AMRITA
VISHWA VIDYAPEETHAM

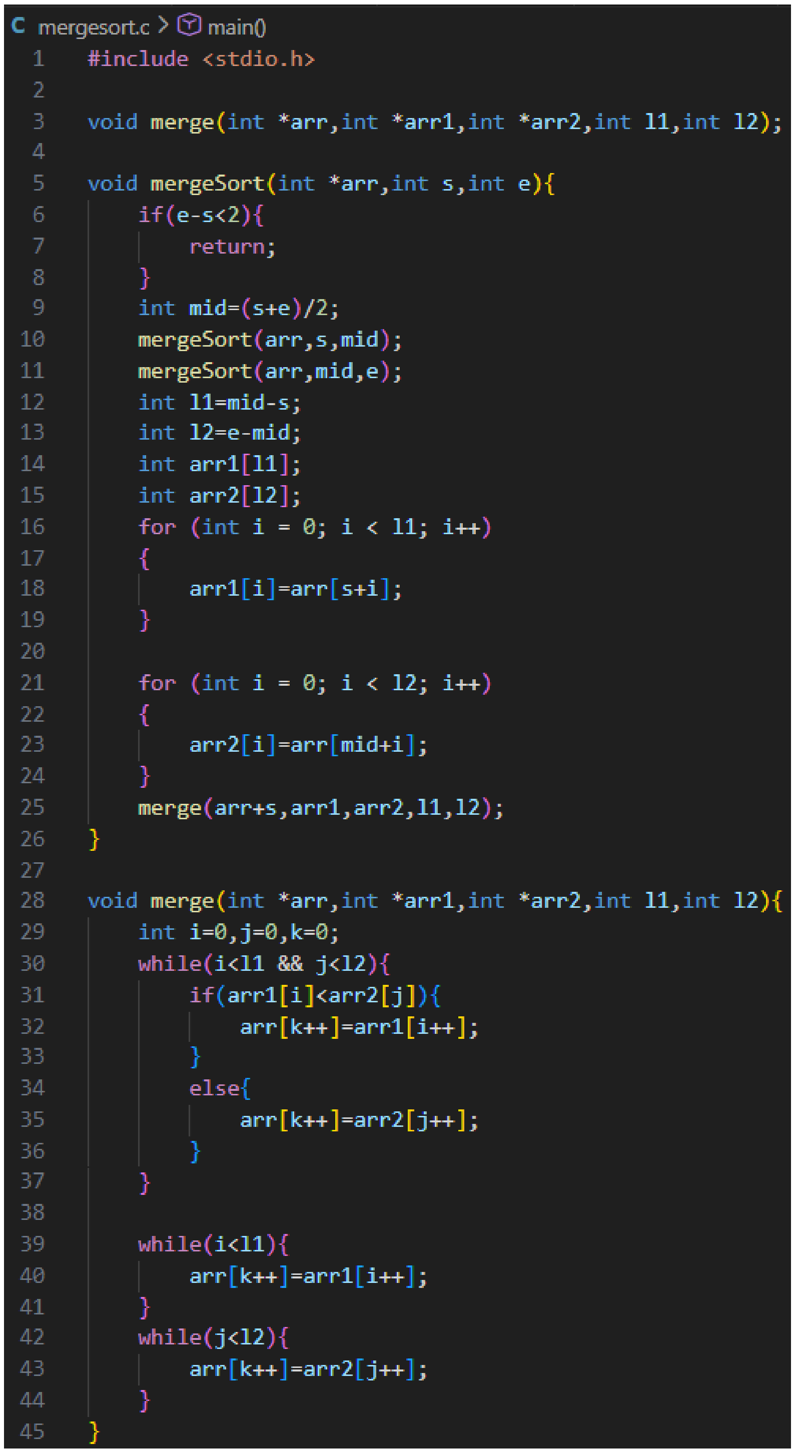
Name: Ganath Avinash G.R

CSE – B

CH.SC.U4CSE24118

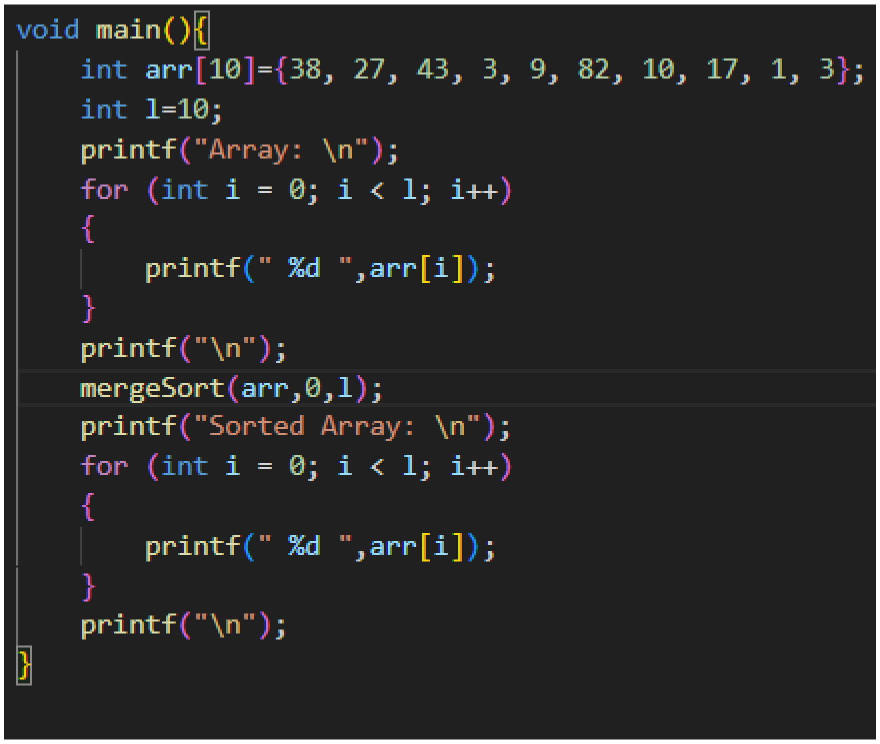
Week –3

1. Merge Sort :



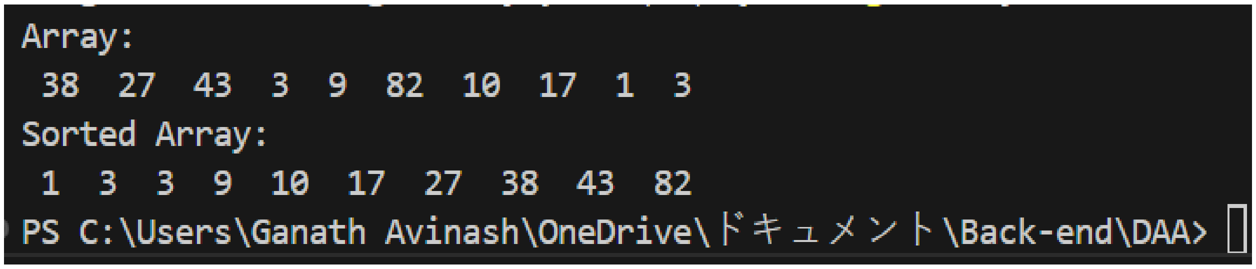
mergesort.c > main()

```c
#include <stdio.h>

void merge(int *arr,int *arr1,int *arr2,int l1,int l2);

void mergeSort(int *arr,int s,int e){
    if(e-s<2){
        return;
    }
    int mid=(s+e)/2;
    mergeSort(arr,s,mid);
    mergeSort(arr,mid,e);
    int l1=mid-s;
    int l2=e-mid;
    int arr1[l1];
    int arr2[l2];
    for (int i = 0; i < l1; i++)
    {
        arr1[i]=arr[s+i];
    }

    for (int i = 0; i < l2; i++)
    {
        arr2[i]=arr[mid+i];
    }
    merge(arr+s,arr1,arr2,l1,l2);
}

void merge(int *arr,int *arr1,int *arr2,int l1,int l2){
    int i=0,j=0,k=0;
    while(i<l1 && j<l2){
        if(arr1[i]<arr2[j]){
            arr[k++]=arr1[i++];
        }
        else{
            arr[k++]=arr2[j++];
        }
    }

    while(i<l1){
        arr[k++]=arr1[i++];
    }
    while(j<l2){
        arr[k++]=arr2[j++];
    }
}
```



```c
void main(){
    int arr[10]={38, 27, 43, 3, 9, 82, 10, 17, 1, 3};
    int l=10;
    printf("Array: \n");
    for (int i = 0; i < l; i++)
    {
        printf(" %d ",arr[i]);
    }
    printf("\n");
    mergeSort(arr,0,l);
    printf("Sorted Array: \n");
    for (int i = 0; i < l; i++)
    {
        printf(" %d ",arr[i]);
    }
    printf("\n");
}
```

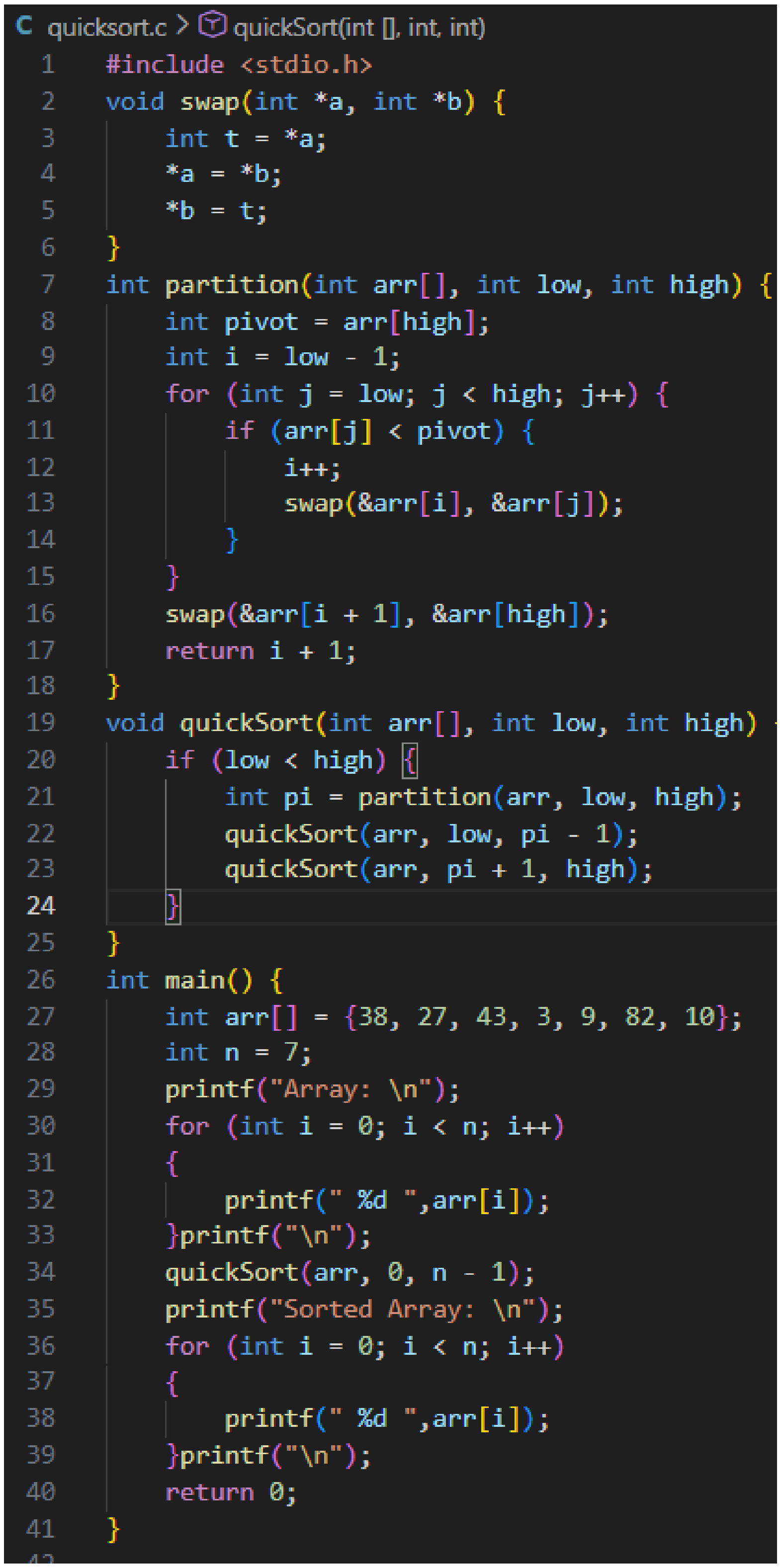
OUTPUT:



```
Array:
 38  27  43  3  9  82  10  17  1  3
Sorted Array:
 1  3  3  9  10  17  27  38  43  82
PS C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\DAA>
```

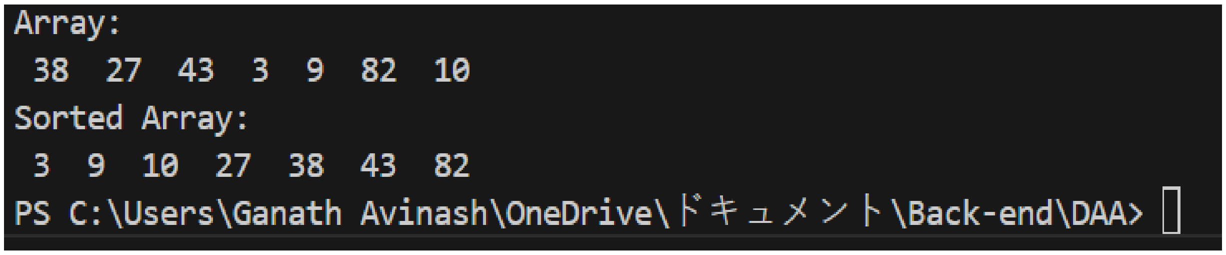
Time Complexity: O(n log n)

2. Quick Sort :



```c
#include <stdio.h>
void swap(int *a, int *b) {
    int t = *a;
    *a = *b;
    *b = t;
}
int partition(int arr[], int low, int high) {
    int pivot = arr[high];
    int i = low - 1;
    for (int j = low; j < high; j++) {
        if (arr[j] < pivot) {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[i + 1], &arr[high]);
    return i + 1;
}
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 arr[] = {38, 27, 43, 3, 9, 82, 10};
    int n = 7;
    printf("Array: \n");
    for (int i = 0; i < n; i++)
    {
        printf(" %d ",arr[i]);
    }printf("\n");
    quickSort(arr, 0, n - 1);
    printf("Sorted Array: \n");
    for (int i = 0; i < n; i++)
    {
        printf(" %d ",arr[i]);
    }printf("\n");
    return 0;
}
```

OUTPUT:



```
Array:
 38  27  43  3  9  82  10
Sorted Array:
 3  9  10  27  38  43  82
PS C:\Users\Ganath Avinash\OneDrive\ドキュメント\Back-end\DAA> 
```

Time Complexity: O(n^2)