Experiment No: 3

Quick Sort

CODE:

```
#include <stdio.h>

void swap(int* a, int* b)

{

	int temp = *a;

	*a = *b;

	*b = temp;

}

int partition(int arr[], int low, int high)

{

	int pivot = arr[low];

	int i = low;

	int j = high;

	while (i < j)

  {

		while (arr[i] <= pivot && i <= high - 1) {

			i++;

		}

		while (arr[j] > pivot && j >= low + 1) {

			j--;

		}

		if (i < j) {

			swap(&arr[i], &arr[j]);

		}
```

```
        }
        swap(&arr[low], &arr[j]);
        return j;
}
void quickSort(int arr[], int low, int high)
{
        if (low < high) {
                int partitionIndex = partition(arr, low, high);
                quickSort(arr, low, partitionIndex - 1);
                quickSort(arr, partitionIndex + 1, high);
        }
}
int main()
{
        int arr[] = { 19, 17, 15, 12, 16, 18, 4, 11, 13 };
        int n = sizeof(arr) / sizeof(arr[0]);
        printf("Original array: ");
        for (int i = 0; i < n; i++) {
                printf("%d ", arr[i]);
        }
        quickSort(arr, 0, n - 1);
        printf("\nSorted array: ");
        for (int i = 0; i < n; i++) {
                printf("%d ", arr[i]);
        }
```

```
    return 0;
}
```

OUTPUT: