

```

#include <iostream>
#include <vector>

using namespace std;

class MaxHeap{
private:
    vector<int> heap;

    void heapifyUp(int index){
        int parent = (index - 1) / 2;

        while (index > 0 && heap[parent] < heap[index]){
            swap(heap[index], heap[parent]);
            index = parent;
            parent = (parent - 1) / 2;
        }
    }

    void heapifyDown(int index){
        int left = 2 * index + 1;
        int right = 2 * index + 2;
        int largest = index;

        if (left < heap.size() && heap[left] > heap[largest]){
            largest = left;
        }
        if (right < heap.size() && heap[right] > heap[largest]){
            largest = right;
        }
        if (largest != index){
            swap(heap[index], heap[largest]);
            heapifyDown(largest);
        }
    }

public:
    MaxHeap() {}

    void insert(int value){
        heap.push_back(value);
        heapifyUp(heap.size() - 1);
    }

    int extractMax(){
        if (heap.empty()){
            return -1;
        }
    }

```

```

        int max = heap[0];
        swap(heap[0], heap[heap.size() - 1]);
        heap.pop_back();
        heapifyDown(0);
        return max;
    }

    int peekMax(){
        if (heap.empty()){
            return -1;
        }
        return heap[0];
    }

    bool isEmpty(){
        return heap.empty();
    }

    void printHeap(){
        for (int value : heap){
            cout << value << " ";
        }
        cout << endl;
    }
};

int main(){
    MaxHeap heap;

    heap.insert(10);
    heap.insert(5);
    heap.insert(15);
    heap.insert(3);
    heap.insert(2);

    cout << "Heap: ";
    heap.printHeap();

    heap.insert(13);
    cout << "Heap After Inserting 13: ";
    heap.printHeap();
    cout << "Max Element: " << heap.extractMax() << endl;

    cout << "Heap After Extract: ";
    heap.printHeap();

    cout << "Max Element: " << heap.peekMax() << endl;
}

```

```
    return 0;  
}
```

```
// Mikhael Setia Budi  
// 2311110033
```