## Programs

## Insertion Sort:

```
#include<stdio.h>
#include<stdlib.h>

void insertion(int[], int);
void main(){
	int *arr, i, n;

	printf("Enter the number of elements in the array :");
	scanf("%d",&n);

	arr = (int*) malloc(sizeof(int));

	printf("Enter the elements to be sorted: ");
	for(i = 0;i < n; i++)
		scanf("%d",&arr[i]);

	insertion(arr, n);

	printf("The sorted elements are : ");
	for(i = 0;i < n; i++)
		printf("%d\t",arr[i]);
}

void insertion(int array[], int size){
	int i, j, temp;
	for(i = 1;i < size; i++){
		temp = array[i];
		for(j = i-1 ;j >= 0; j--){
		   if(array[j] > temp){
		      array[j+1] = array[j];
		   }else{
		      break;
		   }
		}
		array[j+1] = temp;
	}
}
```

## Output:

```
Enter the number of elements in the array :5
Enter the elements to be sorted: 77
53
62
24
7
The sorted elements are : 7     24      53      62      77
Press any key to continue . . .
```

## Heap Sort:

```
#include <stdio.h>
#include <conio.h>
void create(int[]);
void down_adjust(int[], int);

int main()
{
	int heap[50], n, i, lastElem, temp;

	printf("Enter no. of elements: ");
	scanf("%d", &n);

	printf("\nEnter the elements: ");
	for (i = 1; i <= n; i++)
		scanf("%d", &heap[i]);

	heap[0] = n;
	create(heap);

	while (heap[0] > 1)
	{

		lastElem = heap[0];
		temp = heap[1];
		heap[1] = heap[lastElem];
		heap[lastElem] = temp;
		heap[0]--;
		down_adjust(heap, 1);
	}

	printf("\nSorted array: \n");
	for (i = 1; i <= n; i++)
		printf("%d\t", heap[i]);
	getch();
	return 0;
}

void create(int heap[])
{
	int i, n;
	n = heap[0];

	for (i = n / 2; i >= 1; i--)
		down_adjust(heap, i);
}

void down_adjust(int heap[], int i)
{
	int j, temp, n, flag = 1;
	n = heap[0];

	while (2 * i <= n && flag == 1)
	{
		j = 2 * i;
		if (j + 1 <= n && heap[j + 1] > heap[j])
			j = j + 1;
		if (heap[i] > heap[j])
```

```
                        flag = 0;
                else
                {
                        temp = heap[i];
                        heap[i] = heap[j];
                        heap[j] = temp;
                        i = j;
                }
        }
}
```

**Output:**

```
Enter no. of elements: 7

Enter the elements: 88
34
75
22
5
90
47

Sorted array:
5       22      34      47      75      88      90
```