

FORTRAN

①

Η Fortran αρχικά σχεδιάστηκε για επιστήμονες και μηχανικούς χρήστες και ήταν η επικρατούσα αναδιάρθρωση γλώσσα.

Περιλαμβάνει έναν αριθμό από κοινά χαρακτηριστικά που δεν υπάρχουν σε προηγούμενες από αυτήν γλώσσες.

Πως προγραμματίζουμε σε FORTRAN;

Οι πρώτες 6 στήλες χρησιμοποιούνται για ειδικούς σκοπούς ενώ οι στήλες (73-80) λείπουν.

Αν η 1^η στήλη περιέχει ένα C ή * τότε όλη η σειρά που ακολουθεί είναι ένα σχόλιο.

Στην 1^η γραμμή του προγράμματος ορίζεται το όνομα του. Πρέπει να αρχίζει με γράμμα να περιέχει μόνο γράμματα και ψηφία και μόνο 6 χαρακτήρες το πολύ.

Η εντολή είναι η εξής:

```
PROGRAM FIRST  
C THIS IS THE FIRST PROGRAM
```

Η εντολή READ *, input-list όπως input-list για λίστα εισόδου μιας ή περισσότερων μεταβλητών χρησιμοποιούμε με κομμάτι παίρνει το ρόλο της scanf της C.

②

π.χ. READ*, A, B

Το πρόγραμμα μας έχει δύο μεταβλητές, τις A και B και ο υπολογιστής διαβάζει δύο είδη δεδομένων και θα τα αποθηκεύσει στις θέσεις A και B.

Αν οι μεταβλητές μας είναι οι I, J, K, L, M, N τότε είναι integer.

Αλλιώς οι κβες από A ως H και από O ως Z αποθηκεύονται ως Real.

Όποτε στο πρόγραμμα μας οι A και B είναι πραγματικές (Real) κβες.

Ευτολή ενχώρησης

SUM = A + B ενχωρεί μια τιμή σε μια μεταβλητή

Ευτολή εντύπωσης

PRINT *, output-list

Οι λίστες εξόδου είναι περισσότερο πολύπλοκες από τις λίστες εισόδου. Οι λίστες εξόδου δεν περιορίζονται σε αριθμητικά μεταβλητών, αλλά περιέχουν και σταθερές ή παραστάσεις και εκφράσεις.

3

Μπιλιούσης Σπύρος

3 τύποι σταθερών:

i) Πραγματική σταθερά (μπορεί να έχει και αρν. πρόσημο)
π.χ 5.3, -1204, -17, +3.14

ii) Ακέραια σταθερά (μπορεί να έχει και αρν. πρόσημο)
π.χ 5, +32, -11

iii) Σταθερή - χαρακτηρισμός είναι μια αυθαίρετη χαρακτηριστική (συμβολοσειρά) π.χ 'THE SUM IS'

Αν θέλουμε να συμπεριλάβουμε μέσα στη συμβολοσειρά μια απόδειξη, τότε βασώ δυο διαδοχικές αποστρώσεις

π.χ 'TAILOR'S THEOREM'

Πρόγραμμα 1:

```
PROGRAM FIRST
```

```
C THIS IS THE FIRST TEST PROGRAM
```

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

```
SUM = A + B
```

```
PRINT *, 'THE SUM OF', A, 'AND', B, 'IS', SUM
```

```
STOP
```

```
END
```

Η εντολή STOP δηλώνει ~~το~~ ~~σημείο~~ Η/Υ σταματάει το πρόγραμμα
Η εντολή END δηλώνει το φυσικό τέλος των οδηγιών το

4

Διαφορά STOP και END

STOP δηλώνει το τέλος της εκτέλεσης των εντολιών (από το τέλος) του προγράμματος, ενώ το END δηλώνει το φυσικό τέλος και είναι εντολή για να αρχίσει ο μεταφραστής την μεταγλώττιση.

Στην FORTRAN κατά την εκτέλεση ενός προγράμματος μπορεί να "χρησιμοποιήσει" 2 είδη τερματικών:

- Συντακτικά (ή γραμματικά) σφάλματα
- Λογικά σφάλματα


Αριθμητικοί τελεστές Fortran

Στην σύνταξη εκφράσεων στην FORTRAN χρησιμοποιούμε τους αριθμητικούς τελεστές +, -, *, /, **

- + για πρόσθεση
- για αφαίρεση
- * για πολλαπλασιασμό
- / για διαίρεση
- ** για ύψωση σε δύναμη

Προτεραιότητα

**
* και /
+ και -



και από αριστερά προς δεξιά στο ίδιο επίπεδο

5

Μπιλιούσης Σπύρος

Προσοχή: Ενωείται πρώτα οι πράξεις στις παρενθέσεις

π.χ έστω η έκφραση $A*(C+B)**D-E$

θα υπολογισθεί με την εξής σειρά:

- i) $B+C$
- ii) $(B+C)**D$
- iii) $ii-E$
- iv) $A* iii$

Προγραμμα 2: Μετατροπή από Κελσίου σε Fahrenheit με τον τύπο $F=9C/5+32$

```

PROGRAM CONV
C USE FORMULA  $F=9C/5+32 (=1.8C+32)$ 
  READ *, TEMPC
  TEMPF=(9.0 * TEMPC)/5.0 + 32.0
  PRINT *, 'C=', TEMPC, 'F=', TEMPF
  STOP
  END

```

Όταν χρησιμοποιών κβτες ακεραίους δίνω προσοχή στην διαίρεση

έστω 13 και 5

$13+5=18$

$13-5=8$

$13*5=65$

$13/5 = 9$

(7)

Μπιλιούσης Σπύρος

Τα υποπρογράμματα που είναι μέρος της FORTRAN λειτουργούν
ως συναρτήσεις και αναφέρονται ως εσωτερικές

Π.χ τέτοια είναι υποπρογράμματα είναι:

Εστω x αριθμ

$SIN(x)$ δίνει το ημίτιχο του x

$SQRT(x)$ δίνει την ρίζα του x

$LOG(x)$ δίνει τον λογάριθμο του x .

Υπάρχουν πάνω από 40 τέτοιες συναρτήσεις στην FORTRAN

Όταν χρειαστεί να προγραμματίσουμε εφεξής υποπρογράμματα
καλείται υπορουτίνα (Subroutine).

Εκτέλεση προγράμματος

Αποθηκεύω το πρόγραμμα ως askisi1.f

και το εκτελώ ως εξής:

PF7 -o askisi1.exe askisi1.f

8

Η FORTRAN υποστηρίζει τους παρακάτω τύπους δεδομένων:

INTEGER	→	Ακέραιοι
REAL	→	Πραγματικοί
DOUBLE PRECISION	→	Πραγματικοί Διπλής ακρίβειας
COMPLEX	→	Μικαδικοί
CHARACTER	→	Χαρακτήρες
LOGICAL	→	Λογικοί

~~Μικαδικός~~ Μια σταθερά ^{μεταβλητή} τύπου COMPLEX αναπαριστά
είναι μικαδικό αριθμό

Παράδειγμα COMPLEX I

$$I = (1.0, 2.0) \quad \text{τότε ο } I \text{ έχει την μορφή}$$

↑ ↑
πραγμ. φανταστ.
μέρος μέρος

$$I = 1.0 + 2.0 \cdot i$$

Όταν αφορά τις μεταβλητές τα ονόματά τους πρέπει να συμμορφώνονται με τους παρακάτω κανόνες

- 1) Αρχίζουν από αγγλικό γράμμα
- 2) Έχουν το πολύ 6 χαρακτήρες
- 3) Οι υπολοίποι χαρακτήρες είναι γράμματα ή αριθμοί.

(9)

Μπιλιούσης Σπύρος

Διάταξη μεταβλητών

Όπως αριθμούς και στην \mathbb{C}

REAL x, y, z

INTEGER i, j, k

COMPLEX L, N

Για τον τύπο CHARACTER υπάρχει μια διαφοροποίηση π.χ η εντολή

CHARACTER * 8 NAME, MYNAME δηλώνει τις

μετρες NAME, MYNAME

ως τύπου CHARACTER που

έχουν μήκος 8.

Όπως αναφέραμε ήδη, για μεταβλητή όπου λείπει η δήλωση της, υποτίθεται ότι είναι ανεξάρτητη από το αρχικό γράμμα του ονόματός της αρχίζει από τα γράμματα I-N ή αν δεν εμφανίζει αυτό δηλαδή είναι από A-H και O-Z είναι REAL.

Τέτοιες ανεξάρτητες μετρες έχουν είναι IMPLICIT τύπο, που φανερώνεται με το αρχικό γράμμα του ονόματός της μεταβλητής.

Γι' αυτό χρησιμοποιούμε την εντολή IMPLICIT όπως θέλουμε εμείς σε κάθε προγράμμα.

π.χ IMPLICIT REAL (P-X).

10

Πληροφορική Ιωαννίνων
Φροντ. Αρ Ανάλυσης

IMPLICIT DOUBLE PRECISION (A-H, O-Z)

Αν σε ένα πρόγραμμα δεν κάνουμε δηλώσεις τότε προαπορισμένο είναι το εφής:

IMPLICIT REAL (A-H, O-Z)

IMPLICIT INTEGER (I-N)

Εντολή PARAMETER

Η FORTRAN παρέχει την χρήση παραμέτρων. Οι παράμετροι είναι σταθερές με ονόμα

PARAMETER (PI=3.14)

ώστε βω την #define της C.

π.χ σε ένα πρόγραμμα

COMPLEX I

CHARACTER * 8 TOWN

PARAMETER (I = (0., 1.), TOWN = 'IOANNINA', PI = 3.14,
NOS = 20)



Λίστος Υποδείξεων

INTEGER PI

REAL NOS, TOWN

PARAMETER (I=(0, 1.), TOWN='IOANNINA', PI=3.14, NOS=20)

- Η μεταβλητή I δεν έχει δηλωθεί και άρα είναι τύπου integer ενώ είναι τύπου COMPLEX.
- Η PI έχει δηλωθεί INTEGER ενώ είναι REAL
- Η NOS το αμυνόδο
- Η TOWN έχει δηλωθεί REAL ενώ είναι τύπου CHARACTER

Υποσημειώσεις: Πρέπει μεταξύ COMPLEX και REAL
 ή μεταξύ COMPLEX και INTEGER
ΔΙΝΟΥΝ COMPLEX

Μπορείτε να δείτε σημειώσεις για FORTRAN
 στο link

<http://art.cs.uoi.gr/~lagaris/fortran/>

η πιο αλλα <http://www.cs.uoi.gr/~lagaris>
 και στο παλιό courses in Greek
 πατάτε Fortran FF

Μπορείτε να διαβάσετε το βιβλίο ^{Παράλληλες με} Fortran FF
 του ΗΑΙΑ ΑΥΓΙΤΑΚΗ

<http://www.cs.uoi.gr/~develleg>
 πολύ καλές σημειώσεις

No.
Date

ΕΝΤΟΛΕΣ ΕΙΣΟΔΟΥ - ΕΞΟΔΟΥ

Οι απλούστερες μορφές εντολών (I/O) που υποστηρίζει η FORTRAN

Εντολή εισόδου:

READ *, list
η READ (*,*) list

όπου list για ακολουθία
επιφράσεων, μεταβλητών
και χαρακτήρων

Εντολή εξόδου:

PRINT *, list
WRITE (*,*) list

Πρόγραμμα 3: από Fahrenheit σε Celsius

$$C = \frac{5}{9} [F - 32]$$

```
PROGRAM FARCEL
REAL FAR, CELCIU
WRITE (*,*) 'TEMPERATURE IN FAHRENHEIT?'
READ (*,*) FAR
CELCIU = (FAR - 32) * 5. / 9
WRITE (*,*) FAR, CELCIU 'FAHRENHEIT', 'CELCIU'
END WRITE (*,*) FAR, CELCIU
END
```

Λογικές Επιπράξεις και Λογικοί Τελεστές

Λογικοί Τελεστές

- .NOT
- .AND
- .OR
- .EQV ← NOR } Από συμπλοκή
- .NEQV ← XOR } Σχεδίαση

Εστω L, M logical μεταβλητές

L	M	L OR M	L EQV M	L NEQV M
.TRUE	TRUE	TRUE	TRUE	FALSE
.TRUE	FALSE	TRUE	FALSE	TRUE
FALSE	TRUE	TRUE	FALSE	TRUE
FALSE	FALSE	FALSE	TRUE	FALSE

Σχεδίαση Συμπλοκών (Σχεδίαση Τελεστών)

- .LT. → <
 - .LE. → <=
 - .GE. → >=
 - .EQ. → =
 - .NE. → ≠
- A. LT.C αληθής αν A < C
αλλιώς ψευδής

(14)

Πληροφορική Ιωαννίνων
Φροντ. Αρ. Αναλυτής

Μπορείτε να συζητήσετε και παραστάσεις
π.χ $B**2 - 4*A*C \text{ GE } 0$.

Πως και στην C έτσι και στην FORTRAN
τις λογικές εκφράσεις τις χρησιμοποιούμε στις δομές
έλεγχου.

Δομή έλεγχου IF

IF (logical expression) THEN

⋮

ELSE IF (logical expression) THEN

⋮

ELSE IF (logical expression) THEN

⋮
⋮
⋮

ELSE

⋮
⋮

END IF

π.χ

```

IF ( X LT. Y ) THEN
  WRITE (*,*) ' To x', x, ' einai megalutero apo to y', y,
END IF

```

η αληθεια και

```

IF ( X LT. Y ) OR ( W LT. Z ) THEN
  ?
END IF

```

Προτεραιότητα

- . NOT.
- . AND.
- . OR.
- . EQV. και NEQV.



Βαζω παρενθεσεις για να εχω το υπαρτι πιο νωρο

(16)

Προγραμμα 4

Είς σύνολο δεδομένων από μια καταμέτρηση περιέχει για κάθε άτομο τα εξής: την ηλικία του και βάρος του και μία κωδικά που δείχνει το φύλο του (0=άνδρ, 1=θήλυ).
 Να γραφεί ένα πρόγραμμα που να διαβάσει το πλήθος των ατόμων στην καταμέτρηση και μετά να υπολογίσει και να τυπώσει το Μ.Ο του βάρους των ανδρών που είναι ηλικία μεταξύ 21 και 35.

```
PROGRAM AVEWET
```

```
READ *, N
```

```
NUM = 0
```

```
SUMW = 0
```

```
DO 10, I = 1, N
```

```
READ *, MAGE, WT, ISEX
```

```
IF (ISEX .EQ. 0 .AND. MAGE .GE. 21 .AND. MAGE .LE. 35)
```

```
THEN
```

```
    SUMW = SUMW + WT
```

```
    NUM = NUM + 1
```

```
END IF
```

```
10 CONTINUE
```

```
IF (NUM .GT. 0) THEN
```

```
    PRINT *, 'AVERAGE WEIGHT IS', SUMW / NUM
```

```
ELSE
```

```
    PRINT *, 'THERE ARE NO MEN AGED BETWEEN 21  

    AND 35'
```

```
END IF
```

```
STOP
```

```
END
```


ΕΝΤΟΛΗ ΕΠΑΝΑΛΗΨΗΣ DO

Η επανάληψη ενός αριθμού εντολών, που πρέπει να εκτελεστούν πολλές φορές, γίνεται με την εντολή DO

DO N, I = I1, I2, I3

όπου N ο αριθμός της τελευταίας εντολής που ακολουθεί την ομάδα εντολών που επαναλαμβάνεται

I1 αρχική τιμή

I2 τελική τιμή

I3 βήμα

αν το βήμα είναι 1 το I3

παραλείπεται

DO N, I = I1, I2, I3

⋮
⋮
⋮

N CONTINUE

γίνονται πάντα

$INT((I2 - I1 + I3) / I3)$

επανάληψεις

Δηλαδή αν έχουμε DO N, 1, 30, 2

τότε έχουμε $(30 - 1 + 2) / 2 = \frac{31}{2} = 15$ επαν.

το οποίο σίγουρα μέρος

(18)

Δεν επιτρέπονται κάποιες εντολές, να χρησιμοποιηθούν στην τελευταία γραμμή ενός βρόχου DO.

Τέτοιες εντολές είναι: PROGRAM, STOP, END, IF, DO, RETURN.

Η τελική εντολή ενός βρόχου DO είναι η εντολή CONTINUE με μια ενδιάμεση.

Πρόγραμμα 5:

Να γραφεί ένα πρόγραμμα που πρώτα διαβάσει το πλήθος των εφεταγμάτων σε εφεταιίσεις, μετά διαβάσει τους βαθμούς τους και τσπώνει το μεγαλύτερο, το μικρότερο και δίνει ένα Μ.Ο τις βαθμολογίας.

```
PROGRAM EXAMS
```

Ν. Πλήθος μαθητών

```
C  INITIALIZATION
```

```
    SUM=0
```

```
    MAXM=0
```

```
    MINM=100
```

```
    READ *, N
```

```
    DO 5, I=1, N
```

```
        READ *, MARK
```

```
        SUM=SUM+MARK
```

```
        MAXM=MAX(MAXM, MARK)
```

```
        MINM=MIN(MINM, MARK)
```

```
    5 CONTINUE
```

```
AV = REAL (ISUM) / N
```

```
PRINT *, 'HIGHEST MARK IS', MAXM
```

```
PRINT *, 'LOWEST MARK IS', MINM
```

```
PRINT *, 'AVERAGE MARK IS', AV
```

```
STOP
```

```
END
```

ΠΡΟΣΟΧΗ...

Όταν προγραμματίζετε σε FORTRAN να το είναι στην εντολή DO να κλείσετε το εφής:

DO N, 0, 100, 5 τότε ο μεγαλύτερος αριθμός είναι 21 όπως ο Η/Υ μπορεί να δώσει 20,999 και κλείσει σε 20 λόγο ακεραιότητας

Τι αυτό λόγο DO N, 0, 101, 5 ώστε να γίνει ο αριθμός 21 επαναλήψεις.

ΘΥΛΑΚΩΣΗ ΒΡΟΧΩΝ

Πρόγραμμα 6

Να γραφεί ένα πρόγραμμα που τυπώνει ένα σωστό πίνακα πολλαπλασιασμού από 3 γραφές ως 9 γραφές.

(20)

Φροντιστήριο ΑΠΘ.
Ανάλυσης

PROGRAM TABLES

```
DO 5, I=3,9
```

```
  PRINT *, I, 'TIMES TABLE'
```

```
    DO 10, J=1,9
```

```
      10 PRINT *, I, 'TIMES', J, ' IS', I*J
```

```
      5 CONTINUE
```

```
    STOP
```

```
  END
```

3 TIMES TABLE

```
3 TIMES      1  IS  3
```

```
3 TIMES      2  IS  6
```

```
⋮
```

```
3 TIMES      9  IS  27
```

4 TIMES TABLE

```
4 TIMES      1  IS  4
```

Χειρίδιος Συμβολοσειρών

Ανάλυση χαρακτήρων

• CHARACTER * 8 name1, name 2

επει αυτες οι μεταβλητες εχουν 8 χαρακτήρες

• CHARACTER *8 name1 *6, name2, name3.

τότε αυτές οι ηρώσεις έχουν 8 χαρακτήρες εκτός από την name1 που έχει 6.

• CHARACTER name1 *5, name2 *8, name3 *6

τότε η name1 έχει 5 χαρακτήρες, η name2 8 και η name3 6.

Συνένωση συμβολοσειρών

// π.χ 'FIRST' // 'TEST' Δίνει 'FIRSTTEST'

Υποσυμβολοσειρά :

π.χ 'ALPHABETA' (6:9) Δίνει 'BETA'

από τον 6.οσ τον 9.ο χαρακτήρα

Πρόγραμμα F

```
PROGRAM CHARTEST
CHARACTER *5D*8,E,F
D='ALPHABET'
E=D(1:5)
F=D(6:9) // D(5:5)
STOP
END
```

E = 'ALPHA'

F Δίνει 'BETA'

(92)

Φροντιστήριο Αρ.
Αναλυτής

Στην FORTRAN κατά ομάδες σχετιζόμενων μετρώσεων που έχουν το ίδιο όνομα ονομάζεται πίνακας (array).
Οι μετρές ονομάζονται στοιχεία του πίνακα.

Ανάλυση Πίνακα

REAL X(10) έχει 10 στοιχεία τύπου REAL

CHARACTER *10 VAR(50) έχει 50 στοιχεία που είναι strings

INTEGER C(1000)

REAL D(0:30) έχει μετρές 31.

Για να ανάλυσο 2άρο πίνακα

REAL MARK(1:4, 1:20) έχει 4 γραμμές
20 στήλες

η REAL MARK(4, 20) πιο απλά

Η FORTRAN επιτρέπει μέχρι 7 διαστάσεις

Ο MARK έχει $4 \times 20 = 80$ στοιχεία

Αρχικοποίηση στοιχείων

```
PROGRAM TEST
```

```
REAL X(10)
```

```
DATA X(1), X(3), X(5), X(7) / 4 * 1.0 /, X(9) / 4.0 /
```

η εντολές μεταβίβασης

η με DO

PROGRAM TEST

PROGRAM TEST

REAL X(10)

REAL X(10)

X(1) = 1.0

DO 10, I = 1, 10

X(2) = 0.0

X(I) = -1.0

X(3) = 1.0

10 CONTINUE

Συναρτήσεις και Υπορουτίνες

Μια υπορουτίνα χρησιμοποιείται μέσω της εντολής CALL

CALL ονόμα υπορουτίνας (α1, α2, ...)

παραμέτρους

όπως και στην C

Ελέγχος φέρνει από το πρόγραμμα και ενεργοποιούνται οι εντολές της υπορουτίνας. Όταν εντελέστων αυτές, ο έλεγχος μεταβιβάζεται στο 'υπαίτιο πρόγραμμα'.

Οι συναρτήσεις λειτουργούν διαφορετικά

VAR = FUNCT(A1, A2, ...)

PRINT *, FUNCT(A1, A2, ...)

Αν μια υπορουτίνα δεν έχει παραμέτρους:

```
CALL SUBR
```

Αν μια συνάρτηση << <<:

```
VAR = FUNCT1( )
```

- Το "κύριο πρόγραμμα" ξεκινάει με την εντολή PROGRAM
- Μια υπορουτίνα με την εντολή SUBROUTINE
- Μια συνάρτηση με την εντολή FUNCTION!

Υπορουτίνα

```
SUBROUTINE NAME(A1, A2, A3, ...)
```

↓
αντιπροσωπεύουν τις παραμέτρους της CALL

⋮
⋮
⋮ εντολές που κινούνται όπως θα θάθε
⋮
⋮

```
RETURN  
END
```

← επιστρέφει τον έλεγχο

Οι A1, A2, A3... είναι τυπικές λέξεις μόνο μέσα στην υπορουτίνα

ΣΤΗΝ FORTRAN ΔΕΝ ΕΠΙΤΡΕΦΕΤΑΙ ΟΜΟΔΕΛΙΑΣ ΛΕΓΜΕΝΗ

Η ΕΥΚΟΛΗ CALL ΜΕ ΤΗΝ ΟΠΟΙΑ ΚΑΛΕΙΤΑΙ Η ΥΠΟΡΟΥΤΙΝΑ ΠΕΡΙΕΧΕΙ ΕΝΑΝ ΠΑΡΕΛΕΘΕΝ ΕΥΕΡΓΕΤΗ ΠΑΡΑΜΕΤΡΟ ΕΝΩ Η ΕΥΚΟΛΗ SUBROUTINE ΠΕΡΙΕΧΕΙ ΠΛΑΣΤΑΤΙΜΕΣ, ΤΟΤΙΜΕΣ ΓΙΑ ΤΗΝ ΥΠΟΡΟΥΤΙΝΑ.

ΕΥΕΡΓΕΤΗ ΠΑΡΑΜΕΤΡΟΣ:

- i) ΟΥΔΙΑ ΜΕΤΑΒΛΗΤΩΝ, ΣΤΑΘΕΡΩΝ, ΠΙΝΑΚΩΝ
- ii) ΟΥΔΙΑ ΕΣΩΤΕΡΙΩΝ ΣΥΝΑΡΤΗΣΕΩΝ, ΟΥΔΙΑ ΥΠΟΡΟΥΤΙΝΩΝ Η ΕΦΩΤΕΡΙΩΝ ΣΥΝΑΡΤΗΣΕΩΝ

ΠΛΑΣΤΑΤΙΜΗ ΠΑΡΑΜΕΤΡΟΣ:

- (I) ΟΥΔΙΑ ΜΕΛΩΝ (ΓΙΑ ΤΟΥΣ ΤΥΠΟΥΣ (i)) Η ΠΙΝΑΚΩΝ
- (II) ΟΥΔΙΑ ΔΙΑΔΙΑΚΟΝΙΩΝ (ΓΙΑ ΤΟΝ ΤΥΠΟ (ii)).

Π.Χ

```
PROGRAM TEST
  REAL A,B  INTEGER N
  REAL L(10,N)
```

```
CALL SUB2(A,B,L)
```

```
STOP
END
```

```

SUBROUTINE SUB2(Z,X,Y)
REAL Z,X,Y
REAL Z,X,Y

```

```
RETURN
```

Συνάρτηση

Σε αντίθεση με μια υπορουτίνα μια εντολή FUNCTION είναι της μορφής

REAL FUNCTION TEST (A1, A2)
τύπος όνομα παράμετροι

καλείται μέσα σε ένα πρόγραμμα ως εξής:

Επιστροφή σε μεταβλητή VAR = TEST (A1, A2)

σελ 173

Μπιλιούσης Σπυρος
 Παισιόχος Πληροφορικής Ιωαννίνων
 (Σημειώσεις για FORTRAN)
 Φροντιστήριο Αριθμητικής Ανάλυσης