1993 Paper 1 Question 12

A list in Modula-3 can be represented as a sequence of links. Each link is a record containing one value in the list and a reference to the rest of the list following the link. The following `TYPE` declaration specifies the required data structure:

```
TYPE
  List = REF Link;
  Link = RECORD value:CARDINAL; rest: List:= NIL END;
```

A test program which exploits lists of this kind includes:

```
VAR
  start:List;

BEGIN
  start:= NIL;
  Put(10,start);
  Put(100,start);
  Put(1000,start);
  Print(start);
  Print(Reverse1(start));
  Print(reverse2(start));
```

The procedure call `Put(1000,start)` will add a link containing the value `1000` to the *end* of the list which already includes the values `10` and `100`.

The procedure `Print` writes out the values in a list in order.

The procedures `Reverse1` and `Reverse2` reverse a list in two different ways, equivalent to the ML functions:

```
fun Reverse1 [] = []
 |  Reverse1 (value::rest) = Reverse1 (rest) @ [value];

fun Reverse2 list =
  let
    fun rev ([],result) = result
     |  rev (value::rest, result) = rev (rest,value::result)
  in
    rev (list,[])
  end;
```

Write the Modula-3 procedures `Put, Print, Reverse1` and `Reverse2`.