1. Write a program for the Insertion sort algorithm.

```
#include <stdio.h>
int main()
{
	int n, j, temp;
	printf("Enter the size of the array : \n");
	scanf("%d", &n);
	int arr[n];
	for (int i = 0; i < n; i++)
	{
		printf("Enter %d elements\n", i+1);
		scanf("%d", &arr[i]);
	}
	for (int i = 1 ; i <= n - 1; i++)
	{	j = i;
		while ( j > 0 && arr[j-1] > arr[j])
			{
			temp = arr[j];
			arr[j] = arr[j-1];
			arr[j-1] = temp;
			J--;
		}
	}
	printf("Sorted list in ascending order is:\n");
	for (int i = 0; i <= n - 1; i++)
	{
		printf("%d\n", arr[i]);
	}
	return 0;
}
```

2. Write a program for the Selection sort algorithm.

```
#include<stdio.h>
int main(){
	int n, temp, arr[50];
	printf("How many numbers u are going to enter?: ");
	scanf("%d",&n);
	for(int i=0;i<n;i++){
	printf("Enter the %d elements: ", i+1);
	scanf("%d",&arr[i]);
	}
	for(int i=0;i<n;i++){
		for(int j=i+1;j<n;j++){
			if(arr[i]>arr[j]){
```

```
                        temp=arr[i];
                        arr[i]=arr[j];
                        arr[j]=temp;
                    }
                }
        }
        printf("Sorted elements in the list are\n: ");
        for(int i=0;i<n;i++){
                printf(" %d\n",arr[i]);
        }
        return 0;
}
```

3. Write a program for Bubble sort algorithm.

```
#include<stdio.h>
int main(){
        int n, temp, arr[30];
        printf("Enter how many elements are present in your list: ");
        scanf("%d",&n);

        for(int i=0;i<n;i++){
                printf("Enter the %d numbers: ",i+1);
                scanf("%d",&arr[i]);
        }
        for(int i=n-2;i>=0;i--){
                for(int j=0;j<=i;j++){
                        if(arr[j]>arr[j+1]){
                                temp=arr[j];
                                arr[j]=arr[j+1];
                                arr[j+1]=temp;
                        }
                }
        }
        printf("Sorted elements in the list are: ");
        for(int i=0;i<n;i++){
                printf(" %d\n",arr[i]);
        }
        return 0;
}
```

4. Write a program for the Merge sort algorithm.

```
#include<stdlib.h>
#include<stdio.h>
int main()
{
	int arr[] = {12, 11, 13, 5, 6, 7};
	int arr_size = sizeof(arr)/sizeof(arr[0]);
	printf("The total given elements in array is  \n");
	printArray(arr, arr_size);
	mergeSort(arr, 0, arr_size - 1);
	printf("\n The Sorted array is \n");
	printArray(arr, arr_size);
	return 0;
}

void mergeSort(int arr[], int left_index, int right_index)
{
	if (left_index < right_index)
	{
		int m = left_index+(right_index-left_index)/2;
		mergeSort(arr, left_index, m);
		mergeSort(arr, m+1, right_index);
		merge(arr, left_index, m, right_index);
	}
}

void merge(int arr[], int left_index, int m, int right_index)
{
	int i, j, k;
	int n1 = m - left_index + 1; int n2 = right_index - m;
	int L[n1], R[n2];
	for (i = 0; i < n1; i++)
	{
		L[i] = arr[left_index + i];
	}
	for (j = 0; j < n2; j++){
	R[j] = arr[m + 1+ j];
	}
	i = 0;
	j = 0;
	k = left_index;
	while (i < n1 && j < n2)
	{
		if (L[i] <= R[j])
```

```
                {
                        arr[k] = L[i];
                        I++;
                        }
                else
                {
                        arr[k] = R[j];
                        J++;
                }
                k++;
        }
        while (i < n1)
        {
                arr[k] = L[i];
                i++;
                k++;
                } while (j < n2)
                {
                        arr[k] = R[j];
                        j++;
                        k++;
                }
        }
void printArray(int A[], int size)
{
for (int i=0; i < size; i++){
        printf("%d \n", A[i]);
        }
}
```

5. Write a program for the Heap sort algorithm.

```
#include <stdio.h>
void main()
{
        int arr[10], n, i, j, c, root, temp;
        printf("\n Enter no of elements in the array :");
        scanf("%d", &n);
        for (i = 0; i < n; i++){
        printf("\n Enter the %d element  : ",i+1);
        scanf("%d", &arr[i]);
        }
        for (i = 1; i < n; i++)
        {
```

```
        c = i;
        Do
        {root = (c - 1) / 2;
        if (arr[root] < arr[c])
        {
                temp = arr[root];
                arr[root] = arr[c];
                arr[c] = temp;
        }
        c = root;
        } while (c != 0);
        }
        printf("The array is : ");
        for (i = 0; i < n; i++)
        printf("%d\t ", arr[i]);
        for (j = n - 1; j >= 0; j--)
        {
                temp = arr[0];
                arr[0] = arr[j];
                arr[j] = temp;
                root = 0;
                do
        {
        c = 2 * root + 1;
        if ((arr[c] < arr[c + 1]) && c < j-1)
        c++;
        if (arr[root]<arr[c] && c<j)
        {
                temp = arr[root];
                arr[root] = arr[c];
                arr[c] = temp;
        }
        root = c;} while (c < j);
        }
        printf("\n The sorted array is : ");
        for (i = 0; i < n; i++)
        printf("\t %d", arr[i]);
        printf("\n Complexity : \n Best case = Avg case = Worst case = O(n logn) \n");
}
```