```
// Eliza Wanye
// HW3

#include <iostream>
using namespace std;

void merge(int* a, int left, int mid, int right);
void mergeSort(int* a, int left, int right);

int main()
{
    int n;

    cout << endl;
    cout << "Enter number of integers: ";
    cin >> n;

    int* arr = new int[n];

    cout << "Enter the integers: ";
    for (int i = 0; i < n; i++)
        cin >> arr[i];

    mergeSort(arr, 0, n - 1);

    cout << "Sorted list: ";
    for (int i = 0; i < n; i++) {
        if (i) cout << " ";
        cout << arr[i];
    }
    cout << endl << endl;

    delete[] arr;
    return 0;
}

void merge(int* a, int left, int mid, int right)
{
    int n1 = mid - left + 1;
    int n2 = right - mid;

    int* L = new int[n1];
    int* R = new int[n2];

    for (int i = 0; i < n1; i++) L[i] = a[left + i];
    for (int j = 0; j < n2; j++) R[j] = a[mid + 1 + j];

    int i = 0, j = 0, k = left;

    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) {
          a[k] = L[i];
          i++;
        } else {
            a[k] = R[j];
            j++;
        }
        k++;
    }
```

```
    while (i < n1) {
    a[k] = L[i];
    i++;
    k++;
    }

    while (j < n2) {
    a[k] = R[j];
    j++;
    k++;
    }

    delete[] L;
    delete[] R;
}

void mergeSort(int* a, int left, int right)
{
    if (left >= right) return;
    int mid = left + (right - left) / 2;
    mergeSort(a, left, mid);
    mergeSort(a, mid + 1, right);
    merge(a, left, mid, right);
}
```