



ΑΡΙΣΤΟΤΕΛΕΙΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΧΤΑ  
ΑΚΑΔΗΜΑΪΚΑ  
ΜΑΘΗΜΑΤΑ



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

Ενότητα 2: Βασικά στοιχεία ενός προγράμματος στη  
Fortran 90/95/2003 - Δομή Προγράμματος

Νίκος Καραμπετάκης  
Τμήμα Μαθηματικών

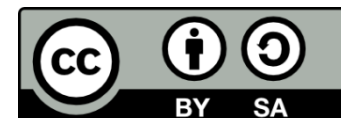


Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Άδειες Χρήσης

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



# Χρηματοδότηση

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



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ  
2007-2013  
πρόγραμμα για την ανάπτυξη  
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



# Περιεχόμενα Ενότητας

1. Πότε δημιουργήθηκε η FORTRAN ;
2. Ποιό είναι το αλφάβητο της FORTRAN ;
3. Ποιό είναι το λεξιλόγιο της FORTRAN ;
4. Ποιά είναι τα δεδομένα που χειρίζεται ένα πρόγραμμα και σε ποιες κατηγορίες χωρίζονται;
5. Εκφράσεις στη FORTRAN.
6. Στοιχειώδεις συναρτήσεις στη FORTRAN.
7. Δομή ενός προγράμματος στη FORTRAN.



# Σκοποί Ενότητας

1. Η παρουσίαση μιας συνοπτικής ιστορίας της γλώσσας προγραμματισμού FORTRAN.
2. Η παρουσίαση του αλφαβήτου και του λεξιλογίου της FORTRAN.
3. Η παρουσίαση των δομών δεδομένων που χρησιμοποιούμε σε ένα πρόγραμμα.
4. Η παρουσίαση του τρόπου δημιουργίας και εκτέλεσης εκφράσεων.
5. Η παρουσίαση των στοιχειωδών συναρτήσεων της FORTRAN.
6. Η παρουσίαση της δομής ενός προγράμματος στη FORTRAN.



# Πότε δημιουργήθηκε η Fortran; (1)

- 1954-1957** IBM (John Backus ).
- 1966** Standard έκδοση της Fortran 66 (διδασκαλία, υπορουτίνες, ανεξάρτητη μετάφραση, ανεξαρτησία από μηχάνημα).
- 1978** Fortran 77 (όχι τόσο ικανοποιητική).
- 1991** Fortran 90 (σημαντικές βελτιώσεις).
- 1997** Fortran 95 (όχι σημαντικές βελτιώσεις).
- 2003** Βελτιωμένη έκδοση με πολλά νέα χαρακτηριστικά που αφορούν μεταξύ άλλων : α) τον αντικειμενοστραφή προγραμματισμό, β) τον χειρισμό δεδομένων, γ) procedure pointers, δ) συνεργασία με την γλώσσα προγραμματισμού C.
- Fortran 2008** Αναμένεται η δημιουργία μιας διορθωτικής έκδοσης της Fortran 2003.



# Πότε δημιουργήθηκε η Fortran; (2)

- Η πρώτη βασική έκδοση της Fortran εμφανίζεται στα μέσα του 1954.
- Η Fortran II δημιουργήθηκε το 1958.
- Η Fortran III δημιουργήθηκε το 1958 και δεν κυκλοφόρησε ποτέ ως προϊόν.
- Η Fortran IV κυκλοφόρησε το 1962.
- Η Fortran 66 αποτελεί την πρώτη έκδοση που κυκλοφόρησε εμπορικά.
- Η Fortran 77 κυκλοφόρησε το 78 χωρίς να περιέχει σημαντικές βελτιώσεις.



# Πότε δημιουργήθηκε η Fortran; (3)

- Η Fortran 90 περιέχει σημαντικές βελτιώσεις.
- Η Fortran 95 κυκλοφορεί το 97 και δεν αποτελεί σημαντική βελτίωση της προηγούμενης.
- Η Fortran 2003 αποτελεί μία νέα βελτιωμένη έκδοση με πολλά νέα χαρακτηριστικά.
- Η Fortran 2008 αναμένεται η δημιουργία μιας διορθωτικής έκδοσης της Fortran 2003.





# Νέα χαρακτηριστικά της Fortran 90/95

- ο τρόπος γραφής του προγράμματος,
- η ευκολία στην διαχείριση πινάκων,
- η δυναμική χρήση της μνήμης του Η/Υ,
- δυνατότητα ορισμού νέων τύπων δεδομένων,
- χρήση αναδρομικών συναρτήσεων και διαδικασιών,
- δυνατότητα δημιουργίας βιβλιοθηκών (Modules),
- νέες δομές ελέγχου.



# Ποιο είναι το αλφάβητο της Fortran 90/95;

- **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 ' " ( ) \* + - / : = \_ ! & \$ ; < > % ? , .



# Ποιο είναι το λεξιλόγιο της Fortran 90/95;

- Ονόματα.
- Αριθμούς.
- Χαρακτήρες.
- Αλυσίδες χαρακτήρων.
- Λογικοί χαρακτήρες.
- Τελεστές.
- Διαχωριστές.
- Λέξεις κλειδιά.
- Σχόλια.



# Επιτρεπτά Ονόματα

Τα **ονόματα** χρησιμοποιούνται για προσδιορισμό μεταβλητών ή σταθερών, ονομασία υπορουτίνων, προγραμμάτων κλπ.

ΕΠΙΤΡΕΠΤΑ ΟΝΟΜΑΤΑ
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.



# Μη Επιτρεπτά Ονόματα

Τα **ονόματα** χρησιμοποιούνται για προσδιορισμό μεταβλητών ή σταθερών, ονομασία υπορουτίνων, προγραμμάτων κλπ.

ΜΗ ΕΠΙΤΡΕΠΤΑ ΟΝΟΜΑΤΑ
Δεν επιτρέπονται τα σύμβολα εκτός από το <code>_</code> .
Δεν επιτρέπονται τα κενά.
Δεν επιτρέπονται τα Ελληνικά γράμματα.

- \* Στην Fortran 2003 τα ονόματα μπορούν να περιέχουν έως 63 χαρακτήρες.



# Παραδείγματα

- 1 VELOCITY  $\rightarrow$  X (Πρώτο γράμμα αριθμός)
- UT  $\rightarrow$   $\surd$
- A\$1  $\rightarrow$  X (Περιέχει σύμβολα)
- ANNA  $\rightarrow$  Εξαρτάται από το αν είναι στα Ελληνικά ή στα Αγγλικά
- U\_T  $\rightarrow$   $\surd$



# Αριθμοί (Ακέραιοι αριθμοί)

## 1. Ακέραιοι Αριθμοί

*Μέγεθος* :  $[-2^7, 2^7 - 1]$ ,  $[-2^{15}, 2^{15} - 1]$ ,  $[-2^{31}, 2^{31} - 1]$

*Σημαντικά ψηφία* : 10

Αριθμοί

Επιτρεπτοί

Μη επιτρεπτοί

23

3.

4.0

123456789012

+45



# 1. Αναπαράσταση ακέραιων αριθμών (1)

$$d = (a_n, a_{n-1}, \dots, a_1, a_0)_{10} = 10^n a_n + \dots + 10^1 a_1 + 10^0 a_0$$

↓

$$(c_m, c_{m-1}, \dots, c_1, c_0)_2 = 2^m c_m + \dots + 2^1 c_1 + 2^0 c_0$$

↓

$$d = 2^m c_m + \dots + 2^1 c_1 + 2^0 c_0 = 2 \left( \underbrace{2^{m-1} c_m + \dots + 2^1 c_2 + 2^0 c_1}_{d_0} \right) + c_0$$

$$= 2 \left( 2 \left( \underbrace{2^{m-2} c_m + \dots + 2^1 c_3 + 2^0 c_2}_{d_1} \right) + c_1 \right) + c_0 = \dots =$$

$$= 2 \left( 2 \left( \dots \left( \underbrace{2c_m + c_{m-1}}_{d_{m-1}} \right) \dots + c_1 \right) + c_0 \right)$$





## 2. Αναπαράσταση ακέραιων αριθμών (2)

Έστω

$$d = (12)_{10}$$

↓

$$12 = 2 * \underbrace{6}_{d_0} + \underbrace{0}_{c_0}$$

$$\underbrace{6}_{d_0} = 2 * \underbrace{3}_{d_1} + \underbrace{0}_{c_1}$$

$$\underbrace{3}_{d_1} = 2 * \underbrace{1}_{d_2} + \underbrace{1}_{c_2}$$

$$\underbrace{1}_{d_2} = 2 * \underbrace{0}_{d_3} + \underbrace{1}_{c_3}$$

$$(c_3 c_2 c_1 c_0)_2 = (1100)_2 = 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 0 * 2^0$$



# 3. Αναπαράσταση ακέραιων αριθμών (3)

Sign	$c_6$	$c_5$	$c_4$	$c_3$	$c_2$	$c_1$	$c_0$
------	-------	-------	-------	-------	-------	-------	-------

+12 →	0	0	0	0	1	1	0	0
-------	---	---	---	---	---	---	---	---

**Αρνητικοί αριθμοί**  
 Συμπλήρωμα ως προς ένα ή *one-pass complement*

-12 →	1	1	1	1	0	0	1	1
-------	---	---	---	---	---	---	---	---

$0 \rightarrow 1 \quad 1 \rightarrow 0$   
 $0 \rightarrow (00000000)_2, (11111111)_2$

*Συμπλήρωμα ως προς δύο ή two-pass complement*  
 $(11110011)_2 + (00000001)_2 = (11110100)_2$   
 $0+0=0, 0+1=1, 1+0=1, 1+1 = 10$  (άθροισμα =0, κρατούμενο=1)



# Δεν μπορούμε να αποθηκεύσουμε στην Fortran όσο μεγάλους ακεραίους θέλουμε!

Sign	$c_6$	$c_5$	$c_4$	$c_3$	$c_2$	$c_1$	$c_0$
$2^8=256$ συνδυασμοί							

**Συμπλήρωμα ως προς ένα ή *one-pass complement***

$$2^8-1=256-1=255 \text{ συνδυασμοί}$$

$$-(2^7-1)=-127 \text{ --- } +(2^7-1)=+127$$

Σημαντικά Ψηφία 3

**Συμπλήρωμα ως προς δύο ή *two-pass complement***

$$-2^7=-128 \text{ --- } +(2^7-1)=+127$$

2 byte δηλαδή 16 bit $\rightarrow [-2^{15}, 2^{15}-1]$ Σημαντικά Ψηφία 5
4 byte δηλαδή 32 bit $\rightarrow [-2^{31}, 2^{31}-1]$ Σημαντικά Ψηφία 10
8 byte δηλαδή 64 bit $\rightarrow [-2^{63}, 2^{63}-1]$ Σημαντικά Ψηφία 19



# Πρόβλημα overflow

<b>+100</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>
+50	0	0	1	1	0	0	1	0
-106	1	0	0	1	0	1	1	0

↓

$$(10010110)_2 - (00000001)_2 = (10010101)_2 = -(01101010)_2 = -106$$

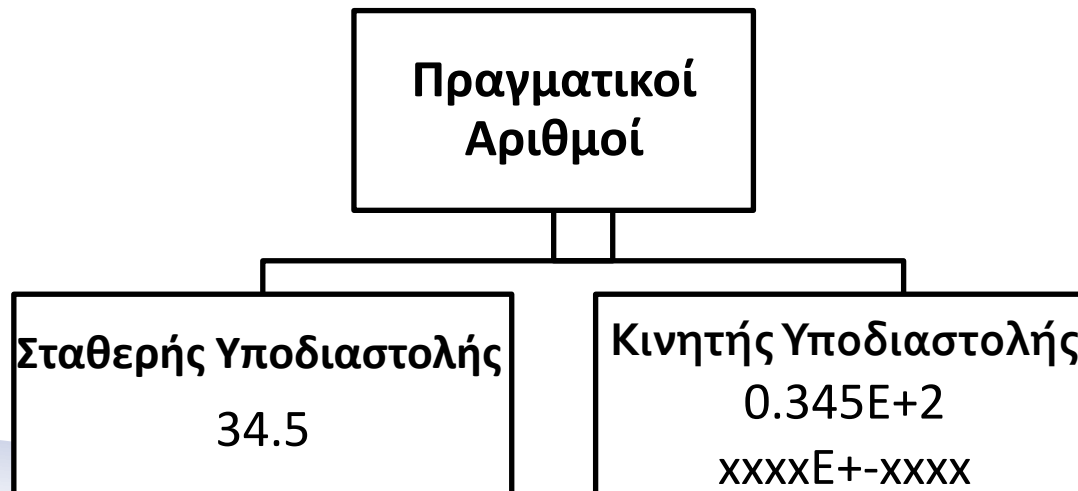
<b>100+50=100+(27+23)</b>
127+1=-128=(-129+1)
127+2=-127=(-129+2)
....
127+23=-106=(-129+23)



# 1. Αριθμοί (Πραγματικοί αριθμοί)

## 2. Πραγματικοί Αριθμοί

Πραγματικοί Αριθμοί	Απλής ακρίβειας	Διπλής ακρίβειας
Μέγεθος:	$[-10^{38}, -10^{-38}],$ $[10^{-38}, 10^{38}]$	$[-10^{308}, -10^{-308}],$ $[10^{-308}, 10^{308}]$
Σημαντικά ψηφία :	6	15



## 2. Αριθμοί (Πραγματικοί αριθμοί)

Αριθμοί	Επιτρεπτοί	Μη επιτρεπτοί
12,3		
3.24		
1.234.345		
3.		
1E+40		
12E+1.2		

**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



# Αναπαράσταση πραγματικών αριθμών

$$d = (a_n a_{n-1} \dots a_1 a_0 . b_0 b_1 \dots)_{10} = 10^n a_n + \dots + 10a_1 + a_0 + 10^{-1}b_0 + 10^{-2}b_1 + \dots$$

↓

$$(c_m c_{m-1} \dots c_1 c_0 . d_0 d_1 \dots)_2 = 2^m c_m + \dots + 2^1 c_1 + c_0 + 2^{-1}d_0 + 2^{-2}d_1 + \dots$$

Πρώτα μετατρέπουμε το ακέραιο μέρος σε δυαδική μορφή.

$$d = (a_n a_{n-1} \dots a_1 a_0)_{10} = 10^n a_n + \dots + 10a_1 + a_0$$

↓

$$(c_m c_{m-1} \dots c_1 c_0)_2 = 2^m c_m + \dots + 2^1 c_1 + c_0$$

Μετά το δεκαδικό μέρος σε δυαδική μορφή.

$$x = (0 . b_0 b_1 \dots)_{10} = 10^{-1}b_0 + 10^{-2}b_1 + \dots$$

↓

$$x = (0 . d_0 d_1 \dots)_2 = 2^{-1}d_0 + 2^{-2}d_1 + \dots$$



# Μετατροπή δεκαδικού μέρους σε δυαδική μορφή (1)

$$x = (0.d_0d_1 \dots)_2 = 2^{-1}d_0 + 2^{-2}d_1 + \dots$$

↓

$$2x = d_0 + 2^{-1}d_1 + 2^{-2}d_2 + \dots = \left( d_0 \cdot \underbrace{d_1d_2 \dots}_{x_0} \right)_2$$

↓

$$2x_0 = 2 \left( \underbrace{2^{-1}d_1 + 2^{-2}d_2 + \dots}_{x_0} \right) = \left( d_1 \cdot \underbrace{d_2d_3 \dots}_{x_1} \right)_2$$

$$12.375 \rightarrow (1100.011)_2$$

$$(12)_{10} = (1100)_2$$

$$x = (0.375)_{10} \rightarrow (0.011)_2$$

$$2x = 2 \times 0.375 = 0.750 \Rightarrow d_0 = 0, x_0 = 0.750$$

$$2x_0 = 2 \times 0.750 = 1.50 \Rightarrow d_1 = 1, x_1 = 0.50$$

$$2x_1 = 2 \times 0.50 = 1.00 \Rightarrow d_2 = 1, x_2 = 0.00$$





# Μετατροπή δεκαδικού μέρους σε δυαδική μορφή (2)

$$x = 0.1$$

$$2x = 2 \times 0.1 = 0.2 \Rightarrow d_0 = 0, x_0 = 0.2$$

$$2x_0 = 2 \times 0.2 = 0.4 \Rightarrow d_1 = 0, x_1 = 0.4$$

$$2x_1 = 2 \times 0.4 = 0.8 \Rightarrow d_2 = 0, x_2 = 0.8$$

$$2x_2 = 2 \times 0.8 = 1.6 \Rightarrow d_3 = 1, x_3 = 0.6$$

$$2x_3 = 2 \times 0.6 = 1.2 \Rightarrow d_4 = 1, x_4 = 0.2$$

↓

$$x = (0.1)_{10} = (0.0001100110011 \dots)_2 = (0.0\overline{0011})_2$$

Υπάρχουν δεκαδικοί αριθμοί που δεν διαθέτουν πεπερασμένη αριθμητική παράσταση στο δυαδικό σύστημα!!



# Κανονικοποιημένη παράσταση κινητής υποδιαστολής (Δεκαδικής μορφής)

$(a_n a_{n-1} \dots a_1 a_0 . b_0 b_1 \dots)_{10}$	12.375
$(a_n a_{n-1} \dots a_1 . a_0 b_0 b_1 \dots)_{10} 10^1$	$1.2375 \times 10^1$
$(a_n a_{n-1} \dots a_2 . a_1 a_0 b_0 b_1 \dots)_{10} 10^2$	$0.12375 \times 10^2$
....	
$(0 . a_n a_{n-1} \dots a_1 a_0 b_0 b_1 \dots)_{10} 10^n$	$0.012375 \times 10^3$

Κανονικοποιημένη παράσταση κινητής υποδιαστολής.  
εκθέτης

$$x = \pm r \times 10^n, \left( \frac{1}{10} \leq r < 1 \right) \quad 0.12375 \times 10^2$$

πρόσημο

κλασματικό μέρος

Το πλήθος των ψηφίων του  $r$  ονομάζονται **σημαντικά ψηφία** (significant digits).



# Κανονικοποιημένη παράσταση κινητής υποδιαστολής (Δυαδικής μορφής)

$(c_m c_{m-1} \dots c_1 c_0 \cdot d_0 d_1 \dots)_2$	$(1100.011)_2$
$(c_m c_{m-1} \dots c_1 \cdot c_0 d_0 d_1 \dots)_2 2^1$	$(110.0011)_2 \times 2^1$
$(c_m c_{m-1} \dots c_2 \cdot c_1 c_0 d_0 d_1 \dots)_2 2^2$	$(11.00011)_2 \times 2^2$
	$(1.100011)_2 \times 2^3$
$(0. c_m c_{m-1} \dots c_2 c_1 c_0 d_0 d_1 \dots)_2 2^{m+1}$	$(0.1100011)_2 \times 2^4$

Κανονικοποιημένη παράσταση κινητής υποδιαστολής.

εκθέτης (δυαδική μορφή)

$$x = \pm q \times 2^m, \left(\frac{1}{2} \leq q < 1\right)$$

$$(0.1100011)_2 \times 2^4 + (0.1100011)_2 \times 2^{(100)_2}$$

πρόσημο

κλασματικό μέρος  
(δυαδική μορφή)



# Κανονικοποιημένη παράσταση κινητής υποδιαστολής (Δυαδικής μορφής) συνέχεια

Κανονικοποιημένη παράσταση κινητής υποδιαστολής.

$$x = \pm q \times 2^m, \left(\frac{1}{2} \leq q < 1\right)$$

← εκθέτης (δυαδική μορφή)

← πρόσημο

← κλασματικό μέρος (δυαδική μορφή)

$$(0.1100011)_2 \times 2^4 + (0.1100011)_2 \times 2^{(100)_2}$$

Το πλήθος των ψηφίων του  $q$  ονομάζονται **σημαντικά ψηφία** (significant digits). Ο αριθμός των bits που χρησιμοποιείται για το κλασματικό μέρος καλείται **ακρίβεια (precision)** της μηχανής.



# Κανονικοποιημένη παράσταση κινητής υποδιαστολής – Παράδειγμα

Πραγματικοί αριθμοί απλής ακρίβειας  $(0.1d_2d_3 \dots)_2$

$$x = (-1)^s * 2^{e-127} * (1.d_2d_3 \dots d_{24})_2$$

$\downarrow$   
 $(1.d_2d_3 \dots)_2$

$s = 0$  ( $s = 1$ ) για θετικό (αρνητικό) κλασματικό μέρος

$$12.375$$

$$+2^4(0.1100011)_2 = +2^3(1.100011)_2 = (-1)^0 2^{130-127}(1.100011)_2$$

$$s = 0, e = 130 = (10000010)_2$$

Πρόσημο κλασματικού μέρους (1 <sup>η</sup> θέση)	Εκθέτης e (θέση 2-9)	Κλασματικό μέρος $d_2d_3\dots$ (θέση 10-32)
0	10000010	00000000000000000000100011



# Αποθήκευση πραγματικού αριθμού απλής ακρίβειας (32 bit)

$$x = (-1)^s \times 2^{e-127} \times (1.d_2d_3 \dots d_{24})_2$$

Πρόσημο κλασματικού μέρους (1 <sup>η</sup> θέση)	Εκθέτης e (θέση 2-9)	Κλασματικό μέρος d <sub>2</sub> d <sub>3</sub> ... (θέση 10-32)
--	----------------------	---

$$0 < (00000000)_2 < e < (11111111)_2 = 255$$

Οι τιμές 0 και 255 δεσμεύονται για τις τιμές 0 και άπειρο.

$$1 \leq e \leq 254 \Leftrightarrow 1 - 127 \leq e - 127 \leq 254 - 127 \Leftrightarrow -126 \leq e - 127 \leq 127$$

$$1 = (1.00 \dots 0)_2 \leq (1.d_2d_3 \dots d_{33})_2 \leq (1.11 \dots 1)_2 = 2^0 + 2^{-1} + \dots + 2^{-23} = 2 - 2^{-23}$$

Μικρότερος	$2^{-126} = 1.17549435 \times 10^{-38}$
Μεγαλύτερος	$(2 - 2^{-23})2^{127} = 3.40282347 \times 10^{38}$



# Αποθήκευση πραγματικού αριθμού διπλής ακρίβειας (64 bit)

$$x = (-1)^s \times 2^{e-1023} \times (1.d_2d_3 \dots)_2$$

Πρόσημο κλασματικού μέρους (1 <sup>η</sup> θέση)	Εκθέτης e (θέση 2-12)	Κλασματικό μέρος $d_2d_3\dots$ (θέση 13-64)
--	-----------------------	---

Μικρότερος	$2^{-1022} = 2.225073858507201 \times 10^{-308}$
Μεγαλύτερος	$(2 - 2^{-52})2^{1023} = 1.797693134862315 \times 10^{308}$



# Απόλυτο σφάλμα (absolute error)

$$x = (-1)^s \times 2^m \times (1.d_2d_3 \dots)_2, -126 \leq m \leq 127$$

Μέθοδος της αποκοπής (chopping ή rounding down).

$$x^* = (-1)^s \times 2^m \times (1.d_2d_3 \dots d_{24})_2$$

Απόλυτο σφάλμα (absolute error)

$$\begin{aligned} |x - x^*| &= \left| 2^m \times \left( 0.\underbrace{00 \dots 0}_{23} d_{25}d_{26} \dots \right)_2 \right| \leq \left| 2^m \times \left( 0.\underbrace{00 \dots 0}_{23} 11 \dots \right)_2 \right| \\ &= 2^m \left( \sum_{i=0}^{+\infty} 2^{-24-i} \right) = 2^m 2^{-23} = 2^{-23+m} \end{aligned}$$

$$\begin{aligned} |q - q^*| &= \left( 0.\underbrace{00 \dots 0}_{23} d_{25}d_{26} \dots \right)_2 \leq \left( 0.\underbrace{00 \dots 0}_{23} 11 \dots \right)_2 = \sum_{i=0}^{+\infty} 2^{-24-i} \\ &= 2^{-23} \end{aligned}$$





# Απόλυτο σχετικό σφάλμα (absolute relative error)

$$\begin{aligned}
 \left| \frac{x - x^*}{x} \right| &= \left| \frac{(-1)^s 2^m \times \left( \underbrace{0.00 \dots 0}_{23} d_{25} d_{26} \dots \right)_2}{(-1)^s \times 2^m \times \left( 1. \underbrace{d_2 d_3 \dots d_{24}}_{23} d_{25} d_{26} \dots \right)_2} \right| = \\
 &= \frac{\left( \underbrace{0.00 \dots 0}_{23} d_{25} d_{26} \dots \right)_2}{\left( 1. \underbrace{d_2 d_3 \dots d_{24}}_{23} 00 \dots \right)_2 + \left( \underbrace{0.00 \dots 0}_{23} d_{25} d_{26} \dots \right)_2} \leq \\
 &\leq \frac{\left( \underbrace{0.00 \dots 0}_{23} 11 \dots \right)_2}{\left( 1. \underbrace{d_2 d_3 \dots d_{24}}_{23} 00 \dots \right)_2 + \left( \underbrace{0.00 \dots 0}_{23} 11 \dots \right)_2} \leq \\
 &\leq \frac{\left( \underbrace{0.00 \dots 0}_{23} 11 \dots \right)_2}{\left( 1. \underbrace{00 \dots 00}_{23} 00 \dots \right)_2 + \left( \underbrace{0.00 \dots 0}_{23} 11 \dots \right)_2} = \frac{\sum_{i=0}^{+\infty} 2^{-24-i}}{1 + \sum_{i=0}^{+\infty} 2^{-24-i}} = \frac{2^{-23}}{1 + 2^{-23}} \leq 2^{-23} = \\
 &= 0.11920928955078125 \times 10^{-6} \text{ έψιλον της μηχανής (machine epsilon)}
 \end{aligned}$$



# Τι γίνεται αν θέλουμε να αποθηκεύσει το 0.1 ;

$x = (0.00011001100110011001100110011 \dots)_2$		
↓		
0	1111011	10011001100110011001100
$x^* = 0.09999999403953552$		

Απόλυτο σφάλμα  $0.596046448 \times 10^{-8}$

Απόλυτο σχετικό σφάλμα  $0.596046448 \times 10^{-7}$

Θα λέμε ότι ο πραγματικός αριθμός  $x^*$  προσεγγίζει τον αριθμό  $x$  σε  $k$  σημαντικά ψηφία (ή αλλιώς είναι ακριβής σε  $k$  σημαντικά ψηφία) αν ο αριθμός  $k$  είναι ο μεγαλύτερος μη-αρνητικός ακέραιος για τον οποίο ισχύει ότι το απόλυτο σχετικό σφάλμα είναι μικρότερο της μισής μονάδας που αντιστοιχεί στο  $k$  σημαντικό ψηφίο.



# Στρογγυλοποίηση προς τα πάνω (rounding up)

$$x = (-1)^s \times 2^m \times (1.d_2d_3 \dots)_2, -126 \leq m \leq 127$$

Αν το  $d_{25} = 0$  τότε το  $d_{24}$  παραμένει ως έχει ενώ στην περίπτωση που  $d_{25} = 1$  τότε το  $d_{24}$  αυξάνει κατά ένα δηλαδή

$$x^{**} = (-1)^s \times 2^m \times [(1.d_2d_3 \dots d_{24})_2 + 2^{-23}]$$

Στην ειδική περίπτωση όμως που  $d_i = 1, i = 1, 2, \dots, 25$  τότε θα πάρουμε

$d_1 = 1, d_i = 0, i = 2, \dots, 24$  και θα αυξηθεί ο εκθέτης κατά ένα.

$$|x - x^{**}| = \left| 2^m \times \left( \underbrace{0.00 \dots 1}_{23} d_{25} d_{26} \dots \right)_2 \right| \leq \left| 2^m \times \left( \underbrace{0.00 \dots 1}_{23} 11 \dots \right)_2 \right|$$

$$= 2^m \left( \sum_{i=0}^{+\infty} 2^{-23-i} \right) = 2^m 2^{-22} = 2^{-22+m}$$

$$\left| \frac{x - x^{**}}{x} \right| = \left| \frac{2^m \times \left( \underbrace{0.00 \dots 1}_{23} d_{25} d_{26} \dots \right)_2}{(-1)^s \times 2^m \times \left( \underbrace{1.d_2 \dots d_{24}}_{23} d_{25} d_{26} \dots \right)_2} \right| \leq \frac{\left( \underbrace{0.00 \dots 1}_{23} d_{25} d_{26} \dots \right)_2}{\left( \underbrace{1.d_2 \dots d_{24}}_{23} d_{25} d_{26} \dots \right)_2}$$

$$\leq \frac{2^{-22}}{1 + 2^{-22}} \leq 2^{-22} = 0.238419 \times 10^{-6}$$



# Τι γίνεται αν θέλουμε να αποθηκεύσει το 0.1 ; ...συνέχεια

$x = (0.00011001100110011001100110011 \dots)_2$		
↓		
0	1111011	10011001100110011001101
↓		
$x^{**} = 0.10000000149011611938$		

Απόλυτο σφάλμα  $1.49011611 \times 10^{-9}$

Απόλυτο σχετικό σφάλμα  $0.149011611 \times 10^{-7}$

Θα λέμε ότι ο πραγματικός αριθμός  $x^*$  προσεγγίζει τον αριθμό  $x$  σε  $k$  σημαντικά ψηφία (ή αλλιώς είναι ακριβής σε  $k$  σημαντικά ψηφία) αν ο αριθμός  $k$  είναι ο μεγαλύτερος μη-αρνητικός ακέραιος για τον οποίο ισχύει ότι το απόλυτο σχετικό σφάλμα είναι μικρότερο της μισής μονάδας που αντιστοιχεί στο  $k$  σημαντικό ψηφίο.



# Συμπέρασμα!!!

- a) Δεν είναι δυνατό να αποθηκεύουμε όσο μεγάλους ή μικρούς αριθμούς θέλουμε (κίνδυνος overflow ή underflow).
- b) Υπάρχει τις περισσότερες φορές μια διαφορά μεταξύ της ακριβής τιμής του αριθμού που δίνουμε και της τιμής που ουσιαστικά δέχεται ο υπολογιστής στην μνήμη του (απόλυτο σφάλμα).

Τα σφάλματα στα αποτελέσματα που θα πάρουμε ακολουθώντας έναν συγκεκριμένο αλγόριθμο μπορεί να παραμείνουν μικρά οπότε μιλάμε για **ευσταθή αλγόριθμο (stable)** ή να διογκωθούν οπότε μιλάμε για **ασταθή αλγόριθμο (unstable)**.

Μπορεί τα αποτελέσματα να εξαρτώνται περισσότερο από τις τιμές των αρχικών δεδομένων που θα εισαχθούν στο προς επίλυση πρόβλημα παρά από τον ίδιο τον αλγόριθμο. Τότε λέμε ότι έχουμε ένα **πρόβλημα κακής κατάστασης (ill-condition problem)**.



# Αριθμοί (Μιγαδικοί αριθμοί)

## 3. Μιγαδικοί Αριθμοί

$$3+4i \rightarrow (3,4)$$

**Μέγεθος:**  $\rightarrow$  Δύο πραγματικοί αριθμοί απλής ή διπλής ακρίβειας

**Σημαντικά ψηφία:**



# Χαρακτήρες

Μεταξύ αποστρόφων π.χ. “a”, “b”

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



# Αλυσίδα χαρακτήρων – Λογικοί χαρακτήρες

**Αλυσίδα χαρακτήρων:** Ακολουθία χαρακτήρων μεταξύ αποστρόφων π.χ. “Fortran90”, “123”, “don’t forget”

‘NIKOS’                                      ?

‘ANNA’                                      ?

‘don’t forget’                              ?

**Λογικοί χαρακτήρες:** .TRUE. .FALSE.





# Τελεστές (αριθμητικοί τελεστές)

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



# Τελεστές (τελεστές χαρακτήρων)

Τελεστής	Λειτουργία	Σύνταξη
//	Συγχώνευση Αλυσίδων Χαρακτήρων	"ABC"//"123" θα μας δώσει "ABC123"



# Τελεστές (τελεστές σύγκρισης)

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



# Σειρά βαρύτητας χαρακτήρων

(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

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

"A1"<"B1"

"A1"<"A2"

"ABC"<"ABC123"



# Δραστηριότητα

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

“GIORGOS” ..... “ ΓΙΩΡΓΟΣ”

“ ΝΙΚΟΣ” ..... “ΝΙΚΟΣ”

“1234” .... “123”

“ANNA” ..... “ANNA”

“ΝΙΚΟΣ” .... “ΝΙΚΟΣ”

123 ..... “123”

“\_ABC” ..... “ABC\_”



# Τελεστές (λογικοί τελεστές) (1)

Τελεστής	Λειτουργία
.NOT.	Λογική άρνηση
.AND.	Λογική Πρόσθεση
.OR.	Διάζευξη
.EQV.	Ισοδυναμία
.NEQV.	Μη ισοδύναμα
.XOR.	ή/και



# Τελεστές (λογικοί τελεστές) (2)

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

(5>7).OR.(9>2)                    ?  
(9>2).AND.(2<3)                   ?  
.NOT.(4>3)                            ?



# Τελεστές (λογικοί τελεστές) (3)

X	Y	X.EQV.Y	X.NEQV.Y	X.XORY
T	T	T	F	F
T	F	F	T	T
F	T	F	T	T
F	F	T	F	F



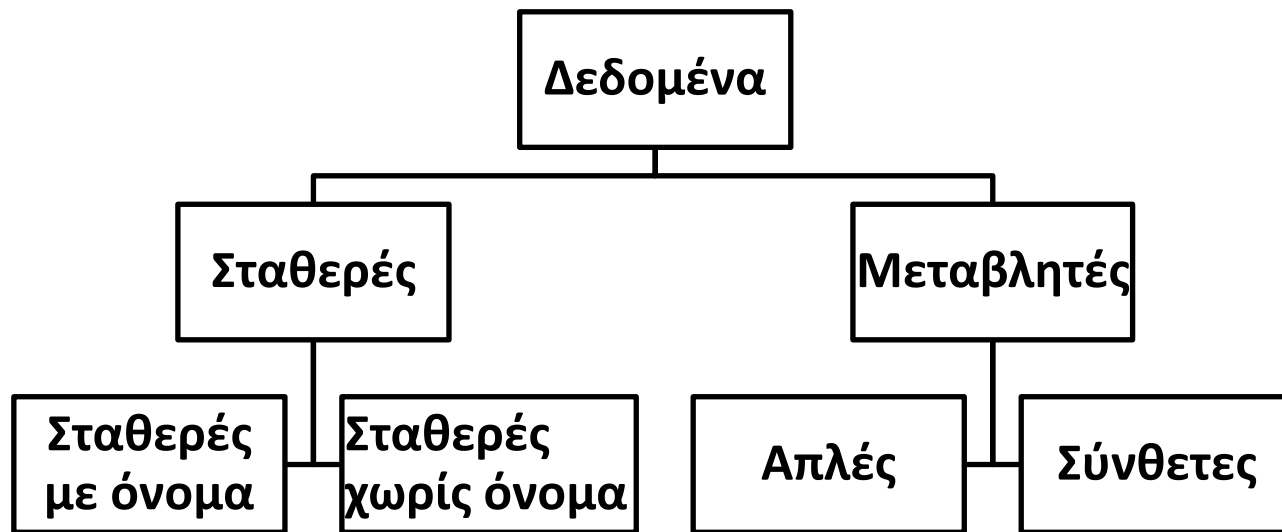


# Διαχωριστές – Λέξεις Κλειδιά – Σχόλια

- **Διαχωριστές** (Χαρακτήρες με συγκεκριμένη λειτουργία)  
= Καταχώριση.  
( ) Παρενθέσεις, για τους πίνακες.  
,. Σημεία στίξης.  
& Σύμβολο συνέχειας γραμμής.  
; Για τον διαχωρισμό εντολών που βρίσκονται στην ίδια γραμμή.
- **Λέξεις Κλειδιά** (Λέξεις με ειδική σημασία)  
PROGRAM, IF, THEN, ELSE, DO, WHILE κλπ.
- **Σχόλια**  
*! Το παρακάτω πρόγραμμα ..*



Ποια είναι τα δεδομένα που χειρίζεται ένα πρόγραμμα και σε ποιες κατηγορίες χωρίζονται;



A	34.4
B(1)	35
B(2)	1-E2



# Ποια είναι τα δεδομένα που χειρίζεται ένα πρόγραμμα και σε ποιες κατηγορίες χωρίζονται; (Ακέραιοι)

## 1. Ακέραιοι

Όνομα	Σύνολο Τιμών	Bytes	Επιτρεπτοί Τελεστές
<b>INTEGER</b>	$[-2^{31}, 2^{31}-1]$	4	Αριθμητικοί, Σύγκρισης
<b>INTEGER(1)</b>	$[-2^7, 2^7-1]$	1	Αριθμητικοί, Σύγκρισης
<b>INTEGER(2)</b>	$[-2^{15}, 2^{15}-1]$	2	Αριθμητικοί, Σύγκρισης
<b>INTEGER(4)</b>	$[-2^{31}, 2^{31}-1]$	4	Αριθμητικοί, Σύγκρισης



# Ποια είναι τα δεδομένα που χειρίζεται ένα πρόγραμμα και σε ποιες κατηγορίες χωρίζονται; (Πραγματικοί)

## 2. Πραγματικοί

<b>REAL</b>	$[-10^{38}, -10^{-38}], [10^{-38}, 10^{38}]$	4	Αριθμητικοί, Σύγκρισης
<b>REAL(4)</b>	$[-10^{38}, -10^{-38}], [10^{-38}, 10^{38}]$	4	Αριθμητικοί, Σύγκρισης
<b>DOUBLE PRECISION</b>	$[-10^{308}, -10^{-308}], [10^{-308}, 10^{308}]$	8	Αριθμητικοί, Σύγκρισης
<b>REAL(8)</b>	$[-10^{308}, -10^{-308}], [10^{-308}, 10^{308}]$	8	Αριθμητικοί, Σύγκρισης



# Ποια είναι τα δεδομένα που χειρίζεται ένα πρόγραμμα και σε ποιες κατηγορίες χωρίζονται; (Μιγαδικοί)

## 3. Μιγαδικοί

<b>COMPLEX</b>	Το πραγματικό και φανταστικό μέρος έχει μέγεθος που ορίζεται στον τύπο REAL	8	Αριθμητικοί
<b>COMPLEX(4)</b>	Το πραγματικό και φανταστικό μέρος έχει μέγεθος που ορίζεται στον τύπο REAL(4)	8	Αριθμητικοί
<b>DOUBLE COMPLEX</b>	Το πραγματικό και φανταστικό μέρος έχει μέγεθος που ορίζεται στον τύπο DOUBLE PRECISION	16	Αριθμητικοί
<b>COMPLEX(8)</b>	Το πραγματικό και φανταστικό μέρος έχει μέγεθος που ορίζεται στον τύπο DOUBLE PRECISION	16	Αριθμητικοί



Ποια είναι τα δεδομένα που χειρίζεται ένα πρόγραμμα και σε ποιες κατηγορίες χωρίζονται; (Χαρακτήρες)

#### 4. Χαρακτήρες

<b>CHARACTER</b>	Ένας χαρακτήρας	1	Αλφαριθμητικοί, Σύγκρισης
<b>CHARACTER*N</b>	Αλυσίδα n χαρακτήρων	N	Αλφαριθμητικοί, Σύγκρισης



Ποια είναι τα δεδομένα που χειρίζεται ένα πρόγραμμα και σε ποιες κατηγορίες χωρίζονται; (Λογικοί Χαρακτήρες)

<b>LOGICAL</b>	Όπως η LOGICAL(4)	4	Λογικοί
<b>LOGICAL(1)</b>	.FALSE.(0) ή .TRUE.(1)	1	Λογικοί
<b>LOGICAL(2)</b>	Το λιγότερο σημαντικό είναι ίδιο με το LOGICAL(1) ενώ το πιο σημαντικό byte είναι μη ορισμένο.	2	Λογικοί
<b>LOGICAL(4)</b>	Η λιγότερο σημαντική λέξη (word) είναι ίδια με το LOGICAL(2) ενώ η πιο σημαντική λέξη είναι μη ορισμένη.	4	Λογικοί



# Εκφράσεις στη Fortran 90/95 – Σειρά προτεραιότητας πράξεων

5+x: 5 → τελεστέος, + → τελεστής

Τύπος	Τελεστής	Σειρά προτεραιότητας για τελεστές με την ίδια προτεραιότητα
Αριθμητικός	** * / + -	Δεξιά προς αριστερά Αριστερά προς τα δεξιά Αριστερά προς τα δεξιά
Σύγκρισης	< <= > >= == /=	Αριστερά προς τα δεξιά
Λογικοί	.NOT. .AND. .OR. .EQV. .NEQV.	Δεξιά προς αριστερά Αριστερά προς τα δεξιά Αριστερά προς τα δεξιά Αριστερά προς τα δεξιά





# Εκφράσεις στη Fortran 90/95 – Ιδιαιτερότητες

Πράξη	Μετατροπή	Αποτέλεσμα
{Ακέραιος} ⊕ {Ακέραιος}		{Ακέραιος}
{Ακέραιος} ⊕ { Πραγματικός}	{Πραγματικός} ⊕ {Πραγματικός}	{Πραγματικός}



# Εκφράσεις στη Fortran 90/95 - Παραδείγματα

- $3^{**}2^{**}3$  ?
- $2*(5+1)*3/4^{**}2$  ?
- $1.2*(1+5/100)^{**}10$  ?
- $3*(6.0-2)^{**}2/4+5$  ?
- $(4>3).OR.(3<1).AND.(.NOT.(3>1))$  ?
- $3^{**}(1/2)$  ?
- $0.25^*E-20/0.25E+20$  ?
- $(-2)^{**}2.0$  ?
- $(2+3)/2$  ?



# Εκφράσεις στη Fortran 90/95 – Λάθη από μη επιτρεπτές πράξεις

1. Division by Zero.
2. Σωστό όρισμα, (π.χ. το υπόριζο να είναι θετικό).
3. Overflow ή Underflow.
4. Η παράσταση των αριθμών στον Η/Υ γίνεται στο δυαδικό σύστημα και συνεπώς μπορεί να έχουμε λάθη.
5. Ύψωση αριθμού σε δύναμη.
6. Προσθέσεις πολύ μικρών και πολύ μεγάλων αριθμών.



# 1. Αριθμητικές Συναρτήσεις

Όνομα	Περιγραφή	Όρισμα	Αποτέλεσμα	Παράδειγμα
<b>ABS(X)</b>	Απόλυτη τιμή του X	INTEGER REAL COMPLEX	INTEGER REAL	ABS(-3.1)=3.1 ABS((3,4))=5
<b>SQRT(X)</b>	Τετραγωνική ρίζα του X	REAL COMPLEX	REAL	SQRT(4)=2.0 SQRT((3,4))=(2.0,1.0)
<b>SIN(X)</b>	Ημίτονο του X σε ακτίνια	REAL COMPLEX	REAL	SIN(3.14)=1.594528E-03 SIN((3,4))=(3.853738,-27.016810)



## 2. Αριθμητικές Συναρτήσεις

Όνομα	Περιγραφή	Όρισμα	Αποτέλεσμα	Παράδειγμα
<b>ASIN(X)</b>	Τόξο ημιτόνου του X	REAL	REAL	ASIN(1)=1.570796
<b>COS(X)</b>	Συνημίτονο του X σε ακτίνια	REAL COMPLEX	REAL	COS(3.141592)=-1.00000 COS((1,1))=(8.337300E-01,-9.888977E-01)
<b>ACOS(X)</b>	Τόξο συνημιτόνου του X	REAL	REAL	ACOS(1)=0
<b>TAN(X)</b>	Εφαπτομένη του X σε ακτίνια	REAL	REAL	TAN(0)=0



# 3. Αριθμητικές Συναρτήσεις

Όνομα	Περιγραφή	Όρισμα	Αποτέλεσμα	Παράδειγμα
<b>ATAN(X)</b>	Τόξο εφαπτομένης του X	REAL	REAL	ATAN(1)= 7.853982E-01
<b>EXP(X)</b>		REAL COMPLEX	REAL COMPLEX	EXP(0)=1 EXP((0,1))=(5.403023E-01,8.414710E-01)
<b>LOG(X)</b>	Φυσικός λογάριθμος του X	REAL COMPLEX (INTEGER δημιουργεί λάθος)	REAL COMPLEX	LOG(1)=0 LOG((0,1))=(0.000000E+00,1.570796)
<b>LOG10(X)</b>	Δεκαδικός λογάριθμος του X	REAL	REAL	LOG10(1)=0



# 4. Αριθμητικές Συναρτήσεις

Όνομα	Περιγραφή	Όρισμα	Αποτέλεσμα	Παράδειγμα
<b>MAX(X1, X2[, X3, ...])</b>	Ο μέγιστος των X1,X2,...	INTEGER REAL	INTEGER REAL	MAX(2,9)=9
<b>MIN(X1, X2[, X3, ...])</b>	Ο ελάχιστος των X1,X2,...	INTEGER REAL	INTEGER REAL	MIN(2,9,4)=2
<b>INT(X)</b>	Μετατροπή σε ακέραιο με αποκοπή	INTEGER REAL	INTEGER	INT(+3.9)=3 INT(-3.9)= -3
<b>REAL(X)</b>	Μετατροπή σε πραγματικό	INTEGER REAL	REAL	REAL(3)=3.0000
<b>NINT(X)</b>	Μετατροπή στον πλησιέστερο ακέραιο	INTEGER REAL	INTEGER	NINT(-3.9)=-4 NINT(+3.9)=4



# 5. Αριθμητικές Συναρτήσεις

Όνομα	Περιγραφή	Όρισμα	Αποτέλεσμα	Παράδειγμα
<b>FLOOR(X)</b>	Μεγαλύτερος ακέραιος $\leq X$	REAL	INTEGER	FLOOR(-3.9)= -4 FLOOR(+3.9)= 3
<b>FRACTION(X)</b>	Κλασματικό μέρος του X	REAL	REAL	FRACTION(3.9)=0.9
<b>MOD(X,Y)</b> ή <b>X-INT(X/Y)*Y</b>	Ακέραιο υπόλοιπο της διαίρεσης X/Y	INTEGER	INTEGER	MOD(7,2)=1





# Παραδείγματα Αριθμητικών Συναρτήσεων

Έκφραση	Έκφραση FORTRAN
$\frac{x + y}{x - y}$	$(X + Y)/(X - Y)$
$K(1 + E)^N$	$K * (1 + E) ** N$
$\sqrt[3]{x^2 + y^2}$	$(X ** 2 + Y ** 2) ** (1/3.0)$
$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$	$(-B + SQRT(B ** 2 - 4 * A * C))/(2.0 * A)$
$\frac{\eta\mu(2\chi)}{\sigma\upsilon\nu^2(\chi - 1) + 1}$	$SIN(2 * X) / ((COS(X - 1)) ** 2 + 1)$



# 1. Αλφαριθμητικές Συναρτήσεις

Όνομα	Περιγραφή	Όρισμα	Αποτέλεσμα	Παράδειγμα
<b>LEN(X)</b>	Μήκος της αλυσίδας χαρακτήρων X	Αλυσίδα χαρακτήρων	INTEGER	LEN("GATE□4")=6 LEN("GATE□□□")=7
<b>LEN_TRIM(X)</b>	Μήκος μέχρι τον τελευταίο μη κενό χαρακτήρα του X	Αλυσίδα χαρακτήρων	INTEGER	LEN_TRIM("GATE□4")=6 LEN_TRIM("GATE□□□")=4
<b>TRIM(X)</b>	Αφαιρεί τα τελευταία κενά	Αλυσίδα χαρακτήρων	Αλυσίδα χαρακτήρων	TRIM("GATE□□□")="GATE"
<b>REPEAT(X,N)</b>	Δημιουργεί N αντίγραφα του X	Αλυσίδα χαρακτήρων	Αλυσίδα χαρακτήρων	REPEAT("A",3)="AAA"



## 2. Αλφαριθμητικές Συναρτήσεις

Όνομα	Περιγραφή	Όρισμα	Αποτέλεσμα	Παράδειγμα
<b>ACHAR(X)</b>	Επιστρέφει τον χαρακτήρα που αντιστοιχεί στον αριθμό x του κώδικα ASCII	INTEGER	CHAR	ACHAR(160)="ι"
<b>CHAR(X)</b>	Επιστρέφει τον χαρακτήρα που αντιστοιχεί στον αριθμό x του επεξεργαστή του συγκεκριμένου Η/Υ	INTEGER	CHAR	CHAR(160)="ι" Επειδή ο κώδικας του συγκεκριμένου επεξεργαστή είναι ASCII.



# 3. Αλφαριθμητικές Συναρτήσεις

Όνομα	Περιγραφή	Όρισμα	Αποτέλεσμα	Παράδειγμα
<b>IACHAR(X)</b>	Επιστρέφει τον αριθμό του κώδικα ASCII που αντιστοιχεί στον χαρακτήρα x	CHAR	INTEGER	IACHAR('ι')=160
<b>ICHAR(X)</b>	Επιστρέφει τον αριθμό του κώδικα του συγκεκριμένου επεξεργαστή του Η/Υ που αντιστοιχεί στον χαρακτήρα x	CHAR	INTEGER	ICHAR('ι')=160



# 4. Αλφαριθμητικές Συναρτήσεις

Όνομα	Περιγραφή	Όρισμα	Αποτέλεσμα	Παράδειγμα
<b>LLT(str1,str2)</b>	Αν $str1 < str2$ σύμφωνα με τον κώδικα ASCII επιστρέφει TRUE	CHAR	LOGICAL	LLT('ΝΙΚΟΣ','ANNA')=F
<b>LLE(str1,str2)</b>	Αν $str1 \leq str2$ σύμφωνα με τον κώδικα ASCII επιστρέφει TRUE	CHAR	LOGICAL	LLE('ΝΙΚΟΣ','ANNA')=F



# 5. Αλφαριθμητικές Συναρτήσεις

Όνομα	Περιγραφή	Όρισμα	Αποτέλεσμα	Παράδειγμα
<b>LGT(str1,str2)</b>	Αν $str1 > str2$ σύμφωνα με τον κώδικα ASCII επιστρέφει TRUE	CHAR	LOGICAL	LGT('ΝΙΚΟΣ','ANNA')=T
<b>LGE(str1,str2)</b>	Αν $str1 \geq str2$ σύμφωνα με τον κώδικα ASCII επιστρέφει TRUE	CHAR	LOGICAL	LGE('ΝΙΚΟΣ','ANNA')=T



# Άσκηση

Να μετατραπούν οι παρακάτω αλγεβρικές εκφράσεις σε μορφή αποδεκτή από τη Fortran :

$$(x^3 - y^2)^{2x-1}$$

$$x - 1 \leq y \leq x + 1$$

$$\frac{\eta\mu(\chi) - \sigma\upsilon\nu(\chi)}{\varepsilon\varphi(\chi)^2 + 1}$$

$$e^x \left( \ln x - \left| x + \sqrt{x^2 + 1} \right| \right)$$

$$\log_{10}(x + \sqrt{e^x + 1})$$



# Δομή ενός προγράμματος στην Fortran 90/95/2003

- Την επικεφαλίδα του προγράμματος.
- Το τμήμα των δηλώσεων.
- Το τμήμα των προτάσεων.
- Το τμήμα των υποπρογραμμάτων.





# Επικεφαλίδα - Τέλος

[PROGRAM όνομα προγράμματος] ← Επικεφαλίδα προγράμματος  
END [PROGRAM [όνομα προγράμματος]] ← Τέλος προγράμματος

## Παράδειγμα.

```
PROGRAM HELLO
```

```
    WRITE(*,*) 'HELLO'
```

```
END PROGRAM HELLO    END
```

```
WRITE(*,*) 'HELLO'
```



# Τμήμα Δηλώσεων – Περιοχή σταθερών

- Περιοχή σταθερών [PROGRAM όνομα προγράμματος]
- Περιοχή μεταβλητών Implicit None
- Περιοχή τύπων → **Τμήμα Δηλώσεων (π.χ. Περιοχή σταθερών)**  
.....  
END [PROGRAM [όνομα προγράμματος]]

## Περιοχή Σταθερών

PARAMETER (*vname=const* [ , *vname=const* ] ...)

Τύπος δεδομένων, PARAMETER :: *vname=const* [ , *vname=const* ] ...

### Παράδειγμα.

```
PARAMETER (PI=3.14159,G=9.81)
```

```
REAL, PARAMETER :: PI=3.14, G=9.81
```



# Τμήμα Δηλώσεων – Περιοχή μεταβλητών

- Μεταβλητές που ορίζονται αυτόματα.  
I-N ακέραιες, υπόλοιπες πραγματικές.

- Μεταβλητές που ορίζονται  
στην περιοχή των  
μεταβλητών. →

[PROGRAM όνομα προγράμματος]

Implicit None

Περιοχή σταθερών

**Τμήμα Μεταβλητών**

...

END[PROGRAM[όνομα προγράμματος]]



# Τμήμα Δηλώσεων – Ακέραιες μεταβλητές

## Ακέραιες μεταβλητές

INTEGER [ ( [ KIND = ] *kind-value* ) ] [ , *attrs* ] :: ] *vname*

*kind-value* : 1 (INTEGER\*1), 2 (INTEGER\*2), 4 (INTEGER\*4)

*attrs* : PARAMETER, DIMENSION, PUBLIC, PRIVATE κλπ.

*vname* : όνομα μεταβλητής

## Παράδειγμα.

INTEGER (1) :: LENGTH, HEIGHT

INTEGER (KIND=2) :: VOL

INTEGER (4), PARAMETER :: A=1     ! Η A είναι ακέραια σταθερά με τιμή 1

INTEGER AB                             ! Είναι τύπου 4



# Τμήμα Δηλώσεων – Πραγματικές μεταβλητές

## Πραγματικές μεταβλητές

REAL [ ( [ KIND = ] *kind-value* ) ] [ , *attrs* ] :: ] *vname*

*kind-value* : 4 (REAL\*4), 8 (REAL\*8)

*attrs* : PARAMETER, DIMENSION, PUBLIC, PRIVATE κ.λ.π.

*vname* : όνομα μεταβλητής

### Παράδειγμα.

REAL (4) :: INTEREST

REAL (KIND=8) :: DISTANCE, FACTOR

REAL (8), PARAMETER :: PI=3.14159\_dp\_kind ! Η PI είναι πραγματική  
&σταθερά με τιμή 3.14159  
! Είναι τύπου 4

REAL X



# Τμήμα Δηλώσεων – Μιγαδικές μεταβλητές

## Μιγαδικές μεταβλητές

COMPLEX [ ( [ KIND = ] *kind-value* ) ] [ , *attrs* ] :: ] *vname*

*kind-value* : 4 (COMPLEX\*4), 8 (COMPLEX\*8)

*attrs* : PARAMETER, DIMENSION, PUBLIC, PRIVATE κ.λ.π.

*vname* : όνομα μεταβλητής

## Παράδειγμα.

COMPLEX (4) :: IA

COMPLEX (KIND(4)) IB

COMPLEX (KIND=8) :: CB,CD

COMPLEX (4), PARAMETER :: CA=(1,1) ! Η CA είναι μιγαδική σταθερά &

& με τιμή 1+1\*I

COMPLEX CH

! Είναι τύπου 4



# 1. Συναρτήσεις για μιγαδικούς αριθμούς

Όνομα	Περιγραφή	Όρισμα	Αποτέλεσμα	Παράδειγμα
<b>CMPLX(A,B, kind)</b>	Δημιουργεί τον μιγαδικό $A+B*i$ . Το kind προσδιορίζει το είδος του μιγαδικού που θα πάρουμε.	REAL	COMPLEX	CMPLX(3,4)= $3+4*i$ ή (3,4)
<b>REAL(A,kind)</b>	Επιστρέφει το πραγματικό μέρος του μιγαδικού A. Το kind προσδιορίζει το είδος του μιγαδικού που θα πάρουμε.	COMPLEX	REAL	REAL(3.1,4.2)=3.1
<b>DBLE(A)</b>	Επιστρέφει το πραγματικό μέρος του μιγαδικού A.	COMPLEX	REAL*8	DBLE(3.1,4.2)=3.1



## 2. Συναρτήσεις για μιγαδικούς αριθμούς

Όνομα	Περιγραφή	Όρισμα	Αποτέλεσμα	Παράδειγμα
<b>INT(A)</b>	Επιστρέφει το πραγματικό μέρος του μιγαδικού A, εφόσον πρώτα το μετατρέψει σε ακέραιο.	COMPLEX	INTEGER	INT(3.1,4.2)=3
<b>AIMG(A)</b>	Επιστρέφει το φανταστικό μέρος του μιγαδικού A.	COMPLEX	REAL	AIMAG(3.1,4.2)=4.2
<b>CABS(A)</b>	Επιστρέφει το μέτρο του μιγαδικού A.	COMPLEX	REAL	CABS(3,4)=5
<b>CONJG(A)</b>	Επιστρέφει τον συζυγή του μιγαδικού A.	COMPLEX	COMPLEX	CONJG(3,4)=(3,-4)





# Τμήμα Δηλώσεων – Μεταβλητές χαρακτήρων

## Μεταβλητές χαρακτήρων

CHARACTER (επιλογή μήκους) [, *attribs*] :: *vname*

*επιλογή μήκους* : το πλήθος των χαρακτήρων που δέχεται η μεταβλητή  
(xx ή LEN=xx)

*attribs* : PARAMETER, DIMENSION, PUBLIC, PRIVATE κλπ.

*vname* : όνομα μεταβλητής

### Παράδειγμα.

CHARACTER (10) :: NAME ! Η μεταβλητή NAME δέχεται 10 χαρακτήρες  
ή

CHARACTER (LEN=10) :: NAME

CHARACTER (LEN=10), PARAMETER :: SURNAME='GEORGIΟΥ'

CHARACTER Z ! Η Z δέχεται μόνο ένα χαρακτήρα



# Τμήμα Δηλώσεων – Λογικές μεταβλητές

## Λογικές μεταβλητές

LOGICAL [ ( [ KIND = ] *kind-value* ) ] [ , *attrs* ] :: ] *vname*

*kind-value* : 1 (LOGICAL\*1), 2 (LOGICAL\*2), 4 (LOGICAL\*4)

*attrs* : PARAMETER, DIMENSION, PUBLIC, PRIVATE κλπ.

*vname* : όνομα μεταβλητής

### Παράδειγμα.

LOGICAL (2) :: CONDITION

LOGICAL (2), PARAMETER :: VALUE=.TRUE.

LOGICAL SELECTION



# Τμήμα Δηλώσεων – Περιοχή τύπων

**TYPE (Όνομα νέου τύπου δεδομένων)**

Τύπος δεδομένων :: Όνομα πεδίου1

Τύπος δεδομένων :: Όνομα πεδίου2

.....

**END TYPE (Όνομα νέου τύπου δεδομένων)**

**Παράδειγμα.**

**TYPE STUDENTS**

CHARACTER (LEN=20) :: SURNAME ! ΕΠΩΝΥΜΟ

CHARACTER (LEN=10) :: NAME ! ΟΝΟΜΑ

INTEGER (KIND=2) :: AEM

INTEGER (KIND=1) :: GRADE

**END TYPE STUDENTS**

**TYPE (STUDENTS) :: STUDENT**

[PROGRAM όνομα προγράμματος]

Implicit None

Περιοχή σταθερών

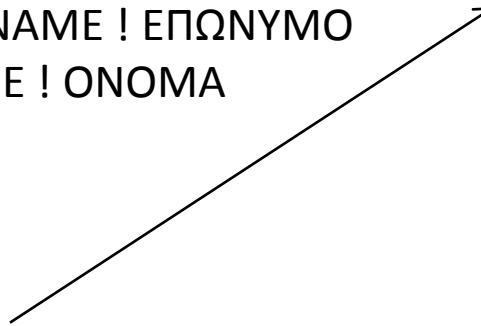
Τμήμα μεταβλητών

**Περιοχή τύπων**

.....

**END [PROGRAM [όνομα προγράμματος]]**

**TYPE (Νέος τύπος δεδομένων) :: Λίστα μεταβλητών**



# Τμήμα Δηλώσεων - Εισαγωγή τιμών σε νέου τύπου μεταβλητές

Μεταβλητή= Όνομα Τύπου Δεδομένων(λίστα τιμών)

ή

Μεταβλητή.Πεδίο=Τιμή

## Παράδειγμα.

```
STUDENT=STUDENTS('Αναστασιάδης', 'Λάζαρος',1234,9)
```

ή

```
STUDENT.SURNAME='Αναστασιάδης'
```

```
STUDENT.NAME='Λάζαρος'
```

```
STUDENT.AEM=1234
```

```
STUDENT.GRADE=9
```



# Επέκταση του τύπου δεδομένων στην Fortran 2003

```
TYPE, EXTENDS(STUDENTS) :: STUDENTS_INFO
```

```
    CHARACTER :: EKSAMINO
```

```
END TYPE STUDENTS_INFO
```

```
TYPE(STUDENTS_INFO) :: FOITHTHS
```

```
FOITHTHS.EKSAMINO="A"
```



# Τμήμα Προτάσεων

[PROGRAM όνομα προγράμματος]

Implicit None

Περιοχή σταθερών

Τμήμα μεταβλητών

Περιοχή τύπων

**Τμήμα προτάσεων**

.....

END [PROGRAM [όνομα προγράμματος]]

**Παράδειγμα.**

! Σχόλια

READ\*,A,B

! Διαβάζω τιμές για A,B

X=A+B ; Y=A-B

T=A+3 &

& \*4



# Τμήμα διαδικασιών και συναρτήσεων

[PROGRAM όνομα προγράμματος]

Implicit None

Περιοχή σταθερών

Τμήμα μεταβλητών

Περιοχή τύπων

Τμήμα προτάσεων

**CONTAINS**

**Διαδικασίες και συναρτήσεις**

END [PROGRAM [όνομα προγράμματος]]

**Παράδειγμα.**

CONTAINS

SUBROUTINE EMBADON(A,B,C)

.....

END SUBROUTINE EMBADON



# Συνολική εικόνα του προγράμματος

**[PROGRAM Όνομα Προγράμματος]**

**USE όνομα module**

**IMPLICIT NONE**

**INTERFACE «όνομα»**

.....

**END INTERFACE**

Δήλωση Τύπων

Δήλωση Σταθερών

Δήλωση Μεταβλητών

Εκτελέσιμες Προτάσεις

**CONTAINS**

Διαδικασίες – Συναρτήσεις

**END [PROGRAM [Όνομα Προγράμματος]]**





# Παράδειγμα

PROGRAM EMBADON

! Το παρακάτω πρόγραμμα υπολογίζει το εμβαδόν ενός κύκλου

!δοθέντος της ακτίνας του

! R=ακτίνα του κύκλου E=εμβαδόν του κύκλου

IMPLICIT NONE

! Δήλωση σταθερών

REAL, PARAMETER :: PI=3.14159

! Δήλωση μεταβλητών

REAL :: R,E

! Εισαγωγή δεδομένων στο πρόγραμμα

PRINT\*, "ΑΚΤΙΝΑ="

READ\*,R

! Υπολογισμός εμβαδού

E=PI\*R\*\*2

! Εκτύπωση αποτελεσμάτων

PRINT\*,"EMBADON=",E

END PROGRAM EMBADON



# Βιβλιογραφία

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



# Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Καραμπετάκης Νικόλαος. «Εισαγωγή στον Προγραμματισμό Η/Υ (Fortran 90/95/2003). Βασικά στοιχεία ενός προγράμματος στη Fortran 90/95/2003 - Δομή Προγράμματος». 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

