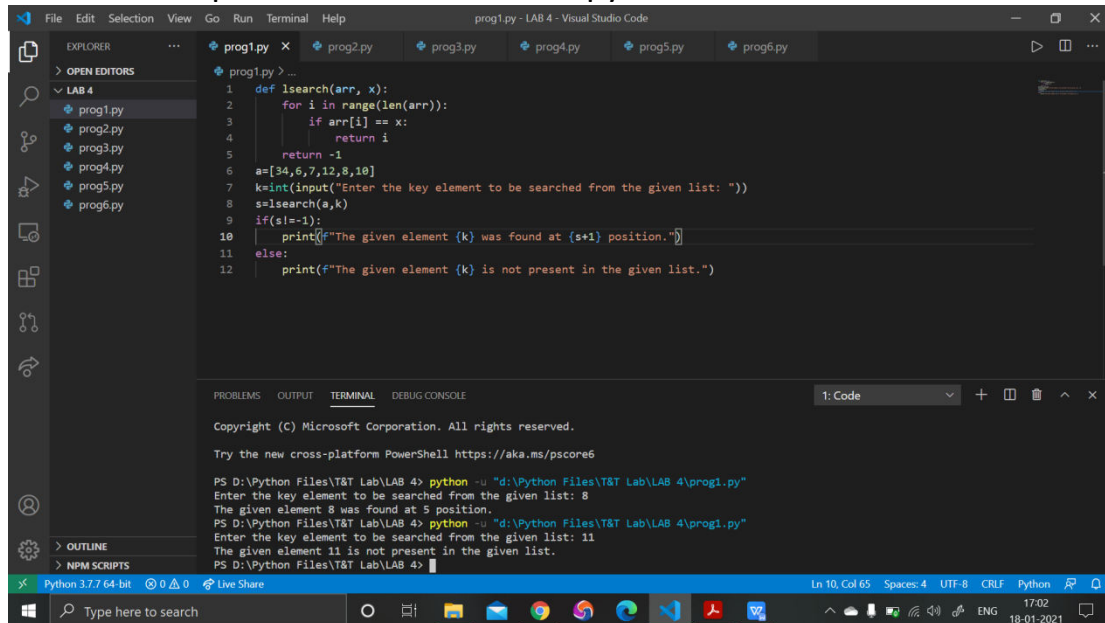


T&T LAB-4

BISWARUP MUKHERJEE

ROLL - 1806468

1. WAP to implement linear search in python.



```
1 def lsearch(arr, x):
2     for i in range(len(arr)):
3         if arr[i] == x:
4             return i
5     return -1
6 a=[34,6,7,12,8,10]
7 k=int(input("Enter the key element to be searched from the given list: "))
8 s=lsearch(a,k)
9 if(s!=-1):
10    print(f"The given element {k} was found at {s+1} position.")
11 else:
12    print(f"The given element {k} is not present in the given list.")
```

Copyright (C) Microsoft Corporation. All rights reserved.

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

PS D:\Python Files\T&T Lab\LAB 4> python -u "d:\Python Files\T&T Lab\LAB 4\prog1.py"

Enter the key element to be searched from the given list: 8

The given element 8 was found at 5 position.

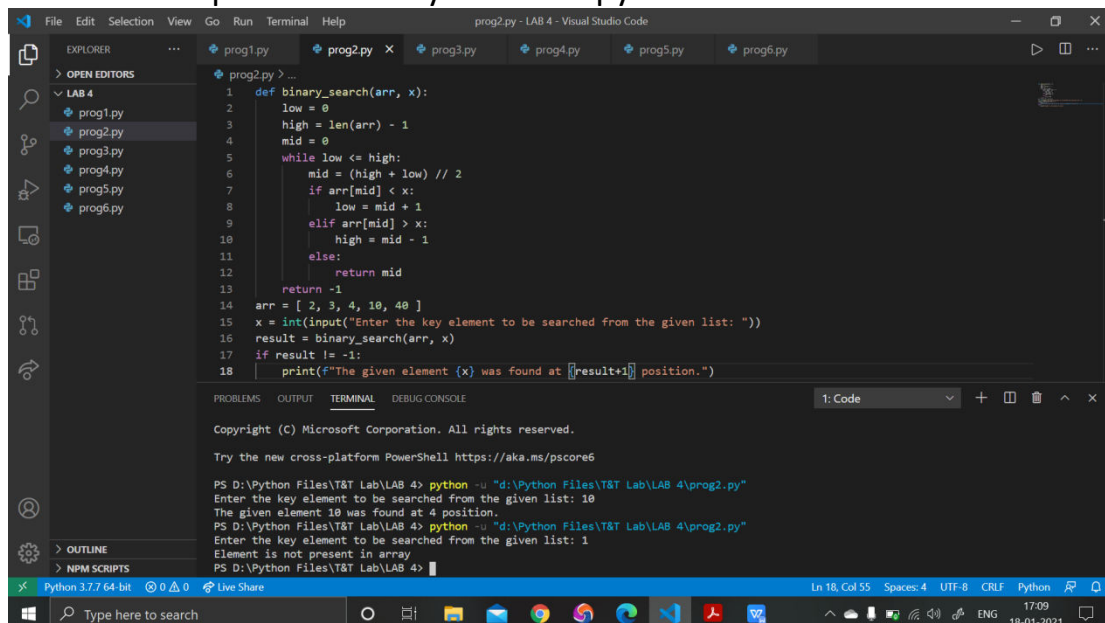
PS D:\Python Files\T&T Lab\LAB 4> python -u "d:\Python Files\T&T Lab\LAB 4\prog1.py"

Enter the key element to be searched from the given list: 11

The given element 11 is not present in the given list.

PS D:\Python Files\T&T Lab\LAB 4>

2. WAP to implement binary search in python.



```
1 def binary_search(arr, x):
2     low = 0
3     high = len(arr) - 1
4     mid = 0
5     while low <= high:
6         mid = (high + low) // 2
7         if arr[mid] < x:
8             low = mid + 1
9         elif arr[mid] > x:
10            high = mid - 1
11        else:
12            return mid
13    return -1
14 arr = [ 2, 3, 4, 10, 40 ]
15 x = int(input("Enter the key element to be searched from the given list: "))
16 result = binary_search(arr, x)
17 if result != -1:
18    print(f"The given element {x} was found at {result+1} position.")
```

Copyright (C) Microsoft Corporation. All rights reserved.

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

PS D:\Python Files\T&T Lab\LAB 4> python -u "d:\Python Files\T&T Lab\LAB 4\prog2.py"

Enter the key element to be searched from the given list: 10

The given element 10 was found at 4 position.

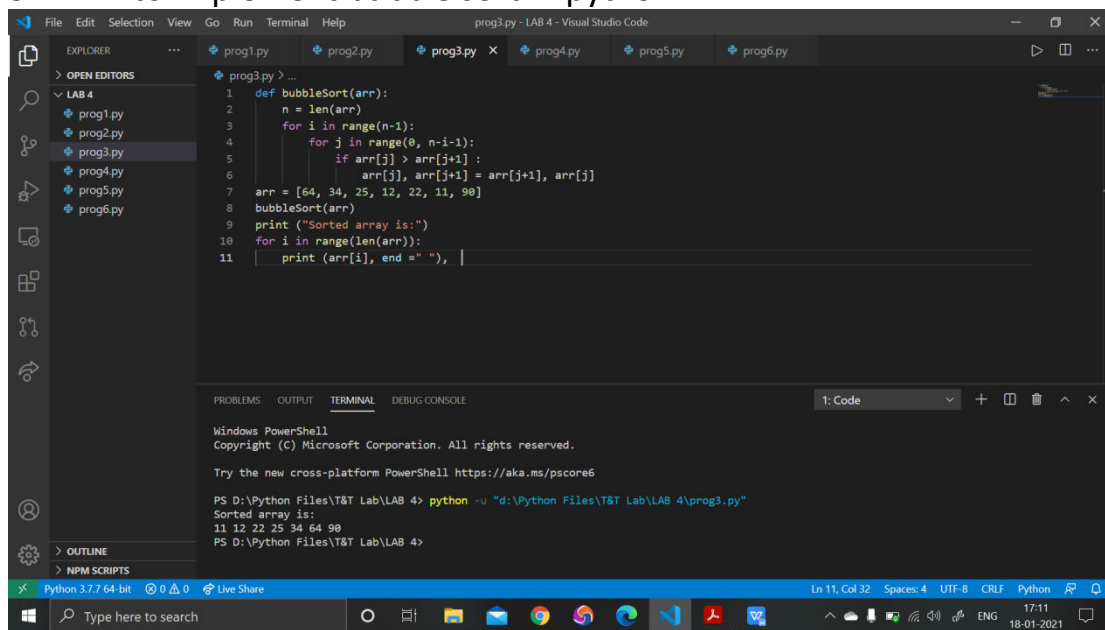
PS D:\Python Files\T&T Lab\LAB 4> python -u "d:\Python Files\T&T Lab\LAB 4\prog2.py"

Enter the key element to be searched from the given list: 1

Element is not present in array

PS D:\Python Files\T&T Lab\LAB 4>

3. WAP to implement bubble sort in python.



The screenshot shows the Visual Studio Code editor with a file explorer on the left containing files prog1.py through prog6.py. The main editor window displays the code for prog3.py, which implements a bubble sort algorithm. The code defines a function bubbleSort(arr) that sorts an array in ascending order. The array is initialized as [64, 34, 25, 12, 22, 11, 98]. The output in the terminal shows the sorted array: 11 12 22 25 34 64 98.

```
1 def bubbleSort(arr):
2     n = len(arr)
3     for i in range(n-1):
4         for j in range(0, n-i-1):
5             if arr[j] > arr[j+1]:
6                 arr[j], arr[j+1] = arr[j+1], arr[j]
7 arr = [64, 34, 25, 12, 22, 11, 98]
8 bubbleSort(arr)
9 print ("Sorted array is:")
10 for i in range(len(arr)):
11     print (arr[i], end = " ")
```

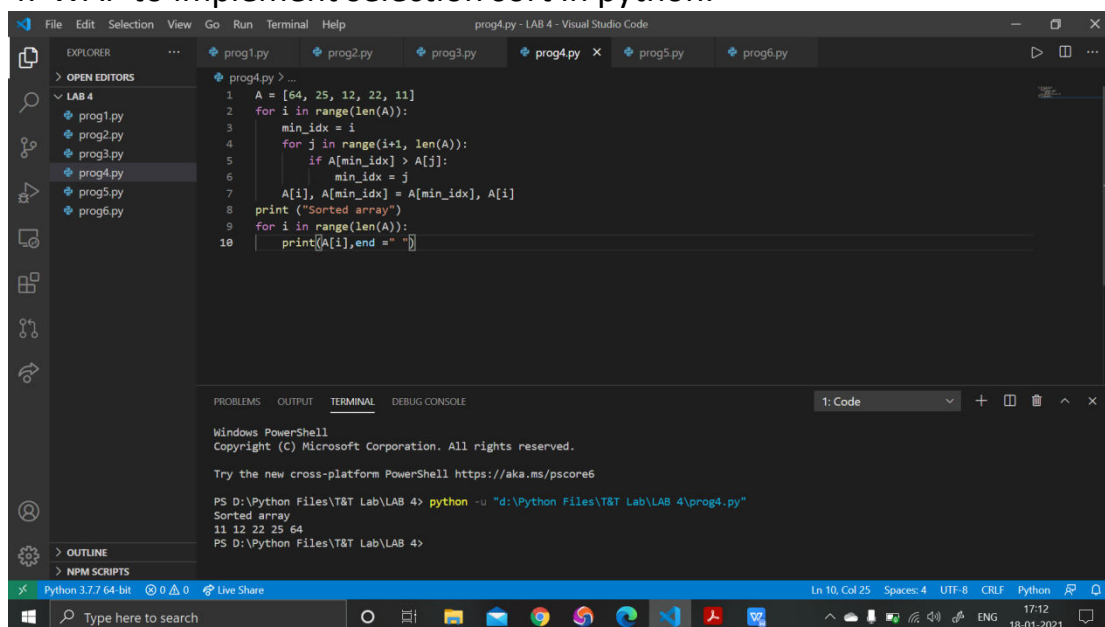
Terminal Output:

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

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

PS D:\Python Files\T&T Lab\LAB 4> python -u "d:\Python Files\T&T Lab\LAB 4\prog3.py"
Sorted array is:
11 12 22 25 34 64 98
PS D:\Python Files\T&T Lab\LAB 4>
```

4. WAP to implement selection sort in python.



The screenshot shows the Visual Studio Code editor with a file explorer on the left containing files prog1.py through prog6.py. The main editor window displays the code for prog4.py, which implements a selection sort algorithm. The array is initialized as [64, 25, 12, 22, 11]. The code finds the minimum element in the unsorted part of the array and swaps it with the first element. The output in the terminal shows the sorted array: 11 12 22 25 64.

```
1 A = [64, 25, 12, 22, 11]
2 for i in range(len(A)):
3     min_idx = i
4     for j in range(i+1, len(A)):
5         if A[min_idx] > A[j]:
6             min_idx = j
7 A[i], A[min_idx] = A[min_idx], A[i]
8 print ("Sorted array")
9 for i in range(len(A)):
10     print(A[i], end = " ")
```

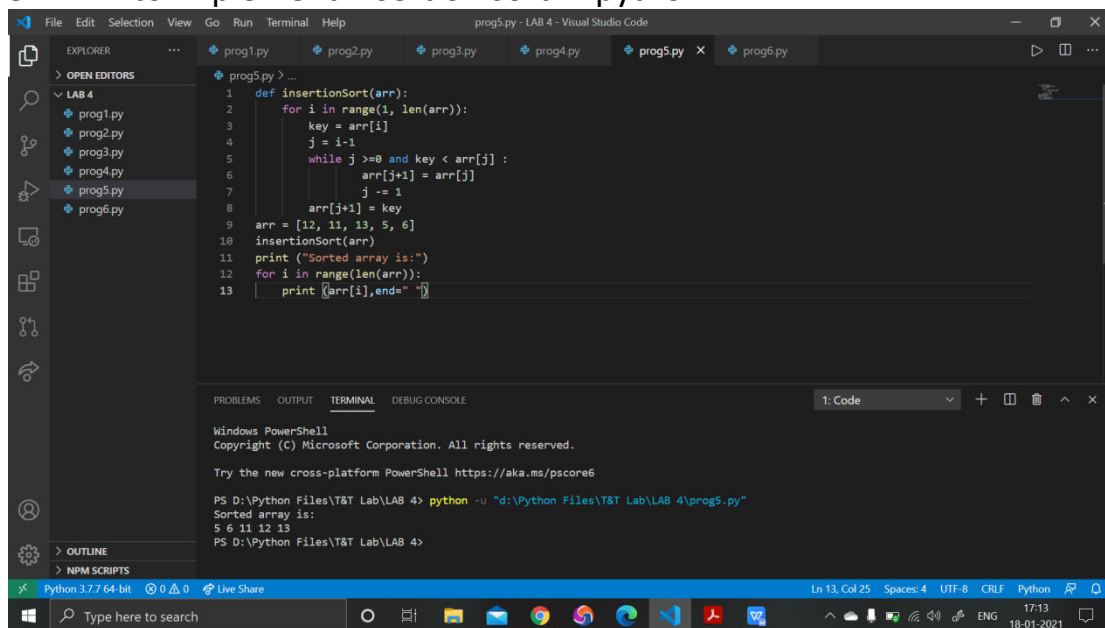
Terminal Output:

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

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

PS D:\Python Files\T&T Lab\LAB 4> python -u "d:\Python Files\T&T Lab\LAB 4\prog4.py"
Sorted array
11 12 22 25 64
PS D:\Python Files\T&T Lab\LAB 4>
```

5. WAP to implement insertion sort in python.



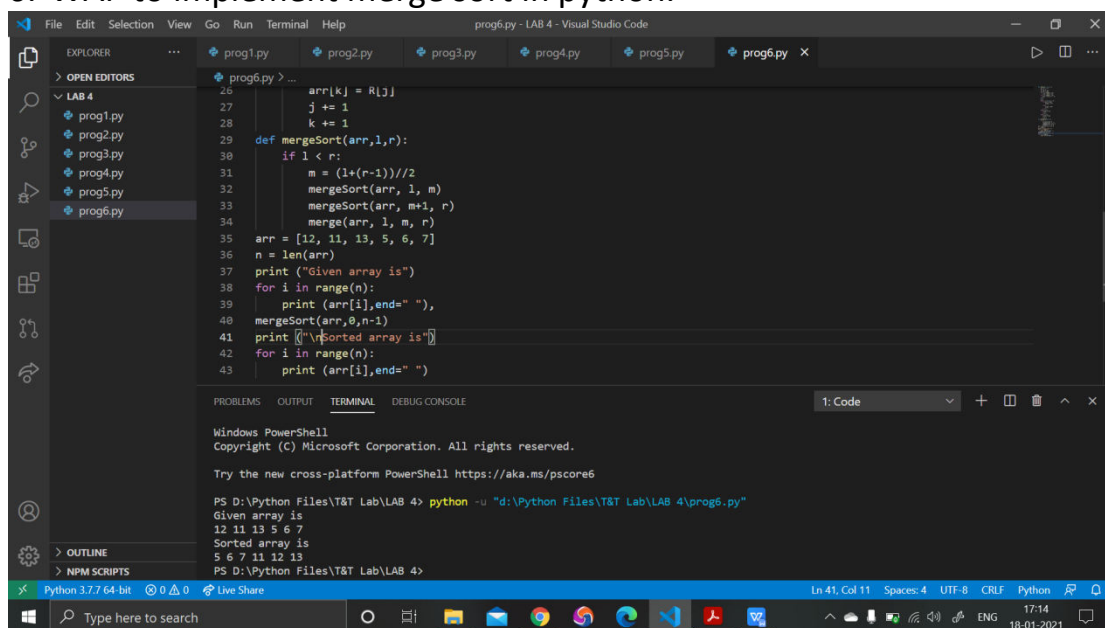
The screenshot shows the Visual Studio Code interface with a Python file named `prog5.py` open. The code implements the Insertion Sort algorithm. The Explorer sidebar on the left shows a folder named `LAB 4` containing files `prog1.py` through `prog6.py`. The `prog5.py` file is selected. The code in the editor is as follows:

```
1 def insertionSort(arr):
2     for i in range(1, len(arr)):
3         key = arr[i]
4         j = i-1
5         while j >= 0 and key < arr[j] :
6             arr[j+1] = arr[j]
7             j -= 1
8         arr[j+1] = key
9 arr = [12, 11, 13, 5, 6]
10 insertionSort(arr)
11 print ("Sorted array is:")
12 for i in range(len(arr)):
13     print (arr[i],end=" ")
```

The TERMINAL panel at the bottom shows the execution of the program using the command `python -u "d:\Python Files\T&T Lab\LAB 4\prog5.py"`. The output is:

```
Sorted array is:
5 6 11 12 13
```

6. WAP to implement merge sort in python.



The screenshot shows the Visual Studio Code interface with a Python file named `prog6.py` open. The code implements the Merge Sort algorithm. The Explorer sidebar on the left shows the same `LAB 4` folder with files `prog1.py` through `prog6.py`. The `prog6.py` file is selected. The code in the editor is as follows:

```
26 arr[k] = R[j]
27 j += 1
28 k += 1
29 def mergeSort(arr,l,r):
30     if l < r:
31         m = (l+(r-1))//2
32         mergeSort(arr, l, m)
33         mergeSort(arr, m+1, r)
34         merge(arr, l, m, r)
35 arr = [12, 11, 13, 5, 6, 7]
36 n = len(arr)
37 print ("Given array is")
38 for i in range(n):
39     print (arr[i],end=" ")
40 mergeSort(arr,0,n-1)
41 print ("\nSorted array is")
42 for i in range(n):
43     print (arr[i],end=" ")
```

The TERMINAL panel at the bottom shows the execution of the program using the command `python -u "d:\Python Files\T&T Lab\LAB 4\prog6.py"`. The output is:

```
Given array is
12 11 13 5 6 7
Sorted array is
5 6 7 11 12 13
```