Computer Science & Engineering 341

Assignment 6: Miranda Warmup May 11, 1998
Due: May 15, 1998

Turn in a listing of your script for Questions 2–4, and sample output showing your function working correctly.

1. *You don't need to hand in anything for Question 1 – this is just to give you some practice.*

   Experiment a bit with Miranda. Try testing some of the built-in functions. Experiment with Miranda's type system: find the type of some constants, of some built-in functions such as member and map, and of some partially applied functions, e.g.

   map (const 1).

   Try to enter some ill-typed expressions, such as

   [1,2,[3],4]

   or

   member [1] "fred".

   Answer the following questions, doing the computation by hand, then checking your answer on the machine.

   Suppose that the following Miranda script has been filed in (it is on orcas on `~borning/miranda/assign6.m`).

```
cube x = x*x*x
my_map f [] = []
my_map f (x:xs) = f x : my_map f xs
my_append [] x = x
my_append (x:xs) ys = x : my_append xs ys
my_map2 f [] [] = []
my_map2 f (a:as) (b:bs) = f a b : my_map2 f as bs
rev f x y = f y x
alligator x y = 3+y
```

   What is the result of evaluating the following Miranda expressions? (If there is a compile-time type error, or a run-time error, or a non-terminating computation, say so.) If the expression is followed by `::`, then give the type, instead of the value.

   (a) `cube 3`

   (b) `cube ::`

   (c) `my_map ::`

   (d) `my_map cube ::`

   (e) `my_map cube [1..]`

   (f) `my_append ::`

   (g) `my_map2 alligator [1..]  [10..]`

   (h) `my_map2 ::`

   (i) `my_map2 alligator ::`

(j) `my_map2 (rev alligator) ::`

(k) `alligator (1/0) (6/2)`

(l) `alligator (6/2) (1/0)`

(m) `alligator ::`

(n) `(rev alligator) 10 20`

(o) `rev ::`

(p) `rev my_map2::`

2. Write and test a Miranda function again that takes a list and returns a new list with each element repeated. For example:

```
again [3,5,7]  => [3,3,5,5,7,7]
again [1..]  => [1,1,2,2,3,3,4,4, ...]
again ["that's", "all", "folks"]  =>
      ["that's", "that's", "all", "all", "folks", "folks"]
```

   What is the type of again? (Try to figure this out by hand first, and then check your answer on the machine.)

3. Write and test a Miranda function average that computes the average of a list of numbers. (Do something sensible if you try to take the average of an empty list.)

4. A *prime pair* is a pair of prime numbers differing by 2, for example (3,5) and (17,19). Write and test a Miranda function prime_pairs that returns the very long list of prime pairs. (It is not known whether there are an infinite number of prime pairs.) Thus:

```
prime_pairs  => [(3,5),(5,7),(11,13),(17,19),(29,31),(41,43),(59,61), ...
```

   (Hint: use the function zip2 in the standard library.)