



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

Ενότητα 1: Εισαγωγή στους Η/Υ και στο περιβάλλον της
Compaq Visual Fortran

Νίκος Καραμπετάκης
Τμήμα Μαθηματικών



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Άδειες Χρήσης

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



Χρηματοδότηση

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



Περιεχόμενα Ενότητας

1. Ποια είναι τα βασικά μέρη από τα οποία αποτελείται ένας Η/Υ;
2. Ποια είναι η δομή του υλικού ενός Η/Υ;
3. Ποιες είναι οι βασικές κατηγορίες λογισμικού;
4. Τι είναι αλγόριθμος και πως μετράμε την απόδοση ενός αλγορίθμου;
5. Τι είναι γλώσσα προγραμματισμού;
6. Φάση υλοποίησης ενός προγράμματος
7. Τι είναι προγραμματισμός;
8. Ποια είναι τα στάδια του προγραμματισμού;
9. Πώς θα μεταφράσουμε και θα εκτελέσουμε ένα πρόγραμμα στη Compaq Visual Fortran;

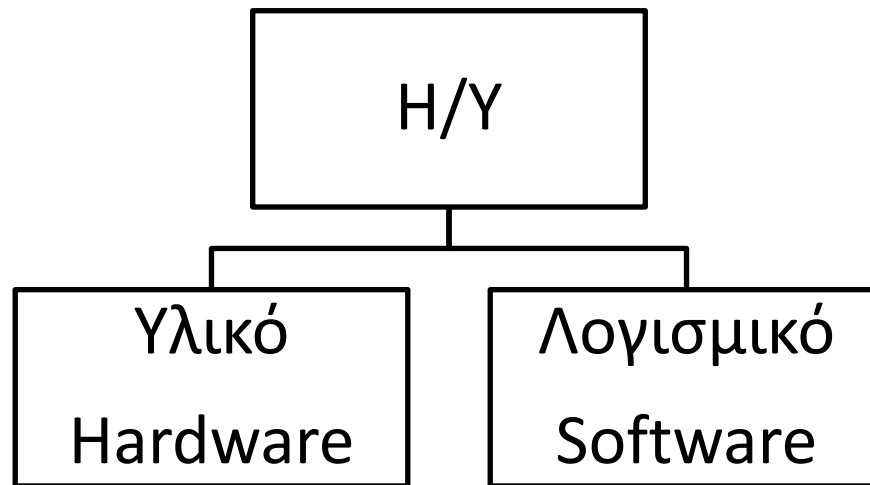


Σκοποί Ενότητας

1. Η παρουσίαση μιας συνοπτικής περιγραφής της δομής των Η/Υ.
2. Η παρουσίαση μιας διαδικασίας επίλυσης προβλημάτων με την βοήθεια του Η/Υ.
3. Η παρουσίαση ενός τρόπου μέτρησης της απόδοσης των αλγορίθμων.
4. Η περιγραφή του περιβάλλοντος της Intel Visual Fortran.



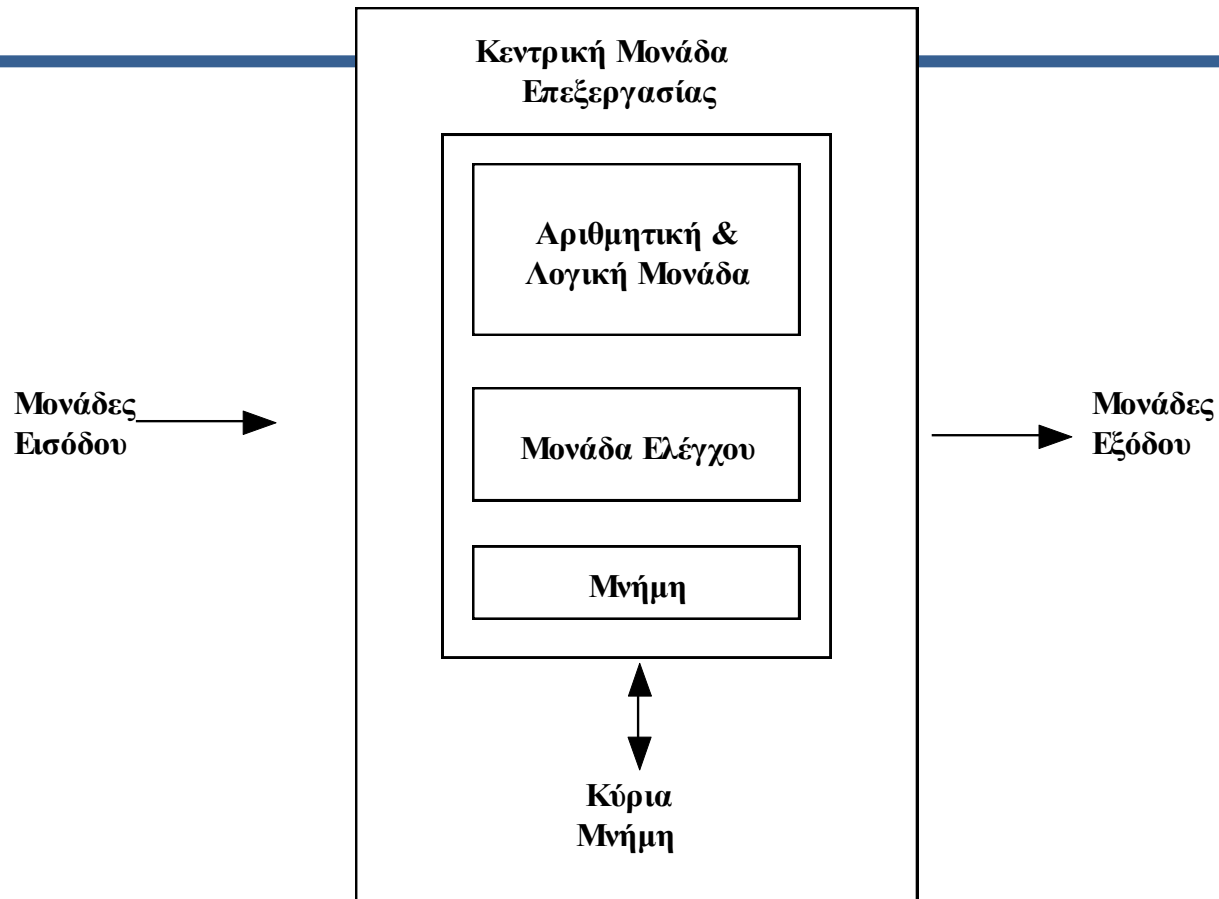
Ποια είναι τα βασικά μέρη από τα οποία αποτελείται ένας Η/Υ;



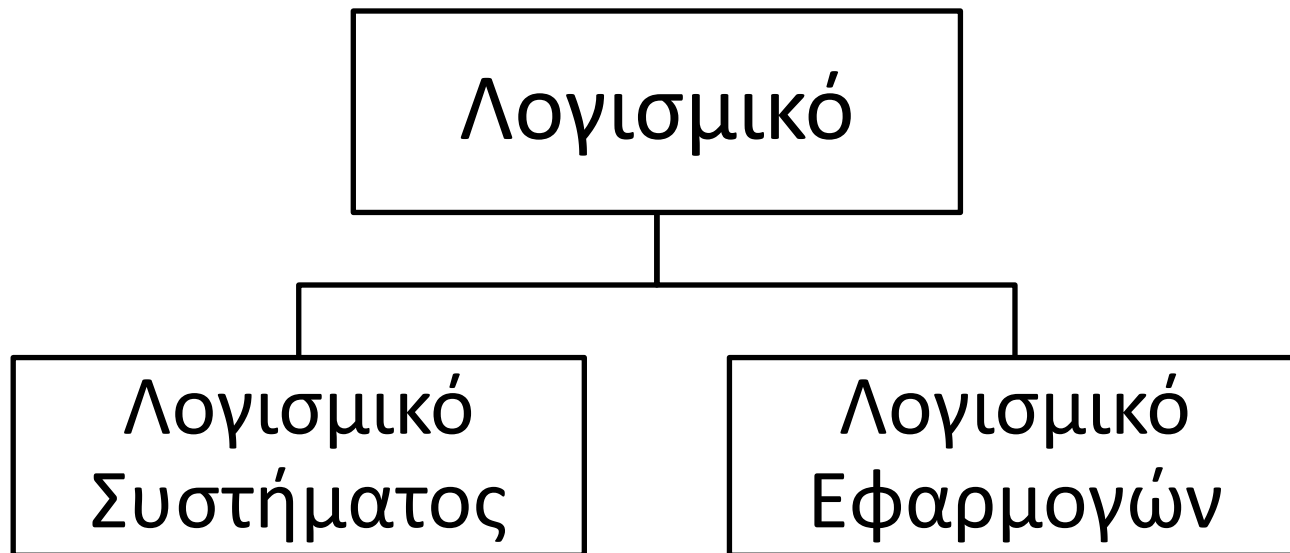
Τα μέρη του Η/Υ



Ποια είναι η δομή του υλικού ενός Η/Υ;



Ποιες είναι οι βασικές κατηγορίες λογισμικού;



Λογισμικό Συστήματος – Λογισμικό Εφαρμογών

➤ Λογισμικό Συστήματος

- Λειτουργικό Σύστημα (Windows XP (Vista, 2000,..), Unix, Linux, Mac OS)
 - 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 ,
 - συντονίζει τα διάφορα μέρη του υπολογιστή που απαιτούνται για την εκτέλεση μιας ολοκληρωμένης εργασίας
- Μεταφραστικά Προγράμματα

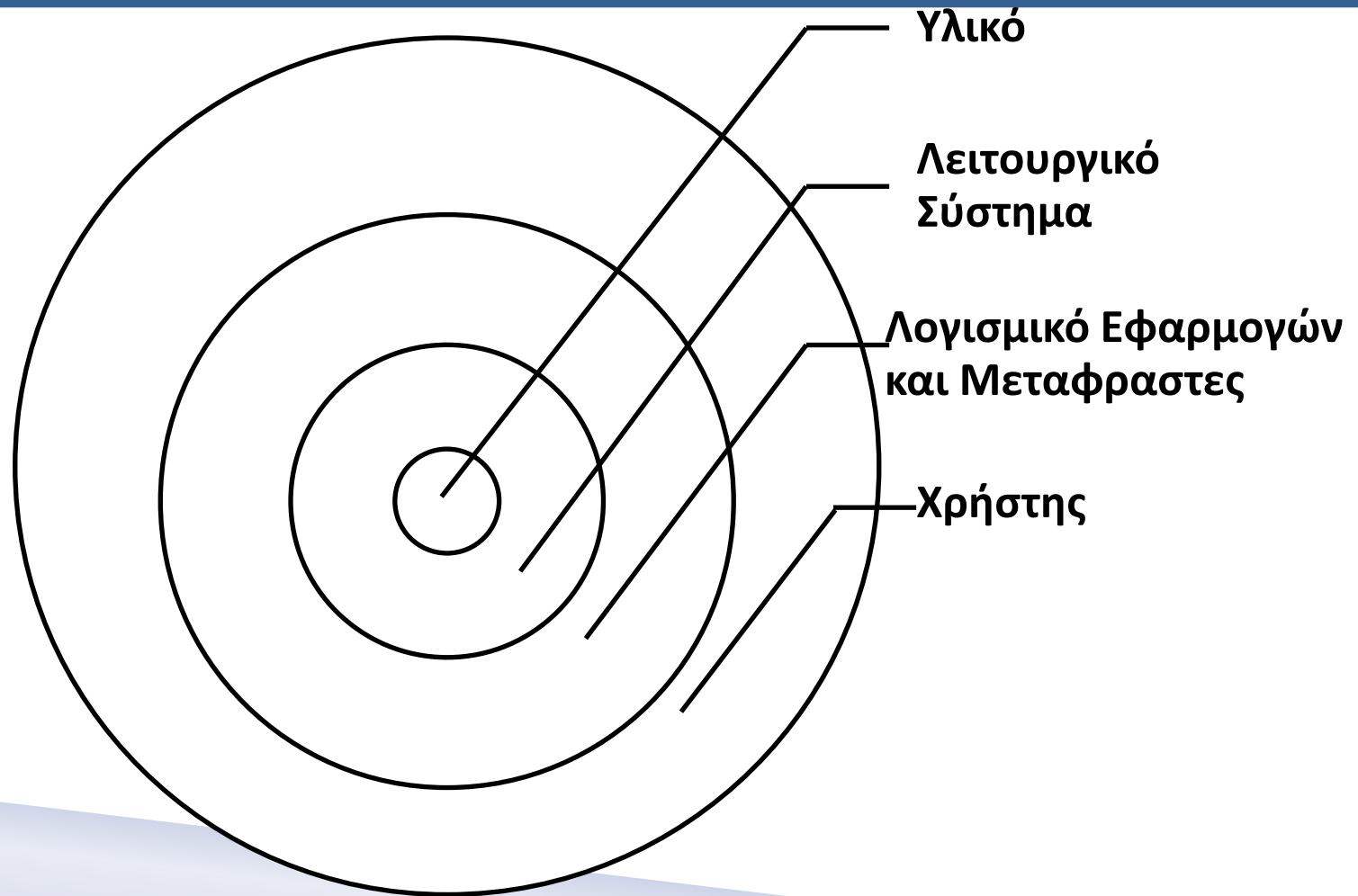


Λογισμικό Εφαρμογών

- **Λογισμικό Εφαρμογών**
 - προγράμματα επεξεργασίας κειμένου,
 - προγράμματα σχεδίασης,
 - λογιστικά φύλλα,
 - προγράμματα επικοινωνίας, κ.τ.λ.



Ποιά είναι η σχέση Υλικού – Λογισμικού – Χρήστη ;



Ορισμός του Αλγόριθμου

Αλγόριθμος είναι η ακριβής περιγραφή μιας αυστηρά καθορισμένης σειράς ενεργειών που πρέπει να ακολουθήσουμε ώστε να φέρουμε σε πέρας μια διαδικασία ή να λύσουμε ένα πρόβλημα, συνήθως με την προϋπόθεση ότι η διαδικασία αυτή θα τερματίσει σε πεπερασμένο χρόνο. Η λέξη **αλγόριθμος** προέρχεται από το όνομα του άραβα μαθηματικού Abu Ja'far Mohammed ibn Musa al Khwarizmi τον 9^ο αιώνα μ.Χ.



Abu Ja'far Mohammed ibn Musa al Khwarizmi



"1983 CPA 5426 (1)" by Unknown –

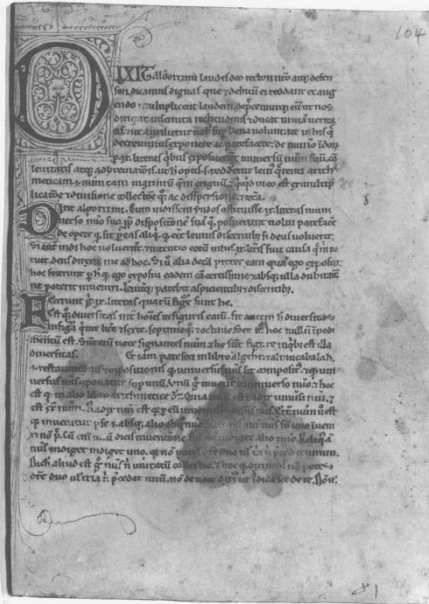
<http://www.muslimheritage.com/topics/default.cfm?ArticleID=631>, [1].

Licensed under Public domain via Wikimedia Commons –

[http://commons.wikimedia.org/wiki/File:1983 CPA 5426 \(1\).png#mediaviewer/File:1983_CPA_5426_\(1\).png](http://commons.wikimedia.org/wiki/File:1983_CPA_5426_(1).png#mediaviewer/File:1983_CPA_5426_(1).png)



"Dixit algorismi..." (όπως είπε ο al Khwarizmi)



Το βιβλίο του al Khwarizmi για το ινδικό σύστημα υπολογισμού «Κανόνες σύνθεσης και αναγωγές» διασώθηκε μόνο στα λατινικά όπου ξεκινάει με τον τίτλο "Dixit algorismi..." (όπως είπε ο al Khwarizmi)

"Dixit algorizmi" από τον Muhammad ibn Musa al-Khwarizmi – scanned from facsimile (1963). Υπό την άδεια Public domain μέσω Wikimedia Commons – http://commons.wikimedia.org/wiki/File:Dixit_algorizmi.png#mediaviewer/File:Dixit_algorizmi.png



Πως μετράμε την απόδοση ενός αλγορίθμου ;

1. Ο **εμπειρικός τρόπος** κατά τον οποίο υπολογίζεται ο χρόνος εκτέλεσης και η χωρητικότητα μνήμης του αλγορίθμου που απαιτούνται για την υλοποίηση του για ένα σύνολο δεδομένων.
2. Ο **θεωρητικός τρόπος** όπου η μέτρηση της αποδοτικότητας εξαρτάται από το μέγεθος της εισόδου το οποίο προσδιορίζεται από μια μεταβλητή n .



Ο θεωρητικός τρόπος

Ας συμβολίσουμε με την συνάρτηση $f(n)$ τον χρόνο εκτέλεσης (**χρονική πολυπλοκότητα** ή time complexity) (ή την χωρητικότητα μνήμης (**χωρική πολυπλοκότητα** ή space complexity)) σε σχέση με το μέγεθος της εισόδου n .

Ο χρόνος εκτέλεσης $f(n)$ εξαρτάται :

1. από τον αριθμό των στοιχειωδών βημάτων που πρέπει να εκτελεστούν σε έναν αλγόριθμο.

Ταξινόμηση \rightarrow συγκρίσεις.

Αντιστροφή πίνακα \rightarrow πολ/μοι, διαιρέσεις.

2. από την δομή των δεδομένων.

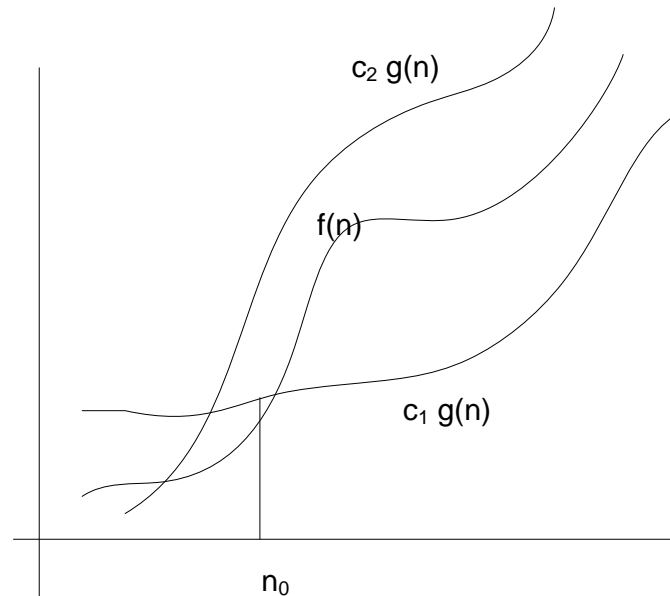
Ταξινόμηση \rightarrow Είναι ταξινομημένος ο πίνακας ;



Μας ενδιαφέρει η ασυμπτωτική εκτίμηση του χρόνου εκτέλεσης $f(n)$ (άνω και κάτω φράγμα)

Ορισμός 1.1 [Knuth,1976] Ορίζουμε ως $\Theta(g(n))$ το σύνολο των συναρτήσεων

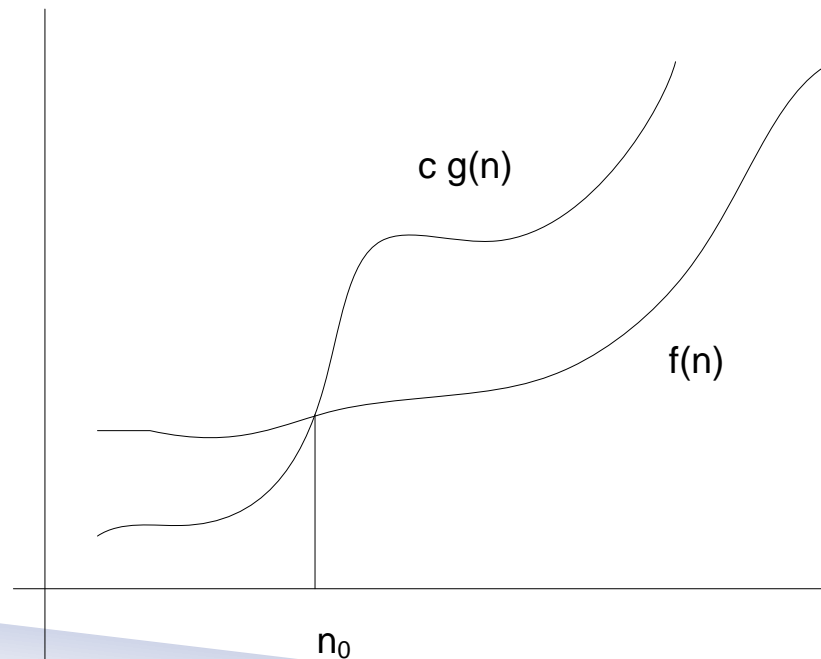
$$\Theta(g(n)) = \{f(n) : \exists c_1, c_2 \in \mathbb{R}^+, n_0 \in \mathbb{N}, 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \forall n \geq n_0\}$$



Μας ενδιαφέρει η ασυμπτωτική εκτίμηση του χρόνου εκτέλεσης $f(n)$ (άνω φράγμα)

Ορισμός 1.2 [P.Bachman,1892] Ορίζουμε ως $O(g(n))$ το σύνολο των συναρτήσεων

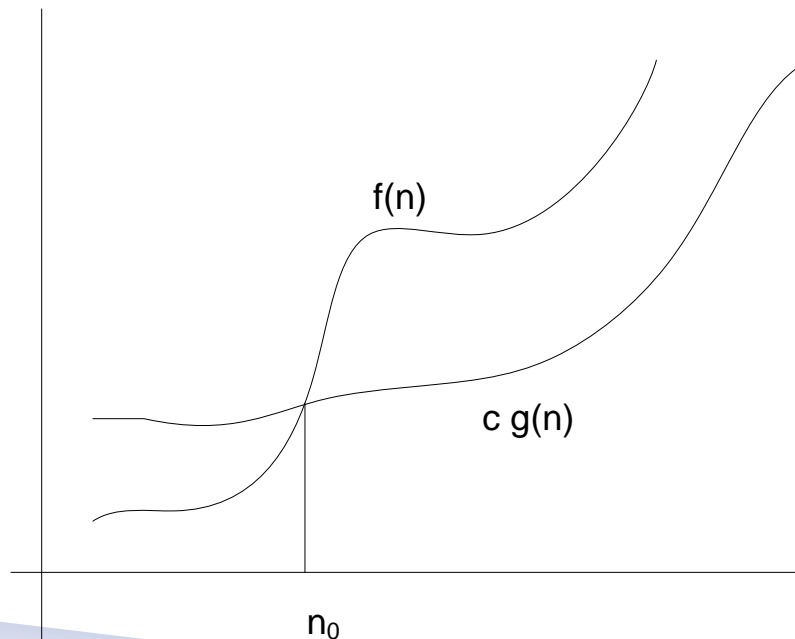
$$O(g(n)) = \{f(n) : \exists c \in \mathbb{R}^+, n_0 \in \mathbb{N}, 0 \leq f(n) \leq cg(n) \forall n \geq n_0\}$$



Μας ενδιαφέρει η ασυμπτωτική εκτίμηση του χρόνου εκτέλεσης $f(n)$ (κάτω φράγμα)

Ορισμός 1.3 [Knuth,1976] Ορίζουμε ως $\Omega(g(n))$ το σύνολο των συναρτήσεων

$$\Omega(g(n)) = \{f(n) : \exists c \in \mathbb{R}^+, n_0 \in \mathbb{N}, 0 \leq cg(n) \leq f(n) \forall n \geq n_0\}$$



Μας ενδιαφέρει η ασυμπτωτική εκτίμηση του χρόνου εκτέλεσης $f(n)$ (αλγόριθμοι)

$$f(n) = \Theta(n^3) \rightarrow f(2n) = \Theta((2n)^3) = \Theta(8n^3)$$
$$f(n) = \Theta(n^3) \rightarrow f(3n) = \Theta((3n)^3) = \Theta(27n^3)$$

Πολυωνυμικοί Αλγόριθμοι $\rightarrow \Theta(n^k), k \in \mathbb{R}$

Μη πολυωνυμικοί Αλγόριθμοι $\rightarrow \Theta(n^n), \Theta(n!), \Theta(2^n)$



Μας ενδιαφέρει η ασυμπτωτική εκτίμηση του χρόνου εκτέλεσης $f(n)$ (πολυπλοκότητα)

Ο χρόνος εκτέλεσης (σε ns) ενός αλγορίθμου σε σχέση με την πολυπλοκότητα του αλγορίθμου αν υποθέσουμε ότι κάθε στοιχειώδης πράξη απαιτεί 1ns της CPU του υπολογιστή μας

Πολυπλοκότητα αλγορίθμου	Μέγεθος προβλήματος (n)		
	10	30	50
$\Theta(n)$	0.00001	0.00003	0.00005
$\Theta(n \log_2 n)$	3.0103×10^{-6}	6.11385×10^{-6}	8.85919×10^{-6}
$\Theta(n^2)$	0.0001	0.0009	0.0025
$\Theta(n^3)$	0.001	0.027	0.125
$\Theta(2^n)$	0.001024	1073.74	1.1259×10^9
$\Theta(n!)$	3.6288	2.65253×10^{26}	3.04141×10^{58}



Μας ενδιαφέρει η ασυμπτωτική εκτίμηση του χρόνου εκτέλεσης $f(n)$

$$3n^2 = O(n^2), 3n = O(n^2)$$

Ορισμός 1.4 [E. Landau, 1909] Ορίζουμε ως $o(g(n))$ το σύνολο των συναρτήσεων

$$o(g(n)) = \{f(n) : \forall c \in \mathbb{R}^+, \exists n_0 \in \mathbb{N}, 0 \leq f(n) \leq cg(n) \forall n \geq n_0\}$$

$$f(n) = o(g(n)) \leftrightarrow \lim_{n \rightarrow +\infty} \frac{f(n)}{g(n)} = 0$$

$$3n = o(n^2), 3n^2 \neq o(n^2)$$



Μας ενδιαφέρει η ασυμπτωτική εκτίμηση του χρόνου εκτέλεσης $f(n)$ (...συνέχεια)

Ορισμός 1.5 Ορίζουμε ως $\omega(g(n))$ το σύνολο των συναρτήσεων

$$\omega(g(n)) = \{f(n) : \forall c \in \mathbb{R}^+, \exists n_0 \in \mathbb{N}, 0 \leq cg(n) \leq f(n) \forall n \geq n_0\}$$

$$f(n) = \omega(g(n)) \leftrightarrow g(n) = o(f(n)) \leftrightarrow \lim_{n \rightarrow +\infty} \frac{f(n)}{g(n)} = \infty$$



Τι είναι γλώσσα προγραμματισμού;

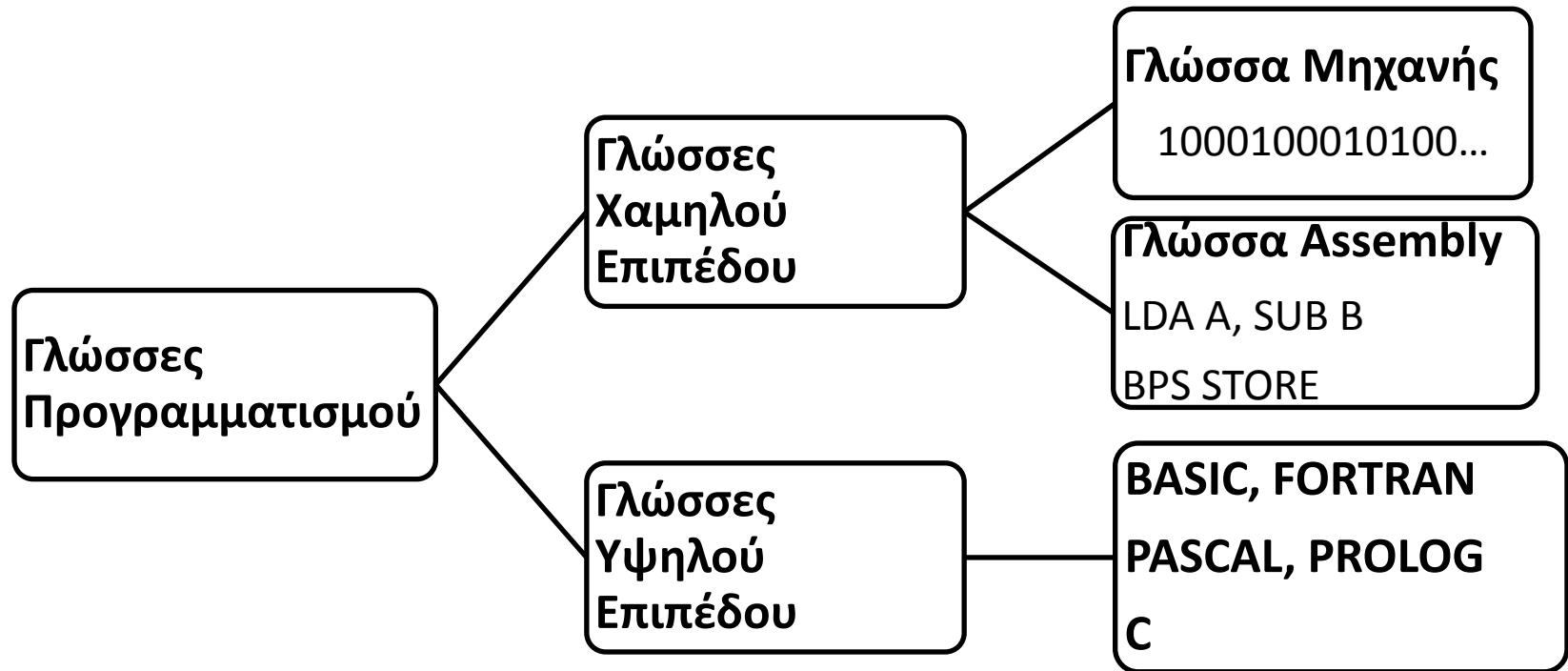
Γλώσσα προγραμματισμού είναι ένα σύνολο γραμμάτων, αριθμών, λέξεων και συντομογραφικών μνημονικών σημείων που διέπονται από ειδικό συντακτικό και χρησιμοποιούνται στην ανάπτυξη αλγορίθμων στον υπολογιστή.

Ποικιλία γλωσσών προγραμματισμού :

- κάποιες γλώσσες υπερτερούν έναντι άλλων σε συγκεκριμένες εφαρμογές.
- προσφέρουν καλύτερη υποστήριξη.
- είναι ευκολότερες στην εκμάθηση τους, κ.τ.λ.



Διαχωρισμός των γλωσσών προγραμματισμού



Ποια είναι τα προγράμματα μετάφρασης;

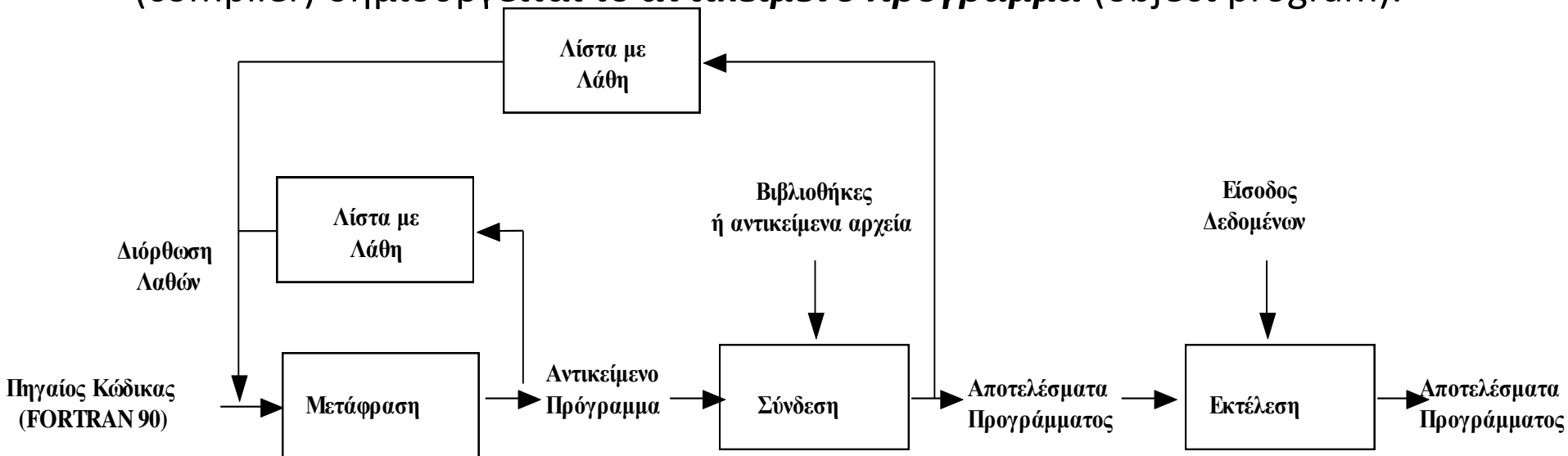
- **Interpreter (Διερμηνέας)**
Μεταφράζει μια-μια τις εντολές συγχρόνως με την εκτέλεση τους.
- **Compiler (Μεταφραστής)**
Μεταφράζει μια φορά ολόκληρο το πρόγραμμα.
Στη συνέχεια το μεταφρασμένο πρόγραμμα εκτελείται.



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).



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.



Ποια είναι τα στάδια του προγραμματισμού;

A. Φάση Ανάλυσης

A. Φάση Ανάλυσης (*Analysis*)

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

Παράδειγμα. *Να υπολογιστεί το εμβαδόν ενός τριγώνου ABΓ.*

- Είναι το πρόβλημα καλά ορισμένο;
- *Επαναδιατύπωση του προβλήματος: «Να υπολογιστεί το εμβαδόν ενός τριγώνου ABΓ εάν είναι γνωστά τα μήκη των πλευρών του AB, BΓ, ΓΑ.»*
- Είσοδος: Πλευρές AB, BΓ, ΓΑ
- Έξοδος: Εμβαδόν E



Ποια είναι τα στάδια του προγραμματισμού;

B. Φάση Σχεδιασμού

B. Φάση Σχεδιασμού (Design)

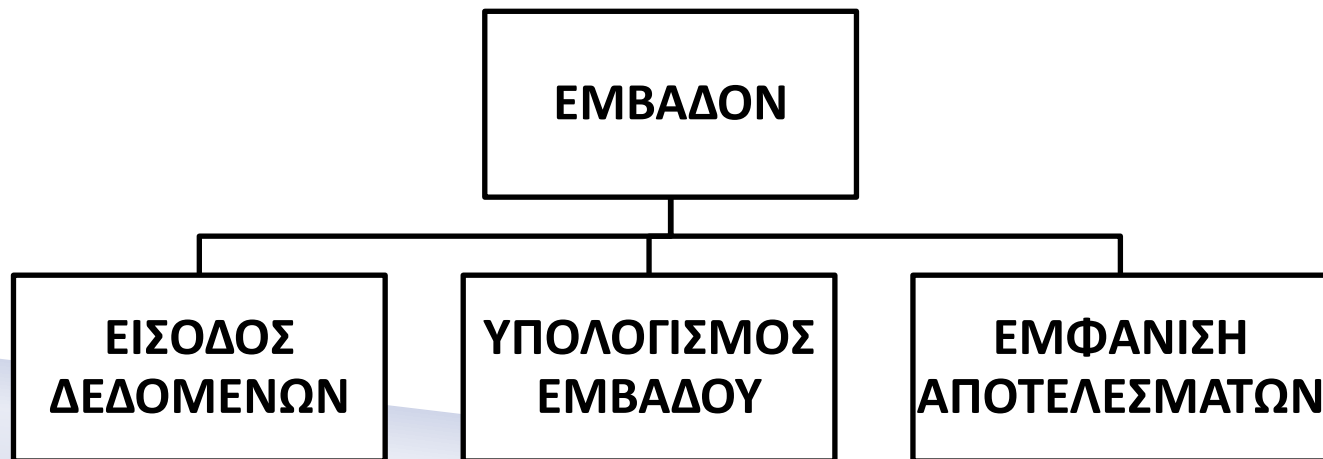
- Σκιαγράφιση της λύσης και διάκριση ανεξαρτήτων διαδικασιών.
- Επιλογή και περιγραφή ενός αλγόριθμου που να είναι περατός, σαφής, γενικός, αποτελεσματικός, αποδοτικός, ευσταθής.



Ποια είναι τα στάδια του προγραμματισμού; B. Φάση Σχεδιασμού – Παράδειγμα (Διαδικασίες)

Παράδειγμα. (B. 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.



Ποια είναι τα στάδια του προγραμματισμού; B. Φάση Σχεδιασμού – Παράδειγμα (Ψευδοκώδικας)

Παράδειγμα. (B. Φάση Σχεδιασμού)

- 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).



Ποια είναι τα στάδια του προγραμματισμού; B. Φάση Σχεδιασμού – Παράδειγμα (Ψευδοκώδικας)

Παράδειγμα. (B. Φάση Σχεδιασμού)

- Ψευδοκώδικας

Βήμα 1^ο. Διάβασε a,b,c.

Βήμα 2^ο. $S \leftarrow \frac{1}{2}(a + b + c)$

Βήμα 3^ο. $E \leftarrow \sqrt{S(S - a)(S - b)(S - c)}$

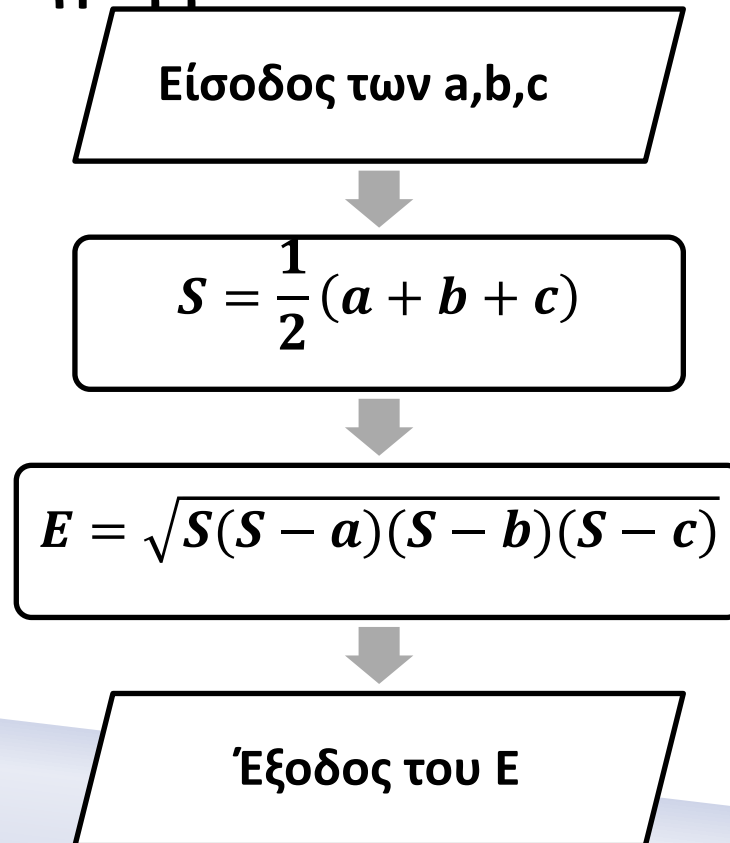
Βήμα 4^ο. Εμφάνισε E.



Ποια είναι τα στάδια του προγραμματισμού; B. Φάση Σχεδιασμού – Παράδειγμα (Λογικό Διάγραμμα)

Παράδειγμα. (B. Φάση Σχεδιασμού)

- Λογικό Διάγραμμα



Λογικό Διάγραμμα

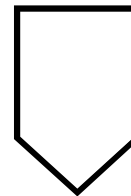
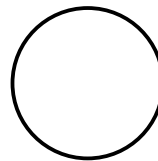
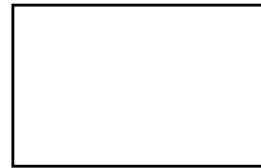
ΑΡΧΗ – ΤΕΛΟΣ

ΕΙΣΟΔΟΣ-ΕΞΟΔΟΣ

ΕΠΕΞΕΡΓΑΣΙΑ

Παραπομπή σε άλλο
σημείο του λογικού
διαγράμματος

Παραπομπή σε άλλη σελίδα.



Ποια είναι τα στάδια του προγραμματισμού;

Γ. Φάση Υλοποίησης

Γ. Φάση Υλοποίησης (*Implementation*)

- Επιλογή της γλώσσας προγραμματισμού.
- Κωδικοποίηση του αλγορίθμου σε πρόγραμμα.



Ποια είναι τα στάδια του προγραμματισμού;

Γ. Φάση Υλοποίησης - Παράδειγμα

Παράδειγμα. Φάση Υλοποίησης (Implementation)

- **Πρόγραμμα**

```
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
```



Ποια είναι τα στάδια του προγραμματισμού;

Δ. Εκσφαλμάτωση (Debugging)

- Έλεγχος προγράμματος για ανίχνευση λαθών.
 - *συντακτικά λάθη* (syntax errors) π.χ. REED αντί για READ
 - *λογικά λάθη* (logical errors) π.χ. $X = -A/B$ χωρίς να ελέγξουμε αν $B \neq 0$,
 - *λάθη εκτέλεσης*, π.χ. λανθασμένο όρισμα σε συνάρτηση.
- Χρήση του προγράμματος με ποικίλα δεδομένα για να διαπιστωθούν τυχόν λάθη.



Ποια είναι τα στάδια του προγραμματισμού; Γ. Φάση Υλοποίησης – Παράδειγμα

Παράδειγμα. Φάση Υλοποίησης (*Implementation*)

- **Πρόγραμμα**

```
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
```

- **Μετάφραση**
Συντακτικά Λάθη
- **Εκτέλεση**
Λογικά Λάθη??



Ποια είναι τα στάδια του προγραμματισμού;

Ε. Τεκμηρίωση

- **Εσωτερική Τεκμηρίωση**
 - Σχόλια όπου επεξηγούνται τα επιμέρους στάδια του προγράμματος.
- **Εξωτερική Τεκμηρίωση** - Ύπαρξη φακέλου που να περιέχει:
 - το πρόγραμμα με την λειτουργική δομή του προγράμματος,
 - τον αλγόριθμο που επιλέχτηκε,
 - τις τεχνικές που χρησιμοποιήθηκαν για την υλοποίηση του αλγορίθμου,
 - τα πλεονεκτήματα και μειονεκτήματα του συγκεκριμένου αλγόριθμου,
 - τον κώδικα,
 - αναλυτική επεξήγηση του κάθε τμήματος του αλγόριθμου κ.α.



Ποια είναι τα στάδια του προγραμματισμού; ΣΤ. Συντήρηση

Ανά τακτά χρονικά διαστήματα :

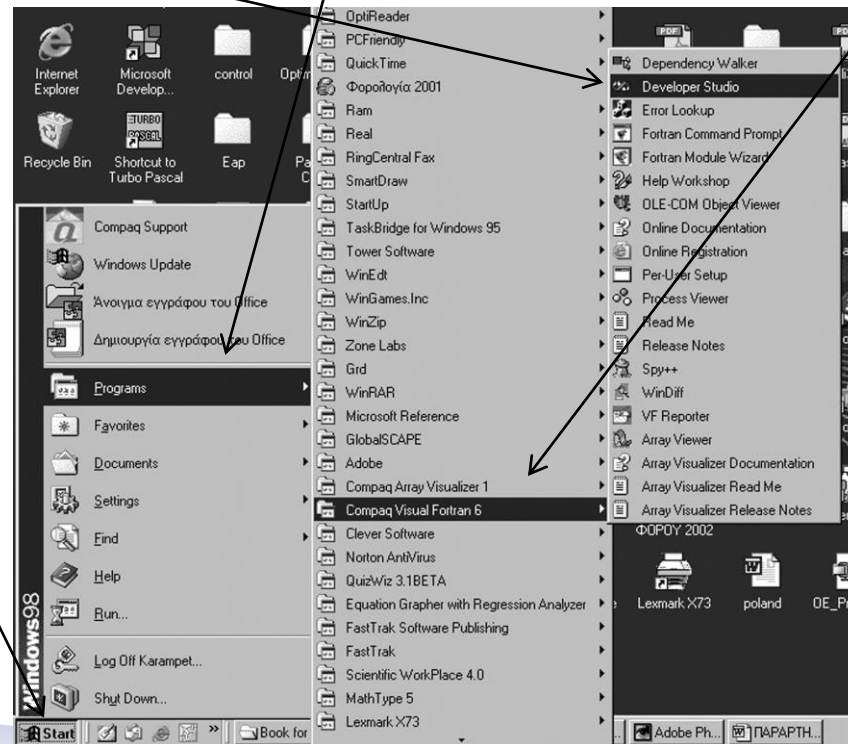
- διόρθωση απρόβλεπτων λαθών που έχουν προκύψει,
- τροποποίηση-βελτίωση του λαμβάνοντας υπόψη νέα δεδομένα και νέες εξελίξεις στον τομέα που το αφορά.



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

Βήμα 1. Εκκίνηση Προγράμματος

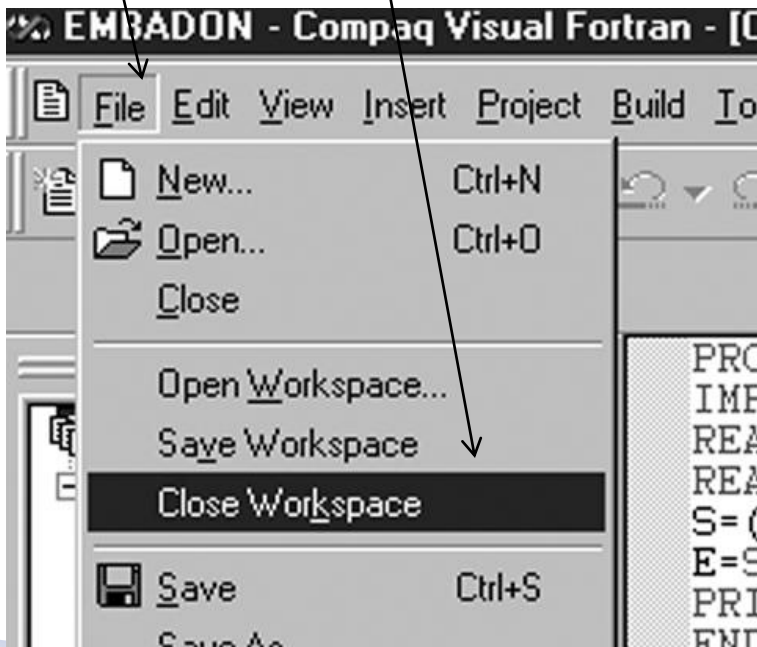
Έναρξη (Start) → Προγράμματα (Programs) → Compaq Visual Fortran 6 → Developer Studio



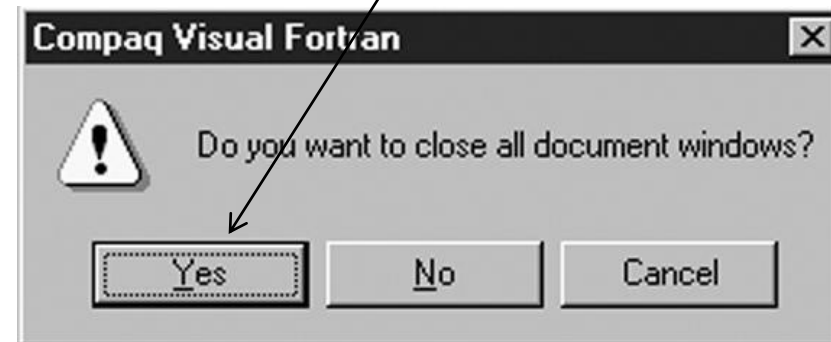
Πώς θα μεταφράσουμε και θα εκτελέσουμε ένα πρόγραμμα στη Compaq Visual Fortran;

Βήμα 2. Κλείσιμο των χώρων εργασίας.

File → Close Workspace



Απαντούμε με Yes στο παρακάτω ερώτημα

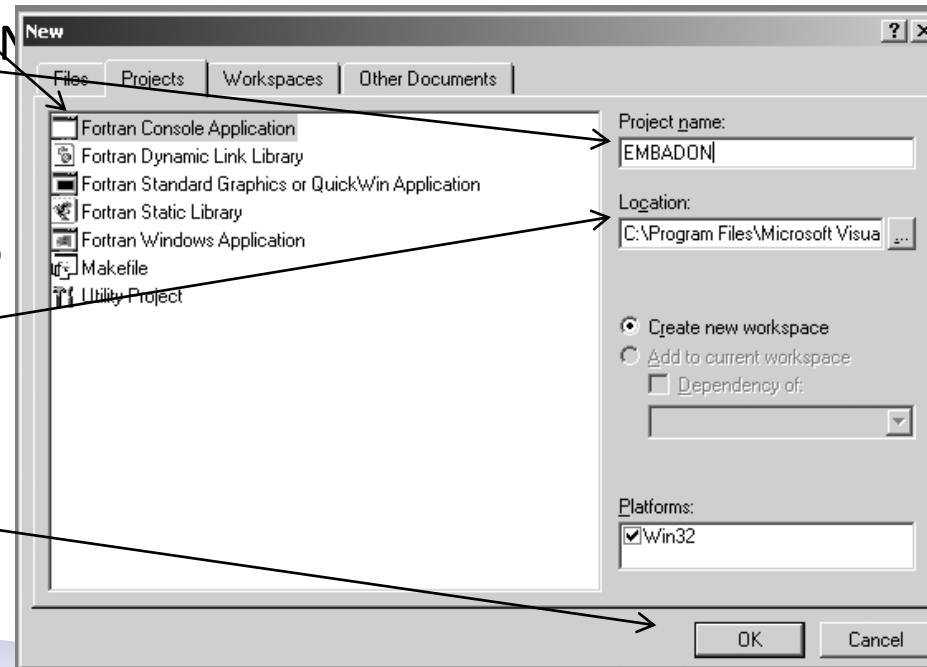
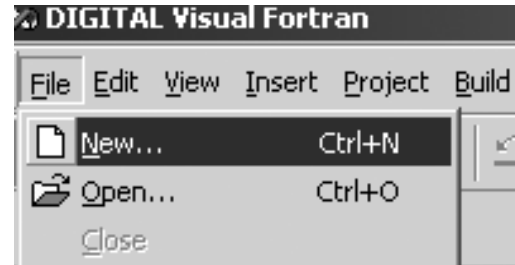


Πώς θα μεταφράσουμε και θα εκτελέσουμε ένα πρόγραμμα στη Compaq Visual Fortran;

Βήμα 3. Δημιουργία νέας εφαρμογής

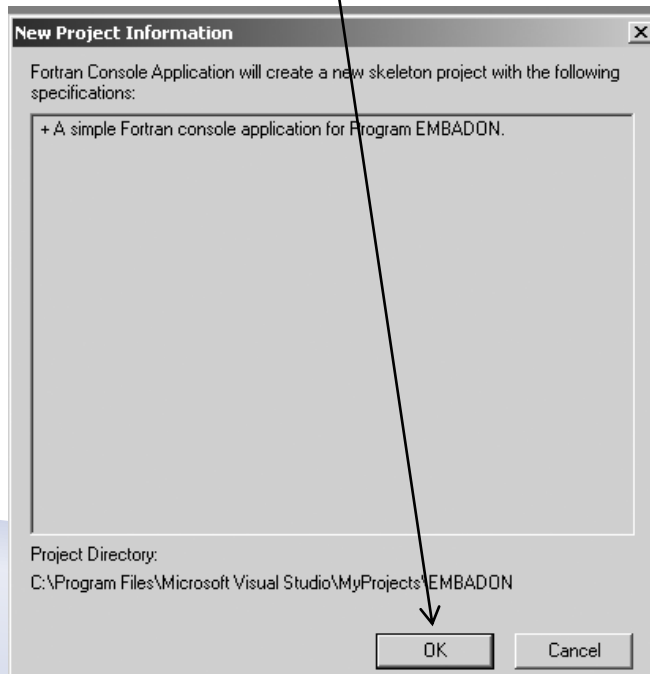
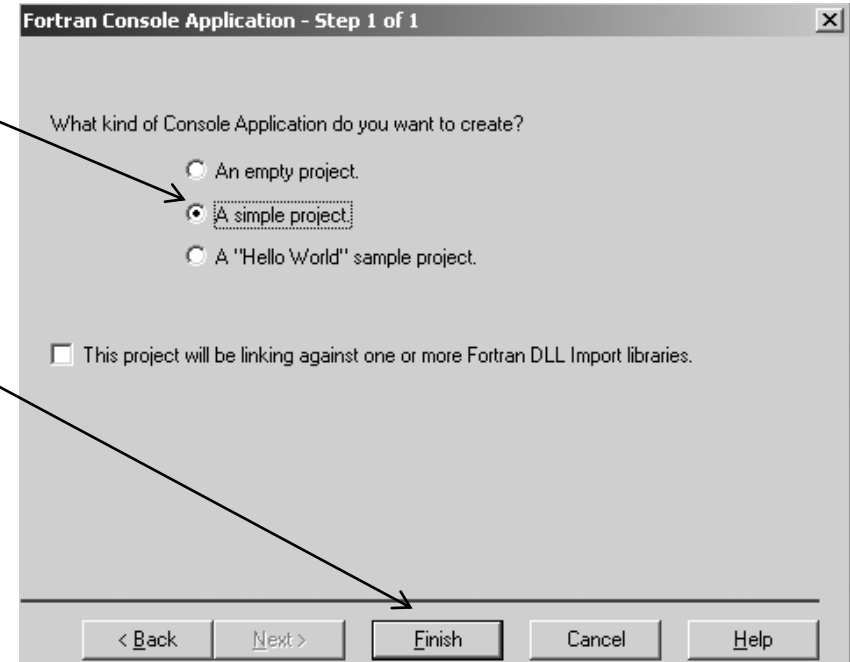
File → New

- α) Fortran Console Application
- β) Project Name → όνομα του Project Workspace (π.χ. EMBADON)
- γ) Location → κατάλογο και δευτερεύουσα μονάδα μνήμης που θέλουμε να δημιουργηθεί το Project Workspace
- δ) Κλικ στο OK.



Πώς θα μεταφράσουμε και θα εκτελέσουμε ένα πρόγραμμα στη Compaq Visual Fortran; Βήμα 4. Δημιουργία νέου Project

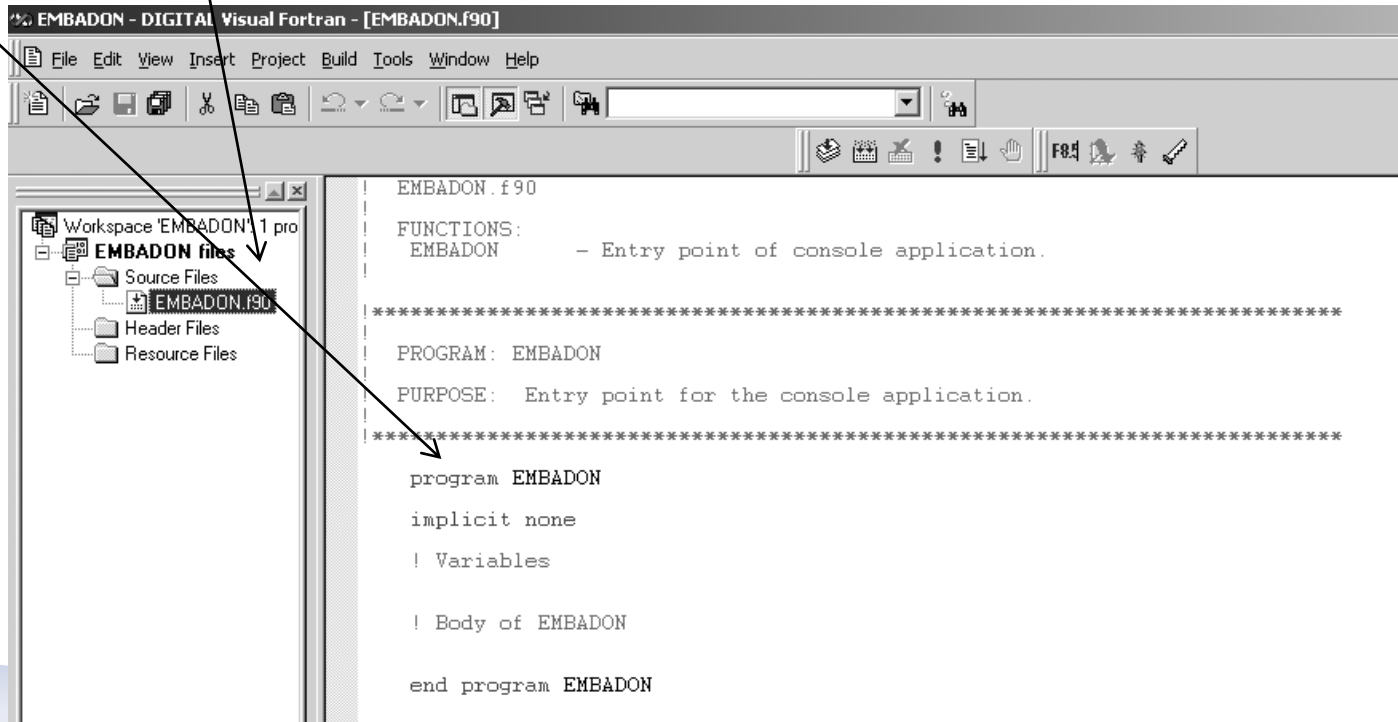
- a) επιλέγουμε *A simple project*
- b) κλικ στο *Finish*
- c) επιλέγουμε *OK*



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

Βήμα 5. Άνοιγμα του αρχείου προγράμματος

Παρατηρούμε ότι έχει δημιουργηθεί ένα αρχείο με το όνομα **EMBADON.F90** στο οποίο και θα τοποθετήσουμε το **πρόγραμμά μας**.



Πώς θα μεταφράσουμε και θα εκτελέσουμε ένα πρόγραμμα στη Compaq Visual Fortran; Βήμα 6. Γράψιμο προγράμματος

Θα γράψουμε το πρόγραμμα μας

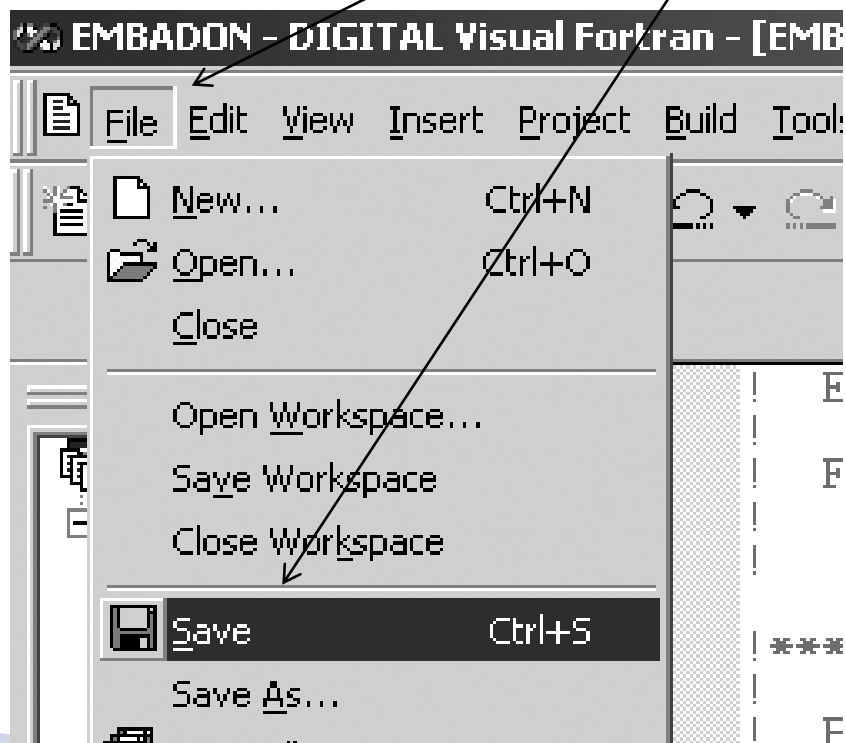
```
| EMBADON.f90
|
| FUNCTIONS:
|   EMBADON      - Entry point of console application.
|
| *****
| PROGRAM: EMBADON
| PURPOSE:  Entry point for the console application.
| *****
|
| program EMBADON
|
| implicit none
|
| ! Variables
| REAL A,B,C,S,E
|
| ! Body of EMBADON
| READ*, A,B,C
|
| S=(1/2)*(A+B+C)
| E=SQRT(S*(S-A)*(S-B)*(S-C))
|
| PRINT*, "-----"
| PRINT*, A,B,C
| PRINT*, "E=",E
|
| end program EMBADON
```



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

Βήμα 7. Αποθήκευση προγράμματος

Αποθηκεύουμε το πρόγραμμά μας. *File* → *Save*

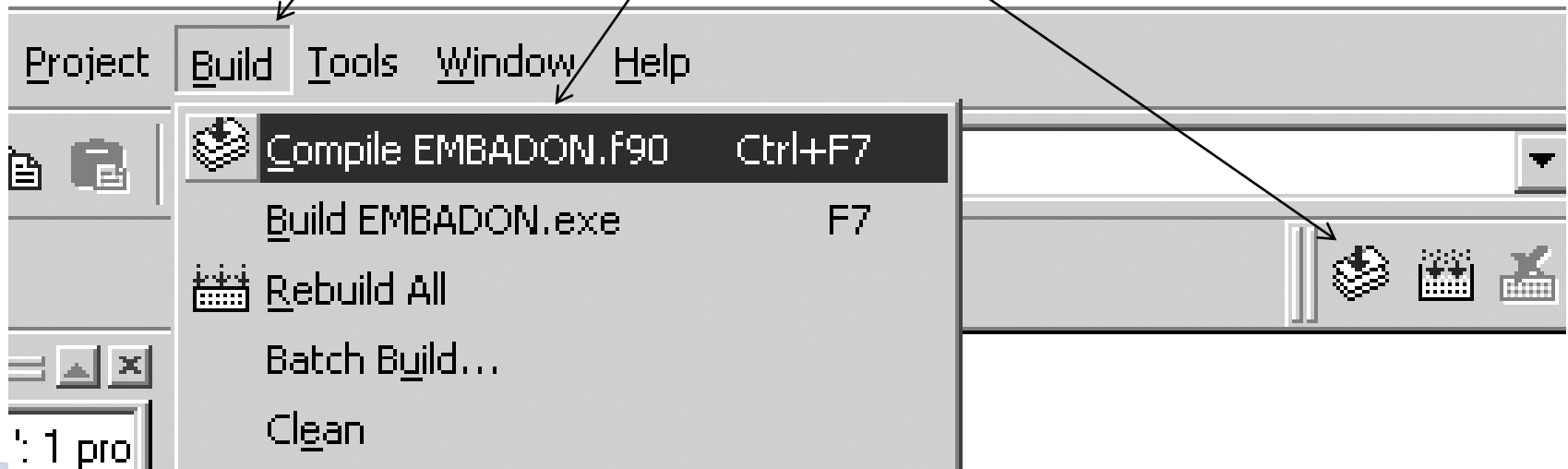


Πώς θα μεταφράσουμε και θα εκτελέσουμε ένα πρόγραμμα στη Compaq Visual Fortran; Βήμα 8. Αποθήκευση προγράμματος

Μετάφραση του προγράμματος.

Build → *Compile EMBADON.f90*

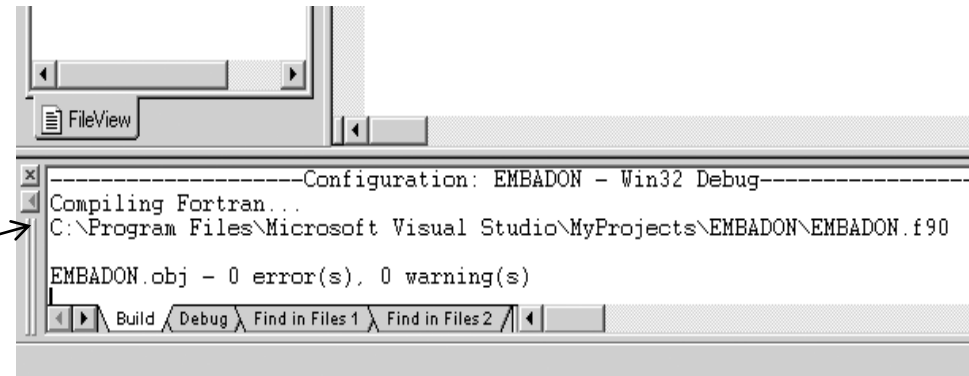
(ή Ctrl+F7 ή το αντίστοιχο **εικονίδιο** από την μπάρα με τα εικονίδια)



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

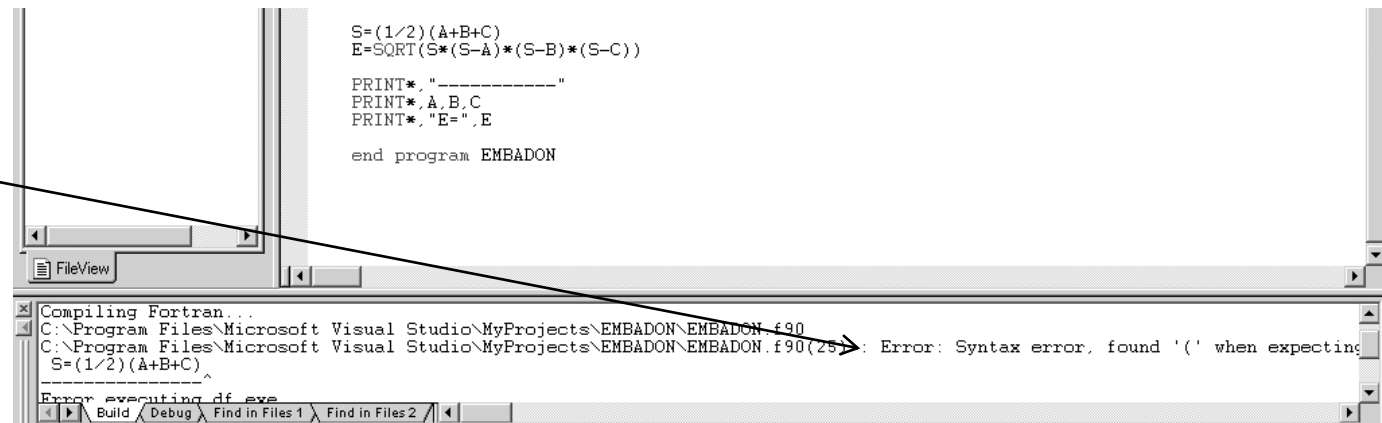
Βήμα 9. Έλεγχος για συντακτικά λάθη

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



```
-----Configuration: EMBADON - Win32 Debug-----
Compiling Fortran...
C:\Program Files\Microsoft Visual Studio\MyProjects\EMBADON\EMBADON.f90
EMBADON.obj - 0 error(s), 0 warning(s)
```

διαφορετικά



```
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
```

```
Compiling Fortran...
C:\Program Files\Microsoft Visual Studio\MyProjects\EMBADON\EMBADON.f90
C:\Program Files\Microsoft Visual Studio\MyProjects\EMBADON\EMBADON.f90(25): Error: Syntax error, found '(' when expecting
S=(1/2)(A+B+C)
^
Error executing df.exe
```

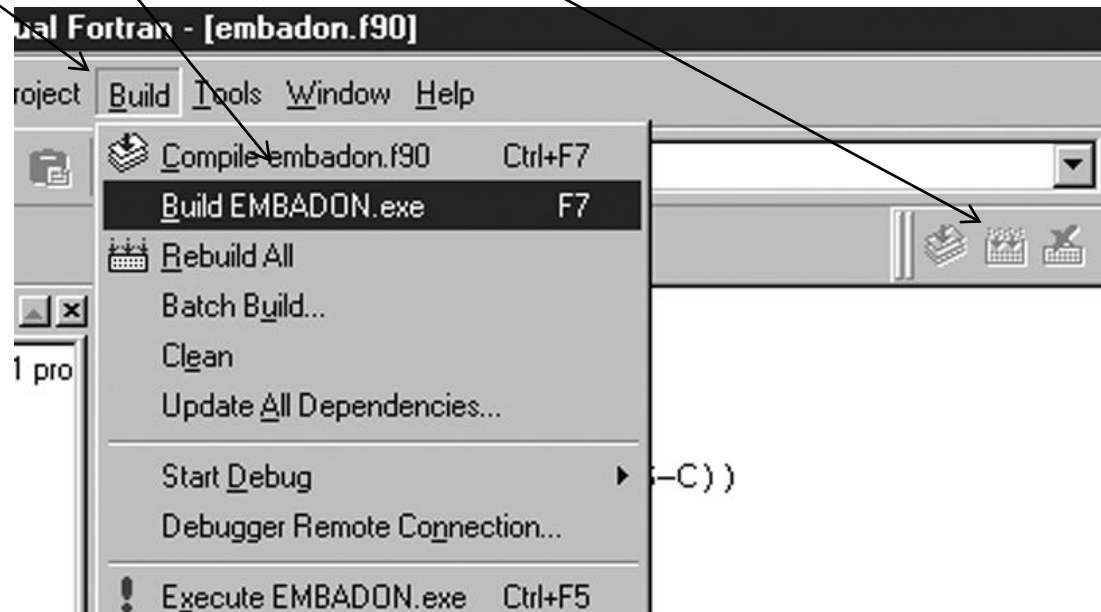


Πώς θα μεταφράσουμε και θα εκτελέσουμε ένα πρόγραμμα
στη Compaq Visual Fortran;

Βήμα 10. Δημιουργία εκτελέσιμου προγράμματος

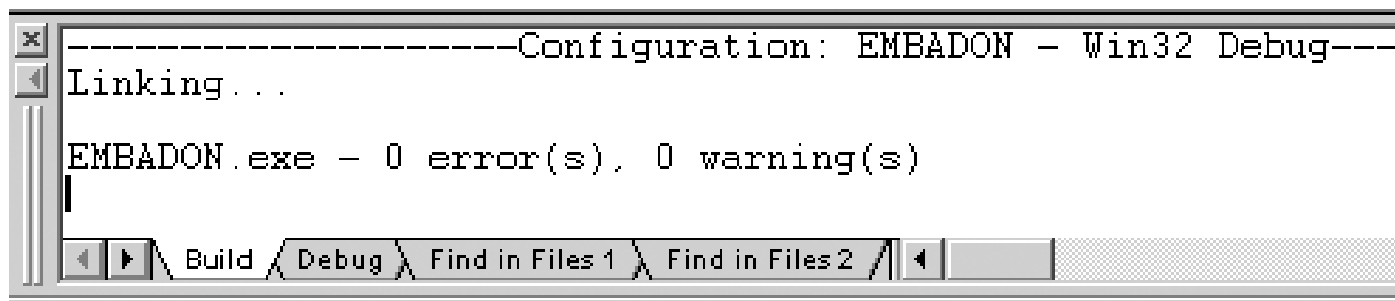
Build → *Build EMBADON.exe*

(ή F7 ή το αντίστοιχο εικονίδιο του BUILD από την μπάρα με τα εικονίδια)



Πώς θα μεταφράσουμε και θα εκτελέσουμε ένα πρόγραμμα στη Compaq Visual Fortran; Βήμα 11. Έλεγχος για επιτυχή σύνδεση

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



```
-----Configuration: EMBADON - Win32 Debug-----  
Linking...  
  
EMBADON.exe - 0 error(s), 0 warning(s)
```

The screenshot shows a window titled "Configuration: EMBADON - Win32 Debug". The main text area displays "Linking..." followed by "EMBADON.exe - 0 error(s), 0 warning(s)". At the bottom, there is a toolbar with buttons for "Build", "Debug", "Find in Files 1", and "Find in Files 2".

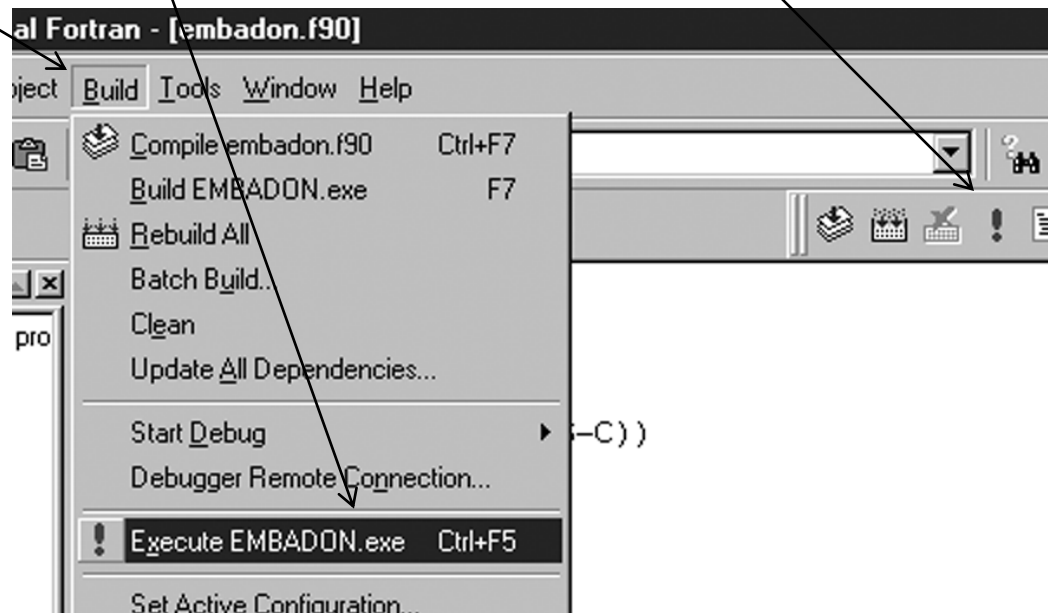


Πώς θα μεταφράσουμε και θα εκτελέσουμε ένα πρόγραμμα στη Compaq Visual Fortran;

Βήμα 12. Εκτέλεση του προγράμματος

Build → *Execute EMBADON.exe*

(ή Ctrl+F5 ή επιλέγουμε το αντίστοιχο εικονίδιο ! από την μπάρα εικονιδίων)



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

Βήμα 13. Εμφάνιση των αποτελεσμάτων

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

```
"C:\Program Files\Microsoft Visual Studio\MyProjects\EMBADO
3,4,5
-----
 3.000000      4.000000      5.000000
E= 0.00000000E+00
Press any key to continue
```



Άσκηση

Προσπαθήστε να γράψετε και να εκτελέσετε το παρακάτω πρόγραμμα.

```
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
```



Βιβλιογραφία

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



Σημείωμα Αναφοράς

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



Σημείωμα Αδειοδότησης

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



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





Τέλος Ενότητας

Επεξεργασία: Αναστασία Γ. Γρηγοριάδου

Θεσσαλονίκη, Χειμερινό Εξάμηνο 2013-2014



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

