1. Insert into an initially empty 2-3-4 tree, in the order given, the following values:

   12, 13, 17, 10, 4, 6, 9, 15, 30, 25, 20, 40.

   Show the intermediate trees after each insertion that causes a split. [6 points]

2. Insert the values above into an initially empty red-black tree. Show the tree after each insertion that causes any color shifts or rotations. [6 points]

3. From the tree derived at the end of the previous problem, delete 13 and then 12. [6 points]

4. Let $T$ and $U$ be two red-black trees storing $n$ and $m$ items, respectively, such that any item in $T$ has a key less than the keys of all items in $U$. Describe an $O(\lg n + \lg m)$ method for joining the trees into a single tree that stores all the items in $T$ and $U$. The original $T$ and $U$ may be destroyed in the process.

   - Give high-level pseudo-code for performing the join. You may use or modify already existing find, insert, and delete routines.
   - Briefly explain why the time is $O(\lg n + \lg m)$.

   [6 points extra credit]

Total: 18 points