SEM 3\Exp8\quick_sort.c

```c
// Function to perform quick sort
int partition(int arr[], int low, int high)
{
    int pivot = arr[high]; // Choosing the rightmost element as pivot
    int i = (low - 1);     // Index of smaller element

    for (int j = low; j < high; j++)
    {
        // If the current element is smaller than or equal to pivot
        if (arr[j] <= pivot)
        {
            i++; // Increment index of smaller element
            int temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
    }
    // Swap the pivot element with the element at i + 1
    int temp = arr[i + 1];
    arr[i + 1] = arr[high];
    arr[high] = temp;
    return i + 1; // Return the partitioning index
}

void quickSort(int arr[], int low, int high)
{
    if (low < high)
    {
        int pi = partition(arr, low, high); // Partitioning index
        quickSort(arr, low, pi - 1);        // Recursively sort elements before partition
        quickSort(arr, pi + 1, high);       // Recursively sort elements after partition
    }
}

```