ALGORITHM 86
PERMUTE
J. E. L. PECK AND G. F. SCHRACK
University of Alberta, Calgary, Alberta, Canada

```
procedure PERMUTE (x, n);
array x; integer n;
comment Each call of PERMUTE executes a permutation of
  the first n components of x. It assumes a nonlocal Boolean
  variable 'first', which when true causes the procedure to initial-
  ise the signature vector p. Thereafter 'first' remains false until
  after n! calls;
begin own integer array p[2:n]; integer i, k;
  if first then
  begin for i := 2 step 1 until n do
            p[i] := i; first := false
  end initialise;
  for k := 2 step 1 until n do
      begin integer km; real t;
        t := x[1]; km := k - 1;
        for i := 1 step 1 until km do
            x[i] := x[i+1];
        x[k] := t; p[k] := p[k] - 1;
        if p[k] ≠ 0 then go to EXIT;
        p[k] := k
      end k;
  first := true;
EXIT: end PERMUTE
```

CERTIFICATION OF ALGORITHM 86
PERMUTE [J. E. L. Peck and G. F. Schrock, *Comm. ACM*, Apr. 1962]
D. M. COLLISON
Elliott Bros. (London) Ltd., Borehamwood, Herts., England

The algorithm was successfully run using the Elliott ALGOL translator on the National-Elliott 803. Values of n used were 0, 1, 2, 3, 4.