```
! Generate all permuations of the numbers 1, 2, ..., NPERM;
!Ref:
 Heap, B. R. (1963). "Permutations by Interchanges". The Computer Journal. 6 (3): 293-4..
 Sedgewick, R. (1977). "Permutation Generation Methods". ACM Computing Surveys. 9 (2):
137-164.;
!Keywords: Enumerate, Heap, LINGO, Ordering, Permutation, Sedgewick;
SETS:
 PSET :  IPERM, CStack;
ENDSETS
DATA:
  NPERM = 4;
  PSET = 1..NPERM;
ENDDATA

PROCEDURE GenPerm:
 ! Generate all permuations of 1, 2, ... , NPERM;
 ! Inputs:
     NPERM = number items to be permuted,
     IsNew = 1 if f1st call, else 0;

!  cStack is an encoding of the state of enumeration;
  PermCnt = 0; ! Count number of permutations;
! Begin special case of first iteration;
  @IFC( IsNew:
    IsNew = 0;
    PERM_i = 0;
   @WHILE( PERM_i #LT# NPERM:
     PERM_i = PERM_i + 1;
     IPERM( PERM_i) = PERM_i;
     cStack( PERM_i) = 1;
        );

 ! Output current permutation;
    Permcnt = PerMcnt + 1;
    @WRITE( Permcnt,') ');
    @FOR( PSET( ii): @WRITE( IPerm( ii), ' '));
    @WRITE( @NEWLINE( 1));
     );  ! End 1st

!   Perm_j is effectively a stack pointer;
    Perm_j = 1;
    @WHILE( Perm_j #LE# NPERM :
      @IFC(  cStack( Perm_j) #LT# Perm_j:
          @IFC( Perm_j #NE# 2*@INT( Perm_j/2):  !is even then;
!                swap(IPerm( 1), IPerm( Perm_j));
                 Temp = IPerm( 1);
                 IPerm( 1) = IPerm( Perm_j);
                 IPerm( Perm_j) = temp;
            @ELSE
 !               swap(IPerm( cStack( Perm_j)), IPerm( Perm_j));
                 Temp = IPerm( cStack( Perm_j));
                 IPerm( cStack( Perm_j)) = IPerm( Perm_j);
                 IPerm( Perm_j) = temp;
             );

! Output current permutation;
    Permcnt = PerMcnt + 1;
    @WRITE( Permcnt,') ');
    @FOR( PSET( ii): @WRITE( IPerm( ii), ' '));
    @WRITE( @NEWLINE( 1));

! Swap has occurred ending the for-loop. Simulate the increment of the for-loop counter;
    cStack( Perm_j) =  cStack( Perm_j) + 1;
! Simulate recursive call reaching the base case by bringing the pointer to the base case
analog in the array;
     Perm_j = 1;
       @ELSE
! Reset the stack pointer;
```

```
        cStack( Perm_j) = 1;
        Perm_j = Perm_j + 1;
      )
   ); ! end while;
ENDPROCEDURE

 CALC:
    NPERM = @SIZE( PSET);
    GenPerm;
 ENDCALC
```