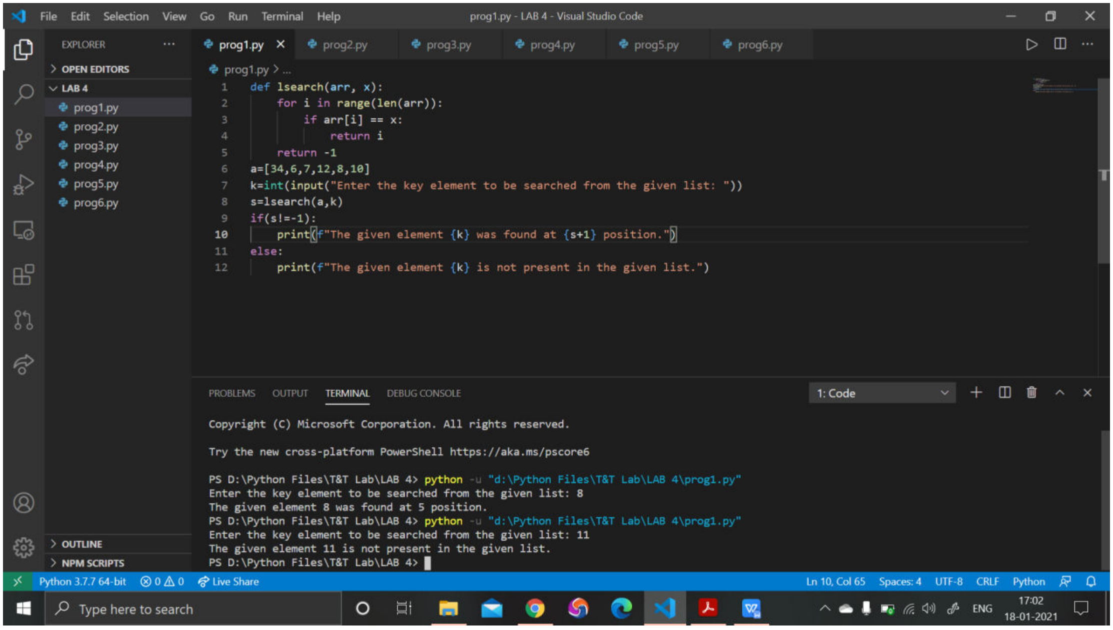
T&T LAB-4
BISWARUP MUKHERJEE
ROLL - 1806468

1. WAP to implement linear search in python.


File Edit Selection View Go Run Terminal Help
prog1.py - LAB 4 - Visual Studio Code
EXPLORER
OPEN EDITORS
LAB 4
prog1.py
prog2.py
prog3.py
prog4.py
prog5.py
prog6.py
OUTLINE
NPM SCRIPTS
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: Code
Python 3.7.7 64-bit ⊗ 0 ⚠ 0 Live Share
Ln 10, Col 65 Spaces: 4 UTF-8 CRLF Python
Type here to search
ENG
17:02
18-01-2021

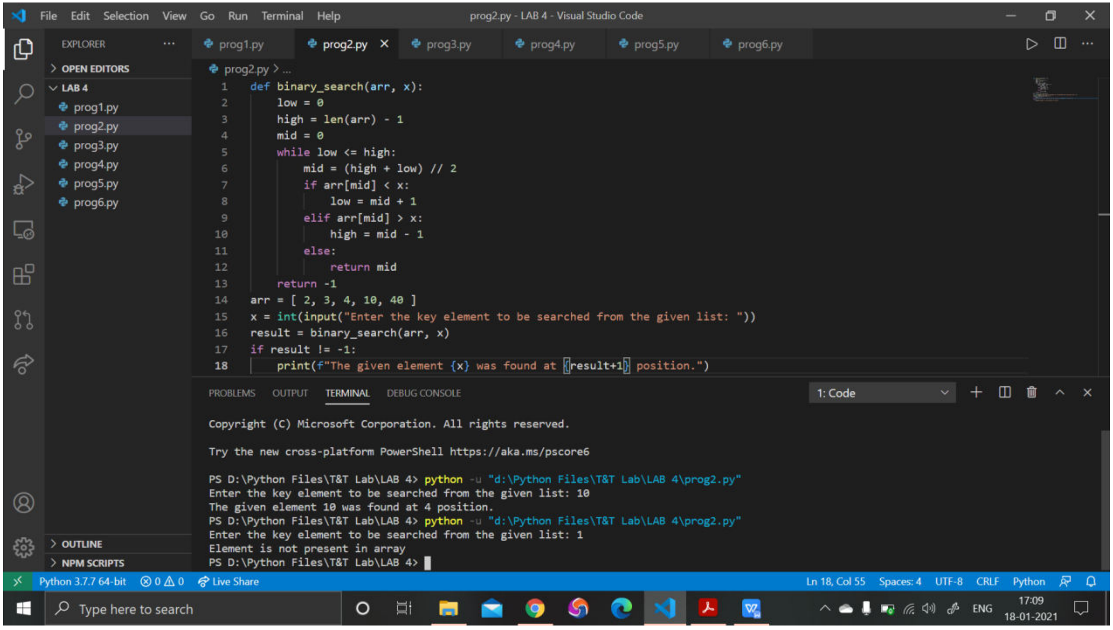
```python
def lsearch(arr, x):
    for i in range(len(arr)):
        if arr[i] == x:
            return i
    return -1
a=[34,6,7,12,8,10]
k=int(input("Enter the key element to be searched from the given list: "))
s=lsearch(a,k)
if(s!=-1):
    print(f"The given element {k} was found at {s+1} position.")
else:
    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.


File Edit Selection View Go Run Terminal Help
prog2.py - LAB 4 - Visual Studio Code
EXPLORER
OPEN EDITORS
LAB 4
prog1.py
prog2.py
prog3.py
prog4.py
prog5.py
prog6.py
OUTLINE
NPM SCRIPTS
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
1: Code
Python 3.7.7 64-bit ⊗ 0 ⚠ 0 Live Share
Ln 18, Col 55 Spaces: 4 UTF-8 CRLF Python
Type here to search
ENG
17:09
18-01-2021

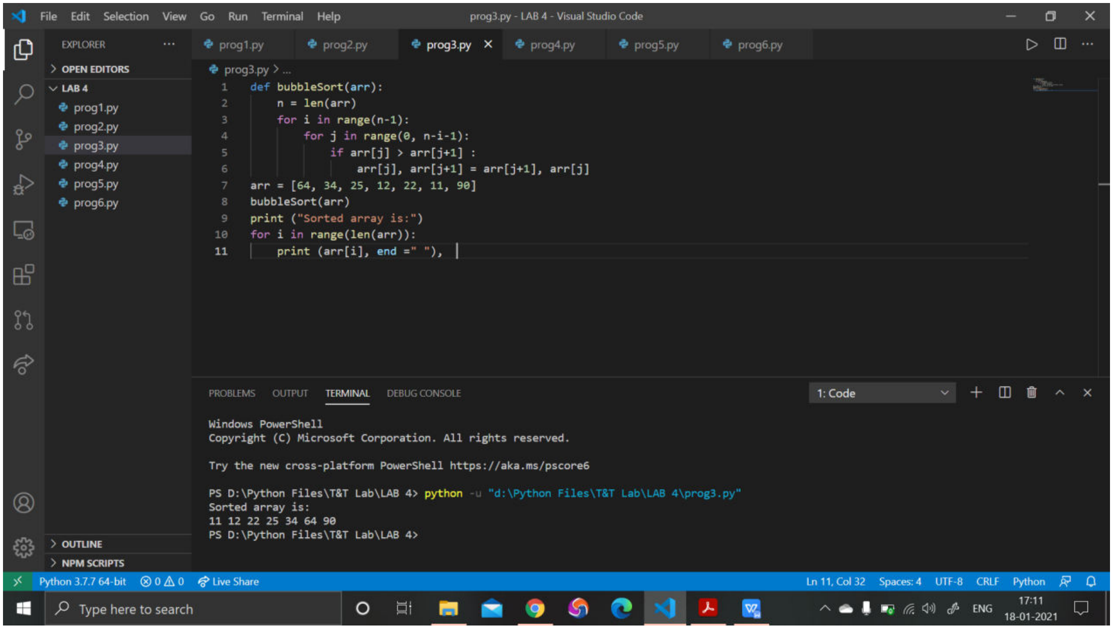
```python
def binary_search(arr, x):
    low = 0
    high = len(arr) - 1
    mid = 0
    while low <= high:
        mid = (high + low) // 2
        if arr[mid] < x:
            low = mid + 1
        elif arr[mid] > x:
            high = mid - 1
        else:
            return mid
    return -1
arr = [ 2, 3, 4, 10, 40 ]
x = int(input("Enter the key element to be searched from the given list: "))
result = binary_search(arr, x)
if result != -1:
    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.



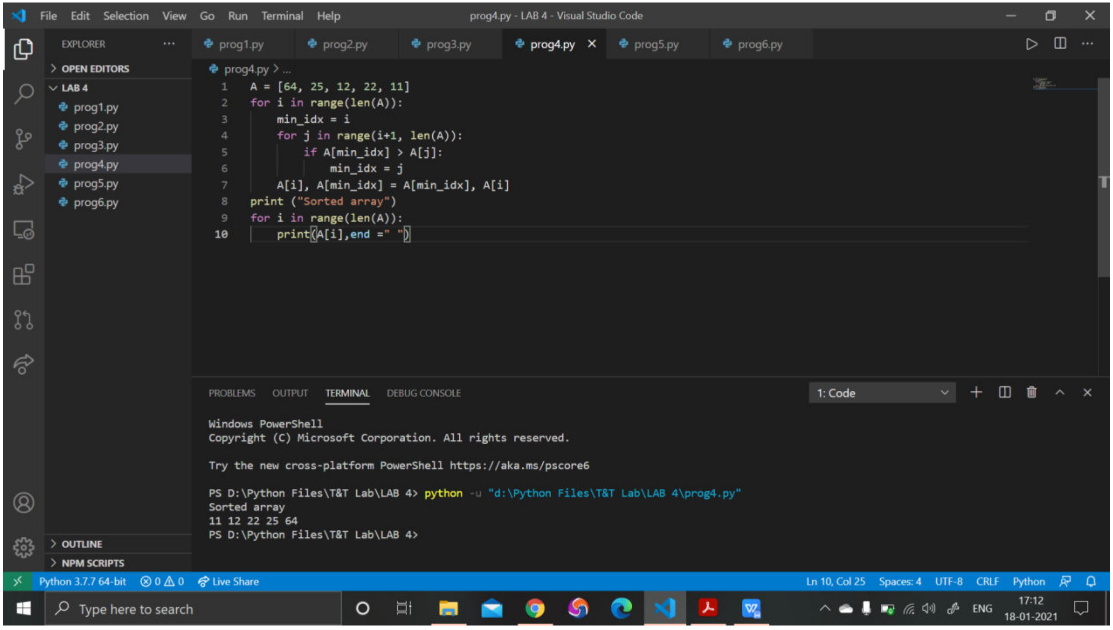
```python
def bubbleSort(arr):
    n = len(arr)
    for i in range(n-1):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1] :
                arr[j], arr[j+1] = arr[j+1], arr[j]
arr = [64, 34, 25, 12, 22, 11, 90]
bubbleSort(arr)
print ("Sorted array is:")
for i in range(len(arr)):
    print (arr[i], end =" "),
```

```
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 90
PS D:\Python Files\T&T Lab\LAB 4>
```

4. WAP to implement selection sort in python.



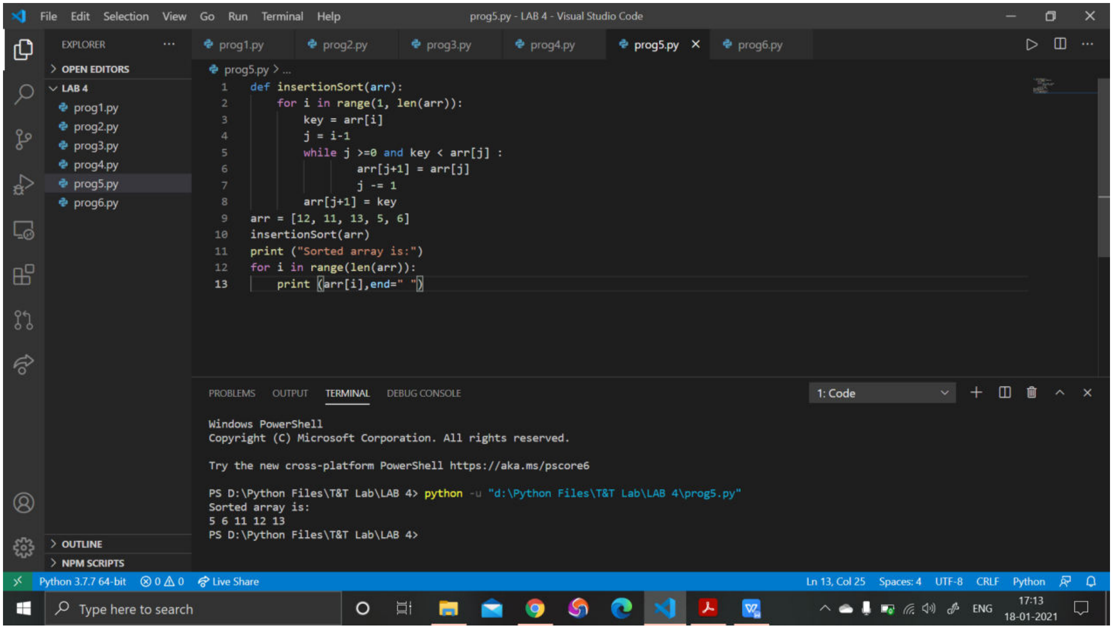
```python
A = [64, 25, 12, 22, 11]
for i in range(len(A)):
    min_idx = i
    for j in range(i+1, len(A)):
        if A[min_idx] > A[j]:
            min_idx = j
    A[i], A[min_idx] = A[min_idx], A[i]
print ("Sorted array")
for i in range(len(A)):
    print(A[i],end =" ")
```

```
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.



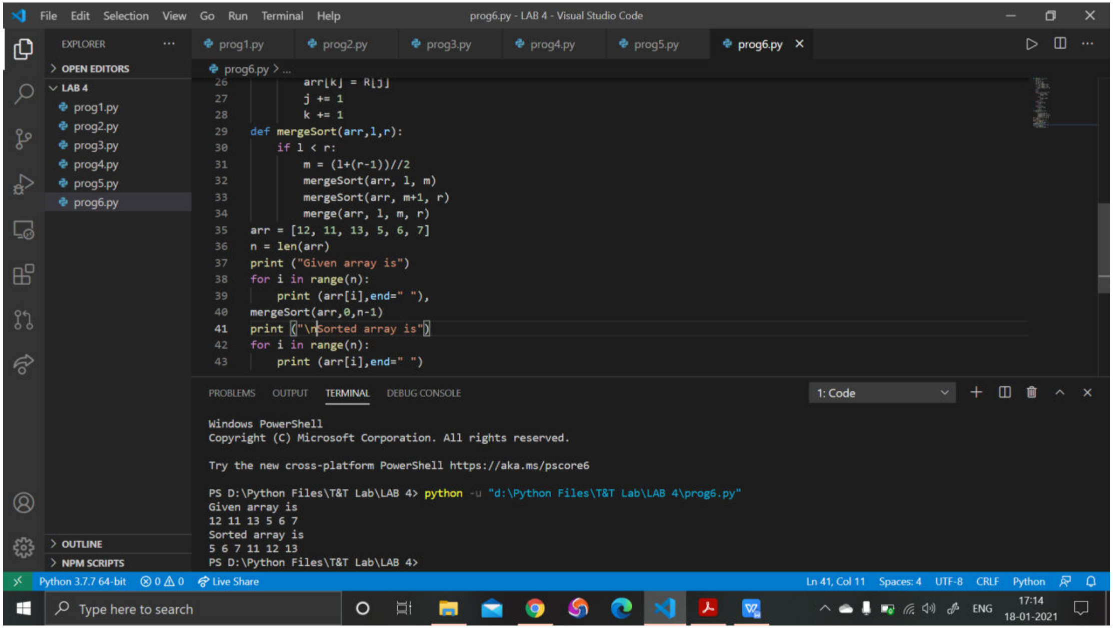
```python
def insertionSort(arr):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i-1
        while j >=0 and key < arr[j] :
                arr[j+1] = arr[j]
                j -= 1
        arr[j+1] = key
arr = [12, 11, 13, 5, 6]
insertionSort(arr)
print ("Sorted array is:")
for i in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

```
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\prog5.py"
Sorted array is:
5 6 11 12 13
PS D:\Python Files\T&T Lab\LAB 4>
```

6. WAP to implement merge sort in python.



```python
            arr[k] = R[j]
            j += 1
            k += 1
def mergeSort(arr,l,r):
    if l < r:
        m = (l+(r-1))//2
        mergeSort(arr, l, m)
        mergeSort(arr, m+1, r)
        merge(arr, l, m, r)
arr = [12, 11, 13, 5, 6, 7]
n = len(arr)
print ("Given array is")
for i in range(n):
    print (arr[i],end=" "),
mergeSort(arr,0,n-1)
print ("\nSorted array is")
for i in range(n):
    print (arr[i],end=" ")
```

```
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\prog6.py"
Given array is
12 11 13 5 6 7
Sorted array is
5 6 7 11 12 13
PS D:\Python Files\T&T Lab\LAB 4>
```