Common mistakes in June 2013 exam paper

1.
a) A common mistake is the students change the code of the function to the point that the function does not produce the desired result.

c) When the tree is not balanced, the operations to balance should start from the deepest unbalanced node. Often students do not start from that node, leading to the wrong resulting tree. Also, students should check whether the tree needs a double rotation for balancing (which branch of the subtree actually makes the subtree unbalanced.

In the case of the hash table, the student enforce an ordering in the lists, which is not in the specifications of the question.

2.

In many cases the function does not check whether the tree is empty or not. This could be easily achieved if a NULL check on the tree pointer is performed. Also, in some cases the students write code such as hdlist->left->left, which assumes the hdlist->left is not null without checking for that condition.

In cases where the students write a function that returns a value, and the whole tree needs to be checked, they wrongly call the recursion using return such as:

```
int sumSalaries(…){

…
        return sumSalaries(hdtree->left);
        return sumSalaries(hdTree->right);
}
```

This code does not traverse the whole tree but only traverses the left branch of the tree.