

solutions.txt	Thu Nov 07 12:21:32 2013	1	Thu Nov 07 12:21:32 2013	2	solutions.txt
---------------	--------------------------	---	--------------------------	---	---------------

```

1:
2: *****
3: ./part3_mod.f90
4:
5: MODULE part3_mod
6:
7: IMPLICIT NONE
8:
9: CONTAINS
10:
11: ! *****
12:
13: FUNCTION getmat(m,n)
14: ***** Function to input a matrix from the keyboard. The number of rows
15: ***** (m) and the number of columns (n) are input arguments to the
16: ***** function
17:
18: INTEGER, INTENT(IN) :: m,n          !**** Dummy declaration
19: REAL, DIMENSION(m,n) :: getmat    !**** Local Declaration
20:
21: INTEGER :: i
22:
23: DO i=1,m
24: PRINT 'Enter matrix row :',i2',i !**** Prompt for row number
25: READ*,getmat(i,:) !**** Read in row
26: ENDDO
27:
28: END FUNCTION getmat
29:
30: ! *****
31:
32: SUBROUTINE outmat(mat)
33: ***** Subroutine to output a matrix to the screen.
34:
35: REAL, DIMENSION(:,:), INTENT(IN) :: mat !**** Dummy declaration
36:
37: INTEGER :: i
38:
39: DO i=1,SIZE(mat,1)
40: PRINT*,mat(i,:)
41: ENDDO
42:
43: END SUBROUTINE outmat
44:
45: ! *****
46:
47: FUNCTION mulmat(mat1,mat2)
48: !**** Function to input two matrices [mat1] & [mat2] and check if
49: !**** [mat1]*[mat2] is a valid matrix multiplication. If it is valid the
50: !**** matrix product [mat1]*[mat2] is returned.
51:
52: REAL, DIMENSION(:,:), INTENT(IN) :: mat1
53: REAL, DIMENSION(:,:), INTENT(IN) :: mat2
54:
55: INTEGER :: m,n,k,i,j,p
56:
57: REAL, DIMENSION(SIZE(mat1,1),SIZE(mat2,2)) :: mulmat
58:
59: m=SIZE(mat1,1) ; n=SIZE(mat1,2) ; k=SIZE(mat2,2)
60:
61: !**** Perform the matrix multiplication
62: !**** using three DO loops
63:
64: IF (SIZE(mat2,1) == n) THEN
65: DO i=1,m
66: DO j=1,k
67: mulmat(i,j)=0
68:
69: DO p=1,n
70: mulmat(i,j)=mulmat(i,j)+mat1(i,p)*mat2(p,j)
71: ENDDO
72: ENDDO
73: ELSE
74: PRINT*, "Size mismatch in mulmat"
75: ENDIF
76:
77: END FUNCTION mulmat
78:
79: ! *****
80:
81: FUNCTION mulmatvec(mat,vec)
82: !**** Function to input a matrix [mat] and a vector [v] check if matrix
83: !**** vector multiplication is valid w.r.t. their sizes. If it is then
84: !**** this function returns the matrix vector product.
85:
86: REAL, DIMENSION(:,:), INTENT(IN) :: mat
87: REAL, DIMENSION(:), INTENT(IN) :: vec
88:
89: INTEGER :: m,n,k,i,j
90: REAL, DIMENSION(SIZE(mat,1)) :: mulmatvec
91:
92: m=SIZE(mat,1) ; n=SIZE(mat,2) ; k=SIZE(mat,1)
93:
94: !**** Perform the matrix multiplication using three DO loops
95: IF (n=k) THEN
96: DO i=1,m
97: mulmatvec(i)=0
98: DO j=1,k
99: mulmatvec(i)=mulmatvec(i)+mat(i,j)*vec(j)
100: ENDDO
101: ENDDO
102: ELSE
103: PRINT*, "Size mismatch in mulmatvec!"
104: ENDIF
105:
106: END FUNCTION mulmatvec
107:
108: ! *****
109:
110:
111: END MODULE part3_mod
112: *****
113:
114:
115: *****
116: ./part3.f90
117:
118: PROGRAM part3
119:
120: !**** Program to multiply together two matrices mat1 & mat2 and store
121: !**** the result in mat3. ie "mat3 = mat1 x mat2" where "x" represents
122: !**** matrix multiplication.
123:
124:
125: USE part3_mod
126:
127: IMPLICIT NONE
128:
129: INTEGER, PARAMETER :: m=3,n=4,k=3
130: REAL, DIMENSION(m,n) :: mat1
131: REAL, DIMENSION(n,k) :: mat2
132: REAL, DIMENSION(m,k) :: mat3,mat4
133:
134: !**** Input the two matrices from the keyboard

```

```

135: !*** Press return after each row
136: !*** matrix one then matrix two
137:
138: mat1=getmat(m,n)
139: mat2=getmat(n,k)
140:
141: !*** Use our matrix mult. function to calc. mat3
142: mat3=mulmat(mat1,mat2)
143:
144: !*** Perform the matrix multiplication
145: !*** using Fortran's MATMUL(A,B) function
146:
147: mat4=MATMUL(mat1,mat2)
148:
149: !*** Print out each matrix to the screen
150: !*** NOTE the use of / to create a blank line
151:
152: PRINT '(/"Matrix One"/)'
153: CALL outmat(mat1)
154:
155: PRINT '(/"Matrix Two"/)'
156: CALL outmat(mat2)
157:
158:
159: PRINT '(/"Matrix1 * Matrix2 (My Answer)"/)'
160: CALL outmat(mat3)
161:
162:
163: PRINT '(/"Matrix1 * Matrix2 (Fortran Answer)"/)'
164: CALL outmat(mat4)
165:
166: END PROGRAM part3
167:
168: *****
169:

```