Karlsruhe Institute of Technology

Fortran 95/2003 Course

Exercises by Robert Barthel
March 25, 2015

STEINBUCH CENTRE FOR COMPUTING - SCC

KIT – University of the State of Baden-Württemberg and
National Laboratory of the Helmholtz Association

www.kit.edu

Exercise 2.1

- Declare the integer array *iarray* which contains 3 rows and 4 columns. Using an array constructor, initialize the first row with integers 1-4 in this order from left to right, the second row with integers 5-8, and the lowest row putting integer -2 to each column. Print out *iarray* row by row, so that each output line contains the elements of one row of the array.
- Build also the 3x8 integer array *bigarray* where the first 4 columns are identical with the array *iarray*, and the 4 last columns are obtained from the columns of the array *iarray* by multiplying them with the number 3 and adding 5 to the product. Use array syntax in the statements. Print out *bigarray* row by row.

Exercise 2.2

- Implement a program that reads three integer numbers *m*, *n* and *p*. Allocate a *mxn* real array *a* and set its array elements to *i/j*, *i=1,…,m*, *j=1,…,n*. Compute *a**p* using the intrinsic function `MATMUL`.
  - Check that *p* is positive or zero
  - Check that the matrix has at least one element
  - Check that the matrix is square
  - Take care of the case *p*=0
  - Allocate an array *tmp* with the same shape as *a* and initialize it to the values of *a*
  - Replace *a* by *a**p* using repeated matrix multiplication `MATMUL` and not element-wise multiplication (Syntax: `C=MATMUL(A,B)`)

Exercise 2.3

- Implement a program that reads many non-negative real numbers. The numbers should be stored in a linked list. Stop reading if a negative number occurs. At the end, the numbers should be printed out.

  Hints:

  - Declare a derived type *node* for the linked list
  - Define two scalar pointers *curr* and *new* of type *node: new* gets the read number, *curr* always points to the last *node* in the list
  - Define a scalar target *first* of type *node*
  - At the beginning, *curr* must point to *first*

Exercise 2.3 (2)

Hints (cont.):

- Use an infinite `DO`-loop: In each cycle, read a real number from STDIN and check if it is non-negative. Allocate a new *node* and set its value to the read number. Assign the pointer component of *curr* and the pointer *curr* itself to the new *node.*
- To print out the read numbers, you have to go back to the first *node* of the list. Afterwards use an infinite `DO`-loop: Check if the current *node* is associated. If yes, print out the value of the current *node*. Then the pointer *curr* must point to the next *node*.