1 Purpose
nag_pack_real_mat_print (x04ccc) is an easy-to-use function to print a double triangular matrix stored in a packed one-dimensional array.

2 Specification
#include <nag.h>
#include <nagx04.h>

void nag_pack_real_mat_print (Nag_OrderType order, Nag_UploType uplo,
    Nag_DiagType diag, Integer n, const double a[], const char *title,
    const char *outfile, NagError *fail)

3 Description
nag_pack_real_mat_print (x04ccc) prints a double triangular matrix stored in packed form. It is an easy-to-use driver for nag_pack_real_mat_print_comp (x04cdc). The function uses default values for the format in which numbers are printed, for labelling the rows and columns, and for output record length.

nag_pack_real_mat_print (x04ccc) will choose a format code such that numbers will be printed with a %8.4f, a %11.4f or a %13.4e format. The %8.4f code is chosen if the sizes of all the matrix elements to be printed lie between 0.001 and 1.0. The %11.4f code is chosen if the sizes of all the matrix elements to be printed lie between 0.001 and 9999.9999. Otherwise the %13.4e code is chosen.

The matrix is printed with integer row and column labels, and with a maximum record length of 80. The matrix is output to the file specified by outfile or, by default, to standard output.

4 References
None.

5 Arguments
1:  order - Nag_OrderType
    Input
    On entry: the order argument specifies the two-dimensional storage scheme being used, i.e., row-major ordering or column-major ordering. C language defined storage is specified by order = Nag_RowMajor. See Section 3.2.1.3 in the Essential Introduction for a more detailed explanation of the use of this argument.
    Constraint: order = Nag_RowMajor or Nag_ColMajor.

2:  uplo - Nag_UploType
    Input
    On entry: indicates the type of the matrix to be printed
    uplo = Nag_Lower
    The matrix is lower triangular
    uplo = Nag_Upper
    The matrix is upper triangular
    Constraint: uplo = Nag_Lower or Nag_Upper.
3: \textbf{diag} – Nag_DiagType \hspace{1cm} \textit{Input}

\textit{On entry:} indicates whether the diagonal elements of the matrix are to be printed.

- \textbf{diag} = Nag_NonRefDiag: The diagonal elements of the matrix are not referenced and not printed.
- \textbf{diag} = Nag_UnitDiag: The diagonal elements of the matrix are not referenced, but are assumed all to be unity, and are printed as such.
- \textbf{diag} = Nag_NonUnitDiag: The diagonal elements of the matrix are referenced and printed.

\textit{Constraint:} \textbf{diag} = Nag_NonRefDiag, Nag_UnitDiag or Nag_NonUnitDiag.

4: \textbf{n} – \texttt{Integer} \hspace{1cm} \textit{Input}

\textit{On entry:} the order of the matrix to be printed.

If \textbf{n} is less than 1, \texttt{nag_pack_real_mat_print (x04ccc)} will exit immediately after printing \texttt{title}; no row or column labels are printed.

5: \textbf{a}[$\dim$] – const double \hspace{1cm} \textit{Input}

\textit{Note:} the dimension, \texttt{dim}, of the array \texttt{a} must be at least $\max(1, \textbf{n} \times (\textbf{n} + 1)/2)$.

\textit{On entry:} the matrix to be printed. Note that \texttt{a} must have space for the diagonal elements of the matrix, even if these are not stored.

The storage of elements $A_{ij}$ depends on the \texttt{order} and \texttt{uplo} arguments as follows:

- if \texttt{order} = Nag_ColMajor and \texttt{uplo} = Nag_Upper, $A_{ij}$ is stored in \texttt{a}[$(j - 1) \times j/2 + i - 1$], for $i \leq j$;
- if \texttt{order} = Nag_ColMajor and \texttt{uplo} = Nag_Lower, $A_{ij}$ is stored in \texttt{a}[$(2n - j) \times (j - 1)/2 + i - 1$], for $i \geq j$;
- if \texttt{order} = Nag_RowMajor and \texttt{uplo} = Nag_Upper, $A_{ij}$ is stored in \texttt{a}[$(2n - i) \times (i - 1)/2 + j - 1$], for $i \leq j$;
- if \texttt{order} = Nag_RowMajor and \texttt{uplo} = Nag_Lower, $A_{ij}$ is stored in \texttt{a}[$(i - 1) \times i/2 + j - 1$], for $i \geq j$.

If \textbf{diag} = Nag_UnitDiag, the diagonal elements of \texttt{A} are assumed to be 1, and are not referenced; the same storage scheme is used whether \textbf{diag} = Nag_NonUnitDiag or \textbf{diag} = Nag_UnitDiag.

6: \textbf{title} – const char * \hspace{1cm} \textit{Input}

\textit{On entry:} a title to be printed above the matrix.

If \textbf{title} = \texttt{NULL}, no title (and no blank line) will be printed.

If \textbf{title} contains more than 80 characters, the contents of \textbf{title} will be wrapped onto more than one line, with the break after 80 characters.

Any trailing blank characters in \textbf{title} are ignored.

7: \textbf{outfile} – const char * \hspace{1cm} \textit{Input}

\textit{On entry:} the name of a file to which output will be directed. If \textbf{outfile} is \texttt{NULL} the output will be directed to standard output.

8: \textbf{fail} – NagError * \hspace{1cm} \textit{Input/Output}

The NAG error argument (see Section 3.6 in the Essential Introduction).
6 Error Indicators and Warnings

**NE_ALLOC_FAIL**
Memory allocation failed.

**NE_BAD_PARAM**
On entry, argument \( \text{\textless value\textgreater} \) had an illegal value.

**NE_INTERNAL_ERROR**
An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

**NE_NOT_APPEND_FILE**
Cannot open file \( \text{\textless value\textgreater} \) for appending.

**NE_NOT_CLOSE_FILE**
Cannot close file \( \text{\textless value\textgreater} \).

**NE_NOT_WRITE_FILE**
Cannot open file \( \text{\textless value\textgreater} \) for writing.

7 Accuracy
Not applicable.

8 Parallelism and Performance
Not applicable.

9 Further Comments
A call to \texttt{nag_pack_real_mat_print (x04ccc)} is equivalent to a call to \texttt{nag_pack_real_mat_print_comp (x04cdc)} with the following argument values:

\[
\begin{align*}
\text{ncolm} &= 80 \\
\text{indent} &= 0 \\
\text{labrow} &= \text{Nag_IntegerLabels} \\
\text{labcol} &= \text{Nag_IntegerLabels} \\
\text{form} &= 0
\end{align*}
\]

10 Example
See Section 10 in \texttt{nag_sum_sqs_update (g02btc)}. 