# Problem 1

Here is a sample solution:

```
fun is_older (date1 : int * int * int, date2 : int * int * i
    let
        val y1 = #1 date1
        val m1 = #2 date1
        val d1 = #3 date1
        val y2 = #1 date2
        val m2 = #2 date2
        val d2 = #3 date2
    in
        y1 < y2 orelse (y1=y2 andalso m1 < m2)
                orelse (y1=y2 andalso m1=m2 andalso d1 < d2)
    end
```

- Be lenient on how let-expressions are used. It is okay if there are no local val bindings. It is also okay if there are more (e.g., to avoid repeating the expression **`y1=y2`**).
- For the logic expression, it is okay to use **`if ... then ... else ...`** instead of **`orelse`** and **`andalso`**, but the logic should still be clear: starting by comparing the year, then the month, then the day. If the logic is hard to follow, give a 4 or 3.

Give a 3 for this sort of more imperative looking code:

```
fun is_older (date1 : int * int * int, date2 : int * int * i
    let val y1 = #1 date1
        val m1 = #2 date1
        val d1 = #3 date1
        val y2 = #1 date2
        val m2 = #2 date2
        val d2 = #3 date2
    in
        let val b1 = y1 < y2
        in
           if b1
           then true
           else let val b2 = y1 > y2
                in
                  if b2
                  then false
                  else ...
           ...
    end
```