Design Analysis and Algorithm – Lab Work

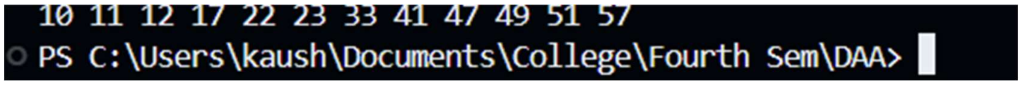
Week 3

Question 1: Write a program to implement merge sort

Code:

```
#include <stdio.h>
void merge(int arr[],int l,int m,int r){
    int n1=m-l+1;
    int n2=r-m;
    int L[n1],R[n2];
    for(int i=0;i<n1;i++)
        L[i]=arr[l+i];
    for(int j=0;j<n2;j++)
        R[j]=arr[m+1+j];
    int i=0,j=0,k=l;
    while(i<n1&&j<n2){
        if(L[i]<=R[j])
            arr[k++]=L[i++];
        else
            arr[k++]=R[j++];
    }
    while(i<n1)
        arr[k++]=L[i++];
    while(j<n2)
        arr[k++]=R[j++];
}
void mergeSort(int arr[],int l,int r){
    if (l<r){
        int m=l+(r-l)/2;
        mergeSort(arr,l,m);
        mergeSort(arr,m+1,r);
        merge(arr,l,m,r);
    }
}
int main(){
    int arr[]={57,10,47,22,11,49,51,41,23,12,17,33};
    int n=sizeof(arr)/sizeof(arr[0]);
    mergeSort(arr,0,n-1);
    for (int i=0;i<n;i++)
        printf("%d ",arr[i]);
    return 0;
}
```

Output:



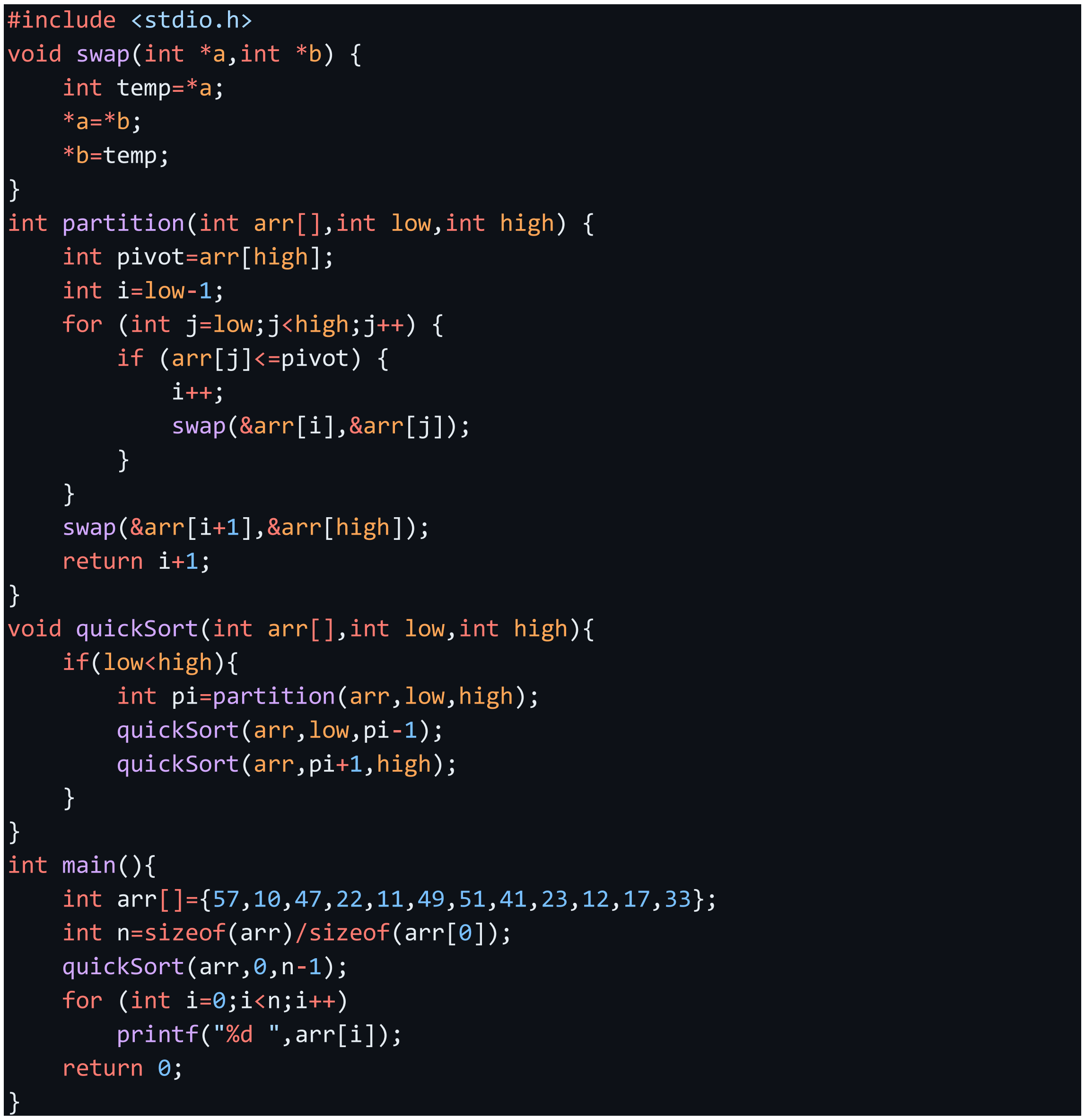
```
10 11 12 17 22 23 33 41 47 49 51 57
PS C:\Users\kaush\Documents\College\Fourth Sem\DAA>
```

Space Complexity: O(n)

Time Complexity: O(nlogn)

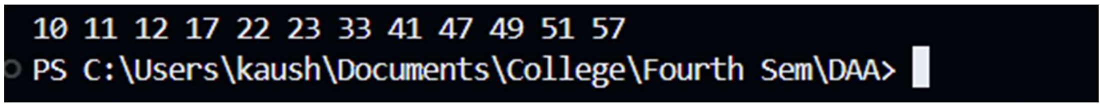
Question 2: Write a program to implement quick sort.

Code:



```c
#include <stdio.h>
void swap(int *a,int *b) {
    int temp=*a;
    *a=*b;
    *b=temp;
}
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[]={57,10,47,22,11,49,51,41,23,12,17,33};
    int n=sizeof(arr)/sizeof(arr[0]);
    quickSort(arr,0,n-1);
    for (int i=0;i<n;i++)
        printf("%d ",arr[i]);
    return 0;
}
```

Output:



```
10 11 12 17 22 23 33 41 47 49 51 57
PS C:\Users\kaush\Documents\College\Fourth Sem\DAA>
```

Space Complexity: O(logn)

Time Complexity: O(n^2)