

**exp\_11/a.c**

```
1 #include <stdio.h>
2
3 void insertionSort(int arr[], int n) {
4     for (int i = 1; i < n; i++) {
5         int key = arr[i], j = i - 1;
6         while (j >= 0 && arr[j] > key) {
7             arr[j + 1] = arr[j];
8             j--;
9         }
10        arr[j + 1] = key;
11    }
12}
13
14 void selectionSort(int arr[], int n) {
15     for (int i = 0; i < n - 1; i++) {
16         int min_idx = i;
17         for (int j = i + 1; j < n; j++)
18             if (arr[j] < arr[min_idx])
19                 min_idx = j;
20         int temp = arr[i];
21         arr[i] = arr[min_idx];
22         arr[min_idx] = temp;
23     }
24}
25
26 void quickSort(int arr[], int low, int high) {
27     if (low < high) {
28         int pivot = arr[high], i = low - 1;
29         for (int j = low; j < high; j++) {
30             if (arr[j] < pivot) {
31                 i++;
32                 int tmp = arr[i];
33                 arr[i] = arr[j];
34                 arr[j] = tmp;
35             }
36         }
37         int tmp = arr[i+1];
38         arr[i+1] = arr[high];
39         arr[high] = tmp;
40         int pi = i+1;
41         quickSort(arr, low, pi-1);
42         quickSort(arr, pi+1, high);
43     }
44 }
45
46 void merge(int arr[], int l, int m, int r) {
47     int n1 = m - l + 1, n2 = r - m;
48     int L[100], R[100];
```

```
49     for (int i = 0; i < n1; i++) L[i] = arr[l + i];
50     for (int j = 0; j < n2; j++) R[j] = arr[m + 1 + j];
51     int i = 0, j = 0, k = l;
52     while (i < n1 && j < n2)
53         arr[k++] = (L[i] <= R[j]) ? L[i++] : R[j++];
54     while (i < n1) arr[k++] = L[i++];
55     while (j < n2) arr[k++] = R[j++];
56 }
57
58 void mergeSort(int arr[], int l, int r) {
59     if (l < r) {
60         int m = l + (r-l)/2;
61         mergeSort(arr, l, m);
62         mergeSort(arr, m+1, r);
63         merge(arr, l, m, r);
64     }
65 }
66
67 void displayArray(int arr[], int n) {
68     for (int i = 0; i < n; i++)
69         printf("%d ", arr[i]);
70     printf("\n");
71 }
72
73 int main() {
74     int arr[100], n, choice;
75     printf("Enter number of elements: ");
76     scanf("%d", &n);
77     printf("Enter elements: ");
78     for (int i = 0; i < n; i++)
79         scanf("%d", &arr[i]);
80     while (1) {
81         printf("1. Insertion Sort\n2. Selection Sort\n3. Quick Sort\n4. Merge
82 Sort\n0. Exit\nEnter choice: ");
83         scanf("%d", &choice);
84         if (choice == 0) break;
85         switch (choice) {
86             case 1:
87                 insertionSort(arr, n);
88                 printf("Sorted array: ");
89                 displayArray(arr, n);
90                 break;
91             case 2:
92                 selectionSort(arr, n);
93                 printf("Sorted array: ");
94                 displayArray(arr, n);
95                 break;
96             case 3:
97                 quickSort(arr, 0, n-1);
98                 printf("Sorted array: ");
```

```
98         displayArray(arr, n);
99         break;
100        case 4:
101            mergeSort(arr, 0, n-1);
102            printf("Sorted array: ");
103            displayArray(arr, n);
104            break;
105        default:
106            printf("Invalid choice\n");
107        }
108    }
109    return 0;
110}
111
```