

**Comments for exam in Dat501—Functional Programming, Mar 21 2003**

1.a The correct solution is

```
type voralonDate = int * int * int;
```

A popular *mistake* is

```
datatype voralonDate = day of int | month of int | year of int
```

1.c `before((d,m,y),(d',m',y')) = (y<y') orelse (y=y' andalso m<m')  
orelse (y=y' andalso m=m' andalso (d<d'));`

1.d `fun tomorrow(30,12,y) = (1, 1, y+1)  
| tomorrow(30,m, y) = (1, m+1,y)  
| tomorrow(d ,m, y) = (d+1,m, y);`

1.e `fun add(d,0) = d  
| add(d,i) = tomorrow(add(d,i-1));`

1.f `add((1,1,1),0), add((1,1,1),1), add((30,1,1),1), add((30,12,1),1)`

1.c–f There a completely different approach to these exercises as well, which involves a function

```
fun toDays(d,m,y) = d + m*30 + y*30*12;
```

that computes the number of days of a given date (since the beginning of the calendar), and a similar inverse function `toVoralonDate`.

2.c `fun polyAdd(p ,[]) = p  
| polyAdd([],p) = p  
| polyAdd(x::xs,y::ys) = [x+y]::polyAdd(xs,ys);`

3.a `Parent("Finw", Childless("Finarfin"), Childless("Fingolfin"))`

3.c `f: person * person -> bool`

3.d `f(x,y)` is true if and only if `x` is a child of `y`.