```c
#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[high]; // choose the rightmost element as the 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 the pivot
        if (arr[j] <= pivot) {
            i++;
            // swap arr[i] and arr[j]
            swap(&arr[i], &arr[j]);
        }
    }

    // swap arr[i+1] and arr[high] (put the pivot element in its correct
position)
    swap(&arr[i + 1], &arr[high]);

    return i + 1;
}

void quickSort(int arr[], int low, int high) {
    if (low < high) {
        // partition the array into two halves and get the index of the pivot
element
        int pi = partition(arr, low, high);

        // recursively sort the subarrays
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main() {
    int n;

    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);
```

```
    int arr[n];

    printf("Enter the elements of the array:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    printf("Given array is \n");
    printArray(arr, n);

    // Perform quicksort
    quickSort(arr, 0, n - 1);

    printf("Sorted array is \n");
    printArray(arr, n);

    return 0;
}
```