

Foundation Technical Training

Data Structures and Algorithms

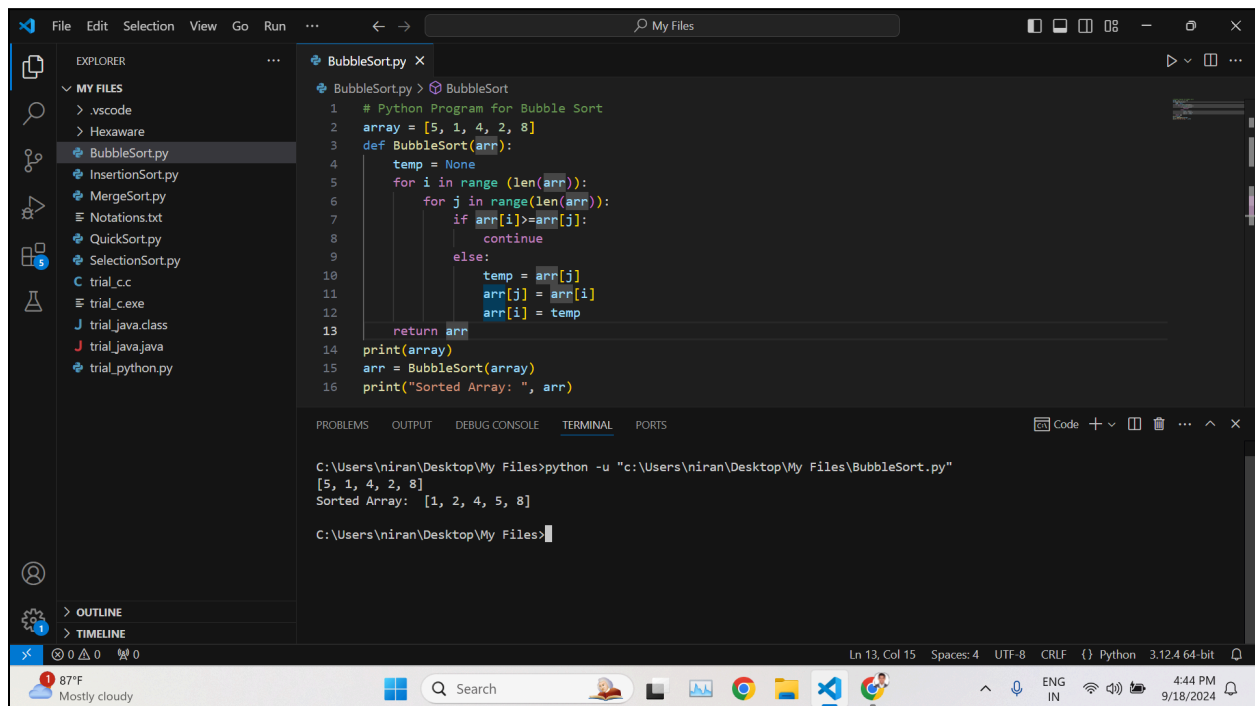
Name: Niranjan Kolpe, Batch: C#-Batch 2

1. Bubble Sort

Program:

```
# Python Program for Bubble Sort
array = [5, 1, 4, 2, 8]
def BubbleSort(arr):
    temp = None
    for i in range (len(arr)):
        for j in range(len(arr)):
            if arr[i]>=arr[j]:
                continue
            else:
                temp = arr[j]
                arr[j] = arr[i]
                arr[i] = temp
    return arr
print(array)
arr = BubbleSort(array)
print("Sorted Array: ", arr)
```

Output:



The screenshot shows a Visual Studio Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'BubbleSort.py' with several files listed. The code editor displays the Python program for Bubble Sort. The terminal at the bottom shows the command to run the program and the output, which is the sorted array [1, 2, 4, 5, 8].

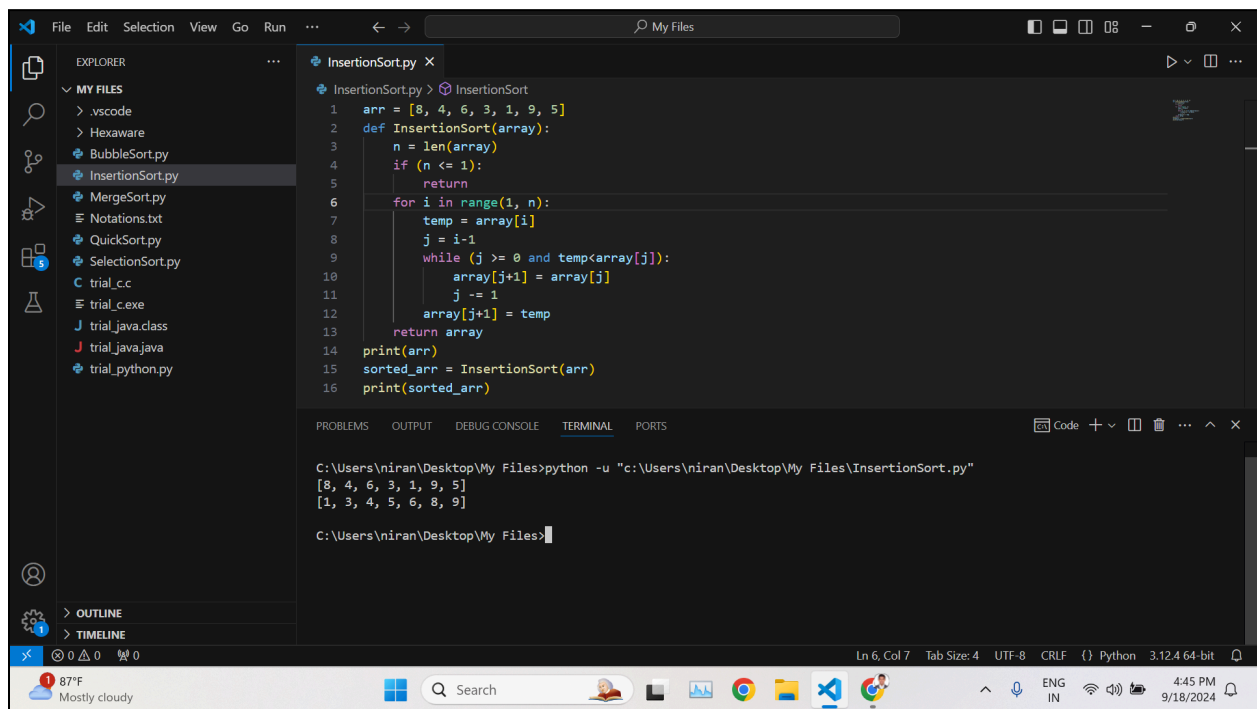
```
C:\Users\niran\Desktop\My Files>python -u "c:\Users\niran\Desktop\My Files\BubbleSort.py"
[5, 1, 4, 2, 8]
Sorted Array: [1, 2, 4, 5, 8]
C:\Users\niran\Desktop\My Files>
```

2. Insertion Sort

Program:

```
arr = [8, 4, 6, 3, 1, 9, 5]
def InsertionSort(array):
    n = len(array)
    if (n <= 1):
        return
    for i in range(1, n):
        temp = array[i]
        j = i-1
        while (j >= 0 and temp<array[j]):
            array[j+1] = array[j]
            j -= 1
        array[j+1] = temp
    return array
print(arr)
sorted_arr = InsertionSort(arr)
print(sorted_arr)
```

Output:



The screenshot shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left displays a file tree with 'MY FILES' containing various files like .vscode, Hexaware, BubbleSort.py, InsertionSort.py (selected), MergeSort.py, Notations.txt, QuickSort.py, SelectionSort.py, trial.cc, trial.exe, trial.java.class, trial.java.java, and trial_python.py. The main editor area shows the code for InsertionSort.py, which is identical to the program provided in the previous block. Below the editor, the TERMINAL panel is active, showing the command 'C:\Users\niran\Desktop\My Files>python -u "c:\Users\niran\Desktop\My Files\InsertionSort.py"' and its output: '[8, 4, 6, 3, 1, 9, 5]' followed by '[1, 3, 4, 5, 6, 8, 9]'. The status bar at the bottom indicates 'Ln 6, Col 7', 'Tab Size: 4', 'UTF-8', 'CRLF', 'Python', and '3.12.4 64-bit'. The Windows taskbar at the very bottom shows the date and time as '4:45 PM 9/18/2024'.

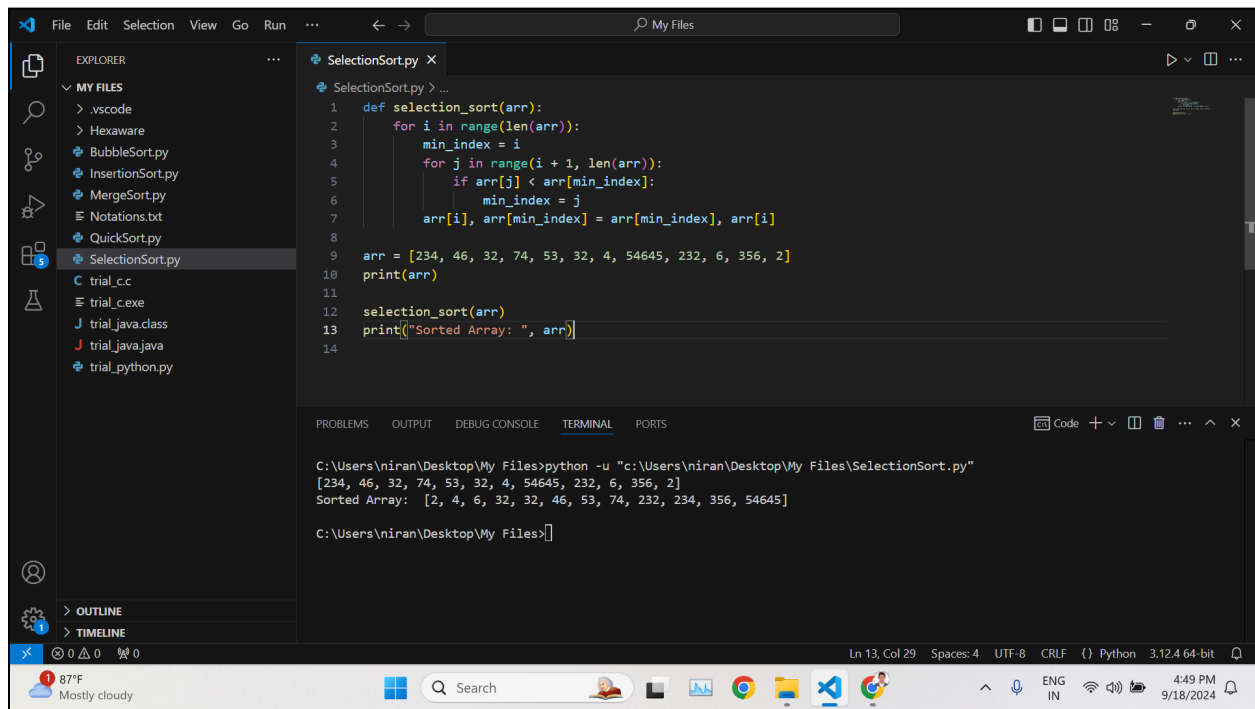
```
File Edit Selection View Go Run ... < -> My Files
EXPLORER
MY FILES
  > .vscode
  > Hexaware
  + BubbleSort.py
  + InsertionSort.py
  + MergeSort.py
  + Notations.txt
  + QuickSort.py
  + SelectionSort.py
  + trial.cc
  + trial.exe
  + trial.java.class
  + trial.java.java
  + trial_python.py
OUTLINE
TIMELINE
InsertionSort.py
1 arr = [8, 4, 6, 3, 1, 9, 5]
2 def InsertionSort(array):
3     n = len(array)
4     if (n <= 1):
5         return
6     for i in range(1, n):
7         temp = array[i]
8         j = i-1
9         while (j >= 0 and temp<array[j]):
10             array[j+1] = array[j]
11             j -= 1
12         array[j+1] = temp
13     return array
14 print(arr)
15 sorted_arr = InsertionSort(arr)
16 print(sorted_arr)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
C:\Users\niran\Desktop\My Files>python -u "c:\Users\niran\Desktop\My Files\InsertionSort.py"
[8, 4, 6, 3, 1, 9, 5]
[1, 3, 4, 5, 6, 8, 9]
C:\Users\niran\Desktop\My Files>
Ln 6, Col 7 Tab Size: 4 UTF-8 CRLF {} Python 3.12.4 64-bit
87°F Mostly cloudy 4:45 PM 9/18/2024
```

3. Selection Sort

Program:

```
def selection_sort(arr):  
    for i in range(len(arr)):  
        min_index = i  
        for j in range(i + 1, len(arr)):  
            if arr[j] < arr[min_index]:  
                min_index = j  
        arr[i], arr[min_index] = arr[min_index], arr[i]  
  
arr = [234, 46, 32, 74, 53, 32, 4, 54645, 232, 6, 356, 2]  
print(arr)  
  
selection_sort(arr)  
print("Sorted Array: ", arr)
```

Output:



The screenshot shows a Visual Studio Code editor window with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'MY FILES' containing several files, including 'SelectionSort.py'. The editor window displays the Selection Sort program code. The terminal shows the output of running the program, which is the sorted array: [2, 4, 6, 32, 32, 46, 53, 74, 232, 234, 356, 54645].

```
def selection_sort(arr):  
    for i in range(len(arr)):  
        min_index = i  
        for j in range(i + 1, len(arr)):  
            if arr[j] < arr[min_index]:  
                min_index = j  
        arr[i], arr[min_index] = arr[min_index], arr[i]  
  
arr = [234, 46, 32, 74, 53, 32, 4, 54645, 232, 6, 356, 2]  
print(arr)  
  
selection_sort(arr)  
print("Sorted Array: ", arr)
```

```
C:\Users\niran\Desktop\My Files>python -u "c:\Users\niran\Desktop\My Files\SelectionSort.py"  
[234, 46, 32, 74, 53, 32, 4, 54645, 232, 6, 356, 2]  
Sorted Array: [2, 4, 6, 32, 32, 46, 53, 74, 232, 234, 356, 54645]  
  
C:\Users\niran\Desktop\My Files>
```

4. Quick Sort

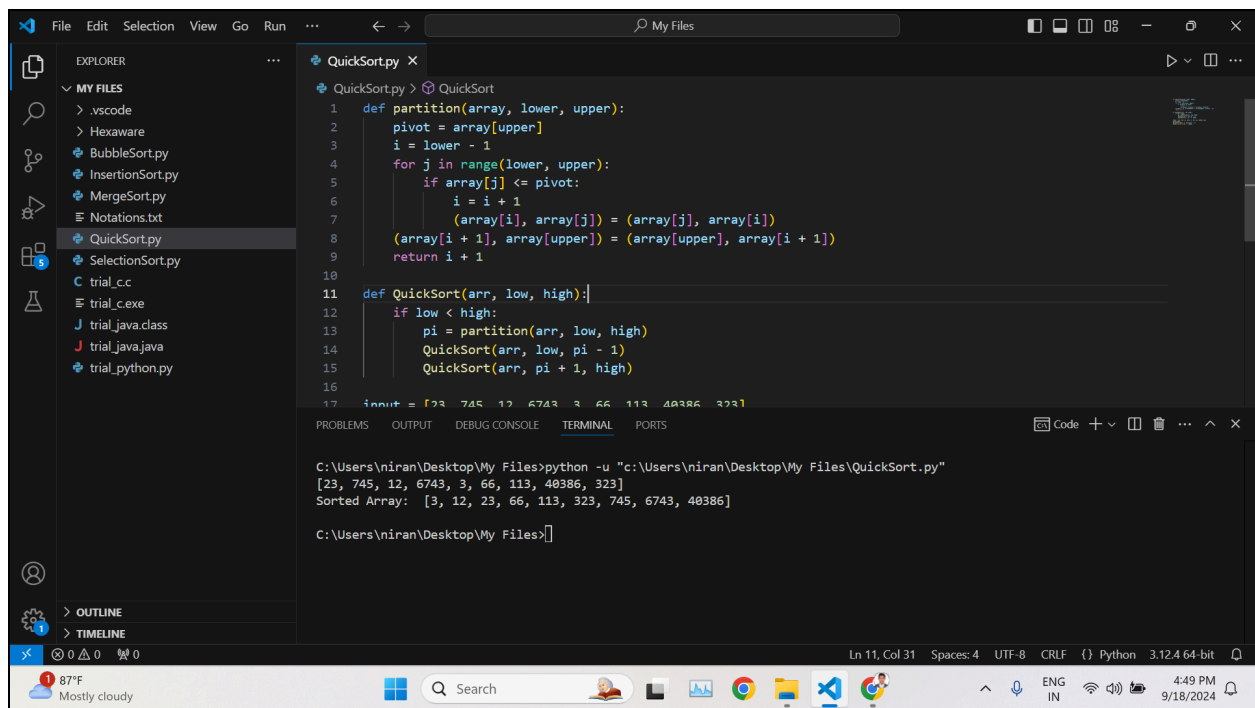
Program:

```
def partition(array, lower, upper):
    pivot = array[upper]
    i = lower - 1
    for j in range(lower, upper):
        if array[j] <= pivot:
            i = i + 1
            (array[i], array[j]) = (array[j], array[i])
    (array[i + 1], array[upper]) = (array[upper], array[i + 1])
    return i + 1

def QuickSort(arr, low, high):
    if low < high:
        pi = partition(arr, low, high)
        QuickSort(arr, low, pi - 1)
        QuickSort(arr, pi + 1, high)

input = [23, 745, 12, 6743, 3, 66, 113, 40386, 323]
print(input)
QuickSort(input, 0, len(input) - 1)
print("Sorted Array: ", input)
```

Output:

A screenshot of a Visual Studio Code editor window. The Explorer sidebar on the left shows a file named 'QuickSort.py' selected. The main editor area displays the Python code for the QuickSort algorithm. Below the code, the 'TERMINAL' tab is active, showing the command 'python -u "c:\Users\niran\Desktop\My Files\QuickSort.py"' and its output: '[23, 745, 12, 6743, 3, 66, 113, 40386, 323]' followed by 'Sorted Array: [3, 12, 23, 66, 113, 323, 745, 6743, 40386]'. The status bar at the bottom indicates the file is at line 11, column 31, using UTF-8 encoding and CRLF line endings, with Python 3.12.4 64-bit running.

5. Merge Sort

Program:

```
def MergeSort(array):
    if len(array)>1:
        mid = len(array)//2
        left_arr = array[:mid]
        right_arr = array[mid:]

        MergeSort(left_arr)
        MergeSort(right_arr)

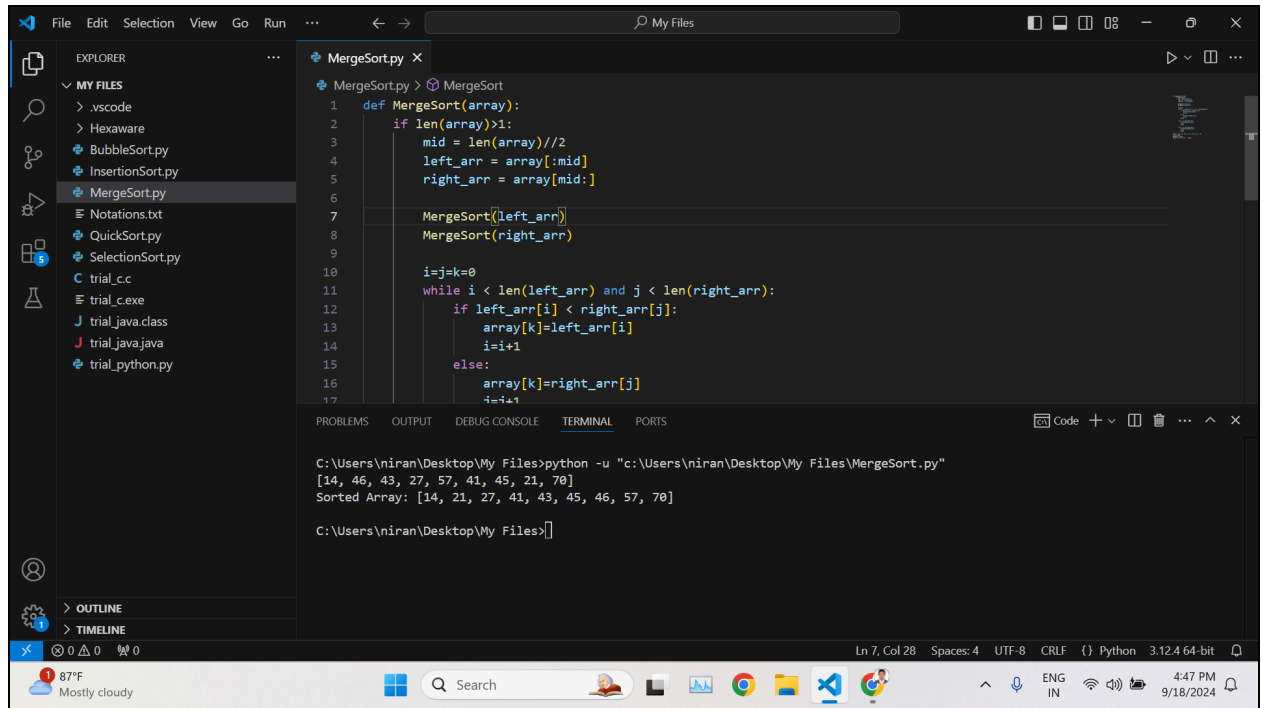
        i=j=k=0
        while i < len(left_arr) and j < len(right_arr):
            if left_arr[i] < right_arr[j]:
                array[k]=left_arr[i]
                i=i+1
            else:
                array[k]=right_arr[j]
                j=j+1
            k=k+1

        while i < len(left_arr):
            array[k]=left_arr[i]
            i=i+1
            k=k+1

        while j < len(right_arr):
            array[k]=right_arr[j]
            j=j+1
            k=k+1

input = [14, 46, 43, 27, 57, 41, 45, 21, 70]
print(input)
MergeSort(input)
print("Sorted Array:", input)
```

Output:



6. Linear Search

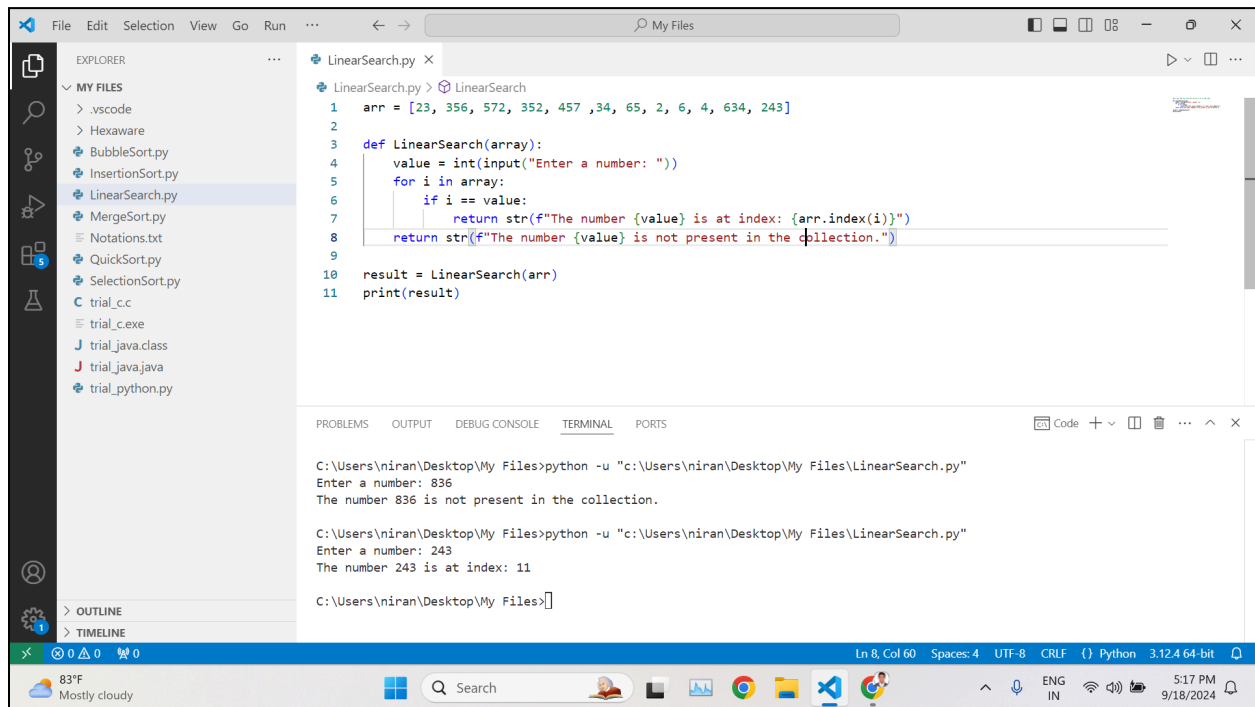
Program:

```
arr = [23, 356, 572, 352, 457, 34, 65, 2, 6, 4, 634, 243]
```

```
def LinearSearch(array):  
    value = int(input("Enter a number: "))  
    for i in array:  
        if i == value:  
            return str(f"The number {value} is at index: {arr.index(i)}")  
    return str(f"The number {value} is not present in the collection.")
```

```
result = LinearSearch(arr)  
print(result)
```

Output:



The screenshot shows a Visual Studio Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'MY FILES' containing several Python files, including 'LinearSearch.py'. The code editor displays the following Python code:

```
1 arr = [23, 356, 572, 352, 457, 34, 65, 2, 6, 4, 634, 243]  
2  
3 def LinearSearch(array):  
4     value = int(input("Enter a number: "))  
5     for i in array:  
6         if i == value:  
7             return str(f"The number {value} is at index: {arr.index(i)}")  
8     return str(f"The number {value} is not present in the collection.")  
9  
10 result = LinearSearch(arr)  
11 print(result)
```

Below the code editor, the 'TERMINAL' tab is active, showing the output of the program. The first run shows that the number 836 is not present in the collection. The second run shows that the number 243 is at index 11.

```
C:\Users\niran\Desktop\My Files>python -u "c:\Users\niran\Desktop\My Files\LinearSearch.py"  
Enter a number: 836  
The number 836 is not present in the collection.  
  
C:\Users\niran\Desktop\My Files>python -u "c:\Users\niran\Desktop\My Files\LinearSearch.py"  
Enter a number: 243  
The number 243 is at index: 11  
  
C:\Users\niran\Desktop\My Files>
```

The status bar at the bottom of the editor shows the current line and column (Ln 8, Col 60), the number of spaces (4), the encoding (UTF-8), the line ending (CRLF), the interpreter (Python 3.12.4 64-bit), and the system status (83°F, Mostly cloudy, 5:17 PM, 9/18/2024).

7. Binary Search

Program:

```
def binary_search(arr, target):
    low = 0
    high = len(arr) - 1
    while low <= high:
        mid = (low + high) // 2
        if arr[mid] == target:
            return mid
        elif arr[mid] < target:
            low = mid + 1
        else:
            high = mid - 1
    return -1

arr = [34, 745, 74, 2, 4, 21, 44]
value = int(input("Enter a number: "))
result = binary_search(arr, value)
print(f"Element found at index {result}") if result != -1 else
print("Element not found in array")
```

Output:

The image shows a Windows 11 desktop environment with the Visual Studio Code (VS Code) editor open. The editor is displaying a Python file named 'BinarySearch.py' within a workspace titled 'My Files'. The Explorer sidebar on the left shows the file structure, including 'MY FILES' and a list of files: '.vscode', 'Hexaware', 'BinarySearch.py', 'BubbleSort.py', 'InsertionSort.py', 'LinearSearch.py', 'MergeSort.py', 'Notations.txt', 'QuickSort.py', 'SelectionSort.py', 'trial_cc', 'trial_c_exe', 'trial_java.class', 'trial_java.java', and 'trial_python.py'. The main editor window shows the following Python code:

```

1 def binary_search(arr, target):
2     low = 0
3     high = len(arr) - 1
4     while low <= high:
5         mid = (low + high) // 2
6         if arr[mid] == target:
7             return mid
8         elif arr[mid] < target:
9             low = mid + 1
10        else:
11            high = mid - 1
12    return -1
13 arr = [34, 745, 74, 2, 4, 21, 44]
14 value = int(input("Enter a number: "))
15 result = binary_search(arr, value)
16 print(f"Element found at index {result}") if result != -1 else print("Element not found in array")

```

The Output console at the bottom shows the execution results of the program. It displays two test cases:

```

C:\Users\niran\My Files>python -u "c:\Users\niran\Desktop\My Files\BinarySearch.py"
Enter a number: 99
Element not found in array

C:\Users\niran\Desktop\My Files>python -u "c:\Users\niran\Desktop\My Files\BinarySearch.py"
Enter a number: 44
Element found at index 6

C:\Users\niran\Desktop\My Files>

```

The Windows taskbar at the bottom shows the system clock as 5:31 PM on 9/18/2024. The taskbar also includes icons for the Start menu, Search, and various applications like File Explorer, VS Code, and a web browser.