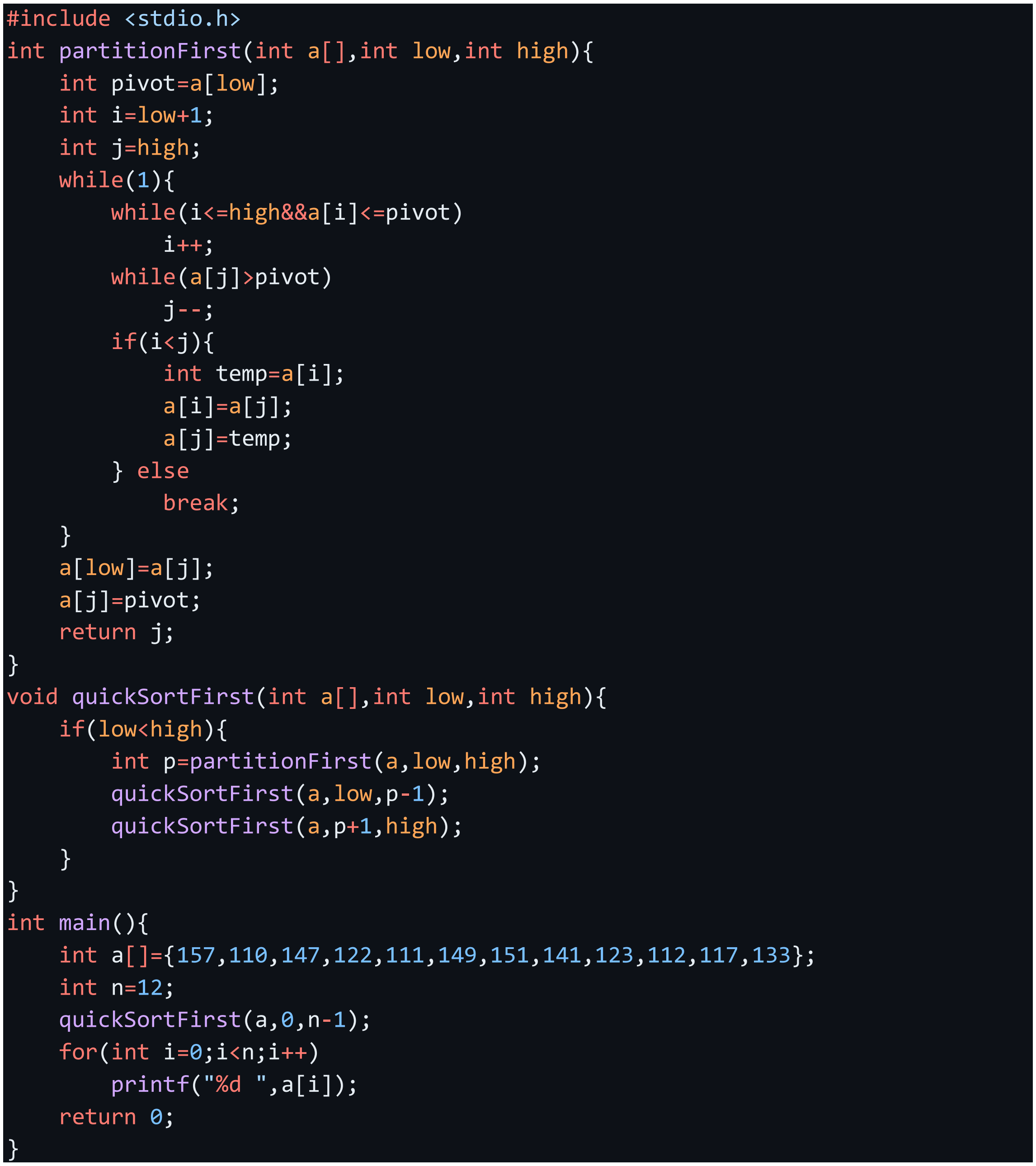
Design Analysis and Algorithm – Lab Work

Week 5

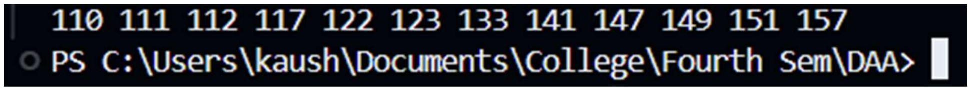
Question 1: Write a program to implement quick sort using first element as pivot.

Code:



```
#include <stdio.h>
int partitionFirst(int a[],int low,int high){
    int pivot=a[low];
    int i=low+1;
    int j=high;
    while(1){
        while(i<=high&&a[i]<=pivot)
            i++;
        while(a[j]>pivot)
            j--;
        if(i<j){
            int temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        } else
            break;
    }
    a[low]=a[j];
    a[j]=pivot;
    return j;
}
void quickSortFirst(int a[],int low,int high){
    if(low<high){
        int p=partitionFirst(a,low,high);
        quickSortFirst(a,low,p-1);
        quickSortFirst(a,p+1,high);
    }
}
int main(){
    int a[]={157,110,147,122,111,149,151,141,123,112,117,133};
    int n=12;
    quickSortFirst(a,0,n-1);
    for(int i=0;i<n;i++)
        printf("%d ",a[i]);
    return 0;
}
```

Output:



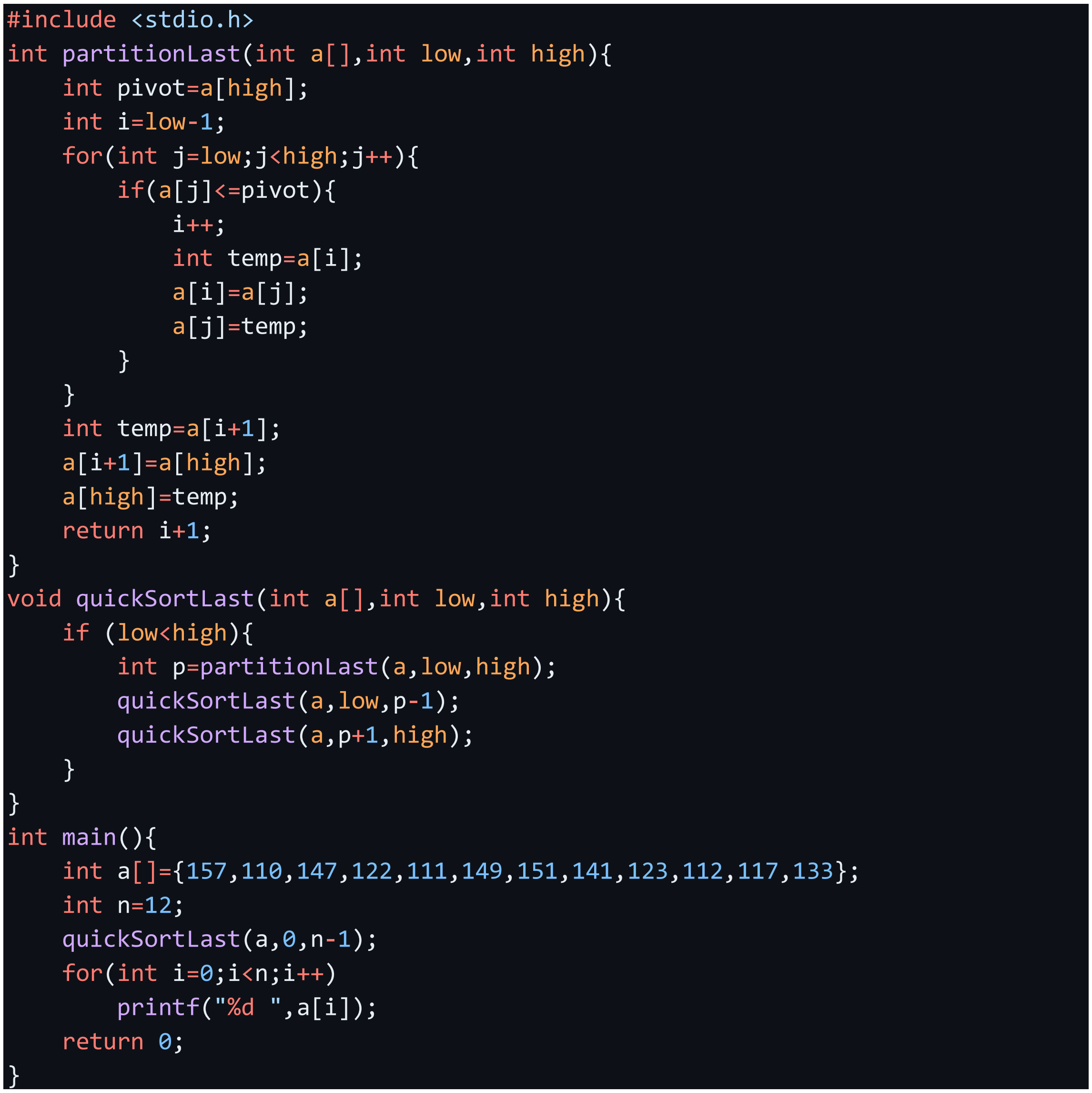
```
110 111 112 117 122 123 133 141 147 149 151 157
PS C:\Users\kaush\Documents\College\Fourth Sem\DAA>
```

Space Complexity: O(n)

Time Complexity: O(nlogn)

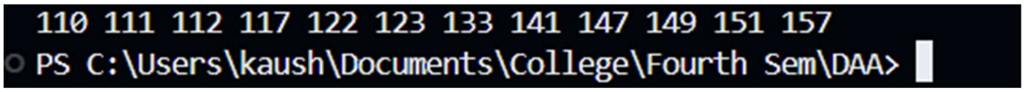
Question 2: Write a program to implement quick sort using last element as pivot.

Code:



```c
#include <stdio.h>
int partitionLast(int a[],int low,int high){
    int pivot=a[high];
    int i=low-1;
    for(int j=low;j<high;j++){
        if(a[j]<=pivot){
            i++;
            int temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }
    }
    int temp=a[i+1];
    a[i+1]=a[high];
    a[high]=temp;
    return i+1;
}
void quickSortLast(int a[],int low,int high){
    if (low<high){
        int p=partitionLast(a,low,high);
        quickSortLast(a,low,p-1);
        quickSortLast(a,p+1,high);
    }
}
int main(){
    int a[]={157,110,147,122,111,149,151,141,123,112,117,133};
    int n=12;
    quickSortLast(a,0,n-1);
    for(int i=0;i<n;i++)
        printf("%d ",a[i]);
    return 0;
}
```

Output:



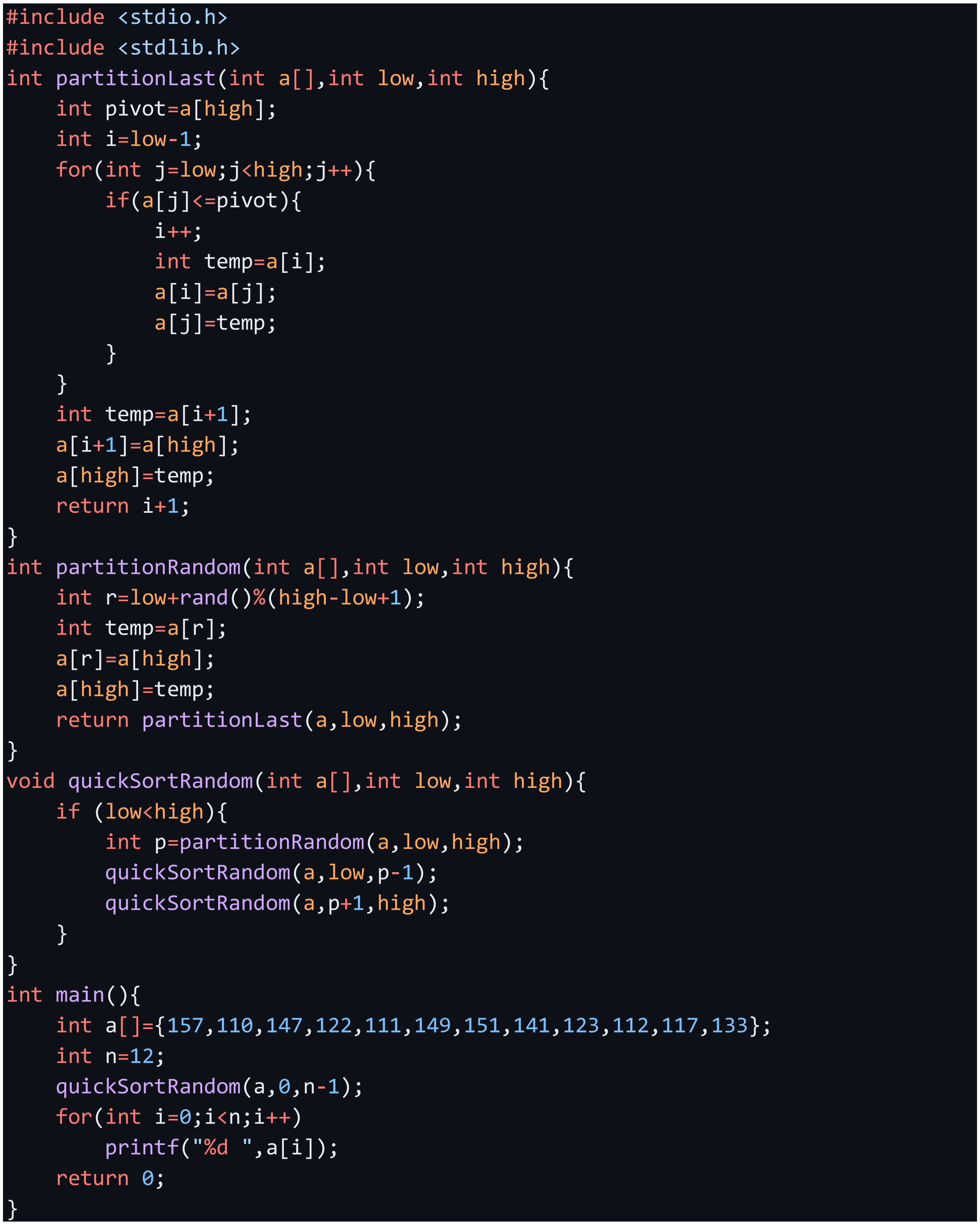
```
110 111 112 117 122 123 133 141 147 149 151 157
PS C:\Users\kaush\Documents\College\Fourth Sem\DAA>
```

Space Complexity: O(n)

Time Complexity: O(nlogn)

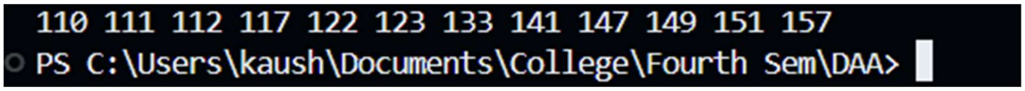
Question 3: Write a program to implement quick sort using random element as pivot.

Code:



```
#include <stdio.h>
#include <stdlib.h>
int partitionLast(int a[],int low,int high){
    int pivot=a[high];
    int i=low-1;
    for(int j=low;j<high;j++){
        if(a[j]<=pivot){
            i++;
            int temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }
    }
    int temp=a[i+1];
    a[i+1]=a[high];
    a[high]=temp;
    return i+1;
}
int partitionRandom(int a[],int low,int high){
    int r=low+rand()%(high-low+1);
    int temp=a[r];
    a[r]=a[high];
    a[high]=temp;
    return partitionLast(a,low,high);
}
void quickSortRandom(int a[],int low,int high){
    if (low<high){
        int p=partitionRandom(a,low,high);
        quickSortRandom(a,low,p-1);
        quickSortRandom(a,p+1,high);
    }
}
int main(){
    int a[]={157,110,147,122,111,149,151,141,123,112,117,133};
    int n=12;
    quickSortRandom(a,0,n-1);
    for(int i=0;i<n;i++)
        printf("%d ",a[i]);
    return 0;
}
```

Output:



```
110 111 112 117 122 123 133 141 147 149 151 157
PS C:\Users\kaush\Documents\College\Fourth Sem\DAA>
```

Space Complexity: O(n)

Time Complexity: O(nlogn)