

Καραμπετάκης Νικόλαος

Εισαγωγή στην Fortran 90/95

```
INTEGER FUNCTION MKD(A,B)
  IMPLICIT NONE
  INTEGER, INTENT(IN) :: A,B
  INTEGER :: ΣΥΖΩΛ, Ζ<A, Ζ<B
  DO WHILE (ΣΥΖΩΛ .NE. 0)
    IF (A .GT. B) THEN
      A = A - B
    ELSE
      B = B - A
    END IF
    ΣΥΖΩΛ = A - B
  END DO
  RETURN A
END FUNCTION MKD
```

ΥΠΟΛΟΓΙΣΜΟΣ

ΕΚΔΟΣΕΙΣ ΖΗΤΗ

ΠΡΟΛΟΓΟΣ

Το βιβλίο αυτό γράφτηκε για τις ανάγκες του μαθήματος «Προγραμματισμός Η/Υ ΙΙ» που διδάσκεται στο Β' εξάμηνο του Τμήματος Μαθηματικών του Α.Π.Θ.. Κύριος στόχος του βιβλίου αυτού είναι να εξοικιώσει τον αρχάριο προγραμματιστή με τις βασικές αρχές του προγραμματισμού των Η/Υ χρησιμοποιώντας την γλώσσα προγραμματισμού FORTRAN 90 καθώς και στοιχεία της FORTRAN 95.

Για την καλύτερη κατανόηση της ύλης οι αναγνώστες παροτρύνονται

- α) να εκτελέσουν τα έτοιμα προγράμματα που παρουσιάζονται στο βιβλίο έτσι ώστε να αποκτήσουν μια εξοικίωση με τον προγραμματισμό σε Fortran,
- β) να προσπαθούν να επιλύουν τις δραστηριότητες αλλά και τις ασκήσεις επανάληψης που προτείνονται σε κάθε ενότητα που έχουν ως στόχο την ενεργό συμμετοχή τους και την εμπέδωση της ύλης. Σε περίπτωση αδυναμίας επίλυσης των δραστηριοτήτων θα πρότεινα μια καλή επανάληψη της ύλης που έχει σχέση με την επίλυση της δραστηριότητας.

Οι λύσεις των δραστηριοτήτων παρουσιάζονται στο τέλος κάθε ενότητας και χρησιμεύουν ως οδηγός αυτοαξιολόγησης για την πορείας εκμάθησης του αναγνώστη.

Θα ήθελα να ευχαριστήσω για την συγγραφή του βιβλίου αυτού τον Καθηγητή κ. Κωνσταντίνο Λάζο του Τμήματος Πληροφορικής του Α.Π.Θ., για τις πολύτιμες συμβουλές του όλα αυτά τα χρόνια που συνεργαστήκαμε, καθώς και την κ. Μπλέρη Καββαδία, τον κ. Γιώργο Ραχώνη και την κ. Κατερίνα Χατζηφωτεινού που συνεργαστήκαμε στενά για μεγάλο διάστημα στη διδασκαλία του μαθήματος «Προγραμματισμός Η/Υ ΙΙ». Θα ήθελα επίσης να ευχαριστήσω την εταιρεία Compaq και την υπεύθυνη του προγράμματος κ. C. Staudinger, οι οποίοι διέθεσαν δωρεάν την Compaq Visual Fortran στο Τμήμα Μαθηματικών για την κάλυψη του μαθήματος «Προγραμματισμός Η/Υ ΙΙ». Τα προγράμματα που παρουσιάζονται στο βιβλίο αυτό έχουν γραφεί και εκτελεστεί στην Compaq Visual Fortran. Τέλος, και πιο πολύ από όλους, θα ήθελα να ευχαριστήσω την γυναίκα μου Άννα και τα δύο παιδιά μου Θεοδώρα και Αθηνά για την υπομονή αλλά και συμπαράσταση που μου δείξανε κατά τη συγγραφή του βιβλίου αυτού.

ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή στους H/Y και στο Περιβάλλον της Compaq Visual Fortan1

1.1 Ποια είναι τα βασικά μέρη από τα οποία αποτελείται ένας H/Y ;	10
1.2 Ποια είναι η δομή του υλικού ενός H/Y ;	11
1.3 Ποιες είναι οι βασικές κατηγορίες λογισμικού ;	12
1.4 Τι είναι γλώσσα προγραμματισμού ;	14
1.5 Τι είναι προγραμματισμός ;	16
1.6 Πώς θα μεταφράσουμε και θα εκτελέσουμε ένα πρόγραμμα στη Compaq Visual Fortan.....	19
1.7 Σύνοψη.....	26
1.8 Απαντήσεις στις δραστηριότητες	26

2. Βασικά στοιχεία ενός προγράμματος στη Fortran 90

2.1 Πότε δημιουργήθηκε ή FORTRAN 90 ;	28
2.2 Ποιο είναι το αλφάβητο της FORTRAN 90 ;	29
2.3 Ποιο είναι το λεξιλόγιο της FORTRAN 90 ;	29
2.4 Ποια είναι τα δεδομένα που χειρίζεται ένα πρόγραμμα και σε ποιες κατηγορίες χωρίζονται;	36
2.5 Εκφράσεις στη Fortran 90	38
2.6 Συναρτήσεις.....	41
2.7 Σύνοψη.....	44
2.8 Απαντήσεις στις δραστηριότητες	45

3. Δομή ενός προγράμματος στην Fortan 90

3.1 Από τι αποτελείται ένα πρόγραμμα FORTRAN 90 ;	48
3.2 Πώς δηλώνουμε την επικεφαλίδα και το τέλος του προγράμματος;	48
3.3 Τμήμα δηλώσεων.....	48
3.4 Πώς δηλώνουμε το τμήμα των προτάσεων ;	54
3.5 Πώς δηλώνουμε το τμήμα των διαδικασιών και συναρτήσεων ;	55
3.6 Σύνοψη.....	56

4. Ο τελεστής ανάθεσης και οι εντολές εισόδου-εξόδου

4.1 Ο τελεστής ανάθεσης.....	58
4.2 Η εντολή εισόδου READ.....	60
4.3 Οι εντολές εξόδου PRINT-WRITE.....	64
4.4 Απλά προβλήματα εισόδου-εξόδου.....	69
4.5 Παρατηρήσεις στις εντολές εισόδου-εξόδου.....	74
4.6 Σύνοψη.....	75
4.7 Επαναληπτικές ασκήσεις.....	76
4.8 Απαντήσεις στις δραστηριότητες	77

5. Εντολές συνθήκης και διακλάδωσης

5.1 Η εντολή GOTO	80
5.2 Η εντολή IF-THEN-ELSE	82
5.3 Η εντολή SELECT-CASE	86
5.4 Παραδείγματα με συνθήκες.....	89
5.5 Σύνοψη.....	99
5.6 Επαναληπτικές ασκήσεις.....	100
5.7 Απαντήσεις στις Δραστηριότητες.....	102

6. Οι εντολές επανάληψης

6.1 Ανακύκλωση και χρήση μετρητών σε προγράμματα.....	106
6.2 Η εντολή DO.....	115
6.3 Σύνοψη.....	136
6.4 Επαναληπτικές ασκήσεις.....	137
6.5 Απαντήσεις στις δραστηριότητες	142

7. Πίνακες

7.1 Τι είναι οι πίνακες και πού χρειάζονται ;	149
7.2 Πίνακες στην Fortran 90.....	151
7.3 Μονοδιάστατοι πίνακες	152
7.4 Πολυδιάστατοι πίνακες.....	183
7.5 Σύνοψη.....	201
7.6 Επαναληπτικές ασκήσεις.....	202
7.7 Απαντήσεις στις δραστηριότητες	207

8. Διαδικασίες και συναρτήσεις

8.1 Τι είναι το υποπρόγραμμα και ποια είδη υποπρογραμμάτων υπάρχουν ;	225
8.2 Συναρτήσεις	227
8.3 Διαδικασίες	240
8.4 Τι είναι τα Modules ;	253
8.5 Συναρτήσεις – Διαδικασίες της αριθμητικής IMSL βιβλιοθήκης της Compaq Visual Fortran	257
8.6 Σύνοψη	262
8.7 Επαναληπτικές ασκήσεις	263
8.8 Απαντήσεις στις δραστηριότητες	267

9. Αρχεία

9.1 Τι είναι αρχείο ;	277
9.2 Αρχεία Σειριακής Προσπέλασης	279
9.3 Αρχεία άμεσης ή τυχαίας προσπέλασης	308
9.4 Σύνοψη	320
9.5 Επαναληπτικές ασκήσεις	321
9.6 Απαντήσεις στις Δραστηριότητες	324

10. Εφαρμογές

10.1 Επίλυση διαφορικών εξισώσεων	334
10.2 Επίλυση εξισώσεων διαφορών καθώς και συστημάτων εξισώσεων διαφορών	340
10.3 Σύνοψη	349
10.4 Ασκήσεις	350
10.4 Απαντήσεις στις Δραστηριότητες	352
Βιβλιογραφία	363
Παράρτημα Α	
Λογικό Διάγραμμα	365
Παράρτημα Β	
Συνοπτικός Πίνακας Εντολών	367
Ευρετήριο	377

1^ο

Κεφάλαιο

Εισαγωγή στους Η/Υ και στο Περιβάλλον της Compaq Visual Fortran

Σκοπός

Σκοπός του κεφαλαίου αυτού είναι να παρουσιάσει

- α) μια συνοπτική περιγραφή της δομής των Η/Υ
- β) μια διαδικασία επίλυσης προβλημάτων με την βοήθεια του Η/Υ και
- γ) το περιβάλλον της Compaq Visual Fortran.

Προσδοκώμενα αποτελέσματα

Έχοντας διαβάσει το κεφάλαιο αυτό θα είσαι σε θέση:

- να δώσεις τον ορισμό του υπολογιστή,
- να περιγράψεις τα δύο βασικά μέρη από τα οποία αποτελείται ο Η/Υ
- να δώσεις το βασικό ορισμό του υλικού (hardware) ενός Η/Υ, να περιγράψεις τη βασική δομή του καθώς και να δώσεις μια σύντομη περιγραφή των μερών που το απαρτίζουν,
- να δώσεις τον ορισμό του λογισμικού, καθώς και να περιγράψεις δύο βασικές κατηγορίες λογισμικού,
- να ορίσεις τι είναι γλώσσα προγραμματισμού και να περιγράψεις τις δύο μεγάλες κατηγορίες στις οποίες χωρίζονται οι γλώσσες προγραμματισμού,
- να περιγράψεις τα 7 στάδια του προγραμματισμού,
- να εισάγεις ένα πρόγραμμα γραμμένο σε FORTRAN 90 στο περιβάλλον της Compaq Visual Fortran και να το εκτελείς.

Έννοιες κλειδιά

- Υλικό (hardware)
- Λογισμικό (software)
- Λειτουργικό σύστημα
- Κύρια Μνήμη (RAM)
- Κεντρική μονάδα επεξεργασίας (CPU)
- Λογισμικό συστήματος
- Λογισμικό εφαρμογών
- Πηγαίος Κώδικας
- Αντικείμενο Πρόγραμμα

Εισαγωγικές Παρατηρήσεις

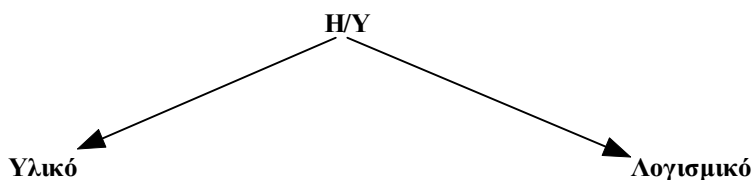
Στο κεφάλαιο αυτό θα εισάγουμε βασικές έννοιες που αφορούν τη δομή του Η/Υ και θα περιγράψουμε τα στάδια υλοποίησης ενός προγράμματος με τη βοήθεια της FORTRAN 90.

Πιο συγκεκριμένα, στην ενότητα 1.1 θα αναφέρουμε ότι ένας υπολογιστής αποτελείται από δύο βασικά μέρη : το υλικό (hardware) και το λογισμικό (software) και θα προχωρήσουμε στη συνοπτική περιγραφή τους στις ενότητες 1.2 και 1.3. Στην ενότητα 1.4 θα δούμε τι είναι γλώσσα προγραμματισμού και σε ποιες βασικές κατηγορίες μπορούμε να χωρίσουμε τις γλώσσες προγραμματισμού, ενώ στην ενότητα 1.5 θα αναφέρουμε τα 7 βασικά στάδια του προγραμματισμού. Τέλος, στην ενότητα 1.6 θα εξηγήσουμε πώς μπορεί να υλοποιηθεί στο περιβάλλον της Compaq Visual Fortran ένα πρόγραμμα που έχουμε γράψει στη FORTRAN 90.

1.1 Ποια είναι τα βασικά μέρη από τα οποία αποτελείται ένας Η/Υ ;

Ο ηλεκτρονικός υπολογιστής (Η/Υ) είναι μια ηλεκτρονική συσκευή, η οποία επεξεργάζεται με μεγάλη ακρίβεια και ταχύτητα έναν τεράστιο όγκο πληροφοριών. Η βασική του διαφορά από άλλες ηλεκτρονικές συσκευές βρίσκεται στη δυνατότητα προγραμματισμού του. Είναι δυνατό ο ίδιος Η/Υ, με τη βοήθεια διαφορετικών προγραμμάτων να είναι το ίδιο χρήσιμος σε ένα Φυσικό, σε ένα Μαθηματικό, σε ένα Μηχανικό κ.τ.λ.

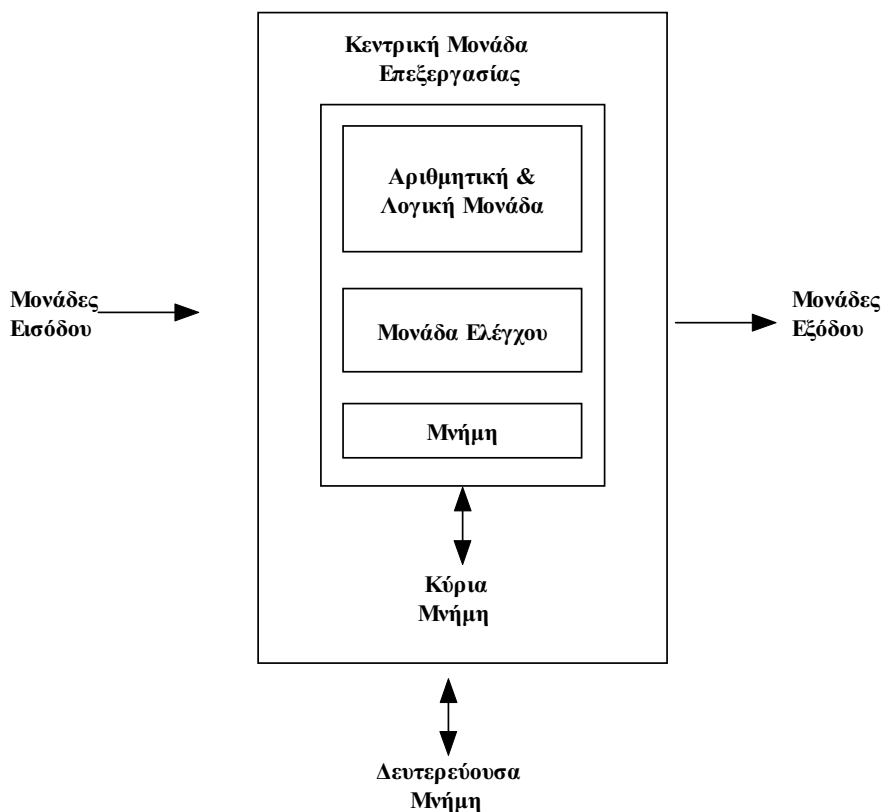
Ο Η/Υ αποτελείται από το *Hardware* ή υλικό και το *Software* ή λογισμικό. Με τον όρο *Hardware* αναφερόμαστε στις διάφορες μονάδες εισόδου, επεξεργασίας και εξόδου του Η/Υ, όπως για πρδ. το πληκτρολόγιο, η κεντρική μονάδα επεξεργασίας, η οθόνη κ.τ.λ., ενώ με τον όρο *Software* στα προγράμματα τα οποία κατευθύνουν τον Η/Υ. Τα προγράμματα αυτά αποτελούν ένα σύνολο από οδηγίες γραμμένες κατά τέτοιο τρόπο, ώστε να είναι άμεσα κατανοητές από τον Η/Υ.



Σχήμα 1.1 Τα μέρη του Η/Υ.

1.2 Ποια είναι η δομή του υλικού ενός Η/Υ ;

Στο παρακάτω σχήμα φαίνονται τα βασικά μέρη από τα οποία αποτελείται το υλικό του Η/Υ :



Σχήμα 2.2 Δομή του υλικού του Η/Υ

όπου :

Μονάδες Εισόδου : Είναι το σύνολο των μονάδων με τις οποίες επιτυγχάνεται η είσοδος των πληροφοριών στον Η/Υ, π.χ. πληκτρολόγιο, ποντίκι, κ.τ.λ.

Κύρια Μνήμη (RAM) : Η μονάδα αυτή χρησιμοποιείται για την αποθήκευση του προγράμματος, των δεδομένων, καθώς επίσης και των ενδιάμεσων και τελικών αποτελεσμάτων του προγράμματος, πριν αυτά εμφανιστούν στην οθόνη ή τυπωθούν στον εκτυπωτή ή γραφτούν σε κάποια μονάδα δευτερεύουσας μνήμης, όπως η μαγνητική δισκέτα ή ο μαγνητικός δίσκος.

Κεντρική Μονάδα Επεξεργασίας (CPU) : Η μονάδα στην οποία γίνονται οι κάθε είδους επεξεργασίες των πληροφοριών, καθώς και ο συντονισμός και έλεγχος των λειτουργιών του Η/Υ. Αποτελείται από την αριθμητική και λογική μονάδα, τη μονάδα ελέγχου και μονάδες μνήμης.

Μονάδες Εξόδου : Είναι το σύνολο των μονάδων με τις οποίες επιτυγχάνεται η έξοδος των αποτελεσμάτων από τον Η/Υ στον εξωτερικό κόσμο π.χ. οθόνη, εκτυπωτής, δισκέτα κ.τ.λ.

Μερικές μονάδες του Η/Υ, όπως οι δισκέτες και το modem, μπορούν να θεωρηθούν ως μονάδες εισόδου-εξόδου γιατί και δίνουν πληροφορίες στον Η/Υ και παίρνουν από αυτόν.

1.3 Ποιες είναι οι βασικές κατηγορίες λογισμικού ;

Ο Η/Υ αποτελεί μια ηλεκτρονική συσκευή, η οποία λειτουργεί μόνο κάτω από ένα σύνολο προγραμμάτων, τα οποία όλα μαζί αποτελούν το λογισμικό ή software του Η/Υ. Το λογισμικό του Η/Υ χωρίζεται σε δύο κατηγορίες :

- α) στο λογισμικό συστήματος και
- β) στο λογισμικό των εφαρμογών.

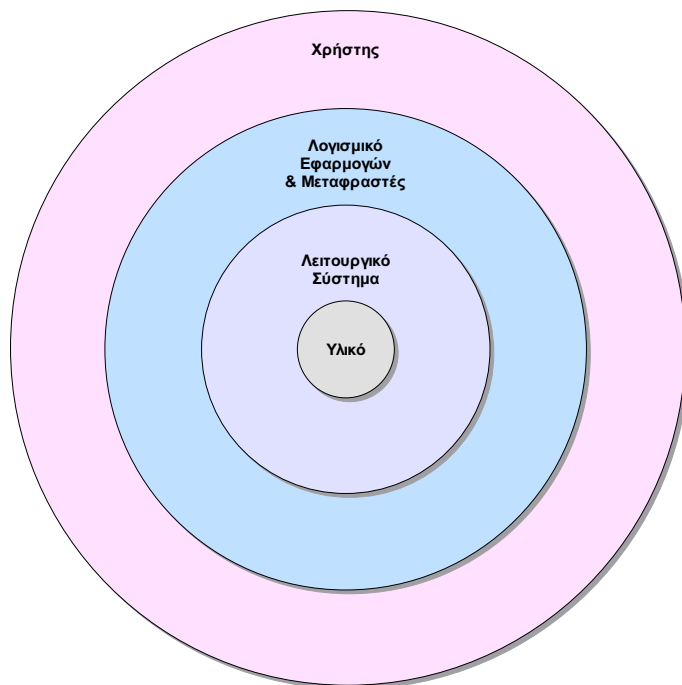


Σχήμα 1.3 Κατηγορίες λογισμικού

Λογισμικό Συστήματος : Αποτελείται από όλα εκείνα τα προγράμματα που εξασφαλίζουν τη σύνδεση όλων των μονάδων μεταξύ τους και τον έλεγχο των λειτουργιών τους. Το λογισμικό συστήματος περιλαμβάνει κυρίως το λειτουργικό σύστημα και τα μεταφραστικά προγράμματα.

Το *λειτουργικό σύστημα* (Λ.Σ.) είναι ο συνδετικός κρίκος μεταξύ του υλικού από τη μια μεριά και των προγραμμάτων εφαρμογής από την άλλη. Πρωταρχικός ρόλος του Λ.Σ. είναι η εύκολη και αποδοτική χρησιμοποίηση του υλικού από το χρήστη. Σε γενικές γραμμές το Λ.Σ. καθορίζει στον υπολογιστή πώς να καλεί κάποιο πρόγραμμα από μια μονάδα αποθήκευσης, πώς να αποθηκεύει δεδομένα σ' αυτές, πώς να χειρίζεται την οθόνη και τον εκτυπωτή, και σε γενικές γραμμές, αναλαμβάνει να συντονίζει τα διάφορα μέρη του υπολογιστή που απαιτούνται για την εκτέλεση μιας ολοκληρωμένης εργασίας. Το Λ.Σ. και η αρχιτεκτονική των Η/Υ έχουν επηρεάσει το ένα το άλλο σε μεγάλο βαθμό. Η ραγδαία εξέλιξη, λοιπόν, στο χώρο της αρχιτεκτονικής των Η/Υ οδήγησε και στην αντίστοιχη εξέλιξη του χώρου των λειτουργικών συστημάτων (αν και πολλές φορές συνέβη και το αντίθετο). Σήμερα υπάρχουν δύο κατηγορίες λειτουργικών συστημάτων :

Αυτά που μπορούν να εξυπηρετούν ένα χρήστη κάθε φορά και χρησιμοποιούνται στους μικροϋπολογιστές. Τέτοια λειτουργικά συστήματα είναι το MS-DOS, τα WINDOWS 3.1, τα WINDOWS 95, τα WINDOWS 98 κ.τ.λ.



Σχήμα 1.4 Σχέση Υλικού – Λογισμικού - Χρήστη.

Αυτά που έχουν την δυνατότητα να εξυπηρετούν συγχρόνως περισσότερους από ένα χρήστες (σε δίκτυα υπολογιστών). Τέτοια λειτουργικά συστήματα είναι το UNIX, τα WINDOWS 98, τα WINDOWS NT, τα WINDOWS 2000 κ.τ.λ.

Λογισμικό Εφαρμογών : Εδώ ανήκουν όλα τα προγράμματα ή σύνολα προγραμμάτων που γράφονται για να διεκπεραιώνουμε με τη βοήθεια του Η/Υ διάφορες εργασίες, πρδ. προγράμματα επεξεργασίας κειμένου, προγράμματα σχεδίασης, λογιστικά φύλλα κ.λ.π.

1.4 Τι είναι γλώσσα προγραμματισμού ;

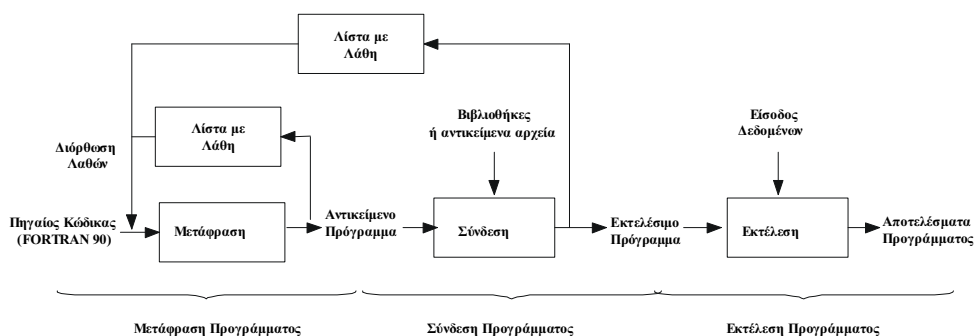
Η ανάγκη επικοινωνίας του ανθρώπου με τον Η/Υ σε μια κοινή γλώσσα ανέπτυξε τις διάφορες γλώσσες προγραμματισμού που υπάρχουν σήμερα. *Γλώσσα προγραμματισμού* είναι ένα σύνολο γραμμάτων, αριθμών, λέξεων και συντομογραφικών μνημονικών σημείων που διέπονται από ειδικό συντακτικό και χρησιμοποιούνται στην ανάπτυξη αλγορίθμων στον υπολογιστή. Η ποικιλία των γλωσσών προγραμματισμού που υπάρχει σήμερα οφείλεται στο ότι κάποιες γλώσσες υπερτερούν έναντι άλλων σε συγκεκριμένες εφαρμογές, ή προσφέρουν καλύτερη υποστήριξη, ή είναι ευκολότερες στην εκμάθησή τους. Γενικά, τις γλώσσες προγραμματισμού μπορούμε να τις χωρίσουμε σε δύο κατηγορίες :

Οι *γλώσσες χαμηλού επιπέδου* που βρίσκονται κοντά στις στοιχειώδεις εντολές μηχανής ενός συγκεκριμένου Η/Υ και γι' αυτό διαφέρουν από Η/Υ σε Η/Υ. Η γλώσσα που είναι πιο κοντά στο επίπεδο της μηχανής είναι η γλώσσα μηχανής. Η ανάγκη επίτευξης μια γλώσσας χαμηλού επιπέδου που να μοιάζει περισσότερο στη γλώσσα του ανθρώπου, χωρίς να ξεφεύγει από το επίπεδο της μηχανής, δημιούργησε τη γλώσσα Assembly. Υπάρχει μάλιστα ειδικό πρόγραμμα που ονομάζεται Assembler, το οποίο αναλαμβάνει να μεταφράσει τα προγράμματα που είναι γραμμένα στη γλώσσα αυτή, σε γλώσσα μηχανής.

Οι *γλώσσες υψηλού επιπέδου*, που προσεγγίζουν την ανθρώπινη γλώσσα περισσότερο από ότι οι γλώσσες χαμηλού επιπέδου και είναι ειδικά σχεδιασμένες, ώστε προγράμματα τα οποία είναι γραμμένα σ' αυτές να εκτελούνται από οποιονδήποτε υπολογιστή. Τέτοιες γλώσσες είναι οι BASIC, FORTRAN, PASCAL, PROLOG, C, κ.τ.λ. Το πρόγραμμα που γράφουμε σε μια γλώσσα υψηλού επιπέδου μεταφράζεται πάντοτε μέσω ειδικών προγραμμάτων στη γλώσσα που καταλαβαίνει ο Η/Υ, δηλαδή τη γλώσσα μηχανής. Τα προγράμματα που κάνουν τη μετάφραση αυτή είναι ο Interpreter ή ο Compiler.

Interpreter (Διερμηνέας)	Μεταφράζει μια-μια τις εντολές συγχρόνως με την εκτέλεση τους.
Compiler (Μεταφραστής)	Μεταφράζει μια φορά ολόκληρο το πρόγραμμα. Στη συνέχεια το μεταφρασμένο πρόγραμμα εκτελείται.

Η Fortran 90, που θα αναλύσουμε παρακάτω, διαθέτει μεταφραστή για τη μετατροπή του προγράμματος σε γλώσσα μηχανής.



Σχήμα 1.5 Φάση υλοποίησης ενός προγράμματος.

Ένα πρόγραμμα γραμμένο σε μια γλώσσα υψηλού επιπέδου ονομάζεται συνήθως *πηγαίος κώδικας* (source code). Μετά τη μετατροπή του σε γλώσσα μηχανής από το μεταφραστή (compiler) δημιουργείται το *αντικείμενο πρόγραμμα* (object program). Το αντικείμενο πρόγραμμα αποθηκεύεται σε αρχείο που έχει το όνομα του αρχείου του πηγαίου κώδικα και επέκταση .obj. Το αντικείμενο πρόγραμμα στη συνέχεια συνδέεται με βιβλιοθήκες της γλώσσας προγραμματισμού και άλλα αντικείμενα προγράμματα προκειμένου να δημιουργηθεί ο εκτελέσιμος κώδικας ή αλλιώς το εκτελέσιμο πρόγραμμα (executable program). Το εκτελέσιμο πρόγραμμα αποθηκεύεται σε αρχείο που έχει το όνομα του αρχείου του πηγαίου κώδικα και επέκταση .exe. Το παραπάνω διάγραμμα δείχνει την διαδικασία υλοποίησης ενός προγράμματος.



Δραστηριότητα 1.1 Να περιγράψετε τη διαδικασία μετάφρασης και εκτέλεσης ενός προγράμματος. Ισχύει η ίδια διαδικασία για προγράμματα που είναι γραμμένα σε γλώσσα μηχανής ;

1.5 Τι είναι προγραμματισμός ;

Προγραμματισμός είναι μια διαδικασία που αποτελείται από καθορισμένα στάδια, σε καθένα από τα οποία γίνονται διάφορες ενέργειες με σκοπό το μετασχηματισμό του προβλήματος και της μεθόδου λύσης του σε μια μορφή που να είναι κατανοητή και αποδεκτή από τον Η/Υ.

Τα στάδια του προγραμματισμού είναι τα ακόλουθα :

Φάση Ανάλυσης (Analysis)

- Η αναγνώριση, ο ορισμός και ο καθορισμός των προδιαγραφών του προβλήματος.
- Εντοπισμός των χρήσιμων εννοιών και απλοποίηση ορισμένων στοιχείων.

Φάση Σχεδιασμού (Design)

- Σκιαγράφηση της λύσης και διάκριση ανεξαρτήτων διαδικασιών.
- Επιλογή και περιγραφή ενός αλγόριθμου που να είναι πραγματοποιήσιμος, σαφής, γενικός, αποτελεσματικός, ταχύς και οικονομικός.

Φάση Υλοποίησης (Implementation)

- Επιλογή της γλώσσας προγραμματισμού βάσει της φύσης του προβλήματος, των γνώσεών μας σε γλώσσες προγραμματισμού και τις δυνατότητες ή περιορισμούς που μας θέτει ο Η/Υ. Κωδικοποίηση του αλγόριθμου σε πρόγραμμα.
- Έλεγχος προγράμματος για ανίχνευση λαθών. Τα λάθη χωρίζονται σε τρεις κατηγορίες : α) *συντακτικά λάθη* (syntax errors), λάθη δηλαδή ως προς τον τρόπο σύνταξης του προγράμματος στην συγκεκριμένη γλώσσα προγραμματισμού π.χ. REED αντί για READ, β) *λογικά λάθη* (logical errors), λάθη που δεν είναι άμεσα αντιληπτά και συνήθως οφείλονται σε λανθασμένη απόδοση του αλγορίθμου ή σε περιορισμούς της γλώσσας ή του συστήματος και οδηγούν σε λάθη εκτέλεσης (run time errors) π.χ. $X = -A/B$ χωρίς να ελέγξουμε αν $B \neq 0$ και γ) *λάθη εκτέλεσης*, λάθη που προκύπτουν κατά την διαδικασία ανάγνωσης/εγγραφής, χρήσης μη επιτρεπτών δεδομένων, χρήσης λανθασμένων ορισμάτων σε συναρτήσεις, αριθμητικών λαθών (π.χ. πολύ μεγάλοι αριθμοί) ή γενικά σε λάθη του συστήματος.
- Χρήση του προγράμματος με ποικίλα δεδομένα για να διαπιστωθούν τυχόν λάθη.

Παρακάτω δίνουμε ένα παράδειγμα όπου εφαρμόζουμε όλα τα στάδια του προγραμματισμού.

Παράδειγμα. Να υπολογιστεί το εμβαδόν ενός τριγώνου ΑΒΓ.

Φάση Ανάλυσης

- Διαπιστώνουμε ότι το παραπάνω πρόβλημα δεν είναι καλά ορισμένο, διότι θα μπορούσαμε να υπολογίσουμε το εμβαδόν του τριγώνου μόνο αν γνωρίζαμε τις 3 πλευρές του, ή τις 2 πλευρές και μια γωνία του, κ.ο.κ.. Επαναδιατυπώνουμε λοιπόν το πρόβλημα με σαφήνεια.
«Να υπολογιστεί το εμβαδόν ενός τριγώνου ΑΒΓ εάν είναι γνωστά τα μήκη των πλευρών του ΑΒ, ΒΓ, ΓΑ.»
- Ξεχωρίζουμε την *είσοδο* στο πρόβλημά μας, που είναι τα μήκη των τριών πλευρών, έστω a, b, c και η *έξοδος* στο πρόβλημα μας θα είναι το εμβαδόν του τριγώνου, έστω E .

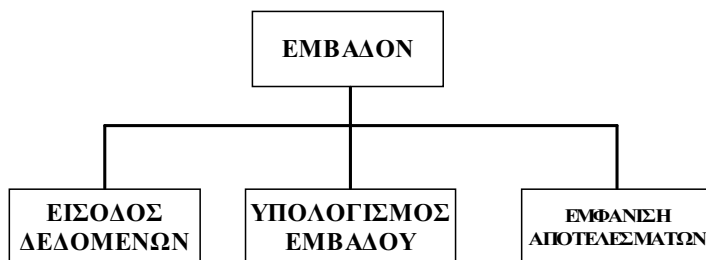
Φάση Σχεδιασμού

- Έχουμε 3 ανεξάρτητες διαδικασίες :

1^η διαδικασία. Είσοδος των δεδομένων.

2^η διαδικασία. Υπολογισμός του εμβαδού μέσω του τύπου του Ήρωνα.

3^η διαδικασία. Έξοδος των αποτελεσμάτων.



- Οι παραπάνω 3 διαδικασίες μπορούν να παρασταθούν μέσω του παρακάτω ψευδοκώδικα :

Ψευδοκώδικας

Βήμα 1^ο. Είσοδος των δεδομένων a, b, c από το χρήστη.

Βήμα 2^ο. Υπολογισμός του

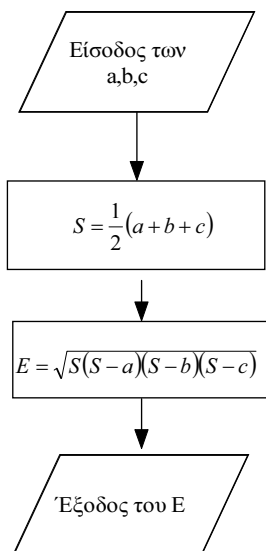
$$S = \frac{1}{2}(a + b + c)$$

Βήμα 3^ο. Υπολογισμός του εμβαδού

$$E = \sqrt{S(S-a)(S-b)(S-c)}$$

Βήμα 4^ο. Εμφάνιση των αποτελεσμάτων (E).

ή του παρακάτω λογικού διαγράμματος (δες Παράρτημα Α, Λογικό Διάγραμμα, σελ. 365)



Φάση Υλοποίησης

- Το πρόγραμμα αυτό είναι πολύ απλό και μπορεί να υλοποιηθεί σε οποιαδήποτε γλώσσα προγραμματισμού, χωρίς κάποια ιδιαίτερα πλεονεκτήματα έναντι κάποιας άλλης γλώσσας. Παρακάτω το πρόγραμμα υλοποιείται στη FORTRAN 90.

```

PROGRAM TEST
  IMPLICIT NONE
  REAL A,B,C,S,E

  READ*, A,B,C

  S=(1/2.0)*(A+B+C)
  E=SQRT(S*(S-A)*(S-B)*(S-C))
  PRINT*, '-----'
  PRINT*, A,B,C
  PRINT*, 'E=',E

```

END

Στη γραμμή 1 και 10 δηλώνεται η αρχή και το τέλος του προγράμματος. Στη γραμμή 3 δηλώνουμε ότι οι μεταβλητές που χρησιμοποιούμε στο πρόγραμμα θα πρέπει να δέχονται πραγματικές τιμές. Στη γραμμή 4 δεχόμαστε τιμές για τις μεταβλητές A,B,C από το χρήστη. Στις γραμμές 5 και 6 υπολογίζουμε το εμβαδόν E, ενώ στις γραμμές 7-9 εκτυπώνουμε τα αποτελέσματα.

- Μεταφράζουμε το πρόγραμμα και εντοπίζουμε τα συντακτικά λάθη, και στη συνέχεια εκτελούμε το πρόγραμμα για να βρούμε τυχόν λογικά λάθη ή λάθη εκτέλεσης. Πράγματι, διαπιστώνουμε λάθη όταν οι τιμές των A,B,C δεν αποτελούν πλευρές τριγώνου. Στην περίπτωση αυτή, η υπόριζη ποσότητα είναι αρνητική και μας οδηγεί σε λάθη εκτέλεσης. Θα έπρεπε να έχουμε έναν επιπλέον έλεγχο για τις πλευρές του τριγώνου. Σε παρακάτω ενότητες θα δούμε την πλήρη επίλυση του προβλήματος αυτού.

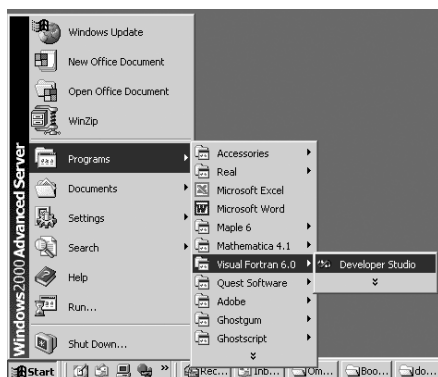
1.6 Πώς θα μεταφράσουμε και θα εκτελέσουμε ένα πρόγραμμα στη Compaq Visual Fortran

Παρακάτω δίνουμε όλα τα βήματα που πρέπει να κάνετε για να εισάγετε ένα πρόγραμμα που έχετε γράψει στο περιβάλλον της Compaq Visual Fortran, να το μεταφράσετε και τέλος να το εκτελέσετε.

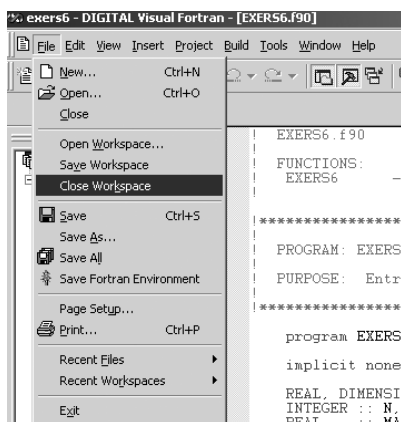
Βήμα 1. Εκκίνηση προγράμματος

Από την επιλογή *Έναρξη (Start)* διαλέγουμε *Προγράμματα (Programs)* στη συνέχεια *Visual Fortran 6.0* και τέλος *Developer Studio*

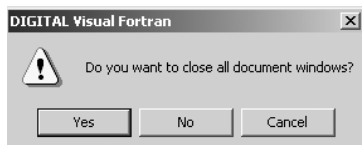
Εναλλακτική λύση: Σε περίπτωση που υπάρχει το εικονίδιο του Microsoft Developer Studio στην επιφάνεια εργασίας, κάνουμε διπλό κλικ επάνω του.



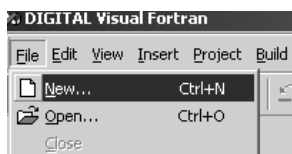
Βήμα 2. Από την επιλογή *File* διαλέγουμε *Close Workspace*. Σε περίπτωση που η επιλογή αυτή είναι απενεργοποιημένη, συνεχίζουμε με το επόμενο βήμα.



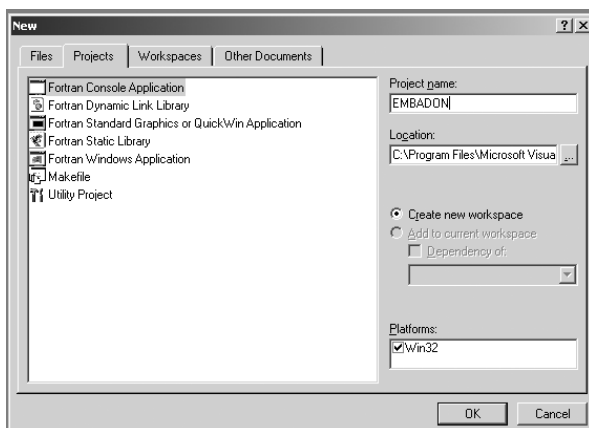
Απαντούμε με Yes στο παρακάτω ερώτημα :



Βήμα 3. Από την επιλογή File διαλέγουμε New



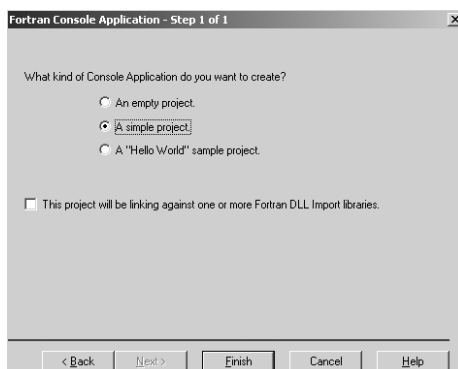
Στο παρακάτω πλαίσιο :



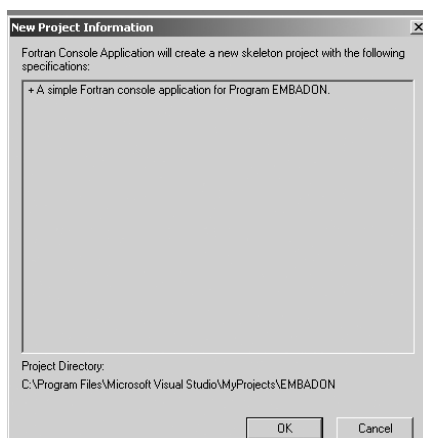
επιλέγουμε *Fortran Console Application*. Στο πλαίσιο *Project Name* συμπληρώνουμε

το όνομα του Project Workspace (π.χ. EMBADON) ενώ στο *Location* συμπληρώνουμε τον κατάλογο και την δευτερεύουσα μονάδα μνήμης που θέλουμε να δημιουργηθεί το Project Workspace (π.χ. A: αν θέλουμε να αποθηκευτεί στη δισκέτα. Τέλος κάνουμε κλικ στο OK. Το Project Workspace δημιουργείται για να τοποθετήσουμε το σύνολο των αρχείων που θα χρειαστούμε στο πρόγραμμα μας. Αν θέλουμε οι εφαρμογές μας να περιέχουν και γραφικά ή να εκτελούνται σε παράθυρο των Windows θα πρέπει να επιλέξουμε μια διαφορετική κατηγορία (δες οδηγό βοήθειας). Στις πρώτες ενότητες που θα μελετήσουμε θα έχουμε ένα μόνο αρχείο που θα είναι το πρόγραμμα μας. Στη συνέχεια, όταν θα αναφερθούμε σε συναρτήσεις/διαδικασίες/modules θα δούμε ότι το πρόγραμμα μας μπορεί να συνδέεται με εξωτερικές συναρτήσεις/διαδικασίες/modules. Στις περιπτώσεις αυτές το Project Workspace θα αποτελείται από περισσότερα από ένα αρχεία, όπως το κυρίως πρόγραμμα και τα αρχεία που περιέχουν τις εξωτερικές συναρτήσεις/διαδικασίες/modules.

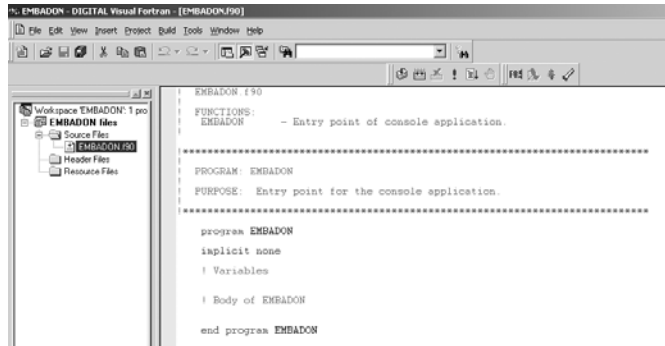
Βήμα 4. Στο πλαίσιο που εμφανίζεται επιλέγουμε το *A simple project* και στη συνέχεια κάνουμε κλικ στο *Finish*.



Βήμα 5. Στο πλαίσιο που εμφανίζεται επιλέγουμε OK.



Βήμα 6. Παρατηρούμε ότι έχει δημιουργηθεί ένα αρχείο με το όνομα EMBADON.F90 στο οποίο και θα τοποθετήσουμε το πρόγραμμα μας.



Βήμα 7. Θα γράψουμε το πρόγραμμα μας

```

EMBADON.F90
FUNCTIONS:
  EMBADON      - Entry point of console application.

*****

PROGRAM: EMBADON
PURPOSE:  Entry point for the console application.
*****

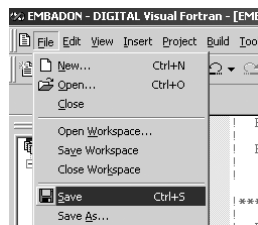
program EMBADON
implicit none
! Variables
REAL A,B,C,S,E
! Body of EMBADON
READ*, A,B,C
S=(1/2)*(A+B+C)
E=SQRT(S*(S-A)*(S-B)*(S-C))

PRINT*, "-----"
PRINT*, A,B,C
PRINT*, "E=", E

end program EMBADON

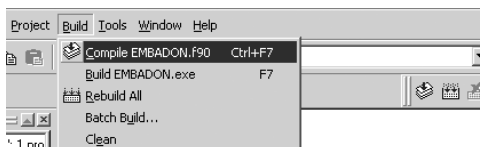
```

Βήμα 8. Αποθηκεύουμε το πρόγραμμά μας. Από την επιλογή File διαλέγουμε Save

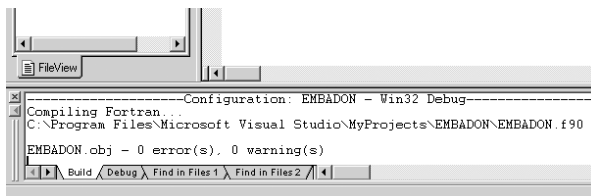


Βήμα 9. Μετάφραση του προγράμματος

Από την επιλογή Build επιλέγουμε Compile EMBADON.f90 (ή Ctrl+F7 ή το αντίστοιχο εικονίδιο από την μπάρα με τα εικονίδια)

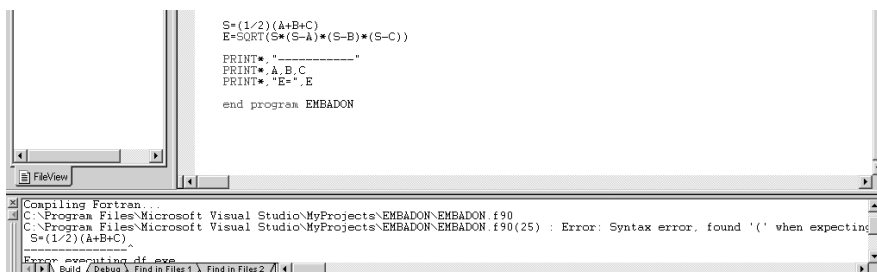


Αν όλα πάνε καλά, δηλαδή δεν βρεθεί κάποιο συντακτικό λάθος τότε θα εμφανισθεί στο τέλος της οθόνης μας το παρακάτω παράθυρο :

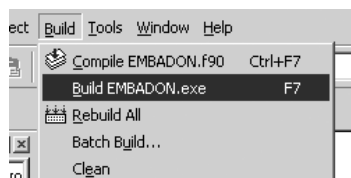


το οποίο δηλώνει ότι δε βρέθηκε κανένα συντακτικό λάθος.

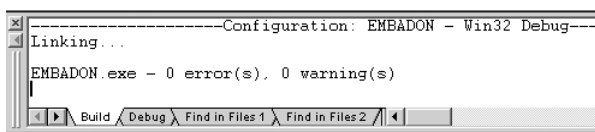
Σε περίπτωση που υπήρχαν λάθη, θα έπρεπε να τα εντοπίσουμε από τα μηνύματα του παραπάνω παράθυρου, να τα διορθώσουμε και να προσπαθήσουμε να ξαναμεταφράσουμε το πρόγραμμά μας. Αν παραλείπαμε το * στην 25 γραμμή του προγράμματος και προσπαθούσαμε να κάνουμε compile θα είχαμε το παρακάτω μήνυμα

**Βήμα 10. Δημιουργία εκτελέσιμου προγράμματος**

Εφόσον έχει μεταφρασθεί χωρίς λάθη το πρόγραμμά μας, διαλέγουμε από το Build την επιλογή Build EMBADON.exe (ή F7 ή το αντίστοιχο εικονίδιο του BUILD από την μπάρα με τα εικονίδια)



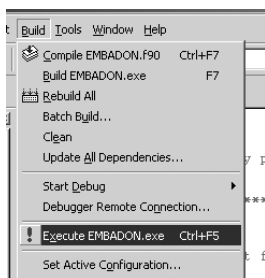
Αν όλα πάνε καλά, χωρίς λάθη έχουμε το παρακάτω μήνυμα :



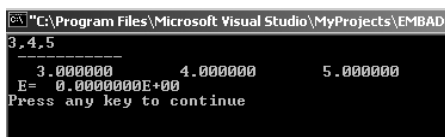
Αν έχουμε λάθη, τα διορθώνουμε και ξαναεκτελούμε τα τελευταία 2 βήματα. Σε περίπτωση που είμαστε σίγουροι ότι δεν έχουμε λάθη, μπορούμε να παραλείψουμε το βήμα 9 και να εκτελέσουμε κατευθείαν το βήμα 10, το οποίο θα εκτελέσει και τα δύο βήματα.

Βήμα 11. Εκτέλεση του προγράμματος

Από την επιλογή Build διαλέγουμε Execute EMBADON.exe (ή Ctrl+F5 ή επιλέγουμε το αντίστοιχο εικονίδιο ! από την μπάρα εικονιδίων)



Εμφανίζεται ένα παράθυρο του MS-DOS όπου και εκτελείται το πρόγραμμά μας :



Η παραπάνω διαδικασία αφορά τα απλά προγράμματα με τα οποία θα ασχοληθούμε στη συνέχεια. Για περισσότερες λεπτομέρειες ο αναγνώστης μπορεί να ανατρέξει στον οδηγό βοήθειας του αντίστοιχου προγράμματος.



Δραστηριότητα 1.2 Προσπάθησε να γράψεις, να μεταφράσεις και να εκτελέσεις το παρακάτω πρόγραμμα στο περιβάλλον της Compaq Visual Fortran :

```
PROGRAM TOKOS
  IMPLICIT NONE
  INTEGER N
  REAL K,E,TK
  PRINT*,'GIVE ME THE INITIAL VALUE'
  READ*, K
  PRINT*,'GIVE ME THE YEARS'
  READ*,N
  PRINT*,'GIVE ME THE PERCENTAGE'
  READ*,E
  TK=K*(1+E)**N
  PRINT*,'THE FINAL VALUE AFTER',N,'YEARS WILL BE',TK
END
```

1.7 Σύνοψη

Ο υπολογιστής αποτελείται από το υλικό και το λογισμικό. Η δομή του υλικού περιγράφηκε στην ενότητα 1.2, ενώ οι βασικές κατηγορίες λογισμικού περιγράφηκαν στην ενότητα 1.3. Στην ενότητα 1.4 προσπαθήσαμε να περιγράψουμε τι είναι γλώσσα προγραμματισμού, καθώς και τις 2 βασικές κατηγορίες στις οποίες χωρίζονται οι γλώσσες προγραμματισμού. Η έννοια του προγραμματισμού αναλύθηκε επιγραμματικά στην ενότητα 1.5 και υλοποιήθηκαν τα στάδια του μέσω ενός απλού παραδείγματος. Η διαδικασία μετάφρασης και εκτέλεσης ενός προγράμματος στο περιβάλλον της Compaq Visual Fortran αναλύθηκε στην ενότητα 1.6.

1.8 Απαντήσεις στις δραστηριότητες

Ενδεικτική απάντηση Δραστηριότητας 1.1

Η διαδικασία είναι αυτή που περιγράφεται στο σχήμα 5. Στην περίπτωση που το πρόγραμμα ήταν γραμμένο σε γλώσσα μηχανής, δεν θα χρειαζόταν η διαδικασία μετάφρασης, η οποία ουσιαστικά μεταφράζει το πρόγραμμα σε γλώσσα μηχανής.

Ενδεικτική απάντηση Δραστηριότητας 1.2

Ακολουθούμε τη διαδικασία που περιγράφηκε στην ενότητα 1.6.

2°

Κεφάλαιο

Βασικά στοιχεία ενός προγράμματος στη Fortran 90

Σκοπός

Σκοπός του κεφαλαίου αυτού είναι να παρουσιάσει

- α) μια συνοπτική ιστορία της γλώσσας προγραμματισμού Fortran
- β) το αλφάβητο και το λεξιλόγιο της Fortran 90,
- γ) τις δομές δεδομένων που χρησιμοποιούμε σε ένα πρόγραμμα
- δ) τον τρόπο δημιουργίας και εκτέλεσης εκφράσεων και τέλος
- ε) στοιχειώδεις συναρτήσεις της Fortran 90.

Προσδοκώμενα Αποτελέσματα

Έχοντας διαβάσει το κεφάλαιο αυτό θα είσαι σε θέση:

- να περιγράψεις το αλφάβητο της Fortran 90,
- να αναφέρεις τις 9 κατηγορίες που απαρτίζουν το λεξιλόγιο της Fortran 90,
- να διακρίνεις αν ένα συγκεκριμένο όνομα αποτελεί όνομα της Fortran 90,
- να χρησιμοποιείς τους τελεστές της Fortran 90 και της ενσωματωμένες συναρτήσεις της για να δημιουργείς εκφράσεις,
- να περιγράψεις τις δύο κατηγορίες στις οποίες χωρίζονται τα δεδομένα,
- να δώσεις τον ορισμό της μεταβλητής,
- να περιγράψεις τους τύπους δεδομένων που υποστηρίζει η Fortran 90,
- να υπολογίσεις το αποτέλεσμα μιας έκφρασης, χρησιμοποιώντας σωστά την προτεραιότητα των πράξεων.

Έννοιες Κλειδιά

- Πραγματικοί αριθμοί σταθερής (κινητής) υποδιαστολής
- Σταθερές (constants)
- Μεταβλητές (variables)
- Τύποι δεδομένων

Εισαγωγικές Παρατηρήσεις

Στο κεφάλαιο αυτό θα ξεκινήσουμε με μια σύντομη αναφορά στην ιστορία της γλώσσας προγραμματισμού Fortran.

Είναι γνωστό ότι σε κάθε φυσική γλώσσα, όπως στην καθομιλουμένη, έχουμε ένα αλφάβητο, ένα σύνολο ψηφίων και ένα σύνολο λέξεων-αριθμών που προκύπτει από συνδυασμούς των γραμμάτων του αλφαβήτου και των γνωστών ψηφίων. Το ίδιο συμβαίνει και σε κάθε τεχνητή γλώσσα, όπως η Fortran 90. Στην ενότητα 2.2 θα περιγράψουμε το αλφάβητο και στην ενότητα 2.3 το λεξιλόγιο της Fortran 90. Στην ενότητα 2.4 θα περιγράψουμε τους τύπους δεδομένων που υποστηρίζει η Fortran 90 ενώ στην ενότητα 2.5 θα αναφέρουμε τι είναι έκφραση και με ποιο τρόπο υπολογίζω το αποτέλεσμα μιας έκφρασης. Τέλος, στοιχειώδεις τύποι συναρτήσεων που είναι χρήσιμοι στην περιγραφή πολύπλοκων αριθμητικών και αλφαριθμητικών εκφράσεων θα περιγραφούν στην ενότητα 2.5.

2.1 Πότε δημιουργήθηκε η FORTRAN 90 ;

Η Fortran ήταν μια από τις πρώτες γλώσσες υψηλού επιπέδου που δημιουργήθηκε για πρώτη φορά το χρονικό διάστημα 1954-1957 από μια ομάδα της IBM υπό την καθοδήγηση του John Backus. Το 1966 δημιουργήθηκε η πρώτη standard έκδοση της (FORTRAN 66) η οποία εφαρμόστηκε ευρέως στη διδασκαλία και η οποία είχε το πλεονέκτημα της χρήσης υπορουτίνων, της ανεξάρτητης μετάφρασης και της ανεξαρτησίας από το συγκεκριμένο μηχάνημα. Το 1978 μετά από πιέσεις για ανταγωνισμό από καινούριες γλώσσες προγραμματισμού, όπως η PASCAL, ADA, Modula 2, C, C++, δημιούργησε μια καινούρια standard έκδοση, τη FORTRAN 77, η οποία παρ' όλα αυτά δε μπόρεσε να ανταποκριθεί ικανοποιητικά έναντι των άλλων γλωσσών. Η επόμενη standard έκδοσή της ήρθε το 1991 (FORTRAN 90), όπου υπήρξαν ουσιαστικές βελτιώσεις, τις οποίες θα δούμε στις παρακάτω ενότητες. Όχι ιδιαίτερα σημαντικές βελτιώσεις παρουσιάστηκαν το 1997 στην επόμενη έκδοση της FORTRAN 95, της οποίας βασικά στοιχεία θα δούμε επίσης στη συνέχεια. Κύρια νέα χαρακτηριστικά της Fortran 90, 95 σε αντίθεση με την Fortran 77 είναι :

- ο τρόπος γραφής του προγράμματος που πλέον θυμίζει όλες τις άλλες γλώσσες προγραμματισμού,
- η ευκολία στην διαχείριση πινάκων (προσθέσεις, αφαιρέσεις, πολλαπλασιασμοί πινάκων, χρήση υποπινάκων, μαζική επεξεργασία στοιχείων πινάκων, συναρτήσεις πινάκων κ.λ.π.),
- η δυναμική χρήση της μνήμης του H/Y (δυναμικοί πίνακες, χρήση δεικτών),
- δυνατότητα ορισμού νέων τύπων δεδομένων (για την δημιουργία λογικών εγγραφών),
- χρήση αναδρομικών συναρτήσεων και διαδικασιών,
- δυνατότητα δημιουργίας βιβλιοθηκών (Modules) από συναρτήσεις, διαδικασίες, ορισμούς μεταβλητών και τύπων κ.λ.π.,

- νέες δομές ελέγχου (DO-END DO, DO – WHILE, EXIT, CYCLE, SELECT CASE) κ.α..

Τη στιγμή αυτή βρίσκεται υπό κατασκευή η FORTRAN 2000.

2.2 Ποιο είναι το αλφάβητο της FORTRAN 90 ;

Όπως κάθε φυσική γλώσσα έχει το δικό της αλφάβητο, έτσι και η Fortran 90 έχει το δικό της αλφάβητο που παρουσιάζεται παρακάτω :

- Γράμματα : Κεφαλαία και μικρά Αγγλικά γράμματα
A,B,...,Z,a,b,...,z
- Ψηφία
0,1,2,3,4,5,6,7,8,9
- Ειδικοί χαρακτήρες
Κενό ‘ “ () * + - / : = _ ! & \$; < > % ? , .

2.3 Ποιο είναι το λεξιλόγιο της FORTRAN 90 ;

Το λεξιλόγιο της Fortran 90 περιλαμβάνει 9 κατηγορίες : ονόματα, αριθμούς, χαρακτήρες, αλυσίδες χαρακτήρων, λογικοί χαρακτήρες, τελεστές, διαχωριστές, λέξεις κλειδιά και σχόλια.

2.3.1 Ονόματα

Τα ονόματα καθορίζονται από τον προγραμματιστή για τον προσδιορισμό διαφόρων ποσοτήτων σε ένα πρόγραμμα : προσδιορισμό μεταβλητών ή σταθερών, ονομασία υπορουτίνων, προγραμμάτων κ.λ.π. Τα ονόματα θα πρέπει να ακολουθούν συγκεκριμένους κανόνες :

ΕΠΙΤΡΕΠΤΑ ΟΝΟΜΑΤΑ	ΜΗ ΕΠΙΤΡΕΠΤΑ ΟΝΟΜΑΤΑ
<ul style="list-style-type: none"> – Περιέχουν 1-31 αλφαριθμητικούς χαρακτήρες (τα 26 γράμματα της αγγλικής αλφαβήτου, τα 10 ψηφία και το σύμβολο _). – Ξεκινούν πάντα με αγγλικό γράμμα και στη συνέχεια ακολουθεί συνδυασμός γραμμάτων και αριθμών. – Κεφαλαία ή πεζά θεωρούνται το ίδιο. – Επιτρέπεται η χρήση ονομάτων που έχουν ειδική σημασία για τη FORTRAN 90 αν και δεν συνιστάται η χρήση τους. 	<ul style="list-style-type: none"> – Δεν επιτρέπονται τα σύμβολα εκτός από το _. – Δεν επιτρέπονται τα κενά. – Δεν επιτρέπονται τα Ελληνικά γράμματα.

Παράδειγμα

Όνομα	Σωστό/Λάθος
X-Y	Λάθος γιατί χρησιμοποιεί μη επιτρεπτά σύμβολα (-).
1AB	Λάθος γιατί ξεκινάει από αριθμό.
HELLO WORLD	Λάθος γιατί περιέχει κενά.
Pay_Day	Σωστό γιατί χρησιμοποιεί επιτρεπτό σύμβολο. Το όνομα PAY_DAY είναι το ίδιο.
Velocity	Σωστό γιατί χρησιμοποιεί από 1-31 αλφαριθμητικούς χαρακτήρες και ξεκινάει από αγγλικό γράμμα.
ΝΙΚΟΣ	Λάθος γιατί περιέχει Ελληνικά γράμματα.

2.3.2 Αριθμοί

Οι αριθμοί που υποστηρίζει η Fortran 90 είναι : ακέραιοι, πραγματικοί και μιγαδικοί:

- **Ακέραιοι αριθμοί.**

Πρόκειται για ένα σύνολο ψηφίων με ένα πρόσημο πριν από αυτά πρδ. -4, +555, 78900. Στην περίπτωση που το πρόσημο είναι + είναι προαιρετικό. Δεν θα πρέπει να χρησιμοποιούμε (.) ή (.) πρδ. 4.5, 5,67, παραπάνω από ένα πρόσημο ή να τοποθετούμε το πρόσημο δεξιά του αριθμού. Το μέγεθος των ακεραίων, ανάλογα με τη δήλωση τους είναι :

$$\left[-2^7, 2^7 - 1\right] \left[-2^{15}, 2^{15} - 1\right] \left[-2^{31}, 2^{31} - 1\right]$$

ενώ ο αριθμός των *σημαντικών*¹ τους ψηφίων είναι το πολύ 10.

Επιτρεπτοί ακέραιοι	Μη επιτρεπτοί ακέραιοι	Αιτιολογία
23	4.0	Δεν πρέπει να υπάρχει υποδιαστολή
+45	5,6	Δεν πρέπει να υπάρχει κόμμα
2.	9999999999	Πολύ μεγάλος αριθμός
0.	++1	Παραπάνω από ένα πρόσημο

1 Με τον όρο αυτό εννοούμε όλα τα ψηφία του αριθμού, εκτός από τυχόν μηδενικά που υπάρχουν μπροστά από το πρώτο μηδενικό ψηφίο ή μηδενικά που υπάρχουν μετά το τελευταίο μηδενικό ψηφίο που υπάρχει μετά από την υποδιαστολή.

-1234	5—	Το πρόσημο δεξιά του αριθμού
-------	----	------------------------------

• Πραγματικοί αριθμοί

Πρόκειται για θετικούς ή αρνητικούς αριθμούς έξω από το διάστημα των ακεραίων ή με δεκαδικά ψηφία. Ανάλογα με το μέγεθος τους διακρίνονται σε πραγματικούς αριθμούς *απλής ακρίβειας* $[-10^{38}, -10^{-38}]$ $[10^{-38}, 10^{38}]$ και πραγματικούς αριθμούς *διπλής ακρίβειας* $[-10^{308}, -10^{-308}]$ $[10^{-308}, 10^{308}]$.

Το πλήθος των *σημαντικών* ψηφίων των πραγματικών αριθμών απλής ακρίβειας είναι 6 ενώ των διπλής ακρίβειας είναι 15. πρδ. Ο πραγματικός αριθμός 123456789.12 θα παρασταθεί στη μνήμη του H/Y ως 123456000.00 αν είναι πραγματικός αριθμός απλής ακρίβειας ή 123456789.12 αν είναι πραγματικός αριθμός διπλής ακρίβειας.

Οι πραγματικοί αριθμοί χωρίζονται σε δύο κατηγορίες :

1. *Πραγματικοί αριθμοί σταθερής υποδιαστολής* (θετικοί ή αρνητικοί αριθμοί που περιέχουν υποδιαστολή) π.χ. 34.5
2. *Πραγματικοί αριθμοί κινητής υποδιαστολής* (θετικοί ή αρνητικοί αριθμοί που είναι εκφρασμένοι σε εκθετική μορφή $\underbrace{xxx}_\text{βάση} E \pm \underbrace{xxx}_\text{εκθέτης}$ όπου η βάση είναι θετικός ή αρνητικός ακέραιος ή δεκαδικός αριθμός, ενώ ο εκθέτης είναι θετικός ή αρνητικός ακέραιος αριθμός) π.χ.

Ο αριθμός $0.345E + 2$ συμβολίζει τον 34.5

Ο αριθμός $0.345E - 2$ συμβολίζει τον 0.00345

Επιτρεπτοί πραγματικοί	Μη επιτρεπτοί πραγματικοί	Αιτιολογία
1.23	5,47	Δεν πρέπει να υπάρχει κόμμα
-12E+3	12.4E+1.2	Ο εκθέτης δεν είναι ακέραιος
1E+34	3.	Δεν έχει δεκαδικά ψηφία
9878.567	1.234.567	Περιέχει δύο υποδιαστολές



Δραστηριότητα 2.1 Ποιοι από τους παρακάτω αριθμούς είναι ίδιοι ;
18.9 ; 10 ; 0.999 ; 0.189E+01 ; 0.189E+02 ; 0.189E-02 ; 0.1E+02 ;
1E+02 ; 9.99E-02 ; 9.99E-01

- **Μιγαδικοί αριθμοί**

Πρόκειται για ένα ζεύγος αριθμών, πρδ. η παράσταση του μιγαδικού αριθμού $3.0 + 4.0I$ είναι (3.0, 4.0). Το πρώτο μέρος του ζεύγους είναι το πραγματικό μέρος, ενώ το δεύτερο μέρος είναι το φανταστικό μέρος. Οι μιγαδικοί αριθμοί χωρίζονται σε δύο κατηγορίες :

- α) τους μιγαδικούς αριθμούς απλής ακριβείας, όπου τα μέλη του ζεύγους είναι και τα δύο πραγματικοί αριθμοί απλής ακριβείας και
- β) τους μιγαδικούς αριθμούς διπλής ακριβείας, όπου τα μέλη του ζεύγους είναι και τα δύο πραγματικοί αριθμοί διπλής ακριβείας.

2.3.3 Χαρακτήρες

Όλοι οι χαρακτήρες που υποστηρίζονται από τον H/Y μέσα σε αποστροφούς πρδ. "a", "γ". Ισχύουν τα εξής :

1. Για να δηλώσουμε το χαρακτήρα απόστροφο πρέπει να τον γράψουμε δύο φορές μεταξύ αποστροφών δηλ. "".
2. Ο χαρακτήρας "A" είναι διαφορετικός από το χαρακτήρα "a".
3. Ο κενός χαρακτήρας συμβολίζεται με δύο αποστροφούς που περικλείουν ένα διάστημα, δηλαδή " ".
4. Ο χαρακτήρας "7" διαφέρει από τον αριθμό 7. Ο πρώτος δε μπορεί να πάρει μέρος σε αριθμητικές πράξεις ενώ ο δεύτερος μπορεί.
5. Πρέπει να δοθεί προσοχή στο διαχωρισμό του μηδέν 0 και του κεφαλαίου όμικρον O.

2.3.4 Αλυσίδες χαρακτήρων

Πρόκειται για ακολουθία χαρακτήρων μεταξύ αποστροφών πρδ. "Fortran 90", "123". Για να τοποθετήσουμε απόστροφο σε μια αλυσίδα χαρακτήρων θα πρέπει να τοποθετήσουμε στη θέση της αποστροφού δύο αποστροφούς, πρδ. "don't forget" έχει περιεχόμενο don't forget.

Επιτρεπτές αλυσίδες χαρακτήρων	Μη επιτρεπτές αλυσίδες χαρακτήρων	Αιτιολογία
"NIKOS"	'NIKOS	Λείπει η απόστροφος.
'ANNA'	'don't forget'	Θέλει διπλά εισαγωγικά μετά το o
' '	'Hello'	Δεν επιτρέπεται η ανάμιξη διπλών εισαγω-

2.3.5 Λογικοί χαρακτήρες

Πρόκειται για δεδομένα τα οποία αποτελούνται από δύο μόνο τιμές .TRUE. και .FALSE. .

2.3.6 Τελεστές

Είναι σύμβολα που δηλώνουν πράξεις μεταξύ τελεστέων, δηλαδή αριθμών, αλυσίδων χαρακτήρων κ.λ.π.. Υπάρχουν 4 κατηγορίες τελεστών : αριθμητικοί, χαρακτήρων, σύγκρισης και λογικοί :

Τελεστές αριθμητικοί

Χρησιμοποιούνται για πράξεις μεταξύ αριθμών.

Τελεστής	Λειτουργία	Σύνταξη
+	Πρόσθεση	X+Y
-	Αφαίρεση	X-Y
*	Πολλαπλασιασμός	X*Y
/	Διαίρεση	X/Y
**	Δύναμη	X**Y

Τελεστές χαρακτήρων

Χρησιμοποιούνται για πράξεις μεταξύ αλυσίδων χαρακτήρων.

Τελεστής	Λειτουργία	Σύνταξη
//	Συνένωση αλυσίδων χαρακτήρων	"ABC"//"123" θα μας δώσει "ABC123"

Τελεστές Σύγκρισης

Χρησιμοποιούνται για σύγκριση μεταξύ αριθμών ή αλυσίδων χαρακτήρων.

Τελεστής	Λειτουργία	Σύνταξη
.EQ. ή ==	Ισότητα	X.EQ.Y ή X=Y
.NE. ή /=	Ανισότητα	X.NE.Y ή X/=Y
.GT. ή >	Μεγαλύτερο	X.GT.Y ή X>Y
.LT. ή <	Μικρότερο	X.LT.Y ή X<Y

.GE. ή >=	Μεγαλύτερο ή ίσο	X.GE.Y ή X>=Y
.LE. ή <=	Μικρότερο ή ίσο	X.LE.Y ή X<=Y

Η σύγκριση αριθμών γίνεται με το γνωστό τρόπο. Η σύγκριση αλυσίδων χαρακτήρων γίνεται συγκρίνοντας έναν έναν τους χαρακτήρες τους. Προκειμένου να γίνει η σύγκριση των χαρακτήρων χρησιμοποιούμε τη θέση τους στον ASCII κώδικα :

Πίνακας 2.1. Σειρά βαρύτητας των βασικών χαρακτήρων ASCII
(κενό=ελαφρύτερος, ω=βαρύτερος)

(κενό) ! " # \$ % & ' () * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
[\] ^ _ ` a b c d e f g h i j k l m n o p q r s t u v w x y z { } ~
Α Β Γ Δ Ε Ζ Η Θ Ι Κ Λ Μ Ν Ξ Ο Π Ρ Σ Τ α Υ Φ Χ Ψ Ω
α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ τ υ φ χ ψ ω

Σε περίπτωση που ο πρώτος χαρακτήρας μιας αλυσίδας χαρακτήρων είναι μικρότερος από τον αντίστοιχο πρώτο χαρακτήρα της δεύτερης αλυσίδας χαρακτήρων, τότε η πρώτη αλυσίδα χαρακτήρων είναι μικρότερη της δεύτερης π.χ. "A1"<"B1". Αν ο πρώτος χαρακτήρας είναι ίδιος, συγκρίνουμε ως προς το δεύτερο κ.ο.κ. πρδ. "A1"<"A2". Αν κατά τη σύγκριση φτάσουμε στο τέλος μιας αλυσίδας χαρακτήρων, τότε αυτή που έχει λιγότερους χαρακτήρες είναι η μικρότερη πρδ. "ABC"<"ABC123".



Δραστηριότητα 2.2 Συμπληρώστε με ένα από τα σύμβολα <, <=, >, >=, =, /= τα παρακάτω κενά ώστε οι παρακάτω προτάσεις να είναι αληθείς :

"GIORGOS" "ΓΙΩΡΓΟΣ"

" NIKOS"..... "ΝΙΚΟΣ"

"1234" "123"

"ANNA" "ANNA"

"ΝΙΚΟΣ" "ΝΙΚΟΣ"

123 "123"

"_ABC" "ABC_"

Τελεστές Λογικοί

Χρησιμοποιούνται για την εκτέλεση λογικών πράξεων.

Τελεστής	Λειτουργία
.NOT.	Λογική άρνηση
.AND.	Λογική πρόσθεση
.OR.	Διάζευξη

.EQV.	Ισοδυναμία
.NEQV.	Μη ισοδύναμα
.XOR.	ή/και

Οι πίνακες αληθείας των τελεστών είναι :

X	Y	X.AND.Y	X.OR.Y	.NOT.X
T	T	T	T	F
T	F	F	T	F
F	T	F	T	T
F	F	F	F	T

X	Y	X.EQV.Y	X.NEQV.Y	X.XOR.Y
T	T	T	F	F
T	F	F	T	T
F	T	F	T	T
F	F	T	F	F

Παράδειγμα

Η λογική παράσταση (5.GT.2).AND.(6.LT.3) είναι ψευδής, διότι η πρώτη πρόταση είναι αληθής και η δεύτερη ψευδής.

2.3.7 Διαχωριστές

Πρόκειται για χαρακτήρες με συγκεκριμένη λειτουργία στη FORTRAN 90. Οι κυριότεροι είναι :

= Καταχώριση

() Παρενθέσεις, για τους πίνακες

.. Σημεία στίξης

& Σύμβολο συνέχειας γραμμής

; Για τον διαχωρισμό εντολών που βρίσκονται στην ίδια γραμμή

2.3.8 Λέξεις κλειδιά

Πρόκειται για λέξεις με ειδική σημασία στη FORTRAN 90, όπως PROGRAM, END, IF, DO, WHILE κ.λ.π.