



Εισαγωγή στον Προγραμματισμό Η/Υ (Fortran 90/95/2003)

Ενότητα 8: Συναρτήσεις

Νίκος Καραμπετάκης
Τμήμα Μαθηματικών



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Περιεχόμενα Ενότητας

1. Η τεχνική του τμηματικού προγραμματισμού.
2. Τι είναι υποπρόγραμμα και ποιοι είναι οι κύριοι λόγοι ανάπτυξης υποπρογραμμάτων.
3. Κατηγορίες υποπρογραμμάτων.
4. Κατηγορίες συναρτήσεων-Σύνταξη Συνάρτησης υποπρογράμματος.
5. Συναρτήσεις – Διαδικασίες της αριθμητικής IMSL βιβλιοθήκης της Compaq Visual Fortran.



Σκοποί Ενότητας

1. Η παρουσίαση των υποπρογραμμάτων.
2. Η παρουσίαση του ορισμού και του τρόπου δήλωσης των συναρτήσεων.
3. Η παρουσίαση των μεθόδων εισαγωγής δεδομένων/επιστροφής τιμών σε συναρτήσεις.



Παραδείγματα όπου απαιτείται η χρήση συναρτήσεων – διαδικασιών

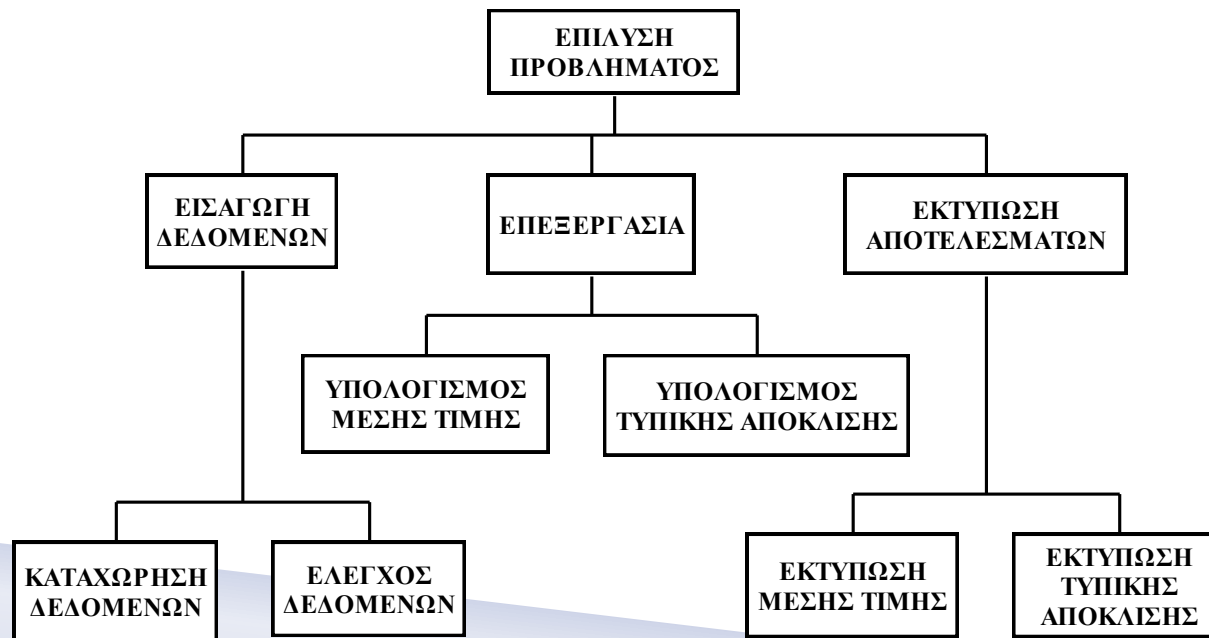
$$e^A = I_n + \frac{1}{1!}A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \dots$$

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}, n > m$$



Η τεχνική του τμηματικού προγραμματισμού

Είναι η τεχνική κατά την οποία η επίλυση ενός προβλήματος ανάγεται στην επίλυση άλλων απλούστερων προβλημάτων, τα οποία με τη σειρά τους μπορούν να ανάγονται σε επίσης απλούστερα προβλήματα.



Τι είναι το υποπρόγραμμα;

Ένα τμήμα προγράμματος που επιτελεί ένα αυτόνομο έργο και έχει γραφτεί χωριστά από το υπόλοιπο πρόγραμμα. Αν έχει γραφεί μέσα στο πρόγραμμα λέγεται **εσωτερικό (internal)** ενώ αν έχει γραφεί εξωτερικά από το πρόγραμμα λέγεται **εξωτερικό (external)**.



Ποιοι είναι οι κύριοι λόγοι ανάπτυξης των υποπρογραμμάτων;

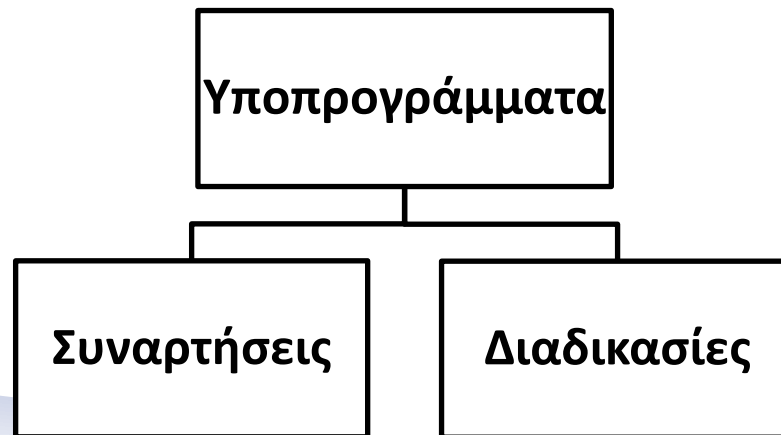
- Είναι ευκολότερο το γράψιμο και ο έλεγχος των προγραμμάτων αυτών, εφόσον γίνεται χώρια από το κυρίως πρόγραμμα.
- Η τεκμηρίωση (έλεγχος των λαθών) είναι ευκολότερη επειδή δουλεύουμε με μικρότερα προγράμματα.
- Τα ανεξάρτητα αυτά προγράμματα μπορούν να χρησιμοποιηθούν και από άλλα προγράμματα.
- Τα προγράμματα είναι πιο αναγνώσιμα λόγω αυτής της δομής.
- Πολλοί προγραμματιστές μπορούν να δουλέψουν σε ξεχωριστά προγράμματα ενός κυρίου προγράμματος, σχεδόν ανεξάρτητα ο ένας από τον άλλο.
- Ανεξάρτητα κομμάτια του προγράμματος γίνονται μικρότερα και συνεπώς απλούστερα.
- Ένα τέτοιο πρόγραμμα μπορεί να χρησιμοποιηθεί πολλές φορές από το ίδιο το πρόγραμμα.



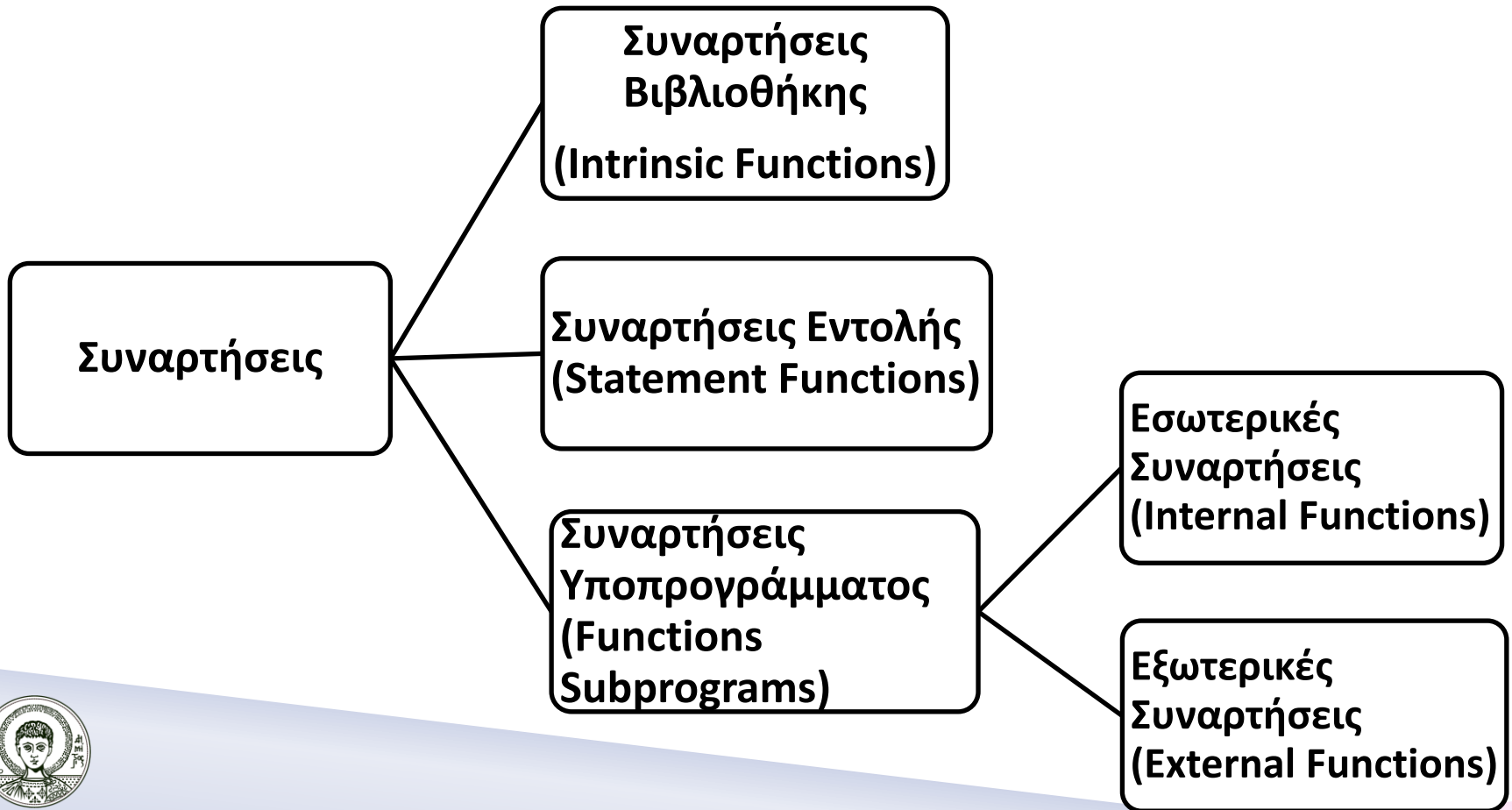
Διαχωρισμός των υποπρογραμμάτων σε δύο κατηγορίες

Συνάρτηση (*function*) : είναι ένα υποπρόγραμμα που στόχο του έχει να υπολογίζει και να επιστρέφει μόνο μια τιμή (αριθμητική, χαρακτήρα, λογική) με το όνομά της όπως οι γνωστές μαθηματικές συναρτήσεις, π.χ. $SIN(X)$, $COS(X)$ κ.λ.π..

Διαδικασία (*subroutine*) : είναι ένα υποπρόγραμμα το οποίο έχει όλα τα χαρακτηριστικά ενός προγράμματος.



Διαχωρισμός των συναρτήσεων σε κατηγορίες



Συναρτήσεις βιβλιοθήκης

Οι συναρτήσεις βιβλιοθήκης είναι οι γνωστές αριθμητικές και αλφαριθμητικές συναρτήσεις που αναφέραμε στο κεφάλαιο 2, π.χ. COS, SIN, TAN, EXP, LEN, TRIM, κλπ.



Συναρτήσεις εντολής

«όνομα συνάρτησης»(λίστα μεταβλητών) = έκφραση

PROGRAM TESTFUNCTION

```
REAL  ::X,Y,DIST
```

```
DIST(X,Y)=SQRT(X**2+Y**2)
```

```
READ*,X,Y
```

```
PRINT*,DIST(X,Y)
```

END PROGRAM TESTFUNCTION



Συναρτήσεις υποπρογράμματος – Σύνταξη

«Τύπ. Δεδομ.» FUNCTION «όνομα συνάρτ.»([Λίστα
Μεταβλητών]) [RESULT=όνομα]

IMPLICIT NONE

Τμήμα Δηλώσεων

Τμήμα Εκτέλεσης

Τμήμα Υποπρογράμματος

END FUNCTION «όνομα συνάρτησης»



Συναρτήσεις υποπρογράμματος – Παράδειγμα

```
REAL FUNCTION DIST(X,Y)
```

```
IMPLICIT NONE
```

```
REAL, INTENT(IN) :: X,Y
```

```
DIST= SQRT(X**2+Y**2)
```

```
END FUNCTION DIST
```

Η δήλωση **INTENT(IN)** σημαίνει ότι η συνάρτηση δέχεται δεδομένα εκτός συνάρτησης, π.χ. από το πρόγραμμα, τα οποία και δεν πρέπει να αλλάξουν τιμή μέσα στη συνάρτηση.



Συναρτήσεις υποπρογράμματος – Παρατηρήσεις

- Η λίστα μεταβλητών (*τυπικές παράμετροι (formal arguments)*) χρησιμοποιείται για την εισαγωγή δεδομένων στη συνάρτηση, ενώ το αποτέλεσμα της συνάρτησης τοποθετείται στο όνομα της συνάρτησης, το οποίο θα πρέπει να λάβει οπωσδήποτε τιμή πριν τον ορισμό του τέλους της συνάρτησης.
- Το όνομα της συνάρτησης ποτέ δεν πρέπει να βρεθεί δεξιά του ίσον.
- Η τοποθέτηση μιας εσωτερικής συνάρτησης γίνεται πριν το τέλος του προγράμματος και μετά την εντολή CONTAINS.
- Η χρήση των συναρτήσεων γίνεται με τον ίδιο τρόπο χρήσης των συναρτήσεων βιβλιοθήκης.
- Μεγάλη προσοχή θα πρέπει να δοθεί στην ταύτιση του πλήθους, της σειράς και του τύπου δεδομένων των ορισμάτων που δέχεται η συνάρτηση.



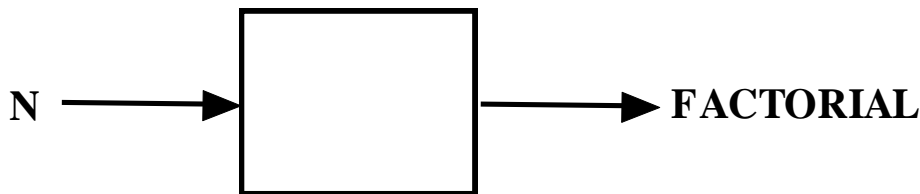
Παράδειγμα 1

(μέσω εσωτερικής συνάρτησης)

Να γραφεί συνάρτηση η οποία να υπολογίζει το $N!$.

PROGRAM FUNCTIONS1

```
IMPLICIT NONE
INTEGER :: N
PRINT*, 'N='
READ*, N
PRINT*, FACTORIAL(N)
```



CONTAINS

INTEGER FUNCTION FACTORIAL(N)

```
IMPLICIT NONE
INTEGER, INTENT(IN)
:: N
INTEGER :: I, F
FACTORIAL=1
F=1
IF (N>0) THEN
    DO I=1,N
        F=F*I
    END DO
    FACTORIAL=F
END IF
```

END FUNCTION FACTORIAL

END PROGRAM FUNCTIONS1



Παράδειγμα 1

(μέσω εξωτερικής συνάρτησης και block interface)

Να γραφεί συνάρτηση η οποία να υπολογίζει το $N!$.

```
Αρχείο 1 : FACTORIAL.F90  
INTEGER FUNCTION  
FACTORIAL(N)  
    IMPLICIT NONE  
    INTEGER, INTENT(IN) :: N  
    INTEGER      :: I,F  
    FACTORIAL=1  
    F=1  
    IF (N>0) THEN  
        DO I=1,N  
            F=F*I  
        END DO  
        FACTORIAL=F  
    END IF  
END FUNCTION FACTORIAL
```



Παράδειγμα 1 (...συνέχεια) (μέσω εξωτερικής συνάρτησης και block interface)

Να γραφεί συνάρτηση η
οποία να υπολογίζει το $N!$.

```
Αρχείο 2 : MAIN.F90  
PROGRAM FUNCTIONS1  
IMPLICIT NONE  
  
INTERFACE  
INTEGER FUNCTION FACTORIAL(N)  
    INTEGER, INTENT(IN) :: N  
END FUNCTION FACTORIAL  
END INTERFACE  
  
    INTEGER :: N  
    PRINT*, 'N='  
    READ*, N  
    PRINT*, FACTORIAL(N)  
END PROGRAM FUNCTIONS1
```



Παράδειγμα 1

(μέσω δήλωσης εξωτερικής συνάρτησης)

Να γραφεί συνάρτηση η οποία να υπολογίζει το N!

```
Αρχείο 1 : FACTORIAL.F90  
INTEGER FUNCTION FACTORIAL(N)  
  IMPLICIT NONE  
  INTEGER, INTENT(IN) :: N  
  INTEGER      :: I,F  
  FACTORIAL=1  
  F=1  
  IF (N>0) THEN  
    DO I=1,N  
      F=F*I  
    END DO  
    FACTORIAL=F  
  END IF  
END FUNCTION FACTORIAL
```



Παράδειγμα 1 (...συνέχεια) (μέσω δήλωσης εξωτερικής συνάρτησης)

Να γραφεί συνάρτηση η οποία να υπολογίζει το N!

```
Αρχείο 2 : MAIN.F90  
PROGRAM FUNCTIONS1  
IMPLICIT NONE  
  
INTEGER, EXTERNAL :: FACTORIAL  
(ή EXTERNAL FACTORIAL )  
  
INTEGER :: N  
PRINT*,'N='  
READ*,N  
PRINT*,FACTORIAL(N)  
END PROGRAM FUNCTIONS1
```



Παράδειγμα 2

Να γραφεί συνάρτηση η οποία να υπολογίζει την δύναμη M ενός πίνακα A , διαστάσεως N .

PROGRAM FUNCTIONS2

```
IMPLICIT NONE
REAL, DIMENSION(:, :), ALLOCATABLE :: A
INTEGER :: I, J, N, M
PRINT*, 'DIMENSION OF THE MATRIX='
READ*, N
ALLOCATE ( A(N, N) )
READ*, ((A(I, J), J=1, N), I=1, N)
PRINT*, 'POWER INDEX ='
READ*, M
PRINT*, MATRIXPOWER(A, N, M)
```



Παράδειγμα 2 (...συνέχεια)

CONTAINS

FUNCTION MATRIXPOWER(A,N,M)

IMPLICIT NONE

INTEGER, INTENT(IN) :: N,M

REAL, DIMENSION(N,N), INTENT(IN) :: A

REAL, DIMENSION(N,N) :: P,MATRIXPOWER

INTEGER :: I

DO I=1,N

P(I,I)=1

END DO

DO I=1,M

P=MATMUL(A,P)

END DO

MATRIXPOWER=P

END FUNCTION MATRIXPOWER

END PROGRAM FUNCTIONS2



Παράδειγμα 2 – ή διαφορετικά

CONTAINS

```
FUNCTION MATRIXPOWER(A,M)  
IMPLICIT NONE  
INTEGER, INTENT(IN) :: M  
REAL, DIMENSION(:,:), INTENT(IN) :: A  
REAL, DIMENSION(SIZE(A,1),SIZE(A,2)) :: P,MATRIXPOWER  
INTEGER :: I  
DO I=1,SIZE(A,1)  
    P(I,I)=1  
END DO  
DO I=1,M  
    P=MATMUL(A,P)  
END DO  
MATRIXPOWER=P  
END FUNCTION MATRIXPOWER
```

```
END PROGRAM FUNCTIONS2
```



Άσκηση 1

Να γραφεί συνάρτηση υποπρογράμματος η οποία θα υπολογίζει το

$$\binom{n}{m} = \frac{n!}{m! (n - m)!}, n > m$$



Προαιρετικές τυπικές παράμετροι

Τύπος δεδομένων, **OPTIONAL** [,άλλα χαρακτηριστικά] :: Λίστα μεταβλητών

FUNCTION MATRIXEXP(A,N,E)

.....

REAL, OPTIONAL, INTENT(IN) :: E

REAL :: E1

.....

IF(.NOT.PRESENT(E)) THEN

E1=1E-05

ELSE

E1=E

END IF

.....

DO WHILE (ANY(ABS(TERM)>E1))

.....

END FUNCTION MATRIXEXP



Αναδρομικές συναρτήσεις

Να υπολογιστεί το παραγοντικό ενός ακεραίου αριθμού N , με την βοήθεια αναδρομικής συνάρτησης, γνωρίζοντας ότι

$$N! = \begin{cases} 1 & N = 0 \\ N * (N - 1)! & N > 0 \end{cases}$$



Αναδρομικές συναρτήσεις (δήλωση)

RECURSIVE Τύπος Δεδομ. «όνομ. συν. 1»([μεταβλητές]) RESULT («όνομ. συν. 2»)



Πρόγραμμα (αναδρομική συνάρτηση)

```
PROGRAM RECFUNCTIONS1
```

```
IMPLICIT NONE
```

```
PRINT*,FACTORIAL(4)
```

```
CONTAINS
```

```
    RECURSIVE INTEGER FUNCTION FACTORIAL(N) RESULT(FACT)
```

```
        IMPLICIT NONE
```

```
        INTEGER, INTENT(IN) :: N
```

```
        IF (N==0) THEN
```

```
            FACT=1
```

```
        ELSE
```

```
            FACT=N*FACTORIAL(N-1)
```

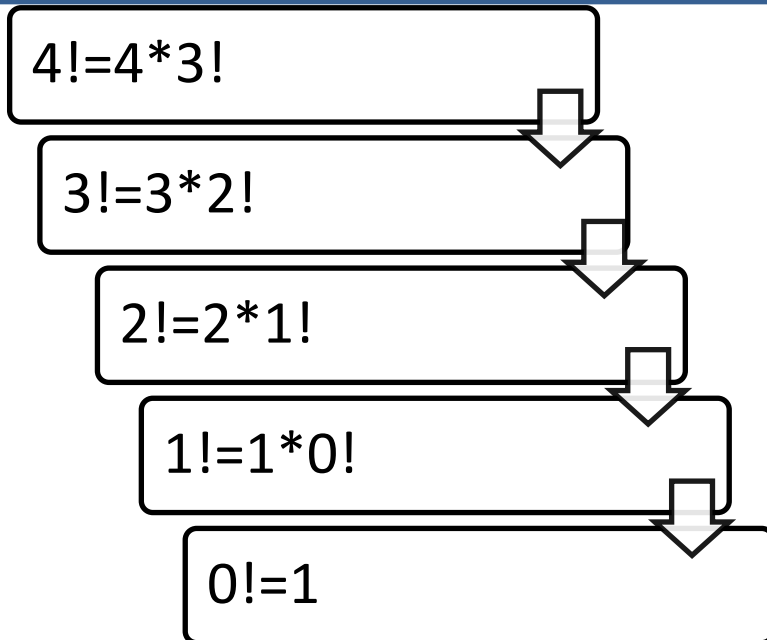
```
        END IF
```

```
    END FUNCTION FACTORIAL
```

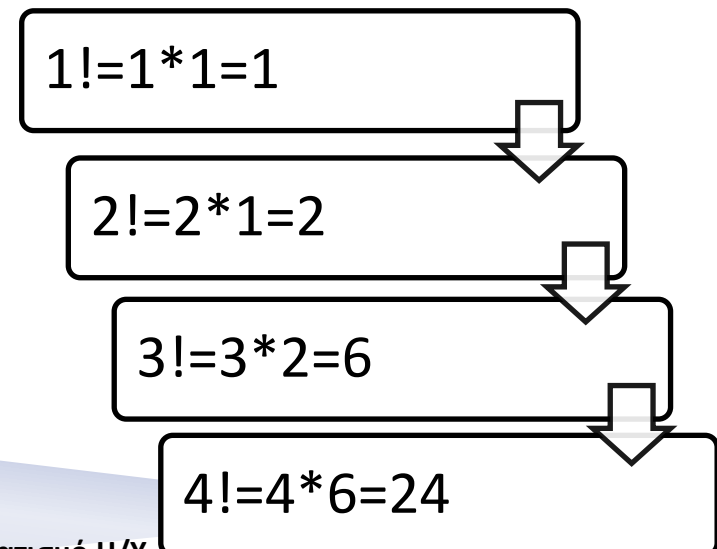
```
END PROGRAM RECFUNCTIONS1
```



Τρόπος ανάκλησης



Τώρα παίρνουμε την τιμή του $0!$
και προχωρούμε προς τα πίσω



Παράδειγμα 2 (Δύναμη πίνακα)

Να γραφεί αναδρομική συνάρτηση υποπρογράμματος που να υπολογίζει την Μ-οστή δύναμη ενός πίνακα Α εκτάσεως /N,N/, χρησιμοποιώντας τον αναδρομικό τύπο :

$$A^n = \begin{cases} I_n & n = 0 \\ A \times A^{n-1} & n \neq 0 \end{cases}$$



Πρόγραμμα (Δύναμη πίνακα)

```
PROGRAM RECFUNCTIONS3  
IMPLICIT NONE  
REAL, DIMENSION(:,,:), ALLOCATABLE :: A  
INTEGER :: I,J,N,M  
PRINT*,'DIMENSION OF THE MATRIX='  
READ*,N  
ALLOCATE ( A(N,N) )  
PRINT*,'ENTER MATRIX'  
READ*,((A(I,J),J=1,N),I=1,N)  
PRINT*,'POWER INDEX='  
READ*,M  
PRINT*,MATRIXPOWER(A,N,M)
```



Πρόγραμμα (Δύναμη πίνακα) (...συνέχεια – 1)

CONTAINS

```
RECURSIVE FUNCTION MATRIXPOWER(A,N,M)  RESULT(MATPOWER)  
  IMPLICIT NONE  
  INTEGER, INTENT(IN) :: N,M  
  INTEGER :: I  
  REAL, DIMENSION(N,N), INTENT(IN) :: A  
  REAL, DIMENSION(N,N) :: MATPOWER
```



Πρόγραμμα (Δύναμη πίνακα) (...συνέχεια – 2)

```
IF (M<0) THEN
    PRINT*, 'ERROR'
ELSE IF (M==0) THEN
    DO I=1,N
        MATPOWER(I,I)=1
    END DO
ELSE
    MATPOWER=MATMUL(MATRIXPOWER(A,N,M-1),A)
END IF
END FUNCTION MATRIXPOWER
END PROGRAM RECFUNCTIONS3
```



Παράδειγμα 3 (Μ.Κ.Δ.)

Να υπολογιστεί ο μέγιστος κοινός διαιρέτης δύο θετικών ακεραίων αριθμών α , β χρησιμοποιώντας τον αναδρομικό τύπο:

$$\text{ΜΚΔ}(\alpha, \beta) = \begin{cases} \text{ΜΚΔ}(\alpha - \beta, \beta) & \text{αν } \alpha > \beta \\ \text{ΜΚΔ}(\alpha, \beta - \alpha) & \text{αν } \alpha < \beta \\ \alpha & \text{αν } \alpha = \beta \end{cases}$$



Πρόγραμμα Μ.Κ.Δ. (1)

```
PROGRAM RECFUNCTIONS2
```

```
IMPLICIT NONE
```

```
INTEGER :: A,B
```

```
PRINT*,'A,B='
```

```
READ*,A,B
```

```
PRINT*,MKD(A,B)
```

```
CONTAINS
```

```
RECURSIVE INTEGER FUNCTION MKD(A,B) RESULT(MKDF)
```

```
IMPLICIT NONE
```

```
INTEGER, INTENT(IN) :: A,B
```



Πρόγραμμα Μ.Κ.Δ. (2)

```
IF (A==B) THEN
    MKDF=A
ELSE IF (A>B) THEN
    MKDF=MKD(A-B,B)
ELSE
    MKDF=MKD(A,B-A)
END IF
END FUNCTION MKD
END PROGRAM RECFUNCTIONS2
```



Παραδείγματα

Παράδειγμα 4

Να υπολογιστεί ο μέγιστος κοινός διαιρέτης δύο θετικών ακεραίων αριθμών a, β χρησιμοποιώντας τον αναδρομικό τύπο :

$$\text{ΜΚΔ}(a, b) = \begin{cases} b & \text{mod}(a, b) = 0 \\ \text{ΜΚΔ}(b, \text{mod}(a, b)) & \text{mod}(a, b) \neq 0, a \geq b \\ a & \text{mod}(a, b) = 0 \\ \text{ΜΚΔ}(\text{mod}(b, a), a) & \text{mod}(a, b) \neq 0, a < b \end{cases}$$

Παράδειγμα 5

Να γραφεί αναδρομική συνάρτηση που θα υπολογίζει το N -οστό όρο της ακολουθίας Fibonacci :

$$F_i = F_{i-1} + F_{i-2}, F_1 = F_2 = 1$$



Συναρτήσεις – Διαδικασίες της αριθμητικής IMSL βιβλιοθήκης της Compaq Visual Fortran

α) **MATH/LIBRARY** : διαθέτει συναρτήσεις χρήσιμες σε περιοχές των εφαρμοσμένων μαθηματικών, όπως γραμμικά συστήματα, ανάλυση ιδιοσυστημάτων, προσέγγιση και παρεμβολή, ολοκλήρωση και διαφόριση, διαφορικές εξισώσεις, μετασχηματισμοί, μη γραμμικές εξισώσεις, βελτιστοποίηση, πράξεις πινάκων, καθώς και άλλες εφαρμογές .

β) **STAT/LIBRARY** : διαθέτει στατιστικές συναρτήσεις που αφορούν βασικά στατιστικά στοιχεία, ανάλυση συσχέτισης, μη παραμετρική στατιστική, συναρτήσεις πιθανότητας, κλπ.



Παράδειγμα με συναρτήσεις της MATH/IMSL βιβλιοθήκης της Compaq Visual Fortran

PROGRAM LIBRARY

```
USE NUMERICAL_LIBRARIES
```

```
IMPLICIT NONE
```

```
REAL, DIMENSION(:,:), ALLOCATABLE :: A, AINV
```

```
INTEGER :: I,J
```

```
INTEGER :: N
```

```
PRINT*,'DIMENSION OF THE SQUARE MATRIX ='
```

```
READ*,N
```

```
ALLOCATE ( A(N,N), AINV(N,N) )
```

```
READ*, ((A(I,J),J=1,N),I=1,N)
```

```
CALL LINRG(N,A,N,AINV,N) ! Υπολογίζει τον αντίστροφο AINV του A
```

```
CALL WRRRN('AINV',N,N,AINV,N,0) ! Εκτυπώνει τον AINV
```

```
END PROGRAM LIBRARY
```



Σύνταξη της LINRG

LINRG (N, A, LDA, AINV, LDAINV)

Ορίσματα

N — Διάσταση του πίνακα *A*. (Είσοδος)

A — Πίνακας *N* επί *N*, του οποίου ζητάμε τον αντίστροφο.
(Είσοδος)

LDA — Η διάσταση του *A* όπως αυτή ορίστηκε στο πρόγραμμα που καλεί τη διαδικασία. (Είσοδος)

AINV — Ο αντίστροφος πίνακας του *A*. (Εξοδος)

Εάν ο *A* δε χρειάζεται μπορούμε να χρησιμοποιήσουμε στην θέση του *AINV* τον *A* και συνεπώς θα μοιραστούν την ίδια θέση μνήμης.

LDAINV — Η διάσταση του *AINV* όπως ακριβώς έχει οριστεί στο πρόγραμμα που καλεί τη διαδικασία. (Είσοδος)



Σύνταξη της WRRRN

WRRRN (TITLE, NRA, NCA, A, LDA, ITRING)

Ορίσματα

- TITLE** — Αλυσίδα χαρακτήρων η οποία προσδιορίζει τον τίτλο που θα εμφανισθεί πριν τον πίνακα που θα εκτυπωθεί. (Είσοδος)
- NRA** — Πλήθος γραμμών. (Είσοδος)
- NCA** — Πλήθος στηλών. (Είσοδος)
- A** — Ο NRA επί NCA πίνακας που πρόκειται να εκτυπωθεί. (Είσοδος)
- LDA** — Η διάσταση του πίνακα A όπως αυτή δηλώθηκε στο πρόγραμμα που καλεί την διαδικασία. (Είσοδος)
- ITRING** — Επιλογή εκτύπωσης. (Είσοδος)



Επιλογή εκτύπωσης ITRING

0	Ο πλήρης πίνακας εκτυπώνεται.
1	Ο πίνακας πάνω από την κύρια διαγώνιο του A εκτυπώνεται,συμπεριλαμβανομένης και της κυρίας διαγωνίου.
2	Ο πίνακας πάνω από την κύρια διαγώνιο του A εκτυπώνεται, μη συμπεριλαμβανομένης της κυρίας διαγωνίου.
-1	Ο πίνακας κάτω από την κύρια διαγώνιο του A εκτυπώνεται, συμπεριλαμβανομένης και της κυρίας διαγωνίου.
-2	Ο πίνακας πάνω από την κύρια διαγώνιο του A εκτυπώνεται, μη συμπεριλαμβανομένης της κυρίας διαγωνίου.



Διαδικασία μετάφρασης/εκτέλεσης προγραμμάτων με συναρτήσεις από την IMSL βιβλιοθήκη

Αφού γράψουμε το πρόγραμμά μας, στη συνέχεια κάνουμε τη μετάφραση του προγράμματος. Πριν τη δημιουργία όμως του εκτελέσιμου προγράμματος, κατά την διάρκεια της σύνδεσης, θα πρέπει να συνδέσουμε την βιβλιοθήκη SMATHS.LIB στο αντικείμενο πρόγραμμα μας, αν και αυτό συνήθως γίνεται αυτόματα. Θα πρέπει λοιπόν να πάμε στην επιλογή PROJECTS-SETTINGS-LINK και να δηλώσουμε επιπλέον στο Object/library modules την βιβλιοθήκη SMATHS.LIB, π.χ. KERNEL32.LIB ; SMATHS.LIB, η οποία περιέχει τις παραπάνω συναρτήσεις (αντίστοιχα την STATS.LIB για στατιστικές συναρτήσεις) που χρησιμοποιήσαμε. (Για περισσότερες πληροφορίες δες εγχειρίδιο χρήσης της Compaq Visual Fortran).



Άσκηση 2

Να γραφεί πρόγραμμα το οποίο θα υπολογίζει τις ιδιοτιμές και τα ιδιοανύσματα του πίνακα

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 3 & 4 & 1 & 2 \\ 4 & 1 & 2 & 3 \end{pmatrix}$$



Βιβλιογραφία

Ν. Καραμπετάκης, *Εισαγωγή στη Fortran 90/95/2003*, 2^η Έκδοση, Θεσσαλονίκη: Εκδόσεις Ζήτη, 2011.



Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Καραμπετάκης Νικόλαος. «Εισαγωγή στον Προγραμματισμό Η/Υ (Fortran 90/95/2003). Συναρτήσεις». Έκδοση: 1.0. Θεσσαλονίκη 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: <http://eclass.auth.gr/courses/OCRS145/>.



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-sa/4.0/>





Τέλος Ενότητας

Επεξεργασία: Αναστασία Γ. Γρηγοριάδου
Θεσσαλονίκη, Χειμερινό Εξάμηνο 2013-2014

