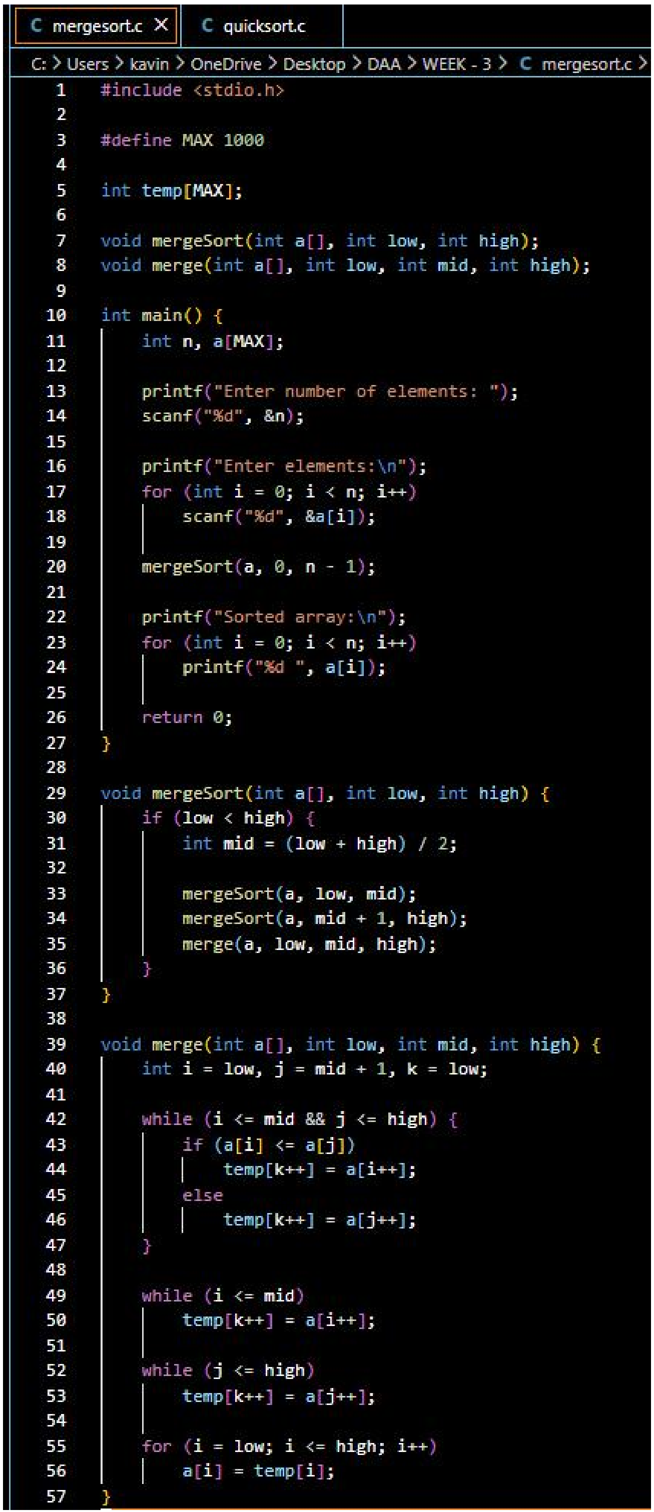
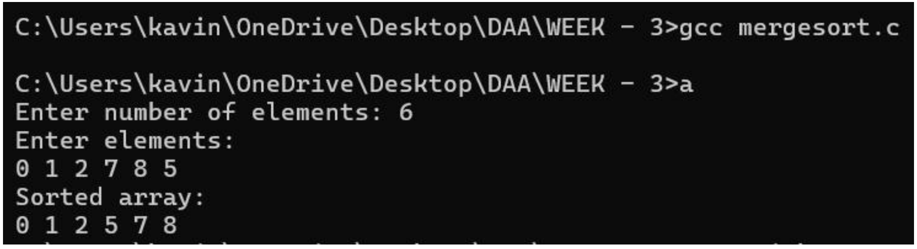
WEEK 3 DAA LAB

Kavin.J.S
CH.SC.U4CSE24119

Merge Sort:

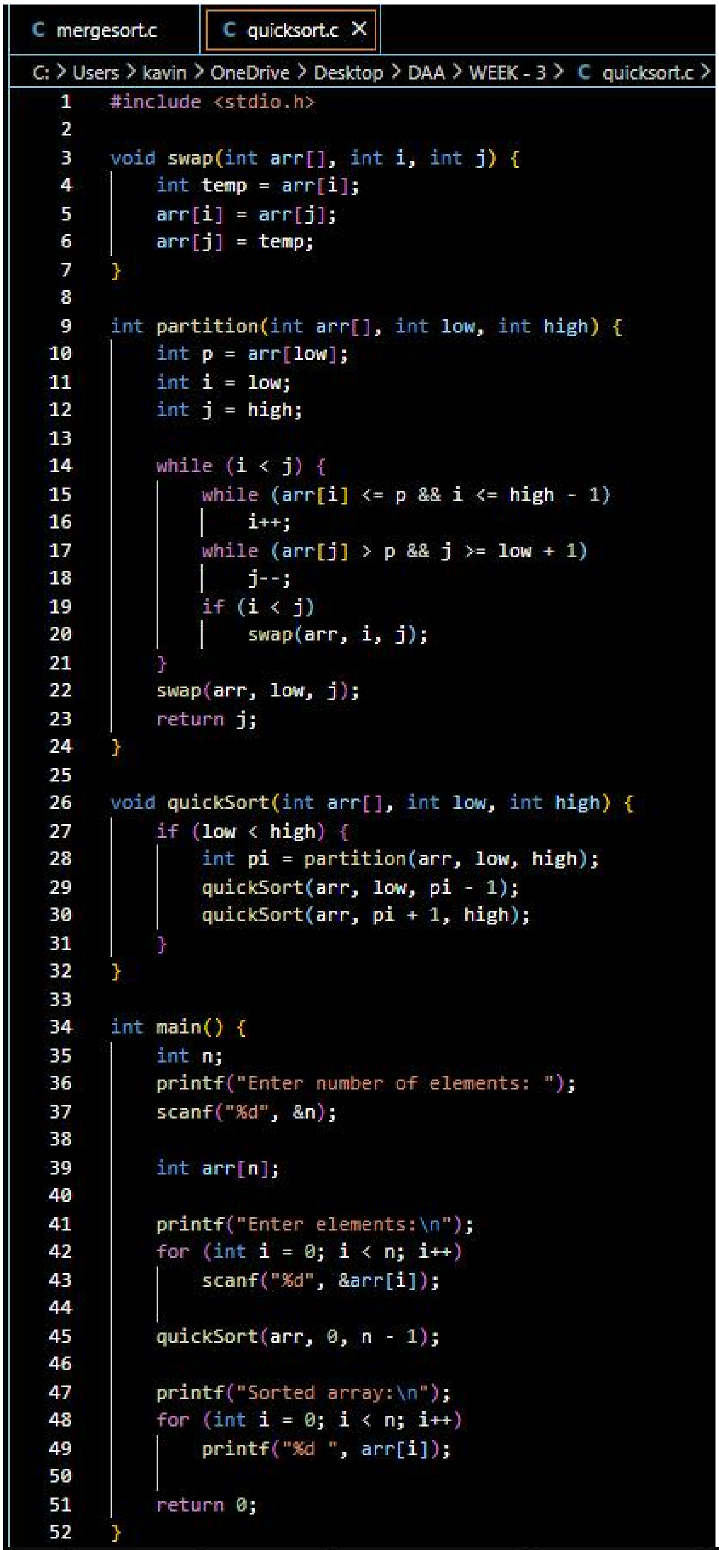

C mergesort.c ×
C quicksort.c
C: > Users > kavin > OneDrive > Desktop > DAA > WEEK - 3 > C mergesort.c >

```c
#include <stdio.h>

#define MAX 1000

int temp[MAX];

void mergeSort(int a[], int low, int high);
void merge(int a[], int low, int mid, int high);

int main() {
    int n, a[MAX];

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter elements:\n");
    for (int i = 0; i < n; i++)
        scanf("%d", &a[i]);

    mergeSort(a, 0, n - 1);

    printf("Sorted array:\n");
    for (int i = 0; i < n; i++)
        printf("%d ", a[i]);

    return 0;
}

void mergeSort(int a[], int low, int high) {
    if (low < high) {
        int mid = (low + high) / 2;

        mergeSort(a, low, mid);
        mergeSort(a, mid + 1, high);
        merge(a, low, mid, high);
    }
}

void merge(int a[], int low, int mid, int high) {
    int i = low, j = mid + 1, k = low;

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

    while (i <= mid)
        temp[k++] = a[i++];

    while (j <= high)
        temp[k++] = a[j++];

    for (i = low; i <= high; i++)
        a[i] = temp[i];
}
```

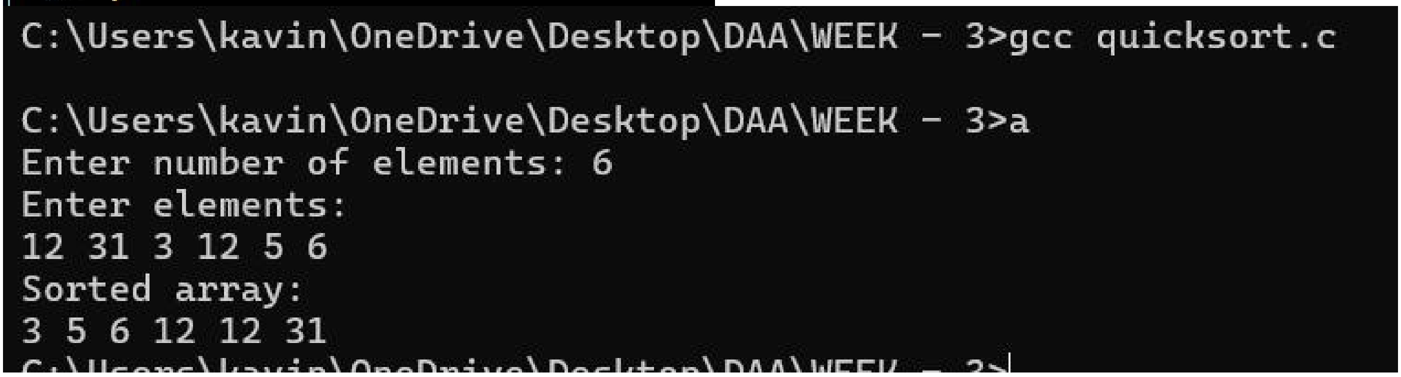


```
C:\Users\kavin\OneDrive\Desktop\DAA\WEEK - 3>gcc mergesort.c

C:\Users\kavin\OneDrive\Desktop\DAA\WEEK - 3>a
Enter number of elements: 6
Enter elements:
0 1 2 7 8 5
Sorted array:
0 1 2 5 7 8
```

Quick Sort:


mergesort.c
quicksort.c
C: > Users > kavin > OneDrive > Desktop > DAA > WEEK - 3 > quicksort.c >

```c
#include <stdio.h>

void swap(int arr[], int i, int j) {
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}

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

    while (i < j) {
        while (arr[i] <= p && i <= high - 1)
            i++;
        while (arr[j] > p && j >= low + 1)
            j--;
        if (i < j)
            swap(arr, i, j);
    }
    swap(arr, low, j);
    return j;
}

void quickSort(int arr[], int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

int main() {
    int n;
    printf("Enter number of elements: ");
    scanf("%d", &n);

    int arr[n];

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

    quickSort(arr, 0, n - 1);

    printf("Sorted array:\n");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);

    return 0;
}
```



```
C:\Users\kavin\OneDrive\Desktop\DAA\WEEK - 3>gcc quicksort.c

C:\Users\kavin\OneDrive\Desktop\DAA\WEEK - 3>a
Enter number of elements: 6
Enter elements:
12 31 3 12 5 6
Sorted array:
3 5 6 12 12 31
C:\Users\kavin\OneDrive\Desktop\DAA\WEEK - 3>
```