Author(s) : F. Beck, T. Lindelöf
Submitter : K.S. Kölbig
Language : Fortran

Library: MATHLIB
Submitted: 15.09.1978
Revised: 07.06.1992

Permutations and Combinations

Successive calls to subroutine subprogram `PERMU` will generate all permutations of a set of integers of total length N consisting of n_1 repetitions of the integer 1, followed by n_2 repetitions of the integer $2, \ldots$ etc, concluding with n_m repetitions of the integer m, where $\sum_{j=1}^{m} n_j = N$.
Subroutine subprogram `PERMUT` generates *directly* a single member of the set of all lexicographically ordered permutations of the first integers $1, 2, \ldots, N$, as specified by its lexicographical ordinal.
Successive calls to subroutine subprogram `COMBI` will generate all the $\binom{N}{J}$ possible combinations without repetition of $J \leq N$ integers from the set $\{1, 2, \ldots, N\}$.

Structure:

`SUBROUTINE` subprogram
User Entry Names: `PERMU`, `PERMUT`, `COMBI`
Files Referenced: `Unit 6`

Usage:

Subroutine PERMU:

```
CALL PERMU(IA,N)
```

`IA` (`INTEGER`) One-dimensional array of length $\geq$ `N`. On entry, `IA(i)`, $(\mathtt{i} = 1, 2, \ldots, \mathtt{N})$, must contain the initial set of integers to be permuted (see **Examples**). A call with `IA(1)` $= 0$ will place the set $\{1, 2, \ldots, \mathtt{N}\}$ in `IA`. On exit, `IA` contains the "next" permutation. If all the permutations have been generated, the next call sets `IA(1)` $= 0$.

`N` (`INTEGER`) Length of the set to be permuted.

Subroutine PERMUT:

```
CALL PERMUT(NLX,N,IP)
```

`NLX` (`INTEGER`) Lexicographical ordinal of the permutation desired.

`N` (`INTEGER`) Length of the set to be permuted.

`IP` (`INTEGER`) One-dimensional array of length $\geq$ `N`. On exit, `IP(i)`, $(\mathtt{i} = 1, 2, \ldots, \mathtt{N})$, contains the `NLX`-th lexicographically ordered permutation of the integers $1, 2, \ldots, N$ (see **Examples**).

Subroutine COMBI:

```
CALL COMBI(IC,N,J)
```

`IC` (`INTEGER`) One-dimensional array of length $\geq \mathtt{N} + 1$. The first call must be made with `IC(1)` $= 0$. This generates the first combination `IC(i)` $=$ `i`, $(\mathtt{i} = 1, 2, \ldots, \mathtt{J})$. Each successive call generates a new combination and places it in the first `J` elements of `IC`. If all the combinations have been generated, the next call sets `IC(1)` $= 0$.

`N` (`INTEGER`) Length of the set from which the combinations are taken.

`J` (`INTEGER`) Length of the combinations.

Examples:

1. Consider the following set of $N = 12$ objects, only 8 are different:

$$\{y_1, y_2, y_3, y, y, r_1, r_2, r, r, b, b, b\}.$$

This set consists of $m = 8$ sequences of length $n_1 = n_2 = n_3 = n_5 = n_6 = 1,\ \ n_4 = n_7 = 2$, $n_8 = 3$. Thus, in order to get the possible permutations, set

$$\texttt{IA} = \{\texttt{1 2 3 4 4 5 6 7 7 8 8 8}\}$$

before calling `PERMU(IA,12)` the first time.

2. To generate all permutations of N indistinguishable objects, set `IA(1)` $= 0$, which is equivalent to `IA(i)` $=$ `i`, (`i` $= 1, 2, \ldots,$ `N`), before calling `PERMU(IA,N)` the first time.

3. To compute the, lexicographically second, third and last $(4! = 24)$ permutions of the set $\{1, 2, 3, 4\}$:

`CALL PERMUT( 2,4,IP)`	sets	`IP` $= \{1, 2, 4, 3\}$
`CALL PERMUT( 3,4,IP)`	sets	`IP` $= \{1, 3, 2, 4\}$
`CALL PERMUT(24,4,IP)`	sets	`IP` $= \{4, 3, 2, 1\}$

4. To generate and print all 20 combinations of 3 integers from the set $\{1, 2, 3, 4, 5, 6\}$ one could write:

```
  ...
  IA(1)=0
1 CALL COMBI(IC,6,3)
  IF(IC(1) .NE. 0) THEN
   PRINT *, IC(1),IC(2),IC(3)
   GO TO 1
  ENDIF
  ...
```

Restrictions:

`PERMUT`: $1 \le$ `NLX` $\le$ `N`!, `N` ≤ 12.
`COMBI`: `J` $\le$ `N`.

Error handling:

If any of the above conditions is not satisfied, a message is written on `Unit 6`.

Notes:

1. If `N` ≤ 0 or `J` ≤ 0, the subprograms return control without action.

2. The number of distinct permutations of a set of N numbers which can be decomposed into m groups of $n_1, n_2, \ldots, n_m$ indistinguishable elements is given by

$$\frac{N!}{n_1!\,n_2!\cdots n_m!}$$

where $n_1 + n_2 + \cdots + n_m = N$. This number can become large even for seemingly simple cases, e.g. in Example 1 above,

$$\frac{12!}{1!\,1!\,1!\,2!\,1!\,1!\,2!\,3!} = 19958400.$$

•