```
//Name: 9. Heap Sort
#include <iostream>
 using namespace std;
 void swap(int* a, int* b) {
    int temp = *a;
    *a = *b;
    *b = temp;
 }
void maxHeapify(int arr[], int n, int i) {
    int largest = i;
    int le = 2 * i + 1;
    int right = 2 * i + 2;
    if (le < n && arr[le ] > arr[largest])
        largest = le ;
    if (right < n && arr[right] > arr[largest])
        largest = right;
    if (largest != i) {
        swap(&arr[i], &arr[largest]);
        maxHeapify(arr, n, largest);
    }
 }

 void minHeapify(int arr[], int n, int i) {
    int smallest = i;
    int le = 2 * i + 1;
    int right = 2 * i + 2;
    if (le < n && arr[le ] < arr[smallest])
        smallest = le ;
    if (right < n && arr[right] < arr[smallest])
        smallest = right;
    if (smallest != i) {
        swap(&arr[i], &arr[smallest]);
        minHeapify(arr, n, smallest);
    }
 }
void maxHeapSort(int arr[], int n) {
    for (int i = n / 2 - 1; i >= 0; i--)
        maxHeapify(arr, n, i);
    for (int i = n - 1; i >= 0; i--) {
        swap(&arr[0], &arr[i]);
        maxHeapify(arr, i, 0);
    }
 }
void minHeapSort(int arr[], int n) {
    for (int i = n / 2 - 1; i >= 0; i--)
        minHeapify(arr, n, i);
    for (int i = n - 1; i >= 0; i--) {
        swap(&arr[0], &arr[i]);
        minHeapify(arr, i, 0);
    }
 }
void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++)
        cout << arr[i] << " ";
    cout << endl;
 }
 int main() {
    int arr[] = {7, 2, 1, 6, 8, 5, 3, 4};
    int n = sizeof(arr) / sizeof(arr[0]);
    cout << "Original array: ";
    printArray(arr, n);
    maxHeapSort(arr, n);
    cout << "Sorted array (Descending order using Max Heap): ";
    printArray(arr, n);
    int arr2[] = {7, 2, 1, 6, 8, 5, 3, 4};
    n = sizeof(arr2) / sizeof(arr2[0]);

    minHeapSort(arr2, n);
    cout << "Sorted array (Ascending order using Min Heap): ";
    printArray(arr2, n);
return 0;
 }
```