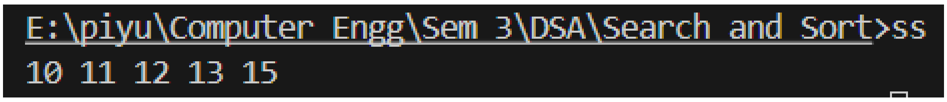
1. Insertion Sort

Code:

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
void main(){
    int a[]={12,11,15,10,13};
    int k,i,key;
    for(k=0;k<=4;k++)
    {
        i=k;
        key=a[i+1];
        while(i>=0 && key<a[i]){
            a[i+1]=a[i];
            i--;
        }
        a[i+1]=key;
    }
    for(k=0;k<=4;k++)
        printf("%d ",a[k]);
}
```

Output:



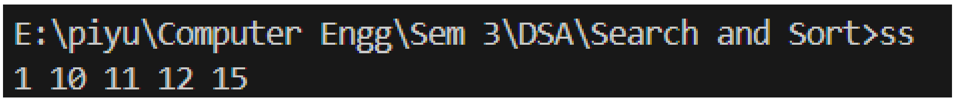
```
E:\piyu\Computer Engg\Sem 3\DSA\Search and Sort>ss
10 11 12 13 15
```

2. Selection Sort

Code:

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
void main(){
    int a[]={12,11,15,10,1};
    int i,j,mindex,temp,min;
    for(i=0;i<4;i++){
        mindex=i;
        for(j=i+1;j<=4;j++){//searching for min
            if(a[j]<a[mindex])
                mindex=j;
        }
        temp=a[i];
        a[i]=a[mindex];
        a[mindex]=temp;
    }
    for(j=0;j<5;j++)
        printf("%d ",a[j]);
}
```

Output:



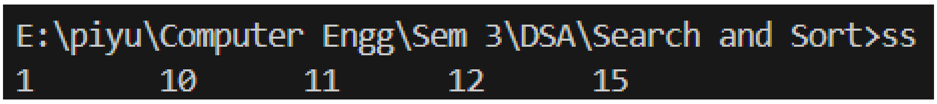
```
E:\piyu\Computer Engg\Sem 3\DSA\Search and Sort>ss
1 10 11 12 15
```

3. Merge Sort

Code:

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
int a[]={12,11,15,10,1};
const int len=sizeof(a) / sizeof(a[0]);
int p[100];
void merge(int low, int mid, int high){
    int i=low,j=mid+1,k=0;
    while(i<=mid && j<=high){
        if(a[i]<a[j])
            p[k++]=a[i++];
        else
            p[k++]=a[j++];
    }
    while(i<=mid){
        p[k++]=a[i++];
    }
    while(j<=high){
        p[k++]=a[j++];
    }
    for(i = low, k = 0; i <= high; i++, k++) //rememberrr
        a[i] = p[k];
}
void mergeSort(int low,int high){
    int mid=(low+high)/2;
    if(low>=high) return;
    mergeSort(low,mid);
    mergeSort(mid+1,high);
    merge(low, mid, high);
}
void main(){
    mergeSort(0,len-1);
    for(int i=0;i<len;i++)
        printf("%d\t",a[i]);
}
```

Output:



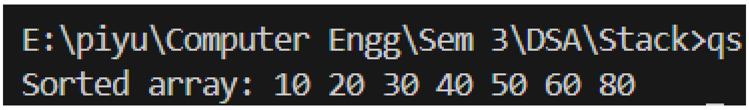
```
E:\piyu\Computer Engg\Sem 3\DSA\Search and Sort>ss
1       10      11      12      15
```

4. Quick Sort

Code:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
int a[7] = {50, 30, 20, 40, 80, 10, 60};
void swap(int x, int y) {
    int temp = a[x];
    a[x] = a[y];
    a[y] = temp;
}
int partition(int low, int high) {
    int pivot = a[high];
    int i = low - 1;
    int j;
    for (j = low; j < high; j++) {
        if (a[j] < pivot) {
            i++;
            swap(i, j);
        }
    }
    swap(i + 1, high);
    return i + 1;
}
void quickSort(int low, int high) {
    if (low < high) {
        int pi = partition(low, high);
        quickSort(low, pi - 1);
        quickSort(pi + 1, high);
    }
}
void main() {
    int n = 7;
    quickSort(0, n - 1);
    printf("Sorted array: ");
    for (int i = 0; i < n; i++)
        printf("%d ", a[i]);
}
```

Output:



```
E:\piyu\Computer Engg\Sem 3\DSA\Stack>qs
Sorted array: 10 20 30 40 50 60 80
```