Solutions for CP Exercises, Week 9

1. Solution for Andrews Ex. 7.3

(a) Assume that the array to be sorted is put in the variable *input*.

```
var input[1..n] : integer := ...;
    output[1..n] : integer;
chan pass[i : 0..n] : integer;
process Feeder =
 for i in 1..n do
    send pass[0](input[i]);
process HoldMin[i : 1..n] =
  var min, x : integer;
 receive pass[i-1](min);
 for j in 1..n - 1 do
    { receive pass[i-1](x);
      if (x \ge min) then send pass[i](x)
                     else {send pass[i](min); min := x}
    };
 send pass[i](min);
process Collector =
 for i in 1..n do
    receive pass[n](output[n + 1 - i]);
```

Now, the values sent on pass[n] will be sorted and can be inserted into the result array *output* by the collector.

Alternatively, the filter processes could directly set "its" element in the result array to the minimum value and just pass on the remaining ones.

2. Solution for Andrews Ex. 7.6

```
type Kind = Read | Write;
chan request : (Kind, integer);
chan release : ();
chan readok[i..n] : ();
chan writeok[i..m] : ();
process Reader[i : 1..n] =
                                   process Writer[j : 1..m] =
  . . .
                                     . . .
  send request(Read, i)
                                     send request(Write, j)
  receive readok[i]();
                                     receive writeok[j]();
  reading;
                                     writing;
  send release();
                                     send release();
  . . .
                                     . . .
process RWControl =
  var k : kind;
      id : integer;
       active : integer := 0;
  receive request(k, id);
  repeat
    if k = \text{Read then}
      while k = \text{Read } \mathbf{do}
         { send okread[id]();
           active := active + 1;
           receive request(k, id);
         }
    else
       { send okwrite[id]();
        active := 1;
        receive request(k, id);
      }
    while active > 0 do
       {receive release(); active := active - 1}
  forever
```

Rather than accepting all request at any time and record pending request, it has here been chosen to serve requests in order of arrival as long as possible, ie. either serve a single write request or a consecutive sequence of read requests. Releases need only be considered when a new request is about to be served.