ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΟΜΕΑΣ ΜΑΘΗΜΑΤΙΚΩΝ

ΕΙΣΑΓΩΓΗ ΣΤΗ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ FORTRAN 77

ΒΑΣΙΛΗΣ ΚΟΚΚΙΝΗΣ

ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ ΕΜΠ

ΓΙΑΝΝΗΣ ΚΟΛΕΤΣΟΣ

ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ ΕΜΠ

ΑΘΗΝΑ, 2002

ΕΙΣΑΓΩΓΗ

Η γλώσσα προγραμματισμού FORTRAN (συντομογραφία του FORmula TRANslation) είναι μια αλγοριθμική γλώσσα προγραμματισμού υψηλού επιπέδου, προσανατολισμένη στον προγραμματισμό επιστημονικών εφαρμογών από τους κλάδους των θετικών επιστημών, χωρίς ιδιαίτερες επιδόσεις στις εμπορικές εφαρμογές.

Παρουσιάστηκε στη δεκαετία 1950-1960 και από τότε έχει δεχθεί μεγάλο πλήθος αναθεωρήσεων και βελτιώσεων. Εδώ παρουσιάζουμε μια εισαγωγή στην έκδοση εκείνη της FORTRAN που έχει επικρατήσει να λέγεται FORTRAN 77 και που επιτρέπει, σε αντίθεση με τις προηγούμενες εκδόσεις της, το δομημένο προγραμματισμό, το χειρισμό χαρακτήρων και τη διαχείριση αρχείων.

Το 1991 παρουσιάστηκε η νέα έκδοση της standard FORTRAN, η FORTRAN 90 που αποτελεί υπερσύνολο της FORTRAN 77 και αναμένεται να επικρατήσει τα επόμενα χρόνια.

1. ΤΟ ΠΡΩΤΟ ΜΑΣ ΠΡΟΓΡΑΜΜΑ

```
      PROGRAM TEST
C     THIS IS OUR FIRST PROGRAM
      READ *, A, B
      SUM = A + B
      PRINT *, 'A=', A, 'B=', B
      PRINT *, 'SUM=', SUM
      STOP
      END
```

Κάθε γραμμή περιέχει 80 θέσεις. Οι εντολές FORTRAN γράφονται από τη θέση 7 (και μετά) μέχρι και τη θέση 72. Σε περίπτωση που δεν χωρέσει μία εντολή σε μια γραμμή συνεχίζουμε στην επόμενη γραμμή βάζοντας στην 6η θέση οποιονδήποτε χαρακτήρα εκτός από το μηδέν και το κενό. Οι θέσεις 1 - 5 χρησιμοποιούνται για την αρίθμηση (προαιρετική) των εντολών.

Αν στη θέση 1 υπάρχει ο χαρακτήρας C ή * τότε η γραμμή αυτή θεωρείται σχόλιο. Κάθε πρόγραμμα αρχίζει με την εντολή PROGRAM και το όνομα του προγράμματος που μπορεί να έχει το πολύ 6 χαρακτήρες. (Η εντολή PROGRAM μπορεί να παραλειφθεί).

Στην εντολή

```
      SUM = A + B
```

το σύμβολο =, δεν έχει την έννοια της μαθηματικής ισότητας αλλά την έννοια της εκχώρησης του αποτελέσματος της παράστασης του δεξιού μέλους στο αριστερό μέλος. Έτσι στη FORTRAN έχουν νόημα εντολές όπως

```
      X = X + 1
```

ενώ απαγορεύονται εντολές όπως

```
3.14 = A
-A = 2*B
```

Σε κάθε πρόγραμμα υπάρχει τουλάχιστον ένα STOP που δηλώνει το λογικό τέλος του προγράμματος και ακριβώς ένα END που είναι η τελευταία εντολή και δηλώνει το φυσικό τέλος του προγράμματος.

2. ΣΤΑΘΕΡΕΣ

Τα ακριβή όρια μέσα στα οποία είναι δυνατόν να κινούνται οι αριθμητικές σταθερές που θα ορίσουμε πιο κάτω εξαρτώνται από τον τύπο του υπολογιστή. Ενδεικτικά θα αναφέρουμε κάποια συνήθη διαστήματα.

2.1. ΑΚΕΡΑΙΕΣ ΣΤΑΘΕΡΕΣ (INTEGER CONSTANTS)

Μία ακέραια σταθερά είναι ένας ακέραιος αριθμός χωρίς δεκαδική υποδιαστολή με σχετικά περιορισμένα όρια.

Επιτρεπτές	**Μη επιτρεπτές**
0	12,35
-1	127.8
157	123456789012

2.2. ΠΡΑΓΜΑΤΙΚΕΣ ΣΤΑΘΕΡΕΣ (ΑΠΛΗΣ ΑΚΡΙΒΕΙΑΣ) (REAL CONSTANTS, REAL * 4)

Είναι κάθε αριθμός πραγματικός της μορφής sm.n ή smpnEx, όπου s το πρόσημο του αριθμού, m, n ακολουθίες δεκαδικών ψηφιών, p δεκαδική υποδιαστολή που μπορεί να παραληφθεί αν δεν υπάρχει n και x (προσημασμένος ή όχι) μονοψήφιος ή διψήφιος ακέραιος εκθέτης.

Επιτρεπτές	Μη επιτρεπτές
1.413	987
-10.317	3,14
8.1 E-7	3.14.5
4 E 10	9.E
5.	3.14 E 85
.5	1.E 1.3
0.0	E + 9
-0.001	4 E 10 E 3
-153.17 E + 37	1.3 E 017

2.3. ΠΡΑΓΜΑΤΙΚΕΣ ΣΤΑΘΕΡΕΣ ΔΙΠΛΗΣ ΑΚΡΙΒΕΙΑΣ (DOUBLE PRECISION CONSTANTS, REAL * 8)

Είναι πραγματικοί αριθμοί, που ορίζονται ανάλογα με τις πραγματικές σταθερές απλής ακρίβειας με την αντικατάσταση του E με D και για τους οποίους ο υπολογιστής συγκρατεί περισσότερα ψηφία και εκτελεί συνεπώς τις πράξεις με μεγαλύτερη ακρίβεια. Είναι δηλαδή της μορφής smpnDx, όπου το Dx είναι εδώ υποχρεωτικό.

Απλή ακρίβεια	Διπλή ακρίβεια
3.	3. D0
4.12 E + 05	4.12 D + 05
-	0.1234567890123 D0

Αν παραληφθεί το Dx σε μια πραγματική σταθερά διπλής ακρίβειας, χάνεται αυτόματα η διπλή ακρίβεια.

2.4. ΠΡΑΓΜΑΤΙΚΕΣ ΣΤΑΘΕΡΕΣ ΤΕΤΡΑΠΛΗΣ ΑΚΡΙΒΕΙΑΣ (QUADRATIC PRECISION CONSTANTS, REAL*16)

Ότι και οι πραγματικές σταθερές διπλής ακρίβειας με τη διαφορά ότι ο δεκαδικός εκθέτης είναι της μορφής Qx.

2.5. ΜΙΓΑΔΙΚΕΣ ΣΤΑΘΕΡΕΣ (COMPLEX CONSTANTS - DOUBLE COMPLEX CONSTANTS)

Είναι διατεταγμένο ζεύγος ακεραίων ή πραγματικών αριθμών της μορφής (α, β), όπου α,β είναι το πραγματικό και το φανταστικό μέρος του μιγαδικού αριθμού αντίστοιχα. Αν α,β ακέραιοι μετατρέπονται αυτόματα σε πραγματικούς απλής ακρίβειας. Αν α,β πραγματικοί απλής ακρίβειας τότε ο μιγαδικός (α,β) είναι μιγαδικός απλής ακρίβειας ενώ αν ένας τουλάχιστον από τους α,β είναι πραγματικός διπλής ακρίβειας τότε και ο (α,β) είναι μιγαδικός διπλής ακρίβειας.

ΤΥΠΟΣ ΣΤΑΘΕΡΑΣ	ΣΥΜΒΟΛΙΣΜΟΣ	ΔΙΑΣΤΗΜΑΤΑ ΚΥΜΑΝΣΗΣ	ΣΗΜΑΝΤΙΚΑ ΨΗΦΙΑ
Ακέραια	I	$0 \le \lvert x \rvert \le 2^{31}-1$	10
Πραγματική	R	$10^{-78} \le \lvert x \rvert \le 10^{75}$	6
Πραγματική διπλής ακρίβειας	DP	$10^{-78} \le \lvert x \rvert \le 10^{75}$	15
Πραγματική τετραπλής ακρίβ.	QP	$10^{-78} \le \lvert x \rvert \le 10^{75}$	32

2.6. ΛΟΓΙΚΕΣ ΣΤΑΘΕΡΕΣ (LOGICAL CONSTANTS)

Κάθε λογική σταθερά έχει μια λογική τιμή αληθή ή ψευδή και έτσι έχουμε μόνο τις εξής δύο λογικές σταθερές:

.TRUE.

.FALSE.

2.7. ΣΤΑΘΕΡΕΣ ΤΥΠΟΥ CHARACTER (CHARACTER CONSTANTS)

Είναι σταθερές που αποτελούνται από σειρά χαρακτήρων της μορφής

΄C_1C_2 ... C_m΄ ή ΄΄C_1C_2 ... C_m΄΄.

Οι απόστροφοι δεν θεωρούνται ότι περιέχονται στη σταθερά και αν θέλουμε να περιλάβουμε απόστροφο βάζουμε δύο συνεχόμενες αποστρόφους.

Σταθερά	**Παράσταση**
SPEED	΄SPEED΄
TODAY΄S DEPOSITS	΄TODAY΄΄S DEPOSITS΄

Η σταθερά CHARACTER πρέπει να περιέχει τουλάχιστον ένα χαρακτήρα.

3. ΜΕΤΑΒΛΗΤΕΣ (VARIABLES)

Μια μεταβλητή παριστάνεται με κάποιο συμβολικό όνομα από το πολύ 6 αλφαριθμητικούς χαρακτήρες (ο πρώτος πρέπει υποχρεωτικά να είναι αλφαβητικός) που αντιστοιχεί σε μια θέση μνήμης και η τιμή της μπορεί να αλλάζει. Οι μεταβλητές ταξινομούνται σε αντίστοιχους τύπους με τις σταθερές.

Με τις εντολές καθορισμού (specification statements) - μη εκτελέσιμες εντολές που απευθύνονται στον compiler και προηγούνται σε ένα πρόγραμμα FORTRAN κάθε εκτελέσιμης εντολής - αποδίδουμε τον τύπο των μεταβλητών. Π.χ.

```
INTEGER I, ALFA
REAL M, SUM
DOUBLE PRECISION SIZE
QUAD PRECISION EPS
COMPLEX ROOT
DOUBLE COMPLEX A
LOGICAL FLAG
CHARACTER * 10 VERA * 6, CALNAT
```

Αν για μια μεταβλητή δεν καθοριστεί με άμεσο τρόπο ο τύπος της τότε έμμεσα θεωρείται η μεταβλητή REAL*4 εκτός αν το όνομά της αρχίζει με ένα από τα γράμματα I, J, K, L, M, N οπότε θεωρείται INTEGER.

4. ΕΣΩΤΕΡΙΚΕΣ Ή ΕΝΥΠΑΡΧΟΥΣΕΣ ΣΥΝΑΡΤΗΣΕΙΣ

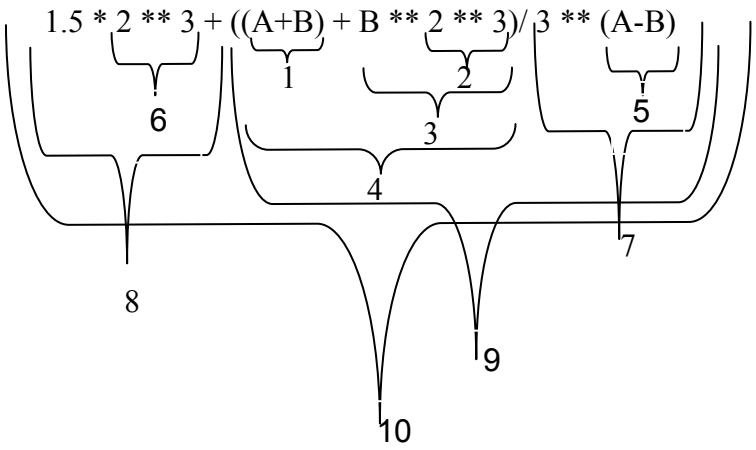
Στη γλώσσα FORTRAN περιέχονται και κάποιες εσωτερικές συναρτήσεις που συχνά χρησιμοποιούνται σε μαθηματικούς υπολογισμούς όπως SIN(X) για το ημίτονο, ABS(X) για την απόλυτη τιμή, SQRT(X) για την τετραγωνική ρίζα κ.λ.π. Αναλυτικός κατάλογός τους παρατίθεται σε παράρτημα στο τέλος.

5. ΑΡΙΘΜΗΤΙΚΕΣ ΕΚΦΡΑΣΕΙΣ

Στη FORTRAN οι αριθμητικοί τελεστές με σειρά προτεραιότητας είναι:

Σύμβολο	**Λειτουργία**
* *	Ύψωση σε δύναμη
* και /	πολλαπλασιασμός και διαίρεση
+ και -	πρόσθεση και αφαίρεση

Σε περίπτωση πράξεων με ίδια τάξη προτεραιότητας αυτές εκτελούνται από αριστερά προς τα δεξιά εκτός από την ύψωση σε δύναμη που εκτελείται από δεξιά προς τα αριστερά. Αν μέρος μιας αριθμητικής παράστασης είναι κλεισμένο σε παρενθέσεις τότε αυτό υπολογίζεται πρώτο. Π.χ.



Μερικές φορές η παρένθεση είναι υποχρεωτική π.χ.

(- B + (B ** 2 - 4 * A * C) ** (1/2.)) / (2 * A)

Σε περίπτωση μικτών πράξεων μεταξύ σταθερών ή μεταβλητών διαφορετικού τύπου ακολουθείται συνήθως ο κάτωθι πίνακας:

x/y	I	R	DP	C	DC
I	I	R	DP	C	DC
R	R	R	DP	C	DC
DP	DP	DP	DP	DC	DC
C	C	C	DC	C	DC
DC	DC	DC	DC	DC	DC

Πάντως μπορεί κανείς να ελέγχει σε κάθε συγκεκριμένο υπολογιστή όπου εργάζεται πως εκτελούνται οι μικτές πράξεις.

Παρατηρήσεις:

i) Το αποτέλεσμα της διαίρεσης δύο ακεραίων αριθμών είναι το ακέραιο μέρος του πηλίκου τους.

ii) Η ύψωση 0^o δεν έχει νόημα. Π.χ. η ύψωση
(3/4) ** (1/2) δεν είναι δεκτή.

iii) Η ύψωση σε εκθέτη τύπου REAL γίνεται βάσει του τύπου $x^y=e^{ylnx}$. Έτσι αρνητική παράσταση δεν μπορεί να υψωθεί σε εκθέτη τύπου REAL.

6. ΛΟΓΙΚΕΣ ΕΚΦΡΑΣΕΙΣ

Στη FORTRAN με τη βοήθεια των τελεστών σύγκρισης .LT.,.LE.,.GT.,.GE., .EQ.,.NE. δημιουργούμε λογικές εκφράσεις της μορφής

A.LT.B	αληθής	αν	A<B
A.LE.B	αληθής	αν	A≤B
A.GT.B	αληθής	αν	A>B
A.GE.B	αληθής	αν	A≥B
A.EQ.B	αληθής	αν	A=B
A.NE.B	αληθής	αν	A≠B

Σύνθετες λογικές εκφράσεις σχηματίζονται με τους λογικούς τελεστές (logical operators) .OR.,.AND.,.EQV.,.NEQV.,.NOT. . Ο πίνακας αληθείας των λογικών τελεστών είναι

A	B	.NOT.A	A.AND.B	A.OR.B	A.EQV.B	A.NEQV.B
T	T	F	T	T	T	F
T	F	F	F	T	F	T
F	T	T	F	T	F	T
F	F	T	F	F	T	F

Οι μόνες επιτρεπτές ακολουθίες δύο λογικών τελεστών είναι

.AND..NOT.
.OR..NOT.
.EQV..NOT.
.NEQV..NOT.

Συνοψίζουμε τη σειρά με την οποία υπολογίζεται μια λογική παράσταση

1. Υπολογισμός συναρτήσεων
2. Παρενθέσεις
3. Ύψωση σε δύναμη
4. Πολλαπλασιασμοί, διαιρέσεις
5. Προσθέσεις, αφαιρέσεις
6. Τελεστές σύγκρισης
7. .NOT.
8. .AND.
9. .OR.
10..EQV.,.NEQV.

Οι τελεστές ίδιας προτεραιότητας υπολογίζονται από αριστερά προς τα δεξιά (εξαιρείται η ύψωση σε δύναμη).

7. ΕΝΤΟΛΕΣ ΕΛΕΓΧΟΥ

7.1. Η ΕΝΤΟΛΗ GOTO

7.1.1. ΑΔΕΣΜΕΥΤΟ (UNCONTITIONAL) GOTO

Σύνταξη: **`GOTO n`**

ή **`GO TO n`**

όπου n ετικέτα (label) εκτελέσιμης εντολής (n είναι ακέραιος μεταξύ του 1 και του 99999).

Λειτουργία: Μεταφέρει τον έλεγχο στην εντολή με ετικέτα n.

7.1.2. ΕΚΤΕΛΕΣΙΜΟ (COMPUTED) GOTO

Σύνταξη: **GOTO (n_1, n_2, ..., n_k) I**

όπου n_1, n_2, ..., n_k ετικέτες εκτελέσιμων εντολών όχι υποχρεωτικά διαφορετικές μεταξύ τους και I ακέραια έκφραση.

Λειτουργία: Υπολογίζει το I (το μετατρέπει σε ακέραιο αν χρειάζεται) και αν το I έχει την τιμή m μεταφέρει τον έλεγχο στην εντολή με ετικέτα n_m αν $1 \le m \le k$ αλλιώς στην επόμενη από το GOTO εντολή.

7.1.3. ΑΠΟΔΙΔΟΜΕΝΟ (ASSIGNED) GOTO

Σύνταξη: **GOTO I, (n_1, n_2, ..., n_k)**

όπου I ακέραια μεταβλητή, n_1, n_2, ..., n_k ετικέτες εκτελέσιμων εντολών.

Λειτουργία: Πρέπει να έχει προηγηθεί η εντολή

```
ASSIGN n TO I
```

που αποδίδει την τιμή n (όπου n είναι ένας από τους n_1, n_2, ..., n_k) στη μεταβλητή I, ώστε το αποδιδόμενο GOTO να μεταφέρει στη συνέχεια τον έλεγχο στην εντολή με ετικέτα n. (Το I δεν μπορεί να χρησιμοποιηθεί σε αριθμητικές πράξεις γιατί περιέχει την τιμή ετικέτας n και όχι τον ακέραιο n).

7.2. Η ΕΝΤΟΛΗ IF

7.2.1. ΑΡΙΘΜΗΤΙΚΟ IF

Σύνταξη: **IF(e) n_1, n_2, n_3**

όπου e αριθμητική έκφραση (όχι μιγαδικής τιμής) και n_1, n_2, n_3 ετικέτες εκτελέσιμων εντολών.

Λειτουργία: Μεταφέρει τον έλεγχο στην εντολή με ετικέτα n_1 αν $e<0$ ή στην n_2 αν $e=0$ ή στην n_3 αν $e>0$.

7.2.2. ΛΟΓΙΚΟ IF

Σύνταξη: **IF(e) st**

όπου e λογική έκφραση και st εντολή FORTRAN εκτός των DO, DO WHILE, Block IF, END, ENDO ή άλλου λογικού IF.

Λειτουργία: Υπολογίζεται η τιμή της e και αν είναι .TRUE. εκτελείται η st, αν είναι .FALSE. ο έλεγχος μεταφέρεται στην επόμενη εντολή FORTRAN.

7.2.3. ΟΜΑΔΙΚΟ (BLOCK) IF

Σύνταξη:

```
IF(e1) THEN
Block of FORTRAN statements
ELSE IF (e2) THEN
Block of FORTRAN statements
          :
          :
ELSE
Block of FORTRAN statements
ENDIF
```

όπου e_1, e_2, ... λογικές εκφράσεις.

Λειτουργία: Σε κάθε στάδιο ελέγχεται η τιμή αλήθειας της λογικής έκφρασης e_k και αν είναι .TRUE. εκτελείται η ομάδα των εντολών FORTRAN που ακολουθεί και ο έλεγχος μεταφέρεται στην πρώτη εντολή μετά το ENDIF, αλλιώς αν είναι .FALSE. ελέγχεται η τιμή της e_{k+1} και συνεχίζει κατά τα γνωστά. Οι εντολές ELSE IF είναι προαιρετικές καθώς και η εντολή ELSE που υπάρχει στο τέλος και που για να εκτελεστεί πρέπει όλες οι λογικές εκφράσεις e_1, e_2, ... να έχουν τιμή .FALSE. .

7.3. ΕΝΤΟΛΕΣ ΕΠΑΝΑΛΗΨΗΣ

7.3.1. Η ΕΝΤΟΛΗ DO

Σύνταξη: **`DO N, I=`e_1`, `e_2`, `e_3**

όπου N ετικέτα εντολής FORTRAN, I ακέραια ή πραγματική μεταβλητή ελέγχου και e_1, e_2, e_3 μη μιγαδικές αριθμητικές εκφράσεις. Η e_1, ορίζει την αρχική τιμή του I, n e_2 την ανωτάτη επιτρεπτή τιμή του I και n e_3 το βήμα μεταβολής του I. Το e_3 είναι δυνατόν να παραληφθεί αλλά τότε θεωρείται ίσο με τη μονάδα. Το N είναι ετικέτα της ειδικής ουδέτερης εντολής CONTINUE ή μιας εκτελέσιμης εντολής FORTRAN (απαγορεύονται οι εντολές PROGRAM, STOP, END, GOTO, οι εντολές IF, άλλο DO, RETURN και FORMAT).

Λειτουργία: Προκαλεί την επαναλαμβανόμενη εκτέλεση των εντολών που περιέχονται μεταξύ της εντολής DO και της εντολής με ετικέτα N. Την πρώτη φορά έχω I=e_1 και στη συνέχεια το I μεταβάλλεται με βήμα e_3 κάθε φορά μέχρις ότου το I να μην μπορεί να μεταβληθεί κατά e_3 συνεχίζοντας να βρίσκεται μεταξύ των e_1, e_2.

Παρατηρήσεις:

i) Το βήμα e_3 δυνατόν να είναι αρνητικό, οπότε η μεταβολή του I είναι ελάττωση.

ii) Η μεταβλητή ελέγχου παίρνει τιμή πριν ληφθεί απόφαση για το αν θα εκτελεστεί ο βρόχος ή όχι.

iii) Απαγορεύεται η αλλαγή των παραμέτρων I, e_1, e_2, e_3 της εντολής DO εντός του βρόχου DO, αλλά επιτρέπεται η χρησιμοποίησή τους.

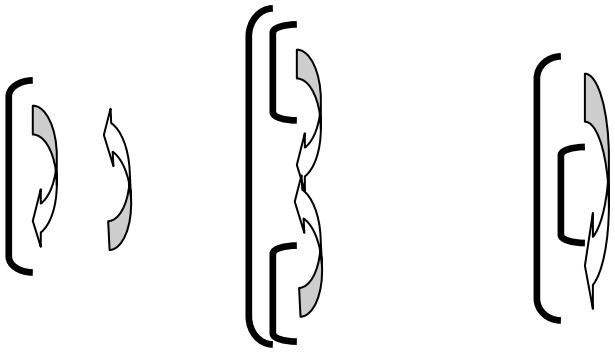
iv) Επιτρέπεται η είσοδος στο βρόχο DO μόνο μέσω της αρχικής εντολής DO.

v) Η έξοδος επιτρέπεται από οποιοδήποτε σημείο.

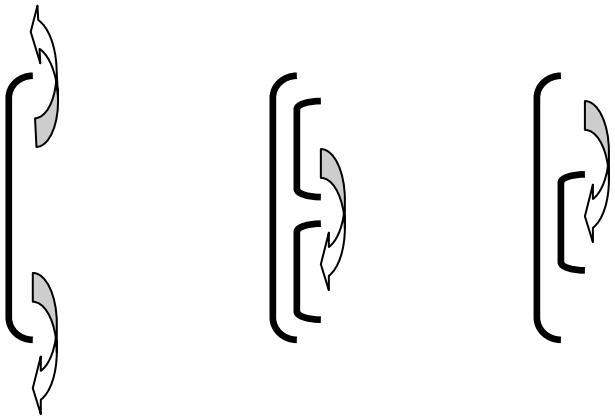
vi) Ο αριθμός επαναλήψεων του βρόχου DO είναι

$\max \{0, INT((e_2-e_1+e_3)/e_3)\}$, όπου $INT(X)=[X]$.

Επιτρεπτές κινήσεις στους βρόχους DO



Μη επιτρεπτές κινήσεις στους βρόχους DO



7.3.2. Η ΕΝΤΟΛΗ DO WHILE

Σύνταξη: **DO N WHILE (e)**

όπου N κατά τα γνωστά ετικέτα εντολής FORTRAN όπως και στην εντολή DO και e λογική έκφραση.

Λειτουργία: Εκτελούνται κατ΄επανάληψη οι εντολές μεταξύ της εντολής DO WHILE και της εντολής με ετικέτα N μέχρις ότου η λογική έκφραση e γίνει ψευδής. Ο έλεγχος της λογικής έκφρασης e προηγείται της εκτέλεσης των εντολών.

7.4. Η ΕΝΤΟΛΗ END DO

Σύνταξη: **END DO**

Λειτουργία: Χρησιμεύει σαν εντολή τερματισμού των βρόχων DO και DO WHILE όταν αυτοί δεν διαθέτουν ετικέτα N. Έτσι οι εντολές DO και DO WHILE γράφονται:

```
DO    I = e1, e2, e3
      :
      :
END DO
```

και

```
DO WHILE (e)
      :
      :
END DO
```

8. ΜΕΤΑΒΛΗΤΕΣ ΜΕ ΔΕΙΚΤΕΣ

8.1. Η ΕΝΤΟΛΗ DIMENSION

Σύνταξη: **DIMENSION name (l_1: q_1, ..., l_k: q_k), ...**

όπου name το συμβολικό όνομα της μεταβλητής με δείκτες (πίνακα), k η διάστασή της (συνήθως $k \leq 7$) και l_i, q_i ακέραιες σταθερές που εκφράζουν την κατώτατη, την

ανώτατη τιμή αντίστοιχα του δείκτη της i διάστασης του πίνακα. Όταν τα l_i παραλείπονται, τότε θεωρούνται ότι έχουν την τιμή 1.

Ο συνολικός αριθμός θέσεων μνήμης που δεσμεύονται από τον υπολογιστή μέσω της εντολής DIMENSION για την αποθήκευση ενός πίνακα δίνεται από τον τύπο

$$\prod_{i=1}^{k}(q_i - l_i + 1)$$

Π.χ. η εντολή

```
DIMENSION A(5), B(-1:1,3)
```

δηλώνει ότι ο A είναι πίνακας μιας διάστασης με στοιχεία A(1), ..., A(5) και ο B πίνακας δύο διαστάσεων με στοιχεία B(-1, 1), B(0, 1), B(1, 1), B(-1, 2), B(0, 2), B(1, 2), B(-1, 3), B(0,3), B(1, 3).

Υπάρχει η δυνατότητα της σύγχρονης δήλωσης των διαστάσεων και του τύπου των στοιχείων ενός πίνακα μέσω των εντολών καθορισμού τύπου, π.χ.

```
INTEGER A(5)
REAL B(-1:1, 3)
```

8.2. ΑΠΟΘΗΚΕΥΣΗ ΣΤΟΙΧΕΙΩΝ ΠΙΝΑΚΑ

Τα στοιχεία ενός πίνακα στη FORTRAN αποθηκεύονται σε ένα διάνυσμα με τρόπο ώστε ο δείκτης που προηγείται να μεταβάλλεται γρηγορότερα από το δείκτη που έπεται. Αυτόν τον τρόπο αποθήκευσης καλό θα είναι να λαμβάνουμε υπόψη μας όταν χρησιμοποιούμε πίνακες σε εντολές επανάληψης. Συμφέρει δηλαδή να γράφουμε:

```
      DO 10 J=1, K
      DO 10 I=1, N
      C(I, J) = A(I, J) + B(I, J)
10    CONTINUE
```

Για παράδειγμα τα στοιχεία ενός πίνακα A(3, 4) τοποθετούνται κατά στήλες σε 12 διαδοχικές θέσεις μνήμης ως εξής:

A (1,1), A(2,1), A(3,1), A(1,2), A(2,2), A(3,2), ..., A(3,4).

Έτσι τα στοιχεία μιας στήλης βρίσκονται σε διαδοχικές θέσεις μνήμης (άρα είναι εύκολο να τα χρησιμοποιούν το ένα μετά το άλλο) ενώ τα στοιχεία της ίδιας γραμμής απέχουν μεταξύ τους τόσες θέσεις μνήμης όσες δείχνει ο πρώτος δείκτης της δήλωσης διάστασης (DIMENSION) στην περίπτωσή μας 3 θέσεις μνήμης.

Γενικά το στοιχείο α_{ijl} ενός MxNxK πίνακα με DIMENSION A(M, N, K) βρίσκεται στη (l-1)MN + (j-1)M + i θέση του διανύσματος όπου αποθηκεύεται ο A, ή για πίνακα δύο διαστάσεων, που είναι και η συνηθέστερη περίπτωση, η θέση του α_{ij} ενός MxN πίνακα είναι η (j-1)M+i θέση του αντίστοιχου διανύσματος. Αντιστρόφως, στη θέση k του διανύσματος όπου είναι αποθηκευμένος ένας MxN πίνακας βρίσκεται το στοιχείο α_{ij} όπου

$$(i, j) = \begin{cases} (M, K/M), \text{ αν υπόλοιπο } (K/M)=0 \\ (\text{υπόλοιπο } (K/M), 1+[K/M]), \text{ αν υπόλοιπο } (K/M)\neq 0 \end{cases}$$

9. ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ

Τα υποπρογράμματα είναι μια εντολή ή ένα σύνολο εντολών FORTRAN, που εκτελούν μια συγκεκριμένη υπολογιστική διαδικασία. Είναι σχετικά ανεξάρτητες μονάδες προγράμματος που μπορούν να κληθούν (και να εκτελεστεί η διαδικασία που περιγράφουν) από διάφορες άλλες μονάδες προγράμματος και περισσότερες από μία φορές.

9.1. Η ΕΝΤΟΛΗ FUNCTION

Σύνταξη: **FNAME (x_1, x_2, ..., x_μ) = e**

όπου FNAME το όνομα της συνάρτησης (σύμφωνα με τους κανόνες της FORTRAN για τα ονόματα), x_1, x_2, ..., x_μ οι μ μεταβλητές της συνάρτησης, που είναι «βουβές», με την έννοια ότι η μόνη χρήση τους είναι να δείχνουν το πλήθος και τον τύπο των μεταβλητών της συνάρτησης και e μια αριθμητική ή λογική παράσταση.

Λειτουργία: Κάθε φορά που παρουσιάζεται στη μονάδα προγράμματος που ορίζεται το όνομα της συνάρτησης FNAME (α_1, ..., α_μ), όπου α_1, ..., α_μ συγκεκριμένες τιμές ή αριθμητικές -λογικές παραστάσεις στη θέση των «βουβών» μεταβλητών της, υπολογίζεται η τιμή της e του ορισμού της συνάρτησης και εκχωρείται στο όνομα της συνάρτησης.

Παρατηρήσεις:

i) Η εντολή FUNCTION γράφεται στην αρχή του προγράμματος και πριν από κάθε άλλη εντολή ή άλλη FUNCTION που πρόκειται να την χρησιμοποιήσει.
ii) Υπάρχει η δυνατότητα του άμεσου καθορισμού του τύπου της FNAME. Π.χ.
```
REAL X, Y, NU
NU (A, X, Y) = COS(X) + A * Y
```
iii) Πρέπει να περιέχει τουλάχιστον μια μεταβλητή.
iv) Πρέπει τα α_i να συμφωνούν με τις βουβές μεταβλητές x_i στο πλήθος και τον τύπο αντίστοιχα.

9.2. ΥΠΟΠΡΟΓΡΑΜΜΑ FUNCTION

Σύνταξη:

```
Type FUNCTION FNAME (x1, x2, ..., xμ)
      :
      :
block of FORTRAN statements
      :
      :
FNAME = e
      :
      :
RETURN
END
```

όπου Type προαιρετικός άμεσος καθορισμός του τύπου της FNAME, FNAME το όνομα της συνάρτησης, x_1, x_2, ..., x_μ βουβές μεταβλητές, RETURN εντολή εκτελέσιμη, που δηλώνει το λογικό τέλος του FUNCTION υποπρογράμματος και END εντολή που δηλώνει το φυσικό τέλος του υποπρογράμματος.

Λειτουργία: Καλείται με το όνομα του υποπρογράμματος με συγκεκριμένες τιμές ή εκφράσεις στη θέση των βουβών μεταβλητών. Ο έλεγχος μεταφέρεται στην πρώτη εκτελέσιμη εντολή του υποπρογράμματος και όταν φτάσει στην εντολή RETURN, ο έλεγχος επιστρέφει στο σημείο του κυρίως προγράμματος που έγινε η κλήση του FUNCTION υποπρογράμματος, με μια μοναδική τιμή στο όνομα της συνάρτησης.

9.3 ΥΠΟΠΡΟΓΡΑΜΜΑ SUBROUTINE

Σύνταξη:

```
SUBROUTINE SUBNAM (x1, x2, ..., xμ)
                :
                :
RETURN
END
```

όπου SUBNAM το όνομα του υποπρογράμματος SUBROUTINE, x_1, x_2, ..., x_μ βουβές μεταβλητές που χρησιμοποιούνται τόσο για την είσοδο, όσο και για την έξοδο τιμών από και προς τη SUBROUTINE.

Η κλήση γίνεται από το κυρίως πρόγραμμα με την εντολή CALL

```
CALL SUBNAM (α1, α2, ..., αμ)
```

όπου α_1, α_2, ..., α_μ συγκεκριμένες παράμετροι, που πρέπει να συμφωνούν στο πλήθος, τον τύπο και τη σειρά με τις βουβές μεταβλητές x_1, ..., x_μ της SUBROUTINE.

Λειτουργία:

ΚΥΡΙΟ ΠΡΟΓΡΑΜΜΑ

```
          :
          :
          :
CALL SUBNAM (A, B, X, N)
          :
          :
         END
```

ΥΠΟΠΡΟΓΡΑΜΜΑ

```
SUBROUTINE SUBNAME (A,BETA,M,K)
           REAL M
             :
             :
           RETURN
            END
```

Παρατηρήσεις:

i) Δεν εκχωρείται τιμή στο όνομα της SUBROUTINE, συνεπώς δεν χρειάζεται άμεσος ή έμμεσος καθορισμός του τύπου του ονόματός της.

ii) Όλα τα υποπρογράμματα γράφονται, εκτός του κυρίως προγράμματος που τα καλεί και σε οποιαδήποτε σειρά.

9.4. ΜΕΤΑΒΛΗΤΗ ΔΙΑΣΤΑΣΗ ΠΙΝΑΚΩΝ

Υπάρχει η δυνατότητα μεταβλητής διάστασης των πινάκων ενός υποπρογράμματος όταν οι διαστάσεις αυτές μεταφέρονται στο υποπρόγραμμα μέσω των παραμέτρων του. Σ΄αυτή την περίπτωση οι διαστάσεις στην εντολή DIMENSION του υποπρογράμματος είναι ακέραιες μεταβλητές και όχι σταθερές.

Παράδειγμα:

```
PROGRAM MANORM
DIMENSION A(10,10)
NDIM =10
READ*,N,((A(I,J),I=1,N),J=1,N)
CALL ENORM (A,NDIM,N,EN)
PRINT*, EN
STOP
END
```

```
      SUBROUTINE ENORM (A,NDIM,K,EN)
      DIMENSION A (NDIM, K)
      SUM = 0
      DO 10 J=1, K
      DO 10 J=1, K
      SUM = SUM+A(I, J) **2
      CONTINUE
   10 EN=SQRT (SUM)
      RETURN
      END
```

9.5. Η ΕΝΤΟΛΗ EXTERNAL

Σύνταξη: **EXTERNAL V_1, V_2, ...**

όπου V_1, V_2, ... ονόματα υποπρογραμμάτων.

Η μη εκτελέσιμη εντολή EXTERNAL απαιτείται να υπάρχει στη μονάδα προγράμματος που καλεί ένα υποπρόγραμμα (FUNCTION ή SUBROUTINE) στις βουβές μεταβλητές του οποίου περιέχεται το όνομα ενός άλλου υποπρογράμματος.

Παράδειγμα χρήσης της εντολής EXTERNAL:

Η πραγματική συνάρτηση υποπρόγραμμα ZEROIN (A, B, F, TOL) υπολογίζει τη ρίζα της πραγματικής συνάρτησης F, που περιέχεται στο διάστημα [A, B].
Το παρακάτω πρόγραμμα καλεί τη ZEROIN για την εύρεση της ρίζας της F(x)=cosx-x στο [0, 1].

```
PROGRAM ROOT
REAL ZEROIN
EXTERNAL F
A = 0.
B = 1.
TOL = 1.0 E-06
R = ZEROIN (A, B, F, TOL)
PRINT*,R
STOP
END
```

```
REAL FUNCTION F(X)
F = COS(X)-X
RETURN
END
```

9.6. Η ΕΝΤΟΛΗ EQUIVALENCE

Είναι η μη εκτελέσιμη εντολή που τίθεται αμέσως μετά τις εντολές καθορισμού τύπου και που αναγκάζει δύο ή περισσότερες μεταβλητές να αντιστοιχηθούν στην ίδια θέση μνήμης.

Σύνταξη: **EQUIVALENCE (L_1), (L_2), ...**

όπου L_1, L_2, ... λίστες από τουλάχιστον 2 μεταβλητές.

Παράδειγμα:

```
INTEGER A, B
REAL X, Y
EQUIVALENCE (I,A,B), (X,Y)
```

Η εντολή EQUIVALENCE συσχετίζει του ακέραιους I, A, B σε μία θέση μνήμης και τους πραγματικούς X, Y επίσης σε μια θέση μνήμης.

Ειδικά για πίνακες, όταν συσχετιστεί ένα στοιχείο τους ενός με κάποιο στοιχείο του άλλου, συσχετίζονται αυτόματα και όλα τα στοιχεία των πινάκων που ακολουθούν.

Παράδειγμα: Με τις εντολές

```
DIMENSION A(5), B(5), C(3,3), D(10)
EQUIVALENCE (A(4), B(1)), (C(1,2), D(7))
```

συσχετίζουμε τα στοιχεία των πινάκων ως εξής

A(4), A(5)
B(1), B(2), B(3), B(4), B(5)
C(1,2), C(2,2), C(3,2), C(1,3), C(2,3), C(3,3)
D(7), D(8), D(9), D(10)

9.7. Η ΕΝΤΟΛΗ COMMON

Επιτρέπει τη συσχέτιση μεταβλητών που βρίσκονται σε διαφορετικές μονάδες ενός προγράμματος με την ίδια θέση μνήμης. Είναι μη εκτελέσιμη εντολή και γράφεται μετά τις εντολές καθορισμού τύπου.

Σύνταξη: **COMMON L**

ή

COMMON / Name$_1$ / L$_1$ / Name$_2$ / L$_2$ ...

όπου L ή L_1, L_2,... λίστες μεταβλητών και $Name_1$, $Name_2$, ... προαιρετικά ονόματα COMMON blocks.

Παράδειγμα 1: Αν στο κυρίως πρόγραμμα υπάρχει η εντολή

```
COMMON A, B, C
```

και στο υποπρόγραμμα υπάρχει η εντολή

```
COMMON X, Y
```

τότε οι μεταβλητές A, X αντιστοιχίζονται στην ίδια θέση μνήμης όπως και οι μεταβλητές B, Y.

Παράδειγμα 2: Αν στο κυρίως πρόγραμμα υπάρχει η εντολή

```
COMMON / BL1 / A,B / BL2 / C, D / / EPS
```

και στο υποπρόγραμμα υπάρχει η εντολή

```
COMMON TOL / BL1 / ALFA, BETA / BL2 / X, Y
```

τότε οι μεταβλητές A, B της COMMON περιοχής με όνομα BL1 αντιστοιχίζονται στις ίδιες θέσεις μνήμης με τις ALFA, BETA όπως και οι C, D με τις X, Y ενώ η μεταβλητή EPS της blank COMMON περιοχής αντιστοιχίζεται στην ίδια θέση μνήμης με τη μεταβλητή TOL.

Αν σε μία λίστα COMMON εντολής συμπεριλαμβάνεται και το όνομα ενός πίνακα τότε η διάστασή του θα δηλωθεί ή στην εντολή COMMON ή στην εντολή DIMENSION αλλά όχι και στις δύο. Οι εντολές

```
DIMENSION A(10, 10), B(5)
```

```
COMMON A, B, C
```

έχουν το ίδιο αποτέλεσμα με την εντολή

```
COMMON A(10, 10), B(5), C
```

Ακόμα οι εντολές

```
COMMON A, B
COMMON C
```

έχουν το ίδιο αποτέλεσμα με την εντολή

```
COMMON A, B, C
```

Η εντολή COMMON αποτελεί έναν εναλλακτικό τρόπο για το πέρασμα όλων ή μερικών τιμών σε ένα υποπρόγραμμα.

Παράδειγμα 3:

Κύριο πρόγραμμα

```
COMMON A(10, 10), B(10)
          :
          :
      CALL TABLE
          :
          :
        END
```

Υποπρόγραμμα

```
   SUBROUTINE TABLE
COMMON C(10,10), D(10)
          :
          :
      RETURN
        END
```

Είναι δυνατόν να εμφανίζονται σε εντολή EQUIVALENCE μεταβλητές που εμφανίζονται επίσης σε εντολή COMMON (αρκεί να μην ανήκουν στο ίδιο COMMON block).

Παράδειγμα 4: Οι εντολές

```
DIMENSION A(2,2), B(5)
COMMON A, Y, Z
EQUIVALENCE (A(1,2), B(1))
```

δίνουν τη συσχέτιση

A(1,1), A(2,1), A(1,2), A(2,2), Y, Z

B(1), B(2), B(3), B(4), B(5)

δηλαδή η COMMON περιοχή επεκτείνεται και κατά μία θέση προς τα δεξιά. Επέκταση προς τα αριστερά δεν επιτρέπεται.

10. ΕΝΤΟΛΕΣ ΕΙΣΟΔΟΥ - ΕΞΟΔΟΥ

10.1. Η ΕΝΤΟΛΗ READ

Ένας απλός τρόπος για είσοδο στοιχείων από την οθόνη είναι με την εντολή READ που έχει τη μορφή:

Σύνταξη: **`READ*, L`**

όπου L λίστα μεταβλητών, που πρέπει να χωρίζονται με κόμμα.

Η FORTRAN 77 μας επιτρέπει και την ελεγχόμενη είσοδο στοιχείων από την οθόνη ή άλλη μονάδα ή από αρχείο, με την εντολή

Σύνταξη: **READ ([UNIT=]u, [FMT=]f [,ERR=s] [,REC=rn]**
[,END=s] [,IOSTAT=IOS]) L

όπου:

u αριθμός μονάδας εισόδου. Αν u=* ή 5 θεωρείται η προσυνδεδεμένη μονάδα εισόδου του συστήματος (συνήθως οθόνη)

f ετικέτα εντολής FORMAT (θα περιγραφεί παρακάτω) ή * για είσοδο μέσω κάποιου ενσωματωμένου FORMAT

s ετικέτα εντολής, όπου μεταφέρεται ο έλεγχος όταν παρουσιαστεί λάθος κατά την ανάγνωση (ERR=s) ή το αρχείο από όπου διαβάζουμε έχει φτάσει στο τέλος του (END=s)

rn αριθμητική έκφραση, που καθορίζει τον αριθμό εγγραφής (record) σε περίπτωση ανάγνωσης από αρχείο τυχαίας προσπέλασης (radom access)

IOS ακέραια μεταβλητή που λαμβάνει:

i) την τιμή 0, αν η ανάγνωση τελείωσε χωρίς σφάλμα και χωρίς να τελειώσει το αρχείο εισόδου δεδομένων.

ii) τιμή >0, αν υπάρχει σφάλμα

iii) τιμή <0, αν δεν υπάρχει σφάλμα ανάγνωσης αλλά υπάρχει τέλος αρχείου.

Παραδείγματα:

```
READ*, A
READ 10, A
READ (*,*) A, B
READ (*, 10) A, B, C
READ (5, 10, END=100) A, B
READ *, A, B(1), B(2), ((C(I,J), I=1,N)J=1,M)
```

10.2. Η ΕΝΤΟΛΗ PRINT

Χρησιμοποιείται για εκτύπωση στην οθόνη.

Σύνταξη: **PRINT f [,L] [,Nu] [,C]**

όπου f ετικέτα εντολής FORMAT ή * για εκτύπωση κατευθυνόμενης λίστας μεταβλητών, L λίστα μεταβλητών, Nu αριθμητικές εκφράσεις και C σειρά χαρακτήρων.

Παραδείγματα:

```
PRINT *, A, B, X
PRINT 10, A
PRINT *, X+Y, A, 2**3, 'STAR'
```

10.3. Η ΕΝΤΟΛΗ WRITE

Σύνταξη: **WRITE ([UNIT=]u, [FMT=]f [,REC=rn]**
[,ERR=s] [,IOSTAT=IOS]) L

όπου

u αριθμός μονάδας εξόδου. Αν u=* ή 6 θεωρείται ότι αναφερόμαστε στην προσυνδεδεμένη μονάδα εξόδου του συστήματος (συνήθως οθόνη) και

f, rn, s, ios ίδιας σημασίας με την εντολή READ.

Παραδείγματα:

```
WRITE (*,*) A
WRITE (6,*) A, B, X
WRITE (6, 10, ERR=100) A
```

10.4. Η ΕΝΤΟΛΗ FORMAT

Είναι μια μη εκτελέσιμη εντολή, που καθορίζει τον τρόπο με τον οποίο ο υπολογιστής θα δεχθεί τα δεδομένα ή θα τυπώσει τα αποτελέσματα.

Σύνταξη: **n FORMAT (q_1 e_1 s_1 e_2 s_2 ... e_n s_n)**

όπου n η ετικέτα της εντολής, q_1 χαρακτήρας ελέγχου εκτύπωσης, e_1, ..., e_n περιγραφείς σύνταξης (edit descriptors) και s_1, ..., s_n διαχωριστές.

Αν η εντολή FORMAT αφορά έξοδο αποτελεσμάτων σε εκτυπωτή, τότε το q_1 ελέγχει την εκτύπωση ως εξής:

κενό	:	εκτύπωση σε νέα γραμμή
+	:	εκτύπωση στην ίδια γραμμή
0	:	αφήνει μία κενή γραμμή και εκτυπώνει στην αρχή της επόμενης
1	:	εκτύπωση στην αρχή νέας σελίδας
άλλος χαρακτήρας	:	αγνοείται και δεν εκτυπώνεται

Οι πιο συνηθισμένοι περιγραφείς σύνταξης είναι:

Aw	Ew.d	Dw.d Ee
Iw	Dw.d	Gw.d Ee
Iw.m	Gw.d	Lw
Fw.d	Ew.d Ee	

όπου A δηλώνει αλφαριθμητικό πεδίο

I δηλώνει ακέραιο πεδίο

F δηλώνει πραγματικό πεδίο

E δηλώνει πραγματικό πεδίο σε εκθετική μορφή

D δηλώνει πραγματικό πεδίο διπλής ακρίβειας σε εκθετική μορφή

G δηλώνει πραγματικό σε εκθετική μορφή μόνο αν είναι απαραίτητο από το μέγεθος του αριθμού

L δηλώνει λογικό πεδίο.

Ακόμη το w δηλώνει το συνολικό μέγεθος του πεδίου, m το ελάχιστο πλήθος των ψηφίων του αριθμού, d είναι ο αριθμός των ψηφίων του δεκαδικού μέρους και e ο αριθμός των ψηφίων του εκθέτη. Πρέπει να δοθεί προσοχή ώστε στο πλάτος του πεδίου να υπάρχει επαρκής χώρος για τα πρόσημα του αριθμού και του εκθέτη (αν υπάρχει), για την υποδιαστολή κλπ.

Π.χ. στο πεδίο Fw.d πρέπει $w \geq d+3$

στο πεδίο Ew.d πρέπει $w \geq d+7$

και στο πεδίο Ew.dEe πρέπει $w \geq d+5+e$.

Δίνουμε τώρα μερικά παραδείγματα εκτύπωσης

Αριθμός	**Περιγραφή**	**Εκτύπωση**
135	I4	b135
135	I2	**
135	I7.5	bb00135
-3.14	F4.1	-3.1
0.6936	E10.3	b0.694E+00
-693.6	E14.5E4	-0.69360E+0003
613.715826957	D20.13	b0.6137158269570D+03
7.1	G11.3	bbb7.10bbbb
0.007	G11.3	bb0.700E-02
.TRUE.	L4	bbbT

Οι διαχωριστές s_1, s_2, ... είναι συνήθως κόμματα. Μπορεί επίσης να χρησιμοποιηθούν:

i) Ο χαρακτήρας / (slash) που προκαλεί τη μεταφορά του ελέγχου για ανάγνωση ή εγγραφή στην αρχή μιας νέας γραμμής αν πρόκειται για είσοδο ή έξοδο από αρχείο ή την εκτύπωση στην αρχή μιας νέας γραμμής αν πρόκειται για έξοδο σε εκτυπωτή.

ii) Ο χαρακτήρας : (colon) που τερματίζει την περιγραφή της εντολής FORMAT, όταν δεν υπάρχουν άλλα στοιχεία για να διαβαστούν ή να εκτυπωθούν.

Παρατηρήσεις:

i) Είναι δυνατόν, η επανάληψη ενός περιγραφέα σύνταξης να γίνει με ένα θετικό ακέραιο που τίθεται μπροστά του. Η επανάληψη μπορεί να αφορά και ομάδες περιγραφέων σύνταξης που βρίσκονται ανάμεσα σε παρενθέσεις.

ii) Η παρουσία του στοιχείου nX σε μια εντολή FORMAT προκαλεί τη μεταφορά του ελέγχου για ανάγνωση ή εκτύπωση n θέσεις προς τα δεξιά.

iii) Η εκτύπωση ενός αλφαριθμητικού πεδίου (μηνύματος, τίτλου κλπ) μπορεί να γίνει μέσω των περιγραφέων

nHs_1, s_2, ...s_n

ή

´s_1s_2 ...s_n´

όπου n το πλήθος των χαρακτήρων και s_1s_2...s_n το αλφαριθμητικό πεδίο που εκτυπώνεται αυτούσιο.

Παραδείγματα:

```
            READ (5,10) N, A, B
         10 FORMAT (I4, F10.5, E16.8)
```

ή

```
         10 FORMAT (I4/2F10.5)
```

ή

```
         10 FORMAT (I5, 5X, F10.5:F10.5)
            WRITE (6,20) N, A, B
         20 FORMAT (1H1, 10X, I5, 2(5X, F10.5)/)
```

ή

```
         20 FORMAT (´ON=´, I5, 5X, 2HA=, G12.5/13X,
2HB=, G12.5//)
```

10.5. Η ΕΝΤΟΛΗ DATA

Σύνταξη: **DATA $V_1/C_1/$, $V_2/C_2/$, ... V_n/C_n**

όπου V_i λίστα μεταβλητών, πινάκων ή στοιχείων πινάκων που χωρίζονται με κόμματα και C_i λίστα σταθερών.

Λειτουργία: Εκχωρεί κατά σειρά σε κάθε μεταβλητή της λίστας V_i τις τιμές της λίστας C_i.

Είναι μη εκτελέσιμη εντολή, που τίθεται μετά τις εντολές δηλώσεως τύπου των μεταβλητών που περιέχει.

Παραδείγματα:

```
    DATA N, K/10,100/, PI/3.1415926/

    DIMENSION A(5), B(5)
    DATA (A(I), I=1,5)/0., 1., 3*0./, B(2)/0./

    COMPLEX A
    CHARACTER*10 TITLE
    DATA TITLE, A/´PROGRAM´, (5., 2.)/

    DATA EPS/1.E-06/
```

αντί

```
    EPS=10.**(-6)

    REAL A(10,10)
    DATA A/100*0./
```

αντί

```
    DO 10 J=1,10
    DO 10 I=1,10
    A(I,J) = 0.
 10 CONTINUE
```

Α Σ Κ Η Σ Ε Ι Σ

ΧΡΗΣΗ ΤΗΣ FORTRAN ΣΤΗΝ ΑΡΙΘΜΗΤΙΚΗ ΑΝΑΛΥΣΗ

1. Αριθμητική Επίλυση Μη Γραμμικών Εξισώσεων

1.1. Γράψτε ένα πρόγραμμα FORTRAN, σε διπλή ακρίβεια, που να προσεγγίζει μία ρίζα x^* της εξίσωσης $f(x)=0$, όπου $f(x)$ δεδομένη συνάρτηση, με την τροποποιημένη μέθοδο του Νεύτωνα (Newton-Raphson), δηλαδή που να υπολογίζει τους όρους x_n της ακολουθίας

$$x_{n+1} = x_n - m\frac{f(x_n)}{f'(x_n)},\ n=0,1,\ldots, \qquad (1)$$

όπου x_0 δεδομένη αρχική προσέγγιση της x^*, και m η πολλαπλότητα της ρίζας x^*. Οι συναρτήσεις $f(x), f'(x)$ πρέπει να δίνονται από ένα υποπρόγραμμα subroutine ή function η κάθε μία. (Έτσι, για κάθε συγκεκριμένη εφαρμογή -δηλαδή για κάθε διαφορετική f- θα αλλάζουν μόνο τα δύο αυτά υποπρογράμματα). Τα δεδομένα εισόδου θα είναι: η πολλαπλότητα m της ρίζας x^*, η αρχική προσέγγιση x_0 και δύο παράμετροι: ε>0 και NMAX (ακέραιος). Η παράμετρος ε (ε «μικρός» θετικός) θα χρησιμοποιείται στο εξής ως κριτήριο τερματισμού για την (1): αν $|x_{n+1}-x_n| \le \varepsilon$ για πρώτη φορά, τότε πάρε το x_{n+1} ως «ρίζα». (Για ασφάλεια, κάντε το πολύ NMAX επαναλήψεις). Με το πρόγραμμα αυτό απαντήστε στα εξής ερωτήματα:

(i) Δοκιμάστε το πρόγραμμά σας στην περίπτωση $f(x)=x^2-2x-3$, για την προσέγγιση της ρίζας $x^*=-1$. Με το $x_0=0$, ε=10^{-6} ή 10^{-8}, NMAX=50, τυπώστε τα x_n και τα σφάλματα $\varepsilon_n = x_n - x^*$ και βεβαιωθείτε υπολογιστικά (πειραματικά) ότι τα x_n «τείνουν» στη x^*. Την εκλογή του m να την κάνετε αφού πρώτα από μια πρόχειρη γραφική παράσταση αποκτήσετε μια ιδέα για την πολλαπλότητα της ρίζας $x^*=-1$.

(ii) Αποδείξτε αναλυτικά ότι η εξίσωση $f(x)=e^x-x-1=0$ έχει μόνο μία πραγματική ρίζα, την $x^*=0$, που είναι διπλή. «Αποδείξτε» το ίδιο πράγμα αριθμητικά δείχνοντας (με $x_0=1$, $\varepsilon=10^{-6}$, $NMAX=150$ και $m=1$) ότι $\varepsilon_{n+1}/\varepsilon_n$ τείνει σε κατάλληλη σταθερά καθώς αυξάνει το n. Στη συνέχεια επαναλάβετε αλλά τώρα με $m=2$. Σημειωτέον ότι η μέθοδος (1) συγκλίνει (στην πράξη) τετραγωνικά μόνο για $m=$ πολλαπλότητα της ζητούμενης ρίζας. Πειραματιστείτε και με άλλες τιμές του m.

(iii) Να γίνει μετατροπή του προγράμματος που έχετε κατασκευάσει για τη μέθοδο Newton-Raphson έτσι ώστε κατά την εφαρμογή του τύπου (1) για k διαδοχικές επαναλήψεις (k=ακέραια σταθερά που θα διαβάζεται από το πρόγραμμα) να διατηρείται η ίδια τιμή για την f'.

Εφαρμογή:

$f(x)=x^4-6x^3+9.1x^2-0.6x+0.9$, $\varepsilon=10^{-6}$, NMAX=150, $m=2$, $k=5$ και $x_0=-1$ ή $x_0=1$ ή $x_0=2$ ή $x_0=4$.

1.2. Έστω η μέθοδος εσφαλμένης θέσης (Regula Falsi) που είναι παραλλαγή της μεθόδου της διχοτόμησης με μόνη διαφορά ότι ανά βήμα το x_k αντί να εκλέγεται σαν το μέσο του διαστήματος (a_k,b_k) δίνεται από τον τύπο

$$x_k=\frac{a_k f(b_k)-b_k f(a_k)}{f(b_k)-f(a_k)}.$$

Να γραφεί ένα πρόγραμμα FORTRAN με την ονομασία REGFA για τη μέθοδο αυτή που να σταματά όταν

$$f(x_k-\varepsilon)f(x_k+\varepsilon)\le 0$$

Να εφαρμοστεί το ανωτέρω πρόγραμμά σας για την εύρεση της θετικής ρίζας της συνάρτησης

$$f(x)=\cos x-\ln x.$$

Η εκλογή του αρχικού διαστήματος (a_0,b_0) να γίνει μετά από πρόχειρη γραφική παράσταση της f. Θεωρείστε $\varepsilon=10^{-6}$.

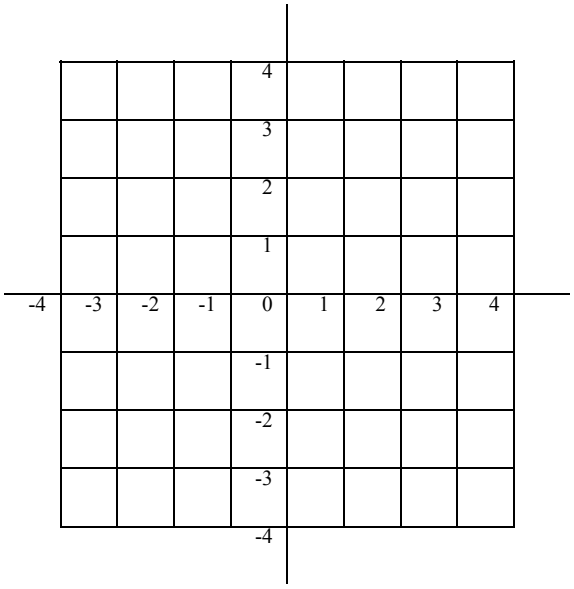
1.3. Να εφαρμοστεί το πρόγραμμα της άσκησης 1.1 (μέθοδος Newton-Raphson) για να βρεθούν οι ρίζες της εξίσωσης

$$f(x) = x^6 - 11x^5 + 49x^4 - 113x^3 + 142x^2 - 92x + 24$$

καθώς και οι πολλαπλότητές τους.

Για τον καθορισμό της πολλαπλότητας κάθε ρίζας να ελεγχθεί η ταχύτητα σύγκλισης της μεθόδου για διάφορες τιμές του $m \le 5$. Σημειωτέον ότι η μέθοδος 1.1(1) συγκλίνει (στην πράξη) τετραγωνικά μόνο για $m =$ πολλαπλότητα της ζητούμενης ρίζας.

1.4. Έστω η μιγαδική μέθοδος Newton-Raphson για την επίλυση της εξίσωσης $f(z) = 0,\ z \in C$. Να γραφεί ένα πρόγραμμα FORTRAN για την εύρεση όσο το δυνατόν περισσότερων ριζών της μιγαδικής συνάρτησης $f(z)$, ξεκινώντας από κάθε σημείο (x_0, y_0) του πλέγματος ,



τυπώνοντας για κάθε αρχικό σημείο $z_0 = \text{DCMPLX}(x_0, y_0)$ την ενδεχόμενη προσέγγιση της ρίζας z_k, αν ικανοποιείται το κριτήριο διακοπής

$$|z_k - z_{k-1}| \le \text{EPS}$$

με EPS=10^{-10} και μέγιστο αριθμό επαναλήψεων NIT=20. Αν σας είναι δυνατό ομαδοποιείστε σε ένα σχήμα τα αρχικά σημεία z_0 του πλέγματος που οδηγούν σε

σύγκλιση στην ίδια ρίζα της συνάρτησης. Είστε ελεύθεροι να πειραματιστείτε με μικρότερο βήμα h για το πλέγμα του σχήματος.

Εφαρμογή:

(i) $f(z) = z^3 - 11.9z^2 + 46.65z - 62.975$

(ii) $f(z) = z^3 - 2.5z^2 - 4z + 21.25$

1.5. Στη λύση με αναλυτικές μεθόδους του προβλήματος ιδιοτιμών

$$y'' + \lambda y = 0,\ y(0) = 0,\ y(1) = y'(1)$$

(βλέπε π.χ. Συνήθεις Διαφορικές Εξισώσεις, Γ. Παντελίδη, Δ. Κραββαρίτη, Ν. Χατζησάββα) χρειάζεται η επίλυση της εξίσωσης

$$\tan x = x,\ x > 0 \qquad (2)$$

Εφαρμόζοντας το πρόγραμμα της άσκησης 1.1 να βρεθούν οι τρεις πρώτες θετικές ρίζες της (2).

1.6. Η συνάρτηση Bessel τάξης 1 ορίζεται από τη σχέση

$$J_1(x) = \tfrac{x}{2} \sum_{k=0}^{\infty} \frac{(-x^2/4)^k}{k!(k+1)!}, \quad x > 0.$$

Να βρεθεί η μικρότερη θετική ρίζα της J_1 εφαρμόζοντας το πρόγραμμα της άσκησης 1.1. Αρχίστε προσεγγίζοντας την J_1 από τους 3 πρώτους όρους της σειράς και συνεχίστε αυξάνοντας το πλήθος των όρων μέχρις ότου πετύχετε ικανοποιητική ακρίβεια στην εύρεση της ρίζας.

2. Επίλυση Γραμμικών Συστημάτων

2.1. Έστω το γραμμικό σύστημα $Ax = b$, όπου A ένας $n \times n$ τετραγωνικός πίνακας και b ένα $n \times 1$ διάνυσμα στήλη. Να γραφεί ένα πρόγραμμα FORTRAN για την επίλυση του γραμμικού συστήματος με τη μέθοδο απαλοιφής του Gauss και με μερική οδήγηση κατά στήλη. Το πρόγραμμα να περιλαμβάνει δύο υπορουτίνες. Η πρώτη με όνομα MAD (Matrix Decomposition) να τριγωνοποιεί τον πίνακα A να αποθηκεύει τους πολλαπλασιαστές στο κάτω τριγωνικό μέρος του A και τις τυχόν εναλλαγές γραμμών σε κατάλληλο διάνυσμα. Η δεύτερη με όνομα SYSO (System

Solving) να χρησιμοποιεί τις πληροφορίες της MAD προκειμένου να τροποποιεί το διάνυσμα b και στη συνέχεια με πίσω αντικατάσταση να λύνει το γραμμικό σύστημα.
Εφαρμογή σε σύστημα 4×4 της επιλογής σας.

2.2. Να γραφεί πρόγραμμα FORTRAN με όνομα MATRIN (Matrix Invert) που να χρησιμοποιεί τα υποπρογράμματα MAD και SYSO της άσκησης 2.1 προκειμένου να υπολογίζει τον αντίστροφο ενός $n\times n$ πίνακα A. Να χρησιμοποιηθεί το MATRIN για την αντιστροφή ενός 4×4 πίνακα A της επιλογής σας. Στη συνέχεια ομοίως να υπολογιστεί ο $(A^{-1})^{-1}$ και να σχολιαστούν τα αποτελέσματα.

2.3. Να γραφεί ένα πρόγραμμα FORTRAN για την επίλυση ενός συστήματος n μη γραμμικών εξισώσεων με n αγνώστους, με τη μέθοδο Newton-Raphson για συστήματα.

Εφαρμογή:

$$\left|\begin{array}{lllll} x_1^2 & + & x_2^2 & - & 1 = 0 \\ x_1^2 & - & x_2^2 & + & 1/2 = 0 \\ (x_1-1)^2 & + & (x_2-1)^2 & - & 4x_3 = 0 \end{array}\right.$$

Αρχικό διάνυσμα (1, 1, 1).

3. Παρεμβολή και Αριθμητική Ολοκλήρωση

3.1. Να γραφεί πρόγραμμα FORTRAN που να υπολογίζει τους συντελεστές του πολυωνύμου παρεμβολής Lagrange P_N σε μορφή Newton με διηρημένες διαφορές μιας συνάρτησης f σε $N+1$ σημεία καθώς και την τιμή του P_N σε κάποιο σημείο x.

3.2. Δίνεται η συνάρτηση $f(x)=\dfrac{1}{1+25x^2}$ (παράδειγμα Runge) ορισμένη στο διάστημα $[-1,1]$. Χρησιμοποιώντας το πρόγραμμα της άσκησης 3.1 να βρεθεί το

πολυώνυμο παρεμβολής Lagrange P_N σε μορφή Newton της f στα ισαπέχοντα σημεία $x_i = -1 + ih,\ i = 0,...N,\ h = 2/N$, καθώς και οι τιμές των σφαλμάτων $\left| f(x_i') - P_N(x_i') \right|$, όπου $x_i' = x_i + h/2$ για $i = 0,...,N-1$. Πως μεταβάλλεται το μέγιστο από αυτά τα σφάλματα όταν το N μεταβάλλεται από N =10 σε $N = 30$;

3.3. Να γραφεί πρόγραμμα FORTRAN για τον υπολογισμό του ολοκληρώματος μιας συνάρτησης στο διάστημα [a,b] με το σύνθετο τύπο αριθμητικής ολοκλήρωσης Simpson.

Εφαρμογή:

$$f(x) = \frac{4}{1+x^2}$$

$a = 0,\ b = 1$

$N = 20, 40.$

Να συγκριθούν τα αποτελέσματα με το ακριβές ολοκλήρωμα που είναι π.

3.4. Να κατασκευαστεί ο διπλός σύνθετος τύπος ολοκλήρωσης Simpson (χωρίς όρο σφάλματος) για τον υπολογισμό ενός ολοκληρώματος

$$\int_c^d \int_a^b f(x,y)dxdy$$

εφαρμόζοντας τον σύνθετο τύπο Simpson στην f για την ολοκλήρωση ως προς y (με βήμα h_y) και μετά ως προς x (με βήμα h_x). Να γραφεί ένα πρόγραμμα FORTRAN για την παραπάνω μέθοδο.

Εφαρμογή:

$$f(x,y) = \sin(x+y),$$
$$a = 0,\ b = \pi,\ c = 0,\ d = \pi,$$
$$h_x = (b-a)/m,\ h_y = (d-c)/n,$$
$$m = n = 3^k,\ k = 1,...,4.$$