TITLE: Using Divide and Conquer Strategies design a class for Concurrent Quick Sort using C++.

THEORY:

Quicksort (sometimes called partition-exchange sort) is an efficient sorting algorithm, serving as a systematic method for placing the elements of an array in order. Developed by Tony Hoare in 1959, with his work published in 1961, it is still a commonly used algorithm for sorting. When implemented well, it can be about two or three times faster than its main competitors, merge sort and heapsort. Quicksort is a comparison sort, meaning that it can sort items of any type for which a "less-than" relation (formally, a total order) is defined. In efficient implementations it is not a stable sort, meaning that the relative order of equal sort items is not preserved. Quicksort can operate in-place on an array, requiring small additional amounts of memory to perform the sorting.

Time Complexity:

- Best Case: O (n log n)
- Worst Case: O(n^2)
- Average Case: O (n log n)

ALGORITHM:

1. Choose an element as pivot.
2. Start indexes at left and (right-1) elements
3. Move left index until we find an element> pivot
4. Move right index until we find an element < pivot
5. If indexes haven’t crossed, swap values and repeat
steps 3 and 4
6. If indexes indexes have crossed crossed, swap pivot and left index
values
7. Call quicksort on the subarrays to the left and right
of the pivot value

```
/********************OUTPUT*****************************

pvg@PC:~/ $ g++ quick.cpp -fopenmp

pvg@PC:~/ $ ./a.out


Enter Number of Elements:7

Enter Elements: 119

90
```

```
85

66

47

39

24


The Entered Array= {119, 90, 85, 66, 47, 39, 24}

24,90,85,66,47,39,119, Thread No. 1

Thread No. 0

24,90,85,66,47,39,119, Thread No. 0

Thread No. 1

24,39,85,66,47,90,119, Thread No. 0

24,39,85,66,47,90,119, Thread No. 1

Thread No. 0

Thread No. 1

24,39,47,66,85,90,119, Thread No. 1

Thread No. 0

24,39,47,66,85,90,119, Thread No. Thread No. 01

24,39,47,66,85,90,119

, Thread No. 1

Thread No. 0

The Sorted Array= {24, 39, 47, 66, 85, 90, 119}

***********************************************************************/
```