import java.util.ArrayList;  
  
class heap<T extends Comparable<T>> {  
 private ArrayList<T> list;  
  
 heap() {  
 list = new ArrayList<>();  
 }  
  
 public void swap(int first, int second) {  
 T temp = list.get(first);  
 list.set(first, list.get(second));  
 list.set(second, temp);  
 }  
  
 public int parent(int index) {  
 return (index - 1) / 2;  
 }  
  
 public int left(int index) {  
 return 2 \* index + 1;  
 }  
  
 public int right(int index) {  
 return 2 \* index + 2;  
 }  
  
 public void insert(T element) {  
 list.add(element);  
 upHead(list.size() - 1);  
 }  
  
 private void upHead(int index) {  
 if (index == 0) {  
 return;  
 }  
 int p = parent(index);  
 if (list.get(index).compareTo(list.get(p)) < 0) {  
 swap(index, p);  
 upHead(p);  
 }  
 }  
  
 public T Remove() throws Exception{  
 if(list.isEmpty()){  
 throw new Exception("list is empty");  
 }  
 T temp=list.get(0);  
 T last=list.remove(list.size()-1);  
 if(!list.isEmpty()){  
 list.set(0,last);  
 downHeap(0);  
 }  
  
  
 return temp;  
 }  
  
 private void downHeap(int index) {  
 if(index==list.size()-1){  
 return;  
 }  
 int min=index;  
 int left=left(index);  
 int right=right(index);  
 if(left<list.size() && list.get(min).compareTo(list.get(left))>0)  
 {  
 min=left;  
 }  
 if(right<list.size() && list.get(min).compareTo(list.get(right))>0)  
 {  
 min=right;  
 }  
 if(min!=index){  
 swap(index,min);  
 downHeap(min);  
 }  
  
 }  
  
 public ArrayList<T> heapSort()throws Exception{  
 ArrayList<T> data=new ArrayList<>();  
 while(!list.isEmpty()){  
 data.add(this.Remove());  
 }  
  
 return data;  
 }  
}  
  
  
  
  
  
public class Main {  
 public static void main(String[] args) throws java.lang.Exception{  
  
 heap<Integer> Heap=new heap<>();  
 Heap.insert(7);  
 Heap.insert(8);  
 Heap.insert(9);  
 Heap.insert(10);  
 Heap.insert(11);  
 Heap.insert(12);  
 Heap.insert(13);  
 Heap.insert(14);  
  
// System.out.println(Heap.Remove());  
// System.out.println(Heap.Remove());  
 ArrayList<Integer> list=Heap.heapSort();  
 System.*out*.println(list);  
  
  
  
 }  
  
}