Name - Sanjoy Saha

Stream - Computer Science & Engineering

Sec - A

Roll no. - 3

University Roll no. :- 10900120003

Subject - DSA Lab

```c
#include<stdio.h>
#include<stdlib.h>

int n;

int main(){
    int i, choice;

    printf("Enter the number of elements in the array: ");
    scanf("%d",&n);
    int arr[n];

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

    printf("\nThe algorithms to perform sorting: \n");
    printf("1. Quick Sort\n");
    printf("2. Merge Sort\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch(choice) {
        case 1:
            quickSort(arr, 0, n-1);
            break;
        case 2:
            mergeSort(arr, 0, n-1);
            break;
        default:
```

```
30          default:
31              printf("Invalid option.");
32      }
33          printf("\nThe sorted array is:\n");
34      for(i=0; i<n;i++)
35          printf("%d ", arr[i]);
36      return 0;
37  }
38  int quickSort(int a[], int start, int end){
39      if(start>=end)
40      return;
41
42      int i,pivot = a[end], pivotIndex = start;
43      for(i=start; i<end; i++){
44          if(a[i]<=pivot)
45          {
46              int temp = a[i];
47              a[i] = a[pivotIndex];
48              a[pivotIndex] = temp;
49
50              pivotIndex++;
51          }
52      }
53      int temp = a[end];
54      a[end] = a[pivotIndex];
55      a[pivotIndex] = temp;
56
57      quickSort(a, start, pivotIndex-1);
58      quickSort(a, pivotIndex+1, end);
59      return 0;
```

```c
int merge(int a[], int start, int mid, int end){
    int i=start,j=mid+1, temp[7],k=start;
    while(i<=mid && j<=end){

        if(a[i]<a[j]){
        temp[k]=a[i];
        i++;
        }

        else{
        temp[k]=a[j];
        j++;
        }
        k++;
    }

    if(i==mid+1){
        for(; j<=end; j++,k++)
        temp[k]=a[j];
    }
    else if(j==end+1){
        for(; i<=mid; i++,k++)
        temp[k]=a[i];
    }
    for(i=start; i<=end; i++)
    a[i] = temp[i];

    return 0;
}
```

```
int mergeSort(int a[], int start, int end){
    if(start>=end)
    return;

    int mid = (start+end)/2;
    mergeSort(a, start, mid);
    mergeSort(a, mid+1, end);

    merge(a, start, mid, end);
    return 0;
}
```

```
Enter the number of elements in the array: 8
Enter the elements of the array:
900 7 12 -4 1 876 -1000 0

The algorithms to perform sorting:
1. Quick Sort
2. Merge Sort
Enter your choice: 1

The sorted array is:
-1000 -4 0 1 7 12 876 900
PS C:\Users\lenovo\Desktop\C DSA lab>
```

```
Enter the number of elements in the array: 8
Enter the elements of the array:
1000 7 56 -10 0 -66 12 456

The algorithms to perform sorting:
1. Quick Sort
2. Merge Sort
Enter your choice: 2

The sorted array is:
-66 -10 0 7 12 56 458 1001
PS C:\Users\lenovo\Desktop\C DSA lab>
```