1. **[30 points]** For the following two SML functions, indicate the type that would be inferred by SML’s type inference and describe how SML’s type inference would compute the type. For your description you should not work through the type inference in detail. You just need to give the intuition.
   (a) fun aggr f lst = if length(lst) = 0 then 0 else f(hd(lst)) + aggr f (tl(lst))
   (b) fun nodup x y = if x = y then x::nil else x::(y::nil)

Assume that `length` takes a list as an argument and returns an integer (`'a list -> int`), `hd` takes a list as an argument and returns the first element (`'a list -> 'a`), and `tl` takes a list as an argument and returns a list of the same type (`'a list -> 'a list`). Also `x::nil` creates a list with a single element `x`, and `x::(y::nil)` creates a list with two elements, `x` and `y`.

2. **[30 points]** Invokeinterface is a Java bytecode which invokes a method on an interface reference. Give an example to illustrate why we cannot use virtual function tables (as in Modula-3) to implement invokeinterface.

3. **[30 points]** What is a narrowing conversion? A widening conversion? Which of these requires a run-time check (if any)?