1) Modify the following to print the indices of all entries in T having the largest value.

```java
int[] T = new int[6];
int m=T[0];
for (int e=0; e<6; e++) {if (T[e]>m) {m=T[e];};};
println("max = " + m);

add this code: for (int e=0; e<6; e++) {if (T[e]==m) {println(e);};};
```

2) Assume that you are using a binary search to find the index of the entry in table T that has value V. What assumption about T must hold? **T must be sorted**

If T has 1000 elements, how many tests will you need? **10 or 11**

3) The pseudo-code below has a bug.

```java
for (i=0; i<n-1; i++) {
    for (j=0; j<n-1-i; j++) {
        if (T[j+1] < T[j]) { # insert "temp=T[j];"
            T[j]=T[j+1];
            T[j+1]=T[j]; # replace by "T[j+1]=temp;"
        };
    }; }
```

What was it supposed to do? **Bubble Sort**

What is the bug? **Swap is overwriting itself**

Indicate how to fix it by marking the code above.

4) You know that algorithm X(n) is O(log(n)). X(u) took 1mn. X(v) took 2 mns.

I conclude that v=2u. Am I right? **No**

Justify my conclusion if correct or rectify it if wrong? **v=u^2**

5) Explain what a greedy algorithm is.

*An algorithm that attempts to minimize some global cost C by selecting at each step the move that minimizes C (the locally optimal choice), without looking ahead to select the globally optimal sequence of moves.*