-----------------------------------------------------

Assignment 4:

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.

-----------------------------------------------------

|  |
| --- |
| """  -----------------------------------------------------  Assignment 4:  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.  -----------------------------------------------------  """  # ====================================================================  # concepts learned  # Quick sort is a recursive method [method that calls itself]  # Divide - and - conquer algorithm  # very efficient for large data sets  # Big Oh Analysis -  # Worst case - O(n^2)  # Average case is - O(nlogn)  # Performance depends on pivot selection  # ====================================================================  def insert(A, marks\_of\_students): # taking user inputs  print("Enter students marks in array: ")  for i in range(marks\_of\_students):  num = float(input(f"Enter Mark{i+1}: "))  A.append(num)  def display(A, marks\_of\_students): # displaying sorted output  for i in range(marks\_of\_students):  print(A[i])  print()  print("Top five students percentage are: ", A[:-6: -1])  def quick\_sort(A, marks\_of\_students):  quick\_sort2(A, 0, len(A) - 1)  print("============= Quick Sort ===============")  print("Sorted marks of students: ")  display(A, marks\_of\_students)  def quick\_sort2(A, low, high):  if low < high:  p = partition(A, low, high)  quick\_sort2(A, low, p - 1)  quick\_sort2(A, p + 1, high)  def get\_pivot(A, low, high):  mid = (high + low) // 2  pivot = high  if A[low] < A[mid]:  if A[mid] < A[high]:  pivot = mid  elif A[low] < A[high]:  pivot = low  return pivot  def partition(A, low, high):  pivotIndex = get\_pivot(A, low, high)  pivotValue = A[pivotIndex]  A[pivotIndex], A[low] = A[low], A[pivotIndex]  border = low  for i in range(low, high + 1):  if A[i] < pivotValue:  border += 1  A[i], A[border] = A[border], A[i]  A[low], A[border] = A[border], A[low]  return border  a = [] # array  m = int(input("How many student's percentage you want to store?: "))  insert(a, m)  quick\_sort(a, m) |