Code

```c
#include <stdio.h>

void mergeSort(int arr[], int si, int ei);
void merge(int arr[], int si, int mid, int ei);

int main() {
    int n;
    printf("Enter the size of the array: ");
    scanf("%d", &n);

    int arr[n];

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

    mergeSort(arr, 0, n - 1);

    printf("Array after Merge Sort: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}

void mergeSort(int arr[], int si, int ei) {
    if(si >= ei) {
        return;
    }

    int mid = si + (ei - si) / 2;
    mergeSort(arr, si, mid);
    mergeSort(arr, mid + 1, ei);
    merge(arr, si, mid, ei);
}

void merge(int arr[], int si, int mid, int ei) {
    int temp[ei - si + 1];
    int i = si;
    int j = mid + 1;
    int k = 0;

    while(i <= mid && j <= ei) {
        if(arr[i] < arr[j]) {
            temp[k++] = arr[i++];
        } else {
            temp[k++] = arr[j++];
        }
    }

    while(i <= mid) {
        temp[k++] = arr[i++];
    }

    while(j <= ei) {
        temp[k++] = arr[j++];
```

```
    }

    for (k = 0, i = si; k < ei - si + 1; k++, i++) {
        arr[i] = temp[k];
    }
}
```

Output

```
Enter the size of the array: 8
Enter the elements of the array: 4 6 2 5 7 9 1 3
Array after Merge Sort: 1 2 3 4 5 6 7 9
```

Code

```c
#include <stdio.h>

void quickSort(int arr[], int low, int high);
int partition(int arr[], int low, int high);
void swap(int arr[], int idx1, int idx2);

int main(){
    int n;
    printf("Enter the size of the array: ");
    scanf("%d", &n);

    int arr[n];

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

    quickSort(arr, 0, n-1);

    printf("Array after Quick Sort: ");
    for (int i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }

    return 0;
}

void quickSort(int arr[], int low, int high){
    if(low < high){
        int pIndex = partition(arr, low, high);

        quickSort(arr, low, pIndex-1);
        quickSort(arr, pIndex+1, high);
    }
}

int partition(int arr[], int low, int high){
    int i=low;
    int j=high;
    int pivot = low;

    while(i<j){
        while(arr[i]<=arr[pivot] && i<=high){
            i++;
        }

        while(arr[j]>arr[pivot] && j>=low){
            j--;
        }

        if(i<j){
            swap(arr, i, j);
        }
    }

    swap(arr, pivot, j);
```

```
    return j;
}

void swap(int arr[], int idx1, int idx2){
    int temp = arr[idx1];
    arr[idx1] = arr[idx2];
    arr[idx2] = temp;
}
```

Output

```
Enter the size of the array: 8
Enter the elements of the array: 4 6 2 5 7 9 1 3
Array after Quick Sort: 1 2 3 4 5 6 7 9
```