Department of Computer Science and Engineering
University of California at San Diego

CSE 230
Winter 1998

Problem Set 1

DUE TUESDAY JANUARY 13 IN CLASS.

1. The task here is to implement, in four different ways using standard ML, a function that performs the mapping
$$\langle x, y\rangle \ \mapsto\ \langle x \ \text{div}\ y,\ x \ \text{mod}\ y\rangle$$
where x and y are integers. The difficulty is to give the function sensible behavior when y is zero.

   First, choose an appropriate value to use as an output "flag" when y is zero. Implement the function to return this value when needed, first using a conditional expression, and then again using pattern-matching.

   In one or two sentences, explain why output flag values are undesirable. Describe an alternative behavior that would be more desirable from the point of view of the user of this function. (There can be differences of opinion on this.) Investigate exceptions in ML and use them to write a third version of the function that behaves in the desirable way you describe.

   Finally, use the ML type system to write a fourth version of the same function that is total, i.e. that always terminates normally. This function should give a sensible result even when y is zero. For this function to interact well with other similar functions, what should its signature be? (Hint: consider the Unix "pipe" idea.)

2. Define an ML function that flattens binary trees. That is, given a binary tree the function should return a list of all atoms at the leaves of the tree, in preorder. (This list is called the fringe of the tree.) Use the following type definition for binary trees:

```
datatype btree = empty | leaf of int | node of btree * btree
```

3. Define an ML function that decides whether two binary trees have the same fringe. Your function should not compute the entire fringe of the two trees unless absolutely necessary. What is the time complexity of your function?

4. Dr. Ludd, your boss at *Hacks'r'us Software* who should take early retirement, says that it is impossible to write efficient code without using jumps, pointers, and other low-level programming language features that are not available in high-level languages like ML and Java. Using the functions that you wrote for questions 1, 2, and 3 above as examples, write a few paragraphs explaining to what extent Dr. Ludd is wrong, and to what extent he is right.