Page No. Date

Assignment 6

class - SE IV ROll NO - 21430

Batch - F4. DOS - 7/10/2020

Problem Statement -:

Write a python program to store first year percentage of student in array of floating point numbers in ascending order using quick sort and display top five scores.

Learning objectives -:

1. TO learn to write simple python program & execute it.
2. TO implement quick sort.

Learning outcomes -:

1. Will be able to implement array data structure
2. Will be able to implement quick sort to sort different elements

S/W and H/W requirements -:

1. Python IDE community version 2020.
2. Window 10 64 bit.

Theory -:

Quick Sort -:

Quick Sort is a highly efficient efficiency algorithm and is based on partioning on array of data into smaller arrays. A large array is participated into 2 arrays one of which holds values smaller than the specified value say pivot, based on which the partition is made and another array holds values greater than the pivot value

* Quick Sort pivot Algorithm -:

Steps:

1. choose highest index value has pivot.
2. Take two variables to point left and right of list excluding pivot.
3. left points to low index.
4. right point to high index.
5. While value at left is less than the pivot move right.
6. While value at right is greater than pivot move left.
7. If both step 5 and step 6 does not match swapleft and right.
8. If left >, right the point where they is new pivot.

Quick Sort Algorithm.

1. Make right - most index value pivot.
2. Partition array using pivot value.
3. Quicksort left partition recursively.
4. Quicksort right partition recursively.

* Pseudocode -:

ADT representation for class search:

```
Class search:
        array = []        # initialize array
        getdata()
          # taking input from user
        Show()
          # showing percentage of students
        partition()
          # making partition for quicksort
        quicksort()
          # sorting function to sort percentage

Main() function

Students = search()
Students. getdata()
students. Show()
students. quicksort (0, len(arr), -1)
i from -1 to 6 by decreminting by -2
        display. Students. arr [i]

procedure quicksort ( self, start, end)
      if  start >= end
            return
      p = self . partition (surt, end)
      self. quicksort ( start, p-1)
      self. quicksort (p+1, end)

END QuickSort.
```

Test Case

NO.	Description	Input	Expected output	Actual output	Result
1.	Enter no. of students	5 30.4 89.2 78.5 87.3 65.2	sorted list top 5	sorted list top 5	Pass

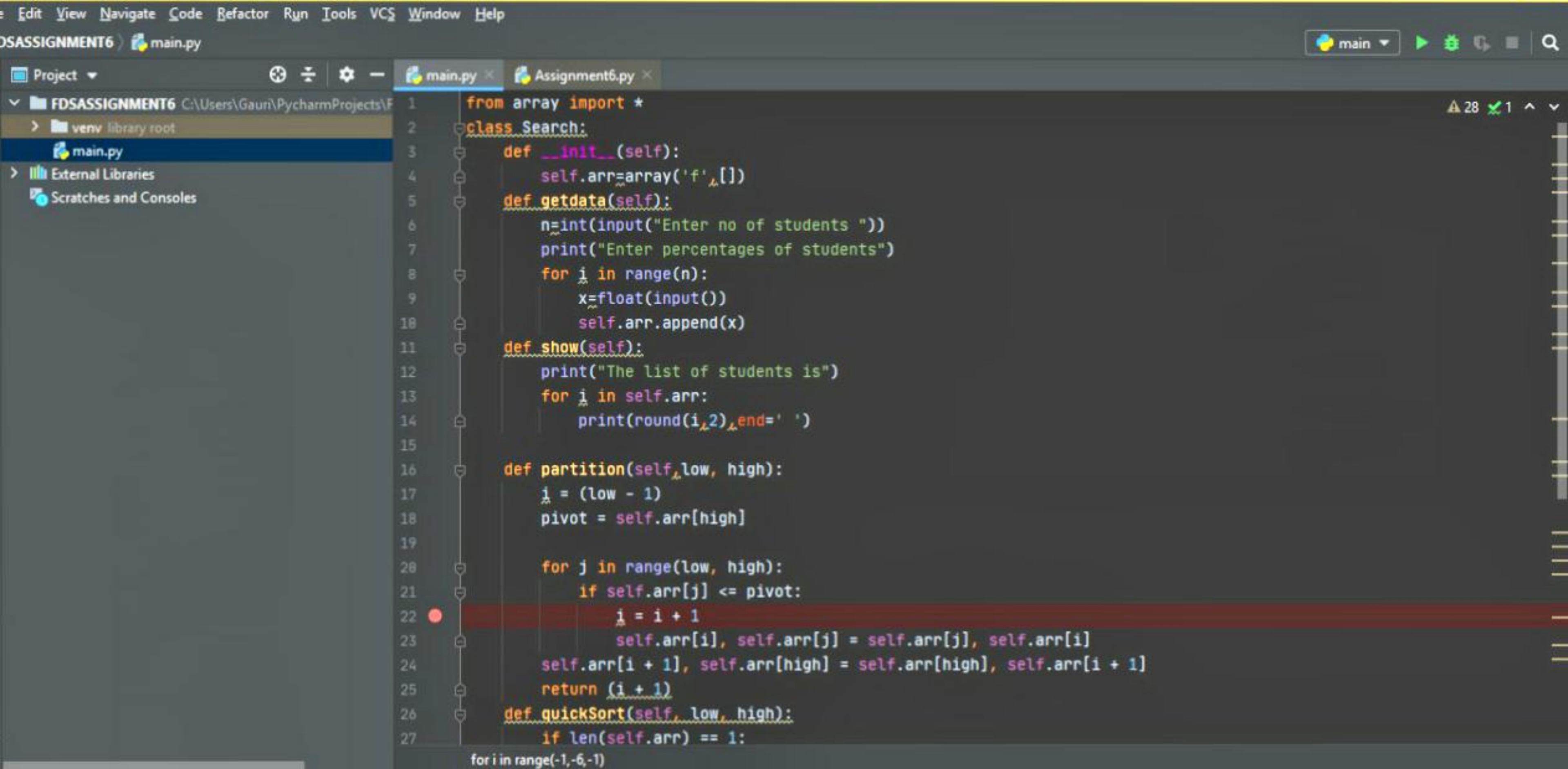
sorted list = [30.4, 65.2, 78.5. 87.3. 89.2].

Conclusion-:
We learnt to implement quick sort algorithm and as well as understand the concept of recursion

* Time Complexity-:

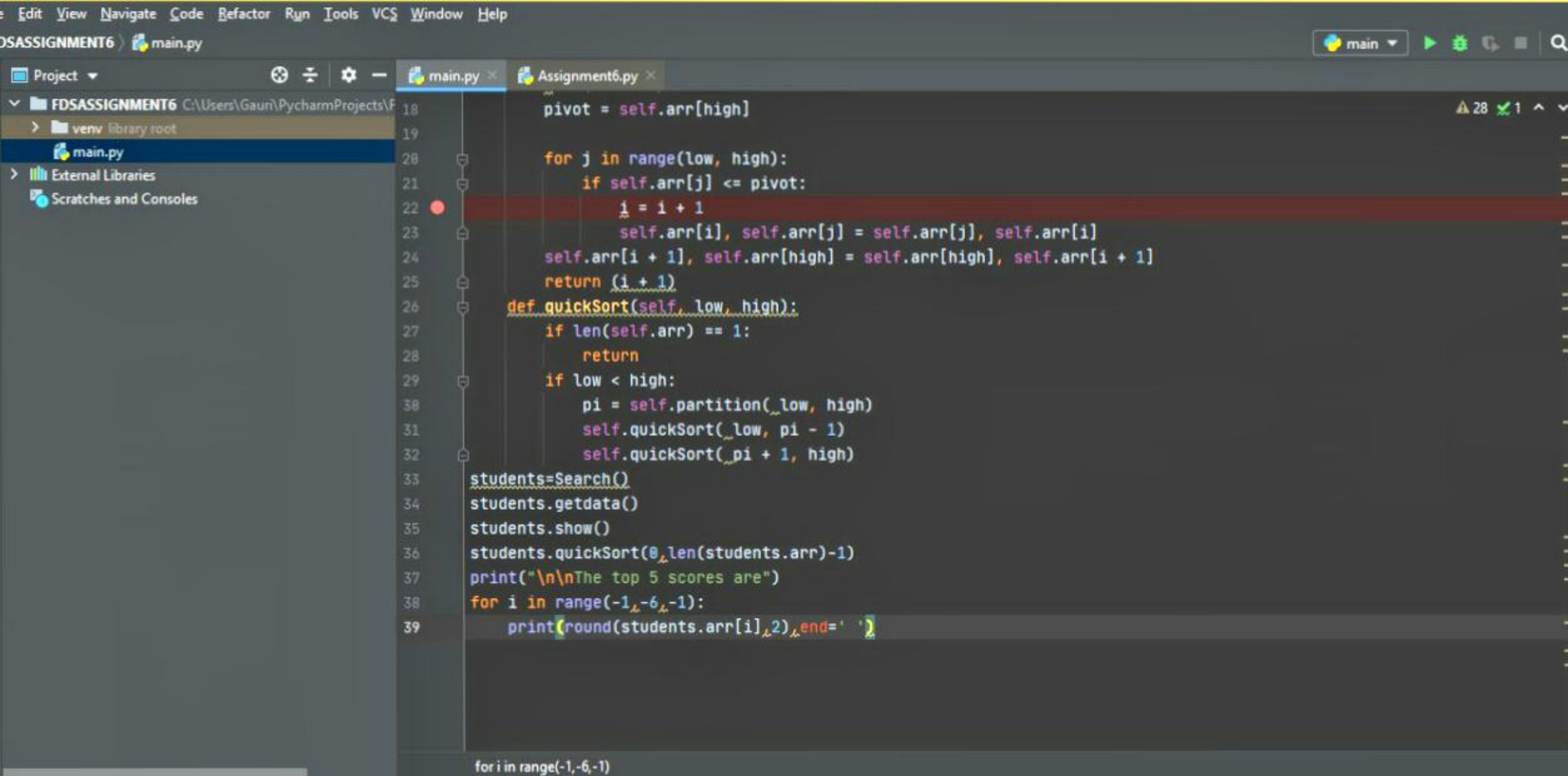
Worst Case	Best Case	Average Case
$O(n^2)$	$O(n\log n)$	$O(n/\log n)$

* Space Complexity - $O(n\log n)$

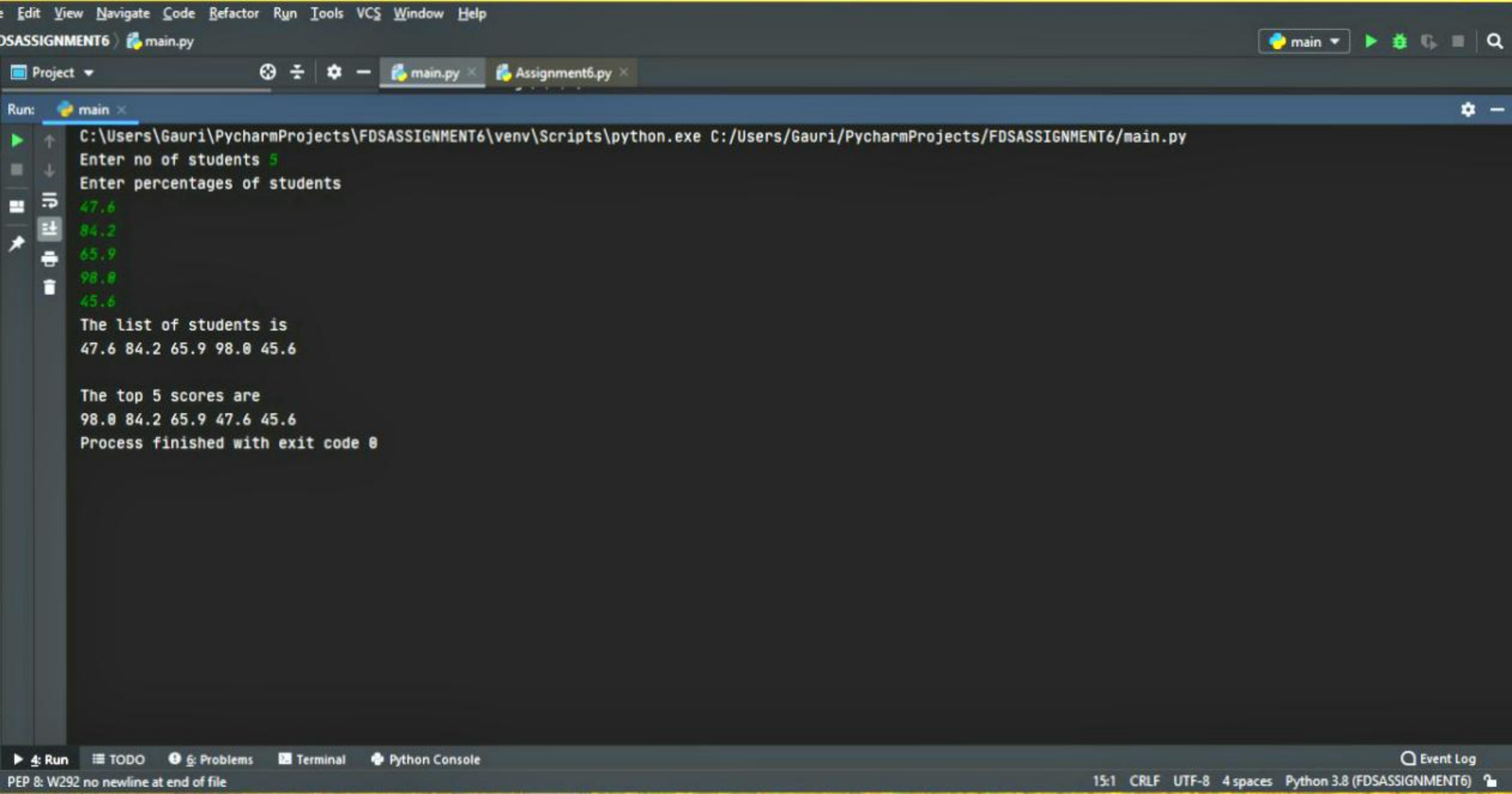


```python
from array import *
class Search:
    def __init__(self):
        self.arr=array('f',[])
    def getdata(self):
        n=int(input("Enter no of students "))
        print("Enter percentages of students")
        for i in range(n):
            x=float(input())
            self.arr.append(x)
    def show(self):
        print("The list of students is")
        for i in self.arr:
            print(round(i,2),end=' ')

    def partition(self,low, high):
        i = (low - 1)
        pivot = self.arr[high]

        for j in range(low, high):
            if self.arr[j] <= pivot:
                i = i + 1
                self.arr[i], self.arr[j] = self.arr[j], self.arr[i]
        self.arr[i + 1], self.arr[high] = self.arr[high], self.arr[i + 1]
        return (i + 1)
    def quickSort(self, low, high):
        if len(self.arr) == 1:
```

for i in range(-1,-6,-1)



```
        pivot = self.arr[high]

        for j in range(low, high):
            if self.arr[j] <= pivot:
                i = i + 1
                self.arr[i], self.arr[j] = self.arr[j], self.arr[i]
        self.arr[i + 1], self.arr[high] = self.arr[high], self.arr[i + 1]
        return (i + 1)
    def quickSort(self, low, high):
        if len(self.arr) == 1:
            return
        if low < high:
            pi = self.partition(low, high)
            self.quickSort(low, pi - 1)
            self.quickSort(pi + 1, high)
students=Search()
students.getdata()
students.show()
students.quickSort(0,len(students.arr)-1)
print("\n\nThe top 5 scores are")
for i in range(-1,-6,-1):
    print(round(students.arr[i],2),end=' ')
```



```
C:\Users\Gauri\PycharmProjects\FDSASSIGNMENT6\venv\Scripts\python.exe C:/Users/Gauri/PycharmProjects/FDSASSIGNMENT6/main.py
Enter no of students 5
Enter percentages of students
47.6
84.2
65.9
98.0
45.6
The list of students is
47.6 84.2 65.9 98.0 45.6

The top 5 scores are
98.0 84.2 65.9 47.6 45.6
Process finished with exit code 0
```

PEP 8: W292 no newline at end of file

15:1 CRLF UTF-8 4 spaces Python 3.8 (FDSASSIGNMENT6)