

```

1: *****
2: /resh.f90
3: *****
4:
5: PROGRAM resh
6: ***
7: *** Example using the intrinsic RESHAPE function
8: ***
9:
10: USE resh_mod
11:
12: IMPLICIT NONE
13:
14: REAL, DIMENSION(16) :: aa !** Declare a 1D array
15: REAL, DIMENSION(4,4) :: bb
16:
17: !** Assign values to the array aa with a 1D array constructor
18: aa=(/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16/)
19:
20: bb=RESHAPE(aa,(/4,4/)) !** Reshape with default (/1,2/) ordering
21: CALL outmat(bb) !** Your own "outmat" subroutine
22: PRINT*, ""
23: PRINT*, ""
24: PRINT*, "And the transpose is : "
25: PRINT*, ""
26: PRINT*, ""
27:
28: bb=RESHAPE(aa,(/4,4/), ORDER=/(2,1/)) !** Reshape with order (/2,1/)
29: CALL outmat(bb) !** Your own "outmat" subroutine
30: PRINT*, ""
31:
32:
33: END PROGRAM resh
34:
35: *****
36:
37: *****
38: /resh_mod.f90
39: *****
40:
41: MODULE resh_mod
42:
43: IMPLICIT NONE
44:
45: CONTAINS
46:
47: ! *****
48:
49: FUNCTION getmat(m,n)
50: !**** Function to input a matrix from the keyboard. The number of rows
51: !**** (m) and the number of columns (n) are input arguments to the
52: !**** function
53:
54: INTEGER, INTENT(IN) :: m,n !**** Dummy declaration
55: REAL, DIMENSION(m,n) :: getmat !**** Local Declaration
56:
57: INTEGER :: i
58:
59: DO i=1,m
60: PRINT 'Enter matrix row :',i2',i !** Prompt for row number
61: READ*,getmat(i,:)
62: ENDDO
63:
64: END FUNCTION getmat
65:
66: ! *****
67:

```

```

68: SUBROUTINE outmat(mat)
69: !**** Subroutine to output a matrix to the screen.
70:
71: REAL, DIMENSION(:,:) , INTENT(IN) :: mat !** Dummy declaration
72:
73: INTEGER :: i
74:
75: DO i=1,SIZE(mat,1)
76: PRINT*,mat(i,:)
77: ENDDO
78:
79: END SUBROUTINE outmat
80:
81: ! *****
82:
83:
84: END MODULE resh_mod
85: *****
86:

```