**Assignment – 4**

**Q1. Implementation of Merge Sort and display the working of it.**

#include <stdio.h>  
#include <stdlib.h>  
  
void merge\_sort(int\*, int, int);  
void merge(int\*, int, int, int);  
int\* create\_array(int);  
void display\_array(int\*, int, int);  
  
int merge\_sort\_call = 1, merge\_call = 1;  
int\* temp\_array;  
  
int main() {  
 int size;  
 printf("Enter the size of array: ");  
 scanf("%d", &size);  
 int\* array = create\_array(size);  
 temp\_array = (int\*)malloc(sizeof(int) \* size);  
 merge\_sort(array, 0, size - 1);  
 printf("Final sorted array: ");  
 display\_array(array, 0, size - 1);  
 free(array);  
 free(temp\_array);  
 return 0;  
}  
  
void merge\_sort(int\* array, int low, int high) {  
 printf("Merge Sort call %d with low=%d and high=%d\n", merge\_sort\_call++, low, high);  
 display\_array(array, low, high);  
  
 if (low < high) {  
 int mid = (low + high) / 2;  
 merge\_sort(array, low, mid);  
 merge\_sort(array, mid + 1, high);  
 merge(array, low, mid, high);  
 }  
}

void merge(int\* array, int low, int mid, int high) {  
 printf("Merge call %d with low=%d, mid=%d, high=%d\n", merge\_call++, low, mid, high);  
 printf("Before merge ");  
 display\_array(array, low, high);  
 int i = low, j = mid + 1, k = low;  
  
 while (i <= mid && j <= high) {  
 if (array[i] <= array[j]) {  
 temp\_array[k++] = array[i++];  
 } else {  
 temp\_array[k++] = array[j++];  
 }  
 }  
 while (i <= mid) {  
 temp\_array[k++] = array[i++];  
 }  
 while (j <= high) {  
 temp\_array[k++] = array[j++];  
 }  
 for (k = low; k <= high; k++) {  
 array[k] = temp\_array[k];  
 }  
 printf("After merge ");  
 display\_array(array, low, high);  
}  
  
int\* create\_array(int n) {  
 int\* array = (int\*)malloc(sizeof(int) \* n), i;  
  
 printf("Enter the array elements: ");  
 for (i = 0; i < n; i++) {  
 scanf("%d", &array[i]);  
 }  
 return array;  
}  
  
void display\_array(int\* array, int low, int high) {  
 int i;  
 printf("The array elements are: ");  
  
 for (i = low; i <= high; i++) {  
 printf("%d ", array[i]);  
 }  
 printf("\n\n");  
}

**Output**:

Enter the size of array: 8  
Enter the array elements: 5 23 3 27 8 77 11 9  
Merge Sort call 1 with low=0 and high=7  
The array elements are: 5 23 3 27 8 77 11 9  
  
Merge Sort call 2 with low=0 and high=3  
The array elements are: 5 23 3 27  
  
Merge Sort call 3 with low=0 and high=1  
The array elements are: 5 23  
  
Merge Sort call 4 with low=0 and high=0  
The array elements are: 5  
  
Merge Sort call 5 with low=1 and high=1  
The array elements are: 23  
  
Merge call 1 with low=0, mid=0, high=1  
Before merge The array elements are: 5 23  
  
After merge The array elements are: 5 23  
  
Merge Sort call 6 with low=2 and high=3  
The array elements are: 3 27  
  
Merge Sort call 7 with low=2 and high=2  
The array elements are: 3  
  
Merge Sort call 8 with low=3 and high=3  
The array elements are: 27  
  
Merge call 2 with low=2, mid=2, high=3  
Before merge The array elements are: 3 27  
  
After merge The array elements are: 3 27  
  
Merge call 3 with low=0, mid=1, high=3  
Before merge The array elements are: 5 23 3 27  
  
After merge The array elements are: 3 5 23 27  
  
Merge Sort call 9 with low=4 and high=7  
The array elements are: 8 77 11 9  
  
Merge Sort call 10 with low=4 and high=5  
The array elements are: 8 77  
  
Merge Sort call 11 with low=4 and high=4  
The array elements are: 8  
  
Merge Sort call 12 with low=5 and high=5  
The array elements are: 77  
  
Merge call 4 with low=4, mid=4, high=5  
Before merge The array elements are: 8 77  
  
After merge The array elements are: 8 77  
  
Merge Sort call 13 with low=6 and high=7  
The array elements are: 11 9  
  
Merge Sort call 14 with low=6 and high=6  
The array elements are: 11  
  
Merge Sort call 15 with low=7 and high=7  
The array elements are: 9  
  
Merge call 5 with low=6, mid=6, high=7  
Before merge The array elements are: 11 9  
  
After merge The array elements are: 9 11  
  
Merge call 6 with low=4, mid=5, high=7  
Before merge The array elements are: 8 77 9 11  
  
After merge The array elements are: 8 9 11 77  
  
Merge call 7 with low=0, mid=3, high=7  
Before merge The array elements are: 3 5 23 27 8 9 11 77  
  
After merge The array elements are: 3 5 8 9 11 23 27 77  
  
Final sorted array: The array elements are: 3 5 8 9 11 23 27 77