QUESTION 1:

#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 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) {

        swap(&arr[i], &arr[smallest]);

        heapify(arr, n, smallest);

    }

}

void insert(int arr[], int \*n, int value) {

    if (\*n == 0) {

        arr[0] = value;

        \*n += 1;

        return;

    }

    int i = \*n;

    arr[i] = value;

    \*n += 1;

    while (i > 0 && arr[i] < arr[(i - 1) / 2]) {

        swap(&arr[i], &arr[(i - 1) / 2]);

        i = (i - 1) / 2;

    }

}

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

    if (\*n <= 0) {

        printf("Heap is empty.\n");

        return;

    }

    if (\*n == 1) {

        \*n = 0;

        return;

    }

    arr[0] = arr[\*n - 1];

    \*n -= 1;

    heapify(arr, \*n, 0);

}

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

    printf("Heap elements: ");

    for (int i = 0; i < n; i++) {

        printf("%d ", arr[i]);

    }

    printf("\n");

}

int main() {

    int heap[100];

    int size = 0;

    // Insert elements

    insert(heap, &size, 4);

    insert(heap, &size, 9);

    insert(heap, &size, 2);

    insert(heap, &size, 7);

    insert(heap, &size, 5);

    display(heap, size);

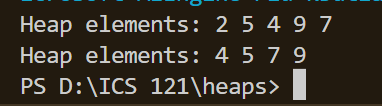
    // Delete minimum element

    deleteMin(heap, &size);

    display(heap, size);

    return 0;

}



QUESTION 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 insert(int arr[], int \*n, int value) {

    if (\*n == 0) {

        arr[0] = value;

        \*n += 1;

        return;

    }

    int i = \*n;

    arr[i] = value;

    \*n += 1;

    while (i > 0 && arr[i] > arr[(i - 1) / 2]) {

        swap(&arr[i], &arr[(i - 1) / 2]);

        i = (i - 1) / 2;

    }

}

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

    if (\*n <= 0) {

        printf("Heap is empty.\n");

        return;

    }

    if (\*n == 1) {

        \*n = 0;

        return;

    }

    arr[0] = arr[\*n - 1];

    \*n -= 1;

    heapify(arr, \*n, 0);

}

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

    printf("Heap elements: ");

    for (int i = 0; i < n; i++) {

        printf("%d ", arr[i]);

    }

    printf("\n");

}

int main() {

    int heap[100];

    int size = 0;

    // Insert elements

    insert(heap, &size, 4);

    insert(heap, &size, 9);

    insert(heap, &size, 2);

    insert(heap, &size, 7);

    insert(heap, &size, 5);

    display(heap, size);

    // Delete maximum element

    deleteMax(heap, &size);

    display(heap, size);

    return 0;

}

