1)

#include <stdio.h>

#include <stdlib.h>

void minHeapify(int arr[], int n, int i) {

    int smallest = i;

    int left = 2 \* i + 1;

    int right = 2 \* i + 2;

    if (left < n && arr[left] < arr[smallest])

        smallest = left;

    if (right < n && arr[right] < arr[smallest])

        smallest = right;

    if (smallest != i) {

        int temp = arr[i];

        arr[i] = arr[smallest];

        arr[smallest] = temp;

        minHeapify(arr, n, smallest);

    }

}

void buildMinHeap(int arr[], int n) {

    for (int i = (n / 2) - 1; i >= 0; i--)

        minHeapify(arr, n, i);

}

void printHeap(int arr[], int n) {

    printf("Min Heap: ");

    for (int i = 0; i < n; i++)

        printf("%d ", arr[i]);

    printf("\n");

}

int main() {

    int arr[] = {1, 5, 6, 8, 9, 7, 3};

    int n = sizeof(arr) / sizeof(arr[0]);

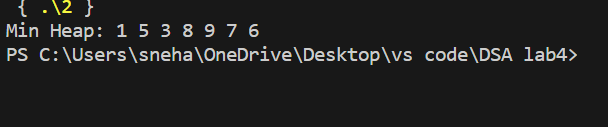
    buildMinHeap(arr, n);

    printHeap(arr, n);

    return 0;

}

Output:



2)

#include <stdio.h>

void swap(int\* a, int\* b) {

    int temp = \*a;

    \*a = \*b;

    \*b = temp;

}

void heapify(int arr[], int n, int i) {

    int largest = i;

    int left = 2 \* i + 1;

    int right = 2 \* i + 2;

    if (left < n && arr[left] > arr[largest])

        largest = left;

    if (right < n && arr[right] > arr[largest])

        largest = right;

    if (largest != i) {

        swap(&arr[i], &arr[largest]);

        heapify(arr, n, largest);

    }

}

void delete\_max(int arr[], int\* n) {

    if (\*n <= 0) {

        printf("Heap is empty.\n");

        return;

    }

    int max = arr[0];

    arr[0] = arr[\*n - 1];

    (\*n)--;

        heapify(arr, \*n, 0);

    printf("Deleted maximum element: %d\n", max);

}

int main() {

    int arr[] = { 44,56,34,12,43,67,89,50,60 };

    int n = sizeof(arr) / sizeof(arr[0]);

    printf("Min-max heap before deletion: ");

    for (int i = 0; i < n; i++) {

        printf("%d ", arr[i]);

    }

    printf("\n");

    delete\_max(arr, &n);

    printf("Min-max heap after deletion: ");

    for (int i = 0; i < n; i++) {

        printf("%d ", arr[i]);

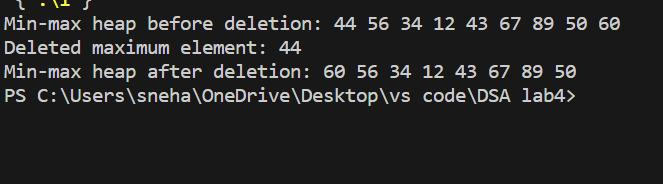
    }

    printf("\n");

      return 0;

}

Output:



3)

#include <stdio.h>

void heapify(int arr[], int n, int i) {

    int largest = i;

    int left = 2 \* i + 1;

    int right = 2 \* i + 2;

    if (left < n && arr[left] > arr[largest])

        largest = left;

    if (right < n && arr[right] > arr[largest])

        largest = right;

    if (largest != i) {

        int temp = arr[i];

        arr[i] = arr[largest];

        arr[largest] = temp;

        heapify(arr, n, largest);

    }

}

void heapSort(int arr[], int n) {

    for (int i = n / 2 - 1; i >= 0; i--)

        heapify(arr, n, i);

    for (int i = n - 1; i > 0; i--) {

        int temp = arr[0];

        arr[0] = arr[i];

        arr[i] = temp;

        heapify(arr, i, 0);

    }

}

int main() {

    int arr[] = {12, 11, 13, 5, 6, 7};

    int n = sizeof(arr) / sizeof(arr[0]);

    printf("Original array: ");

    for (int i = 0; i < n; i++)

        printf("%d ", arr[i]);

    heapSort(arr, n);

    printf("\nSorted array: ");

    for (int i = 0; i < n; i++)

        printf("%d ", arr[i]);

    return 0;

}

Output:

