**CS 337** **Quiz 2** **3/28/03**

Open book and notes.

Max points = 50 Time = 50 min Do all questions.

1. (Programming, 32 points)

   (a) Define function `min2` which accepts a list of two or more distinct integers as input and outputs the second smallest number in the list. Thus,

   ```
   min2 [7,3] = 7
   min2 [7,3,5,1] = 3
   ```

   (b) Define function `apply` which has two inputs, a list of functions and a list of arguments, where each function can be applied to each argument. The output of `apply` is a list of lists, obtained by applying each function, in turn, to each argument. Shown below is the result of running `apply`, where `same` is the identity function, `sq`, `cube` and `quad` are the functions which compute the second, third and fourth power of the argument.

   ```
   Main> apply [same, sq, cube, quad] [2,3,4]
   [[2,3,4],[4,9,16],[8,27,64],[16,81,256]]
   ```

   (c) Define a function which returns `True` if its argument string includes ‘‘`010`’’ as a substring (i.e., a contiguous segment), `False` otherwise.

   (d) Function `flatten`, defined in your notes (section 10.1.3, page 34), takes a list of lists, like

   ```
   [ [1,2,3], [10,20], [], [30] ]
   ```

   and flattens it out by putting all the elements into a single list, like

   ```
   [1,2,3,10,20,30]
   ```

   Write a more efficient version, without using ++ .
   Hint: Consider how you would do it manually.

2. (Types; 8 points) What are the types of the following expressions?

   (a) `(3, "abc",['a','b','c'])`

   (b) Function `apply` of question (1b).

   (c) A function which accepts two strings, $xs$ and $ys$ (of arbitrary elements), and returns a list of positions in $ys$ such that $xs$ occurs as a substring starting at each of these positions.

   (d) `filter even [3,6,7]`.

3. (Proofs; 10 points) Consider the prefix sum algorithm given in your notes (just before section 9.5, page 32). A function defined there is `pt`, which is reproduced below.

```
pt [] c = []
pt (x:xs) c = (c+x) : (pt xs (c+x))
```

Consider the following function, `pd`.

```
pd [] c = []
pd (x:xs) c = (x-c) : (pd xs x)
```

Show that

```
pd (pt xs c) c = xs
```

for any list of integers `xs` and any integer `c`.