print("The Sorted list is: ", selectionSort(list1))
```

4. Result/Output/Writing Summary:

I. Write a Python program to implement linear search.



The screenshot shows a Visual Studio Code editor with a Python file named `WorkSheet8Q1.py`. The code implements a linear search algorithm. The terminal output shows the program being run, with the user entering the size of the list (6), the key to be searched (9), and the program outputting "Element 9 found at index position 2".

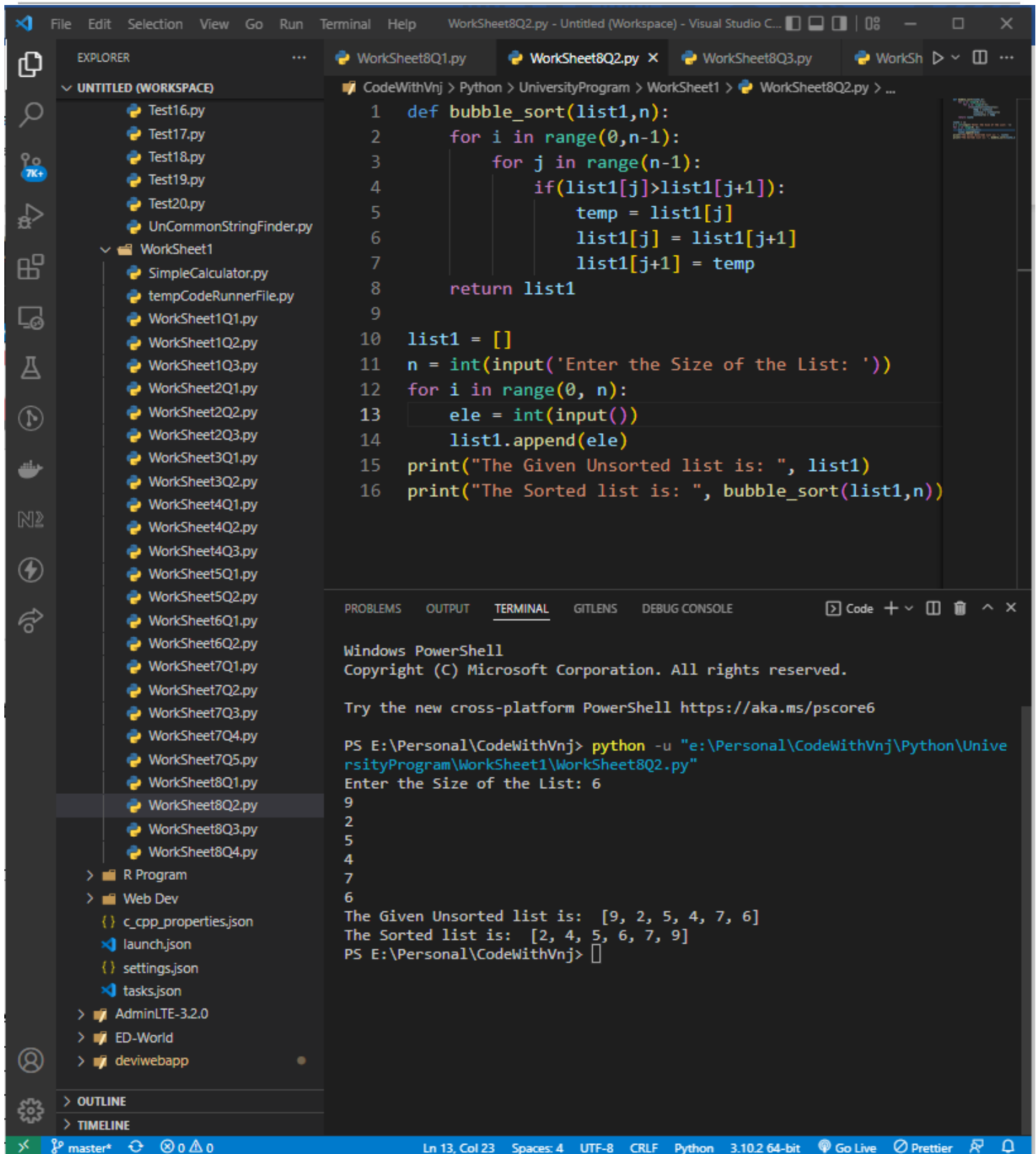
```
def linear_Search(list1, n, key):
    for i in range(0, n):
        if (list1[i] == key):
            return i
    return -1

list1 = []
n = int(input('Enter the Size of the List: '))
for i in range(0,n):
    ele=int(input())
    list1.append(ele)

key = int(input('Enter the Key to be Searched: '))
res = linear_Search(list1, n, key)
if(res == -1):
    print("Element {} not found in the list".format(key))
else:
    print("Element {} found at index position {}".format(key,res))
```

```
PS E:\Personal\CodeWithVnJ> python -u "e:\Personal\CodeWithVnJ\Python\UniversityProgram\WorkSheet1\WorkSheet8Q1.py"
Enter the Size of the List: 6
8
2
4
9
3
7
Enter the Key to be Searched: 9
Element 9 found at index position 2
python -u "e:\Personal\CodeWithVnJ\Python\UniversityProgram\WorkSheet1\WorkSheet8Q1.py"
Enter the Size of the List: 5
8
9
5
7
3
Enter the Key to be Searched: 10
Element 10 not found in the list
PS E:\Personal\CodeWithVnJ>
```

II. Write a Python program to implement bubble sort.



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left displays a project structure with a folder named 'WorkSheet1' containing various Python files. The main editor window shows the code for 'WorkSheet8Q2.py', which implements a bubble sort algorithm. The code defines a 'bubble_sort' function and uses it to sort a list of numbers entered by the user. The bottom panel shows the 'TERMINAL' output, which displays the execution of the program in a Windows PowerShell environment. The user enters the size of the list as 6, and the program outputs the unsorted and sorted lists.

```

1  def bubble_sort(list1,n):
2      for i in range(0,n-1):
3          for j in range(n-1):
4              if(list1[j]>list1[j+1]):
5                  temp = list1[j]
6                  list1[j] = list1[j+1]
7                  list1[j+1] = temp
8      return list1
9
10 list1 = []
11 n = int(input('Enter the Size of the List: '))
12 for i in range(0, n):
13     ele = int(input())
14     list1.append(ele)
15 print("The Given Unsorted list is: ", list1)
16 print("The Sorted list is: ", bubble_sort(list1,n))

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

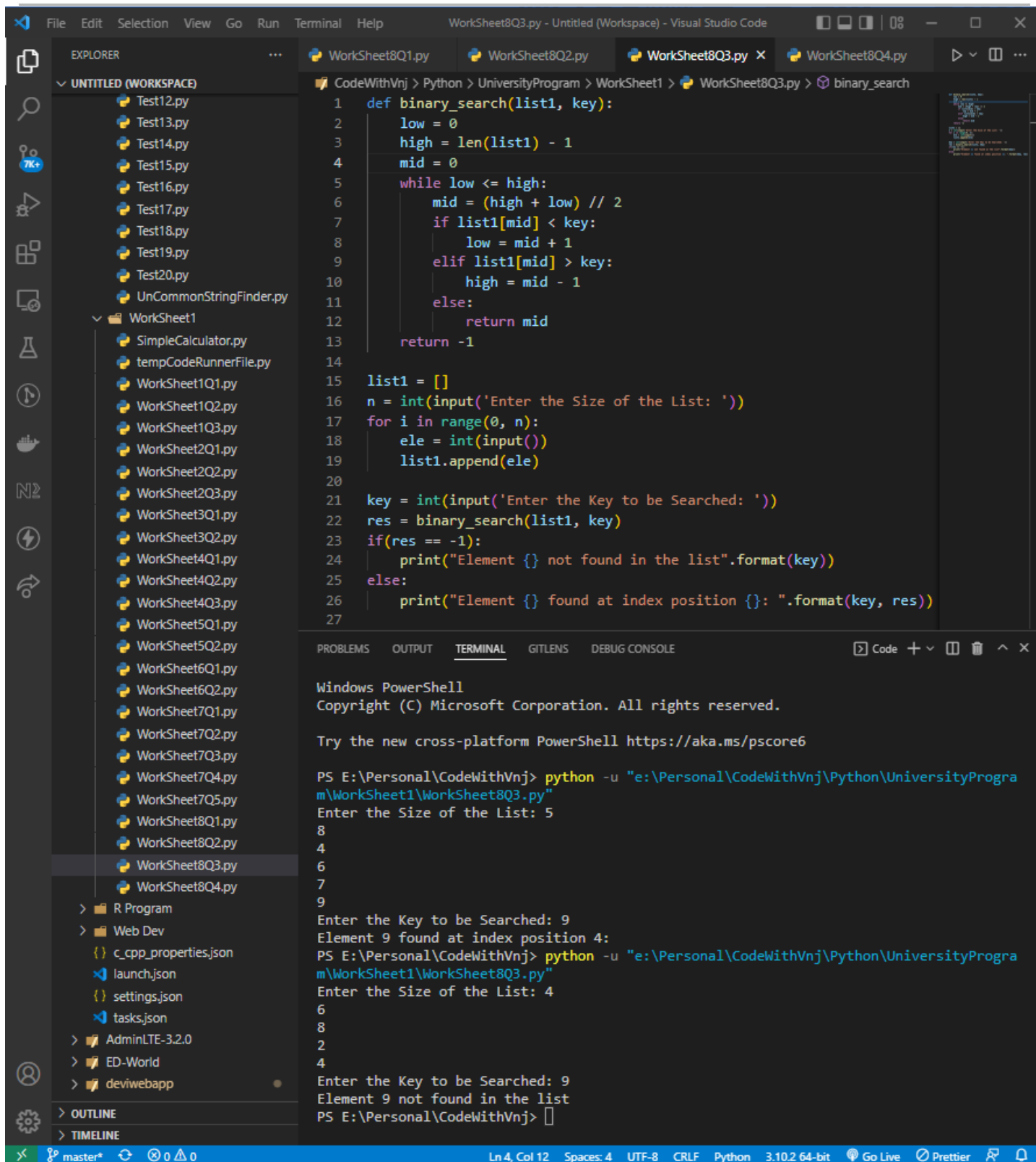
PS E:\Personal\CodeWithVnj> python -u "e:\Personal\CodeWithVnj\Python\UniversityProgram\WorkSheet1\WorkSheet8Q2.py"

Enter the Size of the List: 6

9
2
5
4
7
6

The Given Unsorted list is: [9, 2, 5, 4, 7, 6]
The Sorted list is: [2, 4, 5, 6, 7, 9]
PS E:\Personal\CodeWithVnj>

III. Write a Python program to implement binary search without recursion.



The screenshot shows the Visual Studio Code interface with a workspace named 'WorkSheet8Q3.py - Untitled (Workspace)'. The Explorer panel on the left shows a folder named 'WorkSheet1' containing various Python files, including 'WorkSheet8Q3.py'. The main editor displays the code for 'binary_search.py'.

```

1  def binary_search(list1, key):
2      low = 0
3      high = len(list1) - 1
4      mid = 0
5      while low <= high:
6          mid = (high + low) // 2
7          if list1[mid] < key:
8              low = mid + 1
9          elif list1[mid] > key:
10             high = mid - 1
11          else:
12             return mid
13      return -1
14
15  list1 = []
16  n = int(input('Enter the Size of the List: '))
17  for i in range(0, n):
18      ele = int(input())
19      list1.append(ele)
20
21  key = int(input('Enter the Key to be Searched: '))
22  res = binary_search(list1, key)
23  if(res == -1):
24      print("Element {} not found in the list".format(key))
25  else:
26      print("Element {} found at index position {}".format(key, res))
27

```

The TERMINAL panel at the bottom shows the execution of the program in a Windows PowerShell environment. The user enters the size of the list as 5 and the key as 9. The output is 'Element 9 found at index position 4:'. The user then enters the size of the list as 4 and the key as 9. The output is 'Element 9 not found in the list'.

```

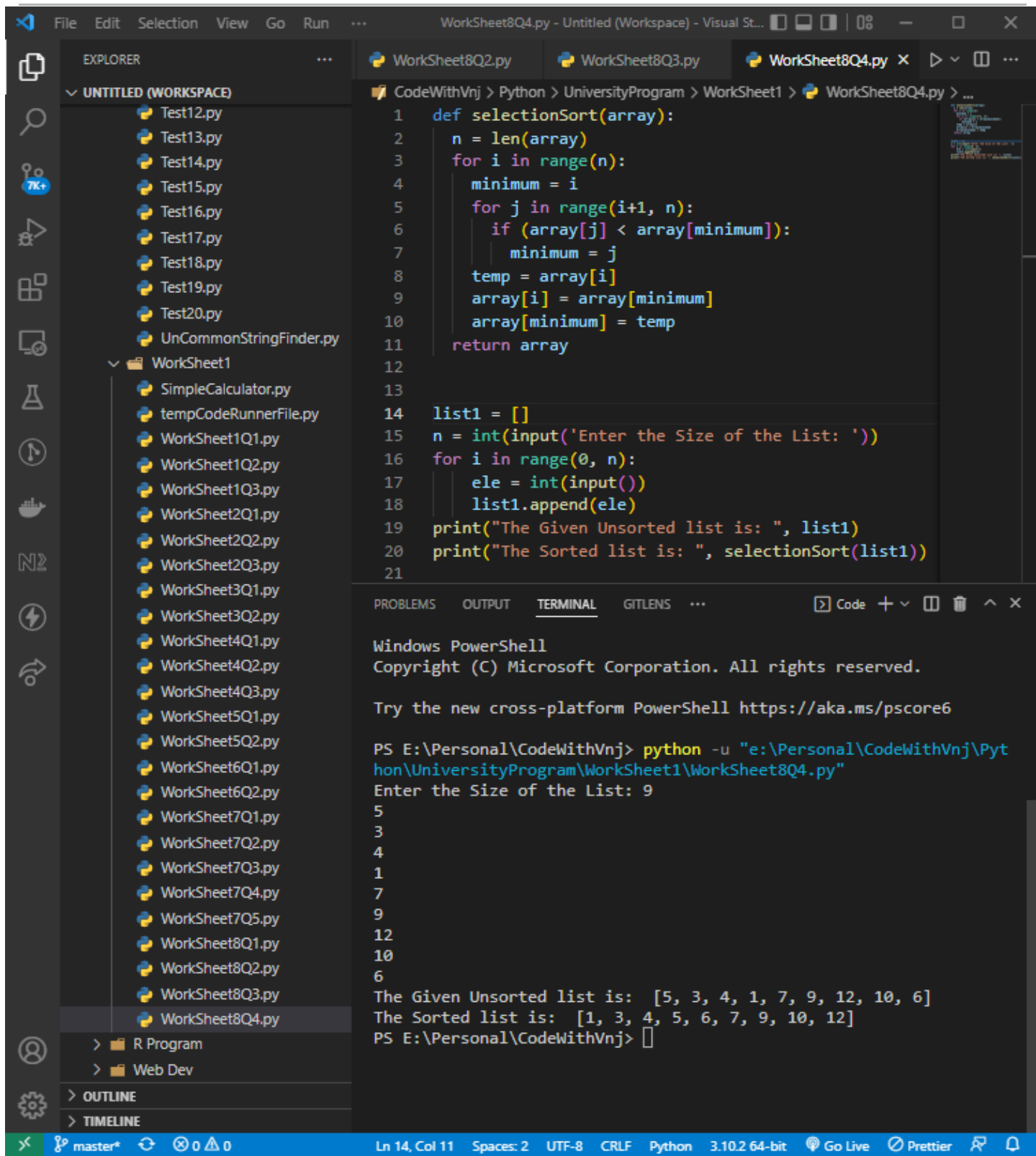
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\Personal\CodeWithVnJ> python -u "e:\Personal\CodeWithVnJ\Python\UniversityProgram\WorkSheet1\WorkSheet8Q3.py"
Enter the Size of the List: 5
8
4
6
7
9
Enter the Key to be Searched: 9
Element 9 found at index position 4:
PS E:\Personal\CodeWithVnJ> python -u "e:\Personal\CodeWithVnJ\Python\UniversityProgram\WorkSheet1\WorkSheet8Q3.py"
Enter the Size of the List: 4
6
8
2
4
Enter the Key to be Searched: 9
Element 9 not found in the list
PS E:\Personal\CodeWithVnJ>

```

IV. Write a Python program to implement selection sort.



The screenshot shows the Visual Studio Code interface with a workspace containing several Python files. The file 'WorkSheet8Q4.py' is open in the editor, displaying a selection sort algorithm. The terminal window at the bottom shows the execution of the program, where the user enters the size of the list (9) and the elements of the list (5, 3, 4, 1, 7, 9, 12, 10, 6). The program outputs the unsorted and sorted lists.

```

1  def selectionSort(array):
2      n = len(array)
3      for i in range(n):
4          minimum = i
5          for j in range(i+1, n):
6              if (array[j] < array[minimum]):
7                  minimum = j
8          temp = array[i]
9          array[i] = array[minimum]
10         array[minimum] = temp
11     return array
12
13
14     list1 = []
15     n = int(input('Enter the Size of the List: '))
16     for i in range(0, n):
17         ele = int(input())
18         list1.append(ele)
19     print("The Given Unsorted list is: ", list1)
20     print("The Sorted list is: ", selectionSort(list1))
21

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS E:\Personal\CodeWithVnj> python -u "e:\Personal\CodeWithVnj\Python\UniversityProgram\WorkSheet1\WorkSheet8Q4.py"

Enter the Size of the List: 9

5

3

4

1

7

9

12

10

6

The Given Unsorted list is: [5, 3, 4, 1, 7, 9, 12, 10, 6]

The Sorted list is: [1, 3, 4, 5, 6, 7, 9, 10, 12]

PS E:\Personal\CodeWithVnj>

Learning outcomes (What I have learnt):

1. I have learnt, how to take List Input from User.
2. Learnt to implement various searching technique in the list.
3. Learnt to implement various sorting technique in the list.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			
4.			