1. Consider the following array of integers:

   13  5  30  12  15  4  10  6  0

   (a) Trace through the above array using selection sort.
   (b) Trace through the above array using insertion sort.
   (c) Trace through the above array using bubble sort.
   (d) Trace through the above array using mergesort.
   (e) Trace through the above array using quicksort (with the first element as the pivot).
   (f) Trace through the above array using radix sort (in particular, the data should all be 2-digit numbers for radix sort to work. How should this be achieved?)
   (g) Place the elements of this array, in order provided, into an initially empty binary search tree.

2. True/False: Performing mergesort on an array of \( n \) elements always takes \( O(n \log_2 n) \) steps.

3. If an array of \( n \) elements is already sorted in ascending order, how many steps (in terms of \( O \) notation) does bubble sort take? Answer the same question for selection sort, mergesort and quicksort.

4. Consider the following array of letters:

   A X Z C V P N O Q E B

   (a) In the order provided, place the elements of this array into an initially empty binary search tree.
   (b) For the tree in part (a), what is the result of a preorder traversal?
   (c) For the tree in part (a), what is the result of a postorder traversal?
5. Write the code for a method

```java
public void bubblesort (Comparable[] ar)
```

which uses bubble sort to sort the array `ar`.

6. Consider a class `BinaryTree` which has as data

```java
private TreeNode<E> root;
```

From inside this class, write a method

```java
public void printPostorder()
```

which prints to the screen the results of a postorder traversal of the tree. (Assume the `TreeNode` class has the usual methods `getLeft()` and `getRight()` to access the left and right child nodes.) You may also write any auxiliary methods as you see fit.