```
'''
//============================================================================
// Name        : 21465_Pract6.py
// Author      : Chaitanya Paraskar
// Roll No.    : 21465
// 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.
//============================================================================

'''


class Array:
    def __init__(self):
        self.n = int(input("Enter Total No. of elements in Array : "))
        self.arr = [-1] * self.n
        for i in range(0, self.n):
            e = int(input(f"Enter {i}th element : "))
            self.arr[i] = e
        print(f"Entered array = {self.arr}")
        self.i = 0
        self.quicksort(0, self.n-1)

    def partition(self, arr, low, high):
        pivot = arr[high]
        i = low - 1
        for j in range(low, high):
            if arr[j] <= pivot:
                i = i + 1
                (arr[i], arr[j]) = (arr[j], arr[i])
        (arr[i + 1], arr[high]) = (arr[high], arr[i + 1])
        return i + 1

    def quicksort(self, low, high):
        self.i = self.i + 1
        print(f"Array after Recursive Call {self.i} : ",
              self.arr, " low = ", low, " high = ", high)
        if low < high:
            pi = self.partition(self.arr, low, high)
            self.quicksort(low, pi - 1)
            self.quicksort(pi + 1, high)


a = Array()


'''
OUTPUT:

$ python pract6.py
```

```
Enter Total No. of elements in Array : 5
Enter 0th element : 5
Enter 1th element : 4
Enter 2th element : 3
Enter 3th element : 2
Enter 4th element : 1
Entered array = [5, 4, 3, 2, 1]
Array after Recursive Call 1 :  [5, 4, 3, 2, 1]  low =  0  high =  4
Array after Recursive Call 2 :  [1, 4, 3, 2, 5]  low =  0  high =  -1
Array after Recursive Call 3 :  [1, 4, 3, 2, 5]  low =  1  high =  4
Array after Recursive Call 4 :  [1, 4, 3, 2, 5]  low =  1  high =  3
Array after Recursive Call 5 :  [1, 2, 3, 4, 5]  low =  1  high =  0
Array after Recursive Call 6 :  [1, 2, 3, 4, 5]  low =  2  high =  3
Array after Recursive Call 7 :  [1, 2, 3, 4, 5]  low =  2  high =  2
Array after Recursive Call 8 :  [1, 2, 3, 4, 5]  low =  4  high =  3
Array after Recursive Call 9 :  [1, 2, 3, 4, 5]  low =  5  high =  4

'''
```