F28PL1 Programming Langauges
Tutorial 8

1) Identify the types of the following functions:

```
a) fun a _ [] = [] |
       a f (h::t) = (2*(f h))::(a f t)

b) fun b _ [] = 0 |
       b f (h::t) = (f h)+(b f t)

c) fun c _ [] = [] |
       c f (h::t) = if f h
                    then (f h)::(c f t)
                    else c f t

d) fun d _ _ [] = [] |
       d p f (h::t) =
        if p h
        then (f h)::(d p f t)
        else d p f t

e) fun e _ _ [] = [] |
       e f1 f2 (h::t) =
       (f1 (f2 h))::(e f1 f2 t)

f) fun f _ _ [] = [] |
       f p1 p2 (h::t) =
        if p1 (p2 h)
        then h::(f p1 p2 t)
        else f p1 p2 t

g) fun g _ [] [] = [] |
       g f (h1::t1) (h2::t2) =
        if f h1
        then h2::(g f t1 t2)
        else g f t1 t2

h) fun h _ _ [] [] = [] |
       h f1 f2 (h1::t1) (h2::t2) =
        if f1 h1
        then (f2 h2)::(f f1 f2 t1 t2)
        else f f1 f2 t1 t2
```

2) Define the following functions. In each case identify the function's type.

a) generate an ascending order list of the first n multiples of 17

b) generate an ascending order list of the even numbers between 1 and n

c) generate an ascending order list of the first n even numbers using the function from b)

d) add 17 to every element in an integer list

e) put "?"s on either side of every element in a string list:

```
ques ["a","b","c"] ==> ["?a?","?b?","?c?"]
```

f) convert every element of a real list to an integer

g) find how many letters are in each element of a string list

h) NAND every element of a boolean list with true

i) from an integer list, generate a list of tuples of elements and their halves

j) select all the elements of an integer list divisible by 3

k) select all the false elements of a boolean list

l) select all the elements of a string list with less than 7 letters

m) select all the lists with more than 2 elements in a list of lists:

```
more2 [[1,2,3],[4,5],[6,7,8],[9,10]] ==>
 [[1,2,3],[6,7,8]
```

n) multiply corresponding elements of two integer lists together:

```
mult2 [1,2,3] [4,5,6] ==> [4,10,18]
```

o) compare corresponding elements of two lists to see if each element of
the first is the same as the corresponding element of the second:

```
comp ["alpha","beta","gamma"]
     ["alpha","delta","gamma"] ==> [true,false,true]
```

p) join each element of a string list onto itself the number of times indicated by the corresponding element of an integer list:

```
copy ["a","b","c","d"] [1,2,3,2] ==>
["a","bb","ccc","dd"]
```