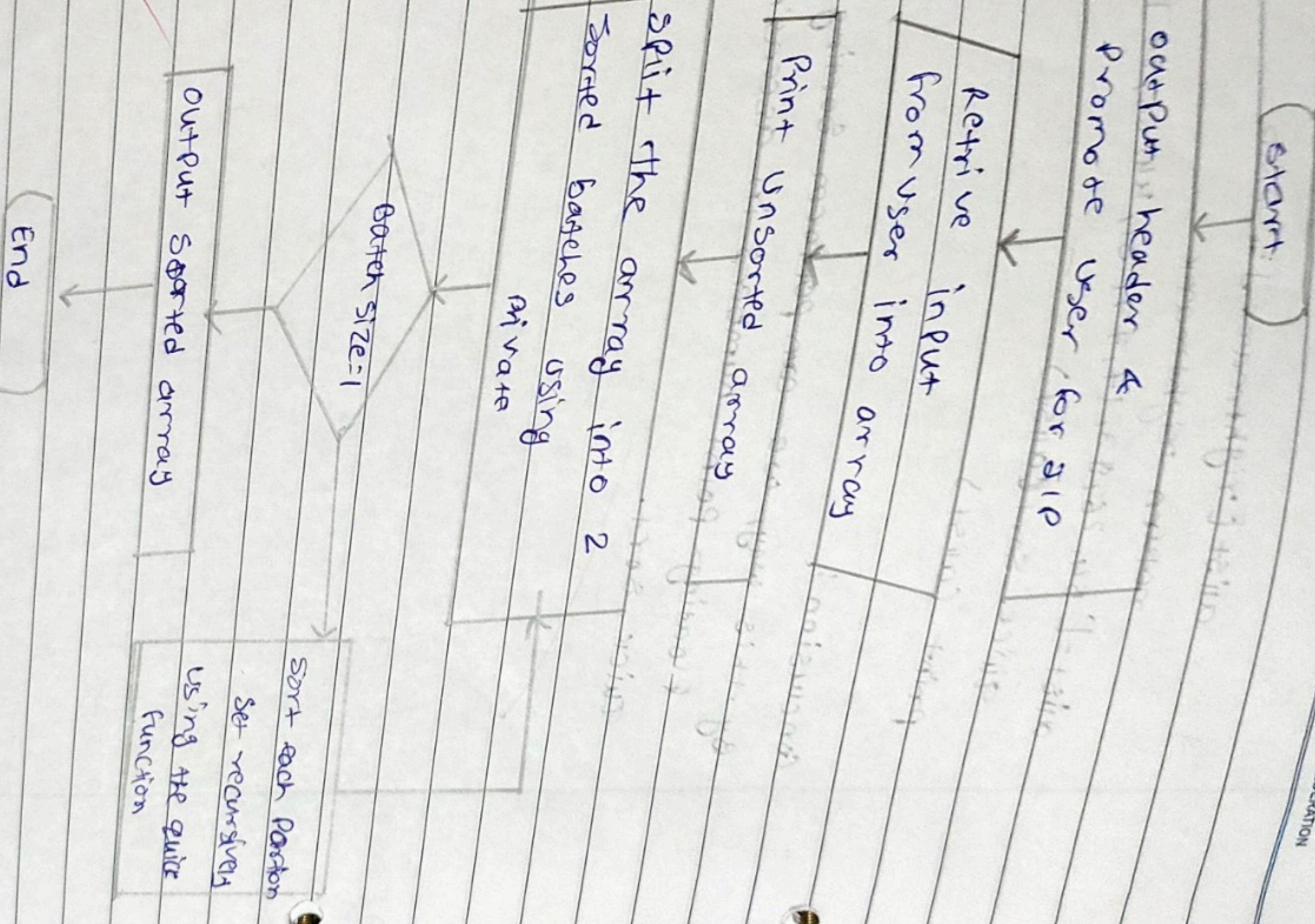




Flowchart :





alist[rightmark] = temp

return rightmark.

alist = [54, 26, 19, 31, 17, 77, 39, 45, 55, 20]

Quick Sort(alist) 26 39 45 54 55 77 17 19 31 39

print (alist)

array divided

Conclusion: array divided

By this way, we can perform sorting array of floating point numbers in ascending order using quick sort.

~~26 39 45 54 55 77 17 19 31 39~~

26 39 45 54 55 77 17 19 31 39

26 39 45 54 55 77 17 19 31 39

26 39 45 54 55 77 17 19 31 39

26 39 45 54 55 77 17 19 31 39

26 39 45 54 55 77 17 19 31 39

26 39 45 54 55 77 17 19 31 39

26 39 45 54 55 77 17 19 31 39

26 39 45 54 55 77 17 19 31 39

26 39 45 54 55 77 17 19 31 39

26 39 45 54 55 77 17 19 31 39

26 39 45 54 55 77 17 19 31 39

26 39 45 54 55 77 17 19 31 39



Algorithm:

```
def quick sort (alist) :
    if quick sort (alist) == len (alist) - 1 :
        quick sort helper (alist, 0, len (alist) - 1)
    else :
        if first < last :
            quick sort helper (alist, first, last)
            quick sort helper (alist, first + 1, last)
        def partition (alist, first, last) :
            pivotvalue = alist [first]
            leftmark = first + 1
            rightmark = last
            done = false
            while not done :
                while leftmark = rightmark and alist [leftmark] <
                    leftmark = leftmark + 1
                while alist [rightmark] >= pivotvalue and rightmark >= leftmark :
                    rightmark = rightmark - 1
                if rightmark < leftmark :
                    done = true
                    temp = alist [first]
                    alist [first] = alist [rightmark]
                    alist [rightmark] = temp
                else :
                    temp = alist [leftmark]
                    alist [leftmark] = alist [rightmark]
                    alist [rightmark] = temp
            else :
                temp = alist [first]
                alist [first] = alist [rightmark]
                alist [rightmark] = temp
```



Practical NO: 6 (B)

- Title:- sorting array of floating point numbers in ascending order - using quick sort.
- Aim:- write a Python program to store first year percentage of students in array. write function for sorting array of floating point numbers in ascending order using quick sort and display top five scores.
- Pre-requisite:- knowledge of python programming.
- Knowledge of sorting technique using quick sort.
- objective:-
- To sort array using quick sort.
- Input:-
Size of array :- 5
First year Percentage of students.
out come:-
To sort array using quick sort
To display top five scores.
- Theory:-
Explain concept of quick sort in details.
- Advantages and disadvantages
- Example of quick sort.
- Time complexity.