McMaster University
Department of Computing and Software
Dr. Wolfram Kahl

SE 3E03, Fall 2002
Assignment 7

Design and Selection of Programming Languages

7th October 2002

Problem 26 (Typing)

The Haskell function definition

```
> f x y = e
```

is equivalent to the following "name binding":

```
> f = \ x -> \ y -> e
```

Analogously,

```
> g (x,y) = e ... x ... y ...
```

is (for most purposes) equivalent to the name binding where the occurrences of the pattern variables `x` and `y` have been replaced by applications of "destructor" or *access* functions, here `fst` and `snd`:

```
> g = \ p -> e ... (fst p) ... (snd p) ...
```

Perform analogous conversions on the following function definitions so that you obtain name bindings. Derive the types of the defined names, calculate the order of those types, and strive to understand the meanings of these functions.

```
    > pupd (f,g) (x,y) = (f x, g y)
    > pupd1 f (x,y) = (f x, y)
    > pupd2 g (x,y) = (x, g y)
    > keep1 h (x,y) = (x, h (x,y))
    > keep2 h (x,y) = (h (x,y), y)
    > pacom (x,y) = (y,x)
    > twist f = f . pacom
    > curry f x y = f (x,y)
    > uncurry g (x,y) = g x y
    > keepof1 k h (x,y) = k (curry (twist h) y) x
    > xxp1 = keepof1 pupd1
    > xxp2 = keepof1 pupd2
    > xxk1 = keepof1 keep1
    > xxk2 = keepof1 keep2
```

Define `keepof2` and derive its typing!

Problem 27

Define Haskell functions for the following purposes:

a) return a list of `Integer` values stripped of leading zeros

b) return the first word of a `String` value containing a sentence

c) calculate the scalar product of two vectors x and y (realized as two lists of equal length)

d) add two matrices (a matrix is a list of lists, as in Problem 24).