1.Maximum Heap

```
#include <stdio.h>

Void swap(int *a, int *b) {
   Int temp = *a;
   *a = *b;
   *b = temp;
}

Void maxHeapify(int heap[], int n, int i) {
   Int largest = I, left = 2 * I + 1, right = 2 * I + 2;
   If (left < n && heap[left] > heap[largest])
      Largest = left;
   If (right < n && heap[right] > heap[largest])
      Largest = right;
   If (largest != i) {
      Swap(&heap[i], &heap[largest]);
      maxHeapify(heap, n, largest);
   }
}

Void insert(int heap[], int *n, int value) {
   Int I = (*n)++;
   Heap[i] = value;
   While (I != 0 && heap[(I – 1) / 2] < heap[i]) {
      Swap(&heap[i], &heap[(I – 1) / 2]);
```

```
      I = (I – 1) / 2;
   }
}

Void printHeap(int heap[], int n) {
   For (int I = 0; I < n; i++)
      Printf(“%d “, heap[i]);
   Printf(“\n”);
}

Int main() {
   Int heap[100], n = 0, numElements, value;
   Printf(“Enter the number of elements to insert: “);
   Scanf(“%d”, &numElements);
   Printf(“Enter the elements:\n”);
   For (int I = 0; I < numElements; i++) {
      Scanf(“%d”, &value);
      Insert(heap, &n, value);
   }
   Printf(“Max Heap:\n”);
   printHeap(heap, n);
   return 0;
}
```

Input

Enter the number of elements to insert: 5

Enter the elements:

3

10

5

1

7

Output

Max Heap:

10 7 5 1 3

2.Minimum Heap

```
#include <stdio.h>

Void swap(int *a, int *b) {
   Int temp = *a;
   *a = *b;
   *b = temp;
}

Void minHeapify(int heap[], int n, int i) {
   Int smallest = I, left = 2 * I + 1, right = 2 * I + 2;
   If (left < n && heap[left] < heap[smallest])
      Smallest = left;
   If (right < n && heap[right] < heap[smallest])
      Smallest = right;
   If (smallest != i) {
      Swap(&heap[i], &heap[smallest]);
```

```
      minHeapify(heap, n, smallest);
   }
}

Void insert(int heap[], int *n, int value) {
   Int I = (*n)++;
   Heap[i] = value;
   While (I != 0 && heap[(I – 1) / 2] > heap[i]) {
      Swap(&heap[i], &heap[(I – 1) / 2]);
      I = (I – 1) / 2;
   }
}

Void printHeap(int heap[], int n) {
   For (int I = 0; I < n; i++)
      Printf(“%d “, heap[i]);
   Printf(“\n”);
}

Int main() {
   Int heap[100], n = 0, numElements, value;
   Printf(“Enter the number of elements to insert: “);
   Scanf(“%d”, &numElements);
   Printf(“Enter the elements:\n”);
   For (int I = 0; I < numElements; i++) {
      Scanf(“%d”, &value);
```

```
    Insert(heap, &n, value);
  }
  Printf(“Min Heap:\n”);
  printHeap(heap, n);
  return 0;
}
```

Input

Enter the number of elements to insert: 5

Enter the elements:

3

10

5

1

7

Output

Min Heap:

1 3 5 10 7

3.Minimum heap Sort

```
#include <stdio.h>

Void swap(int *a, int *b) {
  Int temp = *a;
  *a = *b;
  *b = temp;
}
```

```
Void minHeapify(int heap[], int n, int i) {
   Int smallest = I, left = 2 * I + 1, right = 2 * I + 2;
   If (left < n && heap[left] < heap[smallest])
      Smallest = left;
   If (right < n && heap[right] < heap[smallest])
      Smallest = right;
   If (smallest != i) {
      Swap(&heap[i], &heap[smallest]);
      minHeapify(heap, n, smallest);
   }
}

Void buildMinHeap(int heap[], int n) {
   For (int I = n / 2 – 1; I >= 0; i--)
      minHeapify(heap, n, i);
}

Void heapSort(int heap[], int n) {
   buildMinHeap(heap, n);
   for (int I = n – 1; I > 0; i--) {
      swap(&heap[0], &heap[i]);
      minHeapify(heap, I, 0);
   }
}

Void printArray(int arr[], int n) {
```

```
   For (int I = 0; I < n; i++)
      Printf(“%d “, arr[i]);
   Printf(“\n”);
}

Int main() {
   Int heap[100], n, value;
   Printf(“Enter the number of elements to sort: “);
   Scanf(“%d”, &n);
   Printf(“Enter the elements:\n”);
   For (int I = 0; I < n; i++) {
      Scanf(“%d”, &heap[i]);
   }
   heapSort(heap, n);
   printf(“Sorted elements:\n”);
   printArray(heap, n);
   return 0;
}
```

INPUT

Enter the number of elements to sort: 5

Enter the elements:

3

10

5

1

7

Output

Sorted elements:

10 7 5 3 1

4.Maximum heap Sort

```
#include <stdio.h>


Void swap(int *a, int *b) {
   Int temp = *a;
   *a = *b;
   *b = temp;
}


Void maxHeapify(int heap[], int n, int i) {
   Int largest = I, left = 2 * I + 1, right = 2 * I + 2;
   If (left < n && heap[left] > heap[largest])
      Largest = left;
   If (right < n && heap[right] > heap[largest])
      Largest = right;
   If (largest != i) {
      Swap(&heap[i], &heap[largest]);
      maxHeapify(heap, n, largest);
   }
}


Void buildMaxHeap(int heap[], int n) {
   For (int I = n / 2 – 1; I >= 0; i--)
```

```
        maxHeapify(heap, n, i);
}

Void heapSort(int heap[], int n) {
    buildMaxHeap(heap, n);
    for (int I = n – 1; I > 0; i--) {
        swap(&heap[0], &heap[i]);
        maxHeapify(heap, I, 0);
    }
}

Void printArray(int arr[], int n) {
    For (int I = 0; I < n; i++)
        Printf(“%d “, arr[i]);
    Printf(“\n”);
}

Int main() {
    Int heap[100], n;
    Printf(“Enter the number of elements to sort: “);
    Scanf(“%d”, &n);
    Printf(“Enter the elements:\n”);
    For (int I = 0; I < n; i++) {
        Scanf(“%d”, &heap[i]);
    }
    heapSort(heap, n);
```

```
    printf(“Sorted elements:\n”);
    printArray(heap, n);
    return 0;
}
```

Input

Enter the number of elements to sort: 5

Enter the elements:

3

10

5

1

7

Output

Sorted elements:

1 3 5 7 10