Practical 10

```
#include <iostream>
using namespace std;

void maxHeapify(int arr[], int n, int i) {
    int largest = i;      // Initialize largest as root
    int left = 2 * i + 1; // left = 2*i + 1
    int right = 2 * i + 2;// 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]);
        maxHeapify(arr, n, largest);
    }
}

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) {
        swap(arr[i], arr[smallest]);
        minHeapify(arr, n, smallest);
    }
}

void buildMaxHeap(int arr[], int n) {
    // Start from the last non-leaf node and heapify each
    for (int i = n / 2 - 1; i >= 0; i--) {
        maxHeapify(arr, n, i);
    }
}

void buildMinHeap(int arr[], int n) {
    for (int i = n / 2 - 1; i >= 0; i--) {
        minHeapify(arr, n, i);
    }
}

int main() {
    int n;
    cout << "Enter number of students: ";
    cin >> n;
    int marks[n], maxHeap[n], minHeap[n];
    cout << "Enter marks of " << n << " students:\n";
    for (int i = 0; i < n; i++) {
        cin >> marks[i];
        maxHeap[i] = marks[i];
        minHeap[i] = marks[i];
    }
    buildMaxHeap(maxHeap, n);
    buildMinHeap(minHeap, n);
    cout << "\nMaximum marks: " << maxHeap[0] << endl;
    cout << "Minimum marks: " << minHeap[0] << endl;
    return 0;
}
```