```
#include <iostream>

using namespace std;



// Function to heapify a subtree rooted at index i

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

int largest = i; // Initialize largest as root

int left = 2 * i + 1; // Left child index

int right = 2 * i + 2; // Right child index



// Check if the left child is larger than the root

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

largest = left;



// Check if the right child is larger than the largest so far

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

largest = right;
```

```
// If largest is not root, swap and recursively heapify the affected subtree

if (largest != i) {

swap(arr[i], arr[largest]);

heapify(arr, n, largest);

}

}



// Function to perform heap sort

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

// Build max heap

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

heapify(arr, n, i);



// Extract elements from the heap one by one

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

swap(arr[0], arr[i]); // Move current root to the end
```

```
heapify(arr, i, 0); // Heapify the reduced heap

}

}



// Utility function to print an array

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

for (int i = 0; i < n; i++)

cout << arr[i] << " ";

cout << endl;

}



// Main function

int main() {

int n;



// Take input from the user

cout << "Enter the number of elements: ";
```

```
cin >> n;

int arr[n]; // Declare array of size n

cout << "Enter the elements of the array:\n";
for (int i = 0; i < n; i++)
cin >> arr[i];

cout << "Original array: ";
printArray(arr, n);

heapSort(arr, n);

cout << "Sorted array: ";
printArray(arr, n);
```

```
return 0;

}
```