alg1 submission from James Wood

Name	James Wood (jdw74)
College	ROBIN
Submission contents	uk/ac/cam/jdw74/alg1/MaxHeap.java
Ticker	UNKNOWN
Ticker signature	

MaxHeap.java

```java
package uk.ac.cam.jdw74.alg1;

import java.util.List;
import java.util.ArrayList;

public class MaxHeap {
    private char heapName;
    private List<Character> contents;

    public MaxHeap(char name) {
        heapName = name;
        contents = new ArrayList<>();
    }

    public MaxHeap(char name, String str) {
        heapName = name;
        contents = new ArrayList<>(str.length());
        for (int i = 0; i < str.length(); i++) {
            contents.add(str.charAt(i));
        }

        for (int i = contents.size() - 1; i >= 0; i--) {
            siftDown(i);
        }
    }

    public void insert(char x) {
        contents.add(x);
        siftUp(contents.size() - 1);
    }

    public char getMax() {
        char result = contents.get(0);
        contents.set(0, contents.get(contents.size() - 1));
        siftDown(0);
        return result;
    }

    private void siftDown(int i) {
        int l = leftIndex(i);
        int r = rightIndex(i);
        int n = contents.size();
        if (l >= n) return; // Has no children

        if (contents.get(i) < contents.get(l)) {
            if (r >= n) swap(i, l);
            else if (contents.get(l) > contents.get(r)) {
                swap(i, l);
                siftDown(l);
            }
            else {
                swap(i, r);
                siftDown(r);
            }
        }
        else if (r < n && contents.get(i) < contents.get(r)) {
            swap(i, r);
            siftDown(r);
        }
    }

    private void siftUp(int i) {
        if (i == 0) return; // Is root
        int u = upIndex(i);
        if (contents.get(u) < contents.get(i)) {
            swap(i, u);
            siftUp(u);
        }
    }

    private void swap(int x, int y) {
        char tmp = contents.get(x);
```

```
 72            contents.set(x, contents.get(y));
 73            contents.set(y, tmp);
 74        }
 75
 76        private int upIndex(int i) {
 77            return (i - 1) / 2;
 78        }
 79
 80        private int leftIndex(int i) {
 81            return 2 * i + 1;
 82        }
 83
 84        private int rightIndex(int i) {
 85            return 2 * i + 2;
 86        }
 87
 88        public static void main(String[] args)
 89        {
 90            char c;
 91            MaxHeap h = new MaxHeap('h', "CAMBRIDGEALGORITHMS");
 92            c = h.getMax();
 93            System.out.println(c); // expect T
 94            h.insert('Z');
 95            h.insert('A');
 96            c = h.getMax();
 97            System.out.println(c); // expect Z
 98            c = h.getMax();
 99            System.out.println(c); // expect S
100        }
101    }
```