

```

1: *****
2: *****
3: ./part2.f90
4:
5: PROGRAM part2
6: !*** Program to multiply together two matrices mat1 & mat2 and store
7: !*** the result in mat3, ie "mat3 = mat1 X mat2" where "X" represents
8: !*** matrix multiplication.
9:
10:
11: IMPLICIT NONE
12:
13: INTEGER, PARAMETER :: m=3,n=4,k=3
14: REAL, DIMENSION(m,n) :: mat1
15: REAL, DIMENSION(n,k) :: mat2
16: REAL, DIMENSION(m,k) :: mat3,mat4
17:
18:
19: !*** Input the two matrices from the keyboard
20: !*** Press return after each row
21: !*** matrix one then matrix two
22:
23: mat1=getmat(m,n)
24: mat2=getmat(n,k)
25:
26: !*** Use our matrix mult. function to calc. mat3
27: mat3=mulmat(mat1,mat2)
28:
29: !*** Perform the matrix multiplication
30: !*** using Fortran's MATMUL(A,B) function
31:
32: mat4=MATMUL(mat1,mat2)
33:
34: !*** Print out each matrix to the screen
35: !*** NOTE the use of / to create a blank line
36:
37: PRINT '(//,"Matrix One",//)'
38: CALL outmat(mat1)
39:
40: PRINT '(//,"Matrix Two",//)'
41: CALL outmat(mat2)
42:
43:
44: PRINT '(//,"Matrix1 * Matrix2 (My Answer)",//)'
45: CALL outmat(mat3)
46:
47:
48: PRINT '(//,"Matrix1 * Matrix2 (Fortran Answer)",//)'
49: CALL outmat(mat4)
50:
51: CONTAINS
52:
53: ! *****
54:
55: FUNCTION getmat(m,n)
56: !** Read in matrix with "m" rows and "n" columns
57:
58: INTEGER, INTENT(IN) :: m,n !*** Dummy declaration
59: REAL, DIMENSION(m,n) :: getmat !*** Local Declaration (automatic)
60:
61: INTEGER :: i ! ** Loop variable
62:
63: DO i=1,m
64: PRINT '("Enter matrix row :",i2)',i !** Prompt for row number
65: READ*,getmat(i,:) !** Read in row
66: ENDDO
67:

```

```

68: END FUNCTION getmat
69:
70: ! *****
71:
72: SUBROUTINE outmat(mat)
73: !** Print any matrix out to screen
74:
75: REAL, DIMENSION(:,:), INTENT(IN) :: mat !*** Dummy declaration
76:
77: INTEGER :: i ! ** Loop variable
78:
79: DO i=1,SIZE(mat,1) !*** Loop for each row.
80: PRINT*,mat(i,:) !** Print each row.
81: ENDDO
82:
83: END SUBROUTINE outmat
84:
85: ! *****
86:
87: FUNCTION mulmat(mat1,mat2)
88: !*** Function to premultiply mat2 with mat1
89:
90:
91: !*** Use assumed shape arrays for dummy arrays***
92: REAL, DIMENSION(:,:), INTENT(IN) :: mat1,mat2 !** Dummy declaration
93:
94:
95: !***** Local Declarations *****
96: INTEGER :: m,n,k,i,j,p
97: REAL, DIMENSION(SIZE(mat1,1),SIZE(mat2,2)) :: mulmat
98:
99: !***** Find out the matrix sizes for the DO loop limits *****
100:
101: m=SIZE(mat1,1) ; n=SIZE(mat1,2) ; k=SIZE(mat2,2)
102:
103: !*** Perform the matrix multiplication
104: !*** using three DO loops
105:
106: IF (SIZE(mat2,1) == n) THEN !** For each row of getmat (mat3)
107: DO i=1,m !** For each column of getmat (mat3)
108: DO j=1,k !** initialise to zero
109: mulmat(i,j)=0
110: DO p=1,n
111: mulmat(i,j)=mulmat(i,j)+mat1(i,p)*mat2(p,j)
112: ENDDO
113: ENDDO
114: ENDDO
115: ELSE
116: PRINT*, "Size mismatch in mulmat"
117: ENDIF
118:
119: END FUNCTION mulmat
120:
121: ! *****
122:
123: END PROGRAM part2
124: *****
125:

```