

# ΠΛΗ10

## ΕΝΟΤΗΤΑ 2: Αλγόριθμοι και Ψευδογλώσσα

### Μάθημα 2.3: Τελεστές και η Εντολή Απόφασης

Δημήτρης Ψούνης



[www.psounis.gr](http://www.psounis.gr)



# Περιεχόμενα Μαθήματος

## **A. Θεωρία**

### **1. Τελεστές**

1. Γενικά
2. Ο τελεστής εκχώρησης
3. Αριθμητικοί Τελεστές
  1. Προτεραιότητα Τελεστών
  2. Οι Τελεστές DIV και MOD
4. Σχεσιακοί Τελεστές
5. Λογικοί Τελεστές

### **2. Η Εντολή Απόφασης**

1. Γενικά
2. Πρώτη Μορφή (ΕΪΑΝ)
3. Δεύτερη Μορφή (ΕΪΑΝ...ΑΛΛΙΩΣ)
4. Εμφωλιασμένες Εντολές Απόφασης

## **B. Ασκήσεις**



# A. Θεωρία

## 1. Τελεστές

### 1. Γενικά

- Οι τελεστές είναι σύμβολα που χρησιμοποιούνται στις εντολές ενός προγράμματος.
- Χωρίζονται σε τέσσερις μεγάλες κατηγορίες
  - **Ο τελεστής καταχώρησης ( := )**
    - Χρησιμοποιείται όταν θέλουμε να αποθηκεύσουμε σε μία μεταβλητή μία αριθμητική τιμή.
  - **Αριθμητικοί Τελεστές ( +, -, \*, /, DIV, MOD )**
    - Χρησιμοποιούνται όταν θέλουμε να κάνουμε αριθμητικές πράξεις με τις μεταβλητές
  - **Σχεσιακοί Τελεστές ( =, >, <, >=, <=, <> )**
    - Χρησιμοποιούνται για να γίνει έλεγχος με την εντολή συνθήκης.
  - **Λογικοί Τελεστές ( AND, OR, NOT )**
    - Χρησιμοποιούνται για να γίνει έλεγχος με την εντολή συνθήκης.
- Υπάρχουν και άλλοι πιο προχωρημένοι τελεστές, όπως ο τελεστής έμμεσης προσπέλασης στην μνήμη ( ^ ) που θα δούμε σε επόμενες ενότητες.



# A. Θεωρία

## 1. Τελεστές

### 2. Ο Τελεστής Εκχώρησης (:=)

- Ο τελεστής εκχώρησης (ή καταχώρησης ή αποθήκευσης) ήταν ο 1<sup>ος</sup> τελεστής που μελετήσαμε αναλυτικά στο προηγούμενο μάθημα. Θυμίζουμε το συντακτικό του:

**ΜΕΤΑΒΛΗΤΗ := (ΥΠΟΛΟΓΙΣΜΟΣ) ;**

- Στα αριστερά έχει πάντα μία μεταβλητή (στην οποία θα γίνει η αποθήκευση)
- Στα δεξιά ο υπολογισμός μπορεί να είναι
  - Μία σταθερά (οπότε και αποθηκεύεται η τιμή της σταθεράς)

`X := 5 ;`

- Μία αριθμητική πράξη (πρώτα γίνεται η πράξη και έπειτα αποθηκεύεται το αποτέλεσμα)

`X := 7 + 2 ;`

- Μία πράξη που ενσωματώνει μεταβλητές (πρώτα αντικαθίσταται η τιμή των μεταβλητών στα δεξιά με την τιμή τους, γίνεται η πράξη και έπειτα αποθηκεύεται το αποτέλεσμα)

`X := Y + 2 ;`

- Το αποτέλεσμα της κλήσης μιας συνάρτησης (θα το δούμε σε επόμενο μάθημα)

`X := SQRT ( 4 ) ;`

# A. Θεωρία

## 1. Τελεστές

### 3. Αριθμητικοί Τελεστές

- Οι μαθηματικοί τελεστές εκτελούν τις συνηθισμένες μαθηματικές πράξεις
- Είναι οι εξής:

| Όνομα Τελεστή      | Σύμβολο | Χρήση              | Παράδειγμα<br>(για X=5 και Y=3) |
|--------------------|---------|--------------------|---------------------------------|
| Πρόσθεση           | +       | ΠΑΡ + ΠΑΡ          | $X+Y = 8$                       |
| Πολ/μός            | *       | ΠΑΡ * ΠΑΡ          | $X*Y = 15$                      |
| Αφαίρεση           | -       | ΠΑΡ - ΠΑΡ          | $X-Y = 2$                       |
| Διαίρεση           | /       | ΠΑΡ / ΠΑΡ          | $X/Y = 1.66$                    |
| Ακέραια Διαίρεση   | DIV     | ΠΑΡ <b>DIV</b> ΠΑΡ | $X \text{ DIV } Y = 1$          |
| Υπόλοιπο Διαίρεσης | MOD     | ΠΑΡ <b>MOD</b> ΠΑΡ | $X \text{ MOD } Y = 2$          |

- Όπου ΠΑΡ είναι μία μαθηματική παράσταση



# A. Θεωρία

## 1. Τελεστές

### 3. Αριθμητικοί Τελεστές (Προτεραιότητα Τελεστών)

- Σε μία μαθηματική παράσταση που χρησιμοποιούμε σε ένα πρόγραμμα ακολουθούνται κανόνες προτεραιότητας των πράξεων.
- Συγκεκριμένα (για παραστάσεις που χρησιμοποιούν +, -, \*, /) και εφόσον δεν υπάρχουν παρενθέσεις που να καθορίζουν την σειρά των πράξεων:
  - Πρώτα γίνονται οι πολλαπλασιασμοί και οι διαιρέσεις (από αριστερά προς τα δεξιά)
  - Έπειτα γίνονται οι αφαιρέσεις και οι προσθέσεις (από αριστερά προς τα δεξιά)

(Άσκηση) Ποια θα είναι η τιμή των μεταβλητών μετά την εκτέλεση των ακόλουθων γραμμών κώδικα;

$X := 5 * 3 + 4 * 8 ;$

$Y := 5 + 3 * 4 - 8 ;$

$Z := (5 + 3) * (4 + 8) ;$

$W := 12 * 12 / 6 - 3 ;$

Ελέγξτε την ορθότητα των απαντήσεων σας, κατασκευάζοντας κατάλληλο πρόγραμμα στον μεταγλωττιστή ψευδογλώσσας.



# A. Θεωρία

## 1. Τελεστές

### 3. Αριθμητικοί Τελεστές (Οι Τελεστές DIV και MOD)

- Όταν εκτελούμε μια ακέραια διαίρεση έχουμε δύο αποτελέσματα:
  - Το πηλίκο της διαίρεσης (υπολογίζεται μέσω του τελεστή DIV)
  - Το υπόλοιπο της διαίρεσης (υπολογίζεται μέσω του τελεστή MOD)

(Άσκηση) Ποια θα είναι η τιμή των μεταβλητών μετά την εκτέλεση των ακόλουθων γραμμών κώδικα;

```
X := 55 DIV 12;
```

```
Y := 55 MOD 12;
```

```
Z := Y DIV X;
```

Ελέγξτε την ορθότητα των απαντήσεων σας, κατασκευάζοντας κατάλληλο πρόγραμμα στον μεταγλωττιστή ψευδογλώσσας.



# A. Θεωρία

## 1. Τελεστές

### 4. Σχεσιακοί Τελεστές

- Οι Σχεσιακοί Τελεστές ελέγχουν αν αληθεύει μία συνθήκη σύγκρισης τιμών. Αν αληθεύει απαντούν true (λογικό 1) ενώ αν δεν αληθεύει απαντούν false (λογικό 0)
- Συνήθως χρησιμοποιούνται για να ελεγχθεί μια συνθήκη από την εντολή απόφασης:

| Τελεστής | Επεξήγηση            | Παράδειγμα Χρήσης | Αποτέλεσμα (Για X=5, Y=3) | Αποτέλεσμα (Για X=2, Y=7) |
|----------|----------------------|-------------------|---------------------------|---------------------------|
| =        | Ισότητα              | X=Y               | False                     | False                     |
| >        | Μεγαλύτερο από       | X>Y               | True                      | False                     |
| <        | Μικρότερο από        | X<Y               | False                     | True                      |
| >=       | Μεγαλύτερο ή ίσο από | X>=Y              | True                      | False                     |
| <=       | Μικρότερο ή ίσο από  | X<=Y              | False                     | True                      |
| <>       | Διάφορο              | X<>Y              | True                      | Tue                       |



# A. Θεωρία

## 1. Τελεστές

### 5. Λογικοί Τελεστές

➤ Οι λογικοί τελεστές εκτελούν λογικές πράξεις, συνήθως με το αποτέλεσμα της πράξης δύο ελέγχων που έχουμε κάνει με άλλους σχεσιακούς ή λογικούς τελεστές:

| Όνομα Τελεστή | Χρήση                | Αληθεύει (=1) όταν   | Ψευδής (=0) όταν   |
|---------------|----------------------|--|--|
| AND           | ΠΑΡ1 <b>AND</b> ΠΑΡ2 | Η ΠΑΡ1 είναι αληθής (1)<br><b>ΚΑΙ</b><br>Η ΠΑΡ2 είναι αληθής (1) | Η ΠΑΡ1 είναι ψευδής (1)<br><b>Ή</b><br>Η ΠΑΡ2 είναι ψευδής (1)   |
| OR            | ΠΑΡ1 <b>OR</b> ΠΑΡ2  | Η ΠΑΡ1 είναι αληθής (1)<br><b>Ή</b><br>Η ΠΑΡ2 είναι αληθής (1)   | Η ΠΑΡ1 είναι ψευδής (0)<br><b>ΚΑΙ</b><br>Η ΠΑΡ2 είναι ψευδής (0) |
| NOT           | <b>NOT</b> ΠΑΡ       | Η ΠΑΡ είναι ψευδής (0)   | Η ΠΑΡ είναι αληθής (1)   |

➤ Όπου ΠΑΡ είναι μία μαθηματική παράσταση



# A. Θεωρία

## 1. Τελεστές

### 5. Λογικοί Τελεστές

- Οι σχεσιακοί και λογικοί τελεστές μπορούν να δημιουργήσουν παραστάσεις που θα χρησιμοποιηθούν έπειτα στην εντολή απόφασης που χρησιμοποιεί το πρόγραμμά μας.
- Είναι σημαντικό να θυμόμαστε ότι το αποτέλεσμα της εφαρμογής των τελεστών θα είναι τελικά είτε TRUE (1) είτε FALSE (0)

(Άσκηση) Ποιο θα είναι το αποτέλεσμα της εφαρμογής των τελεστών αν  $X=1$  και  $Y=2$  στις παρακάτω δύο παραστάσεις που χρησιμοποιούν σχεσιακούς / λογικούς αλλά και αριθμητικούς τελεστές.

$$(X <> Y) \text{ AND } (\text{NOT } X < Y)$$
$$(X >= X) \text{ OR } ((\text{NOT } (Y <> X)) \text{ AND } (X+1 < Y+2))$$

Επαναλάβετε εάν  $X=5$  και  $Y=2$

Συμβουλές:

- Φροντίζουμε να παρενθετοποιούμε πλήρως τις παραστάσεις μας, ώστε να είμαστε σίγουροι για το αποτέλεσμα. Π.χ. η προτεραιότητα των λογικών τελεστών είναι NOT, AND, OR αλλά είναι προτιμότερο να έχουμε παρενθετοποιημένες τις παραστάσεις μας.
- Οι λογικοί τελεστές και οι αντίστοιχες λογικές πράξεις θα είναι απαραίτητες στον προγραμματισμό. Κάντε μια υπενθύμιση αν απαιτείται στην Άλγεβρα Boole του Μαθήματος 1.3



# A. Θεωρία

## 2. Η Εντολή Απόφασης

### 1. Γενικά

- Η **εντολή απόφασης** είναι από τα πιο σημαντικά προγραμματιστικά εργαλεία
- Οι προτάσεις σε ένα πρόγραμμα εκτελούνται κανονικά από την αρχή έως το τέλος με την ίδια σειρά που εμφανίζονται στον αλγόριθμο.
- Μια εντολή απόφασης τροποποιεί την σειρά εκτέλεσης των προτάσεων. Μέσω αυτής:
  - Μπορούμε να προκαλέσουμε την εκτέλεση άλλων προτάσεων ή την μη εκτέλεση τους ανάλογα με τον υπολογισμό που εκτελούμε.
- Η σύνταξη της εντολής απόφασης στην γενική μορφή της είναι:

**ΕΑΝ** (συνθήκη) **ΤΟΤΕ**  
(Εντολή ή Εντολές)  
**[ΑΛΛΙΩΣ**  
(Εντολή ή Εντολές)**]**  
**ΕΑΝ-ΤΕΛΟΣ ;**

### Σημείωση:

- Στους ορισμούς των εντολών θα βλέπουμε συχνά ένα μέρος να είναι σε αγκύλες [...]
- Αυτό σημαίνει ότι το μέρος αυτό είναι προαιρετικό. Το χρησιμοποιούμε μόνον αν κρίνουμε ότι αυτό είναι απαραίτητο.



# A. Θεωρία

## 2. Η Εντολή Απόφασης

### 2. Πρώτη Μορφή (EAN)

- Η εντολή απόφασης στην 1<sup>η</sup> της χρήση συντάσσεται ως εξής:

[προηγούμενες εντολές]

EAN (συνθήκη)  
(Εντολές)

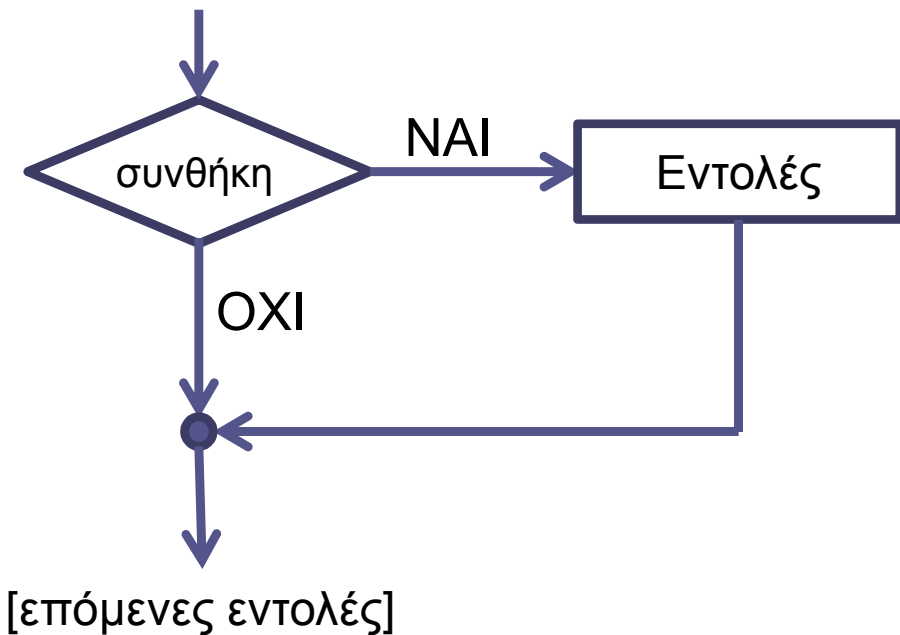
EAN-ΤΕΛΟΣ

[επόμενες εντολές]

- Αφού εκτελεστούν οι προηγούμενες εντολές
- Ελέγχεται η συνθήκη (συνήθως θα είναι μια παράσταση που θα χρησιμοποιεί κάποιον σχεσιακό/λογικό τελεστή)
  - Αν η συνθήκη είναι ΑΛΗΘΗΣ εκτελούνται οι (εντολές) και έπειτα οι [επόμενες εντολές]
  - Αν η συνθήκη είναι ΨΕΥΔΗΣ εκτελούνται απευθείας οι (επόμενες εντολές)

### ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

[προηγούμενες εντολές]





# A. Θεωρία

## 2. Η Εντολή Απόφασης

### 2. Πρώτη Μορφή (ΕΑΝ)

- Μεταγλωττίστε, εκτελέστε και μελετήστε το παρακάτω πρόγραμμα που δείχνει την 1<sup>η</sup> χρήση της εντολής if
- Πόσες συγκρίσεις θα γίνουν κατά την εκτέλεση του προγράμματος;

**ΑΛΓΟΡΙΘΜΟΣ** if1

**ΔΕΔΟΜΕΝΑ**

X, Y: INTEGER;

**ΑΡΧΗ**

ΤΥΠΩΣΕ ("ΔΩΣΕ ΤΟΝ ΑΚΕΡΑΙΟ X: ");

ΔΙΑΒΑΣΕ (X);

ΤΥΠΩΣΕ ("ΔΩΣΕ ΤΟΝ ΑΚΕΡΑΙΟ Y: ");

ΔΙΑΒΑΣΕ (Y);

**ΕΑΝ** (X<Y) **TOTE**

ΤΥΠΩΣΕ ("X<Y", EOLN);

**ΕΑΝ-ΤΕΛΟΣ**

**ΕΑΝ** (X=Y) **TOTE**

ΤΥΠΩΣΕ ("X=Y", EOLN);

**ΕΑΝ-ΤΕΛΟΣ**

**ΕΑΝ** (X>Y) **TOTE**

ΤΥΠΩΣΕ ("X>Y", EOLN);

**ΕΑΝ-ΤΕΛΟΣ**

**ΤΕΛΟΣ**



# A. Θεωρία

## 2. Η Εντολή Απόφασης

### 3. Δεύτερη Μορφή (ΕΑΝ...ΑΛΛΙΩΣ)

- Η δομή ελέγχου, στην 2η της χρήση συντάσσεται ως εξής:

[προηγούμενες εντολές]

**ΕΑΝ** (συνθήκη) **ΤΟΤΕ**

(ΕντολέςΑ)

**ΑΛΛΙΩΣ**

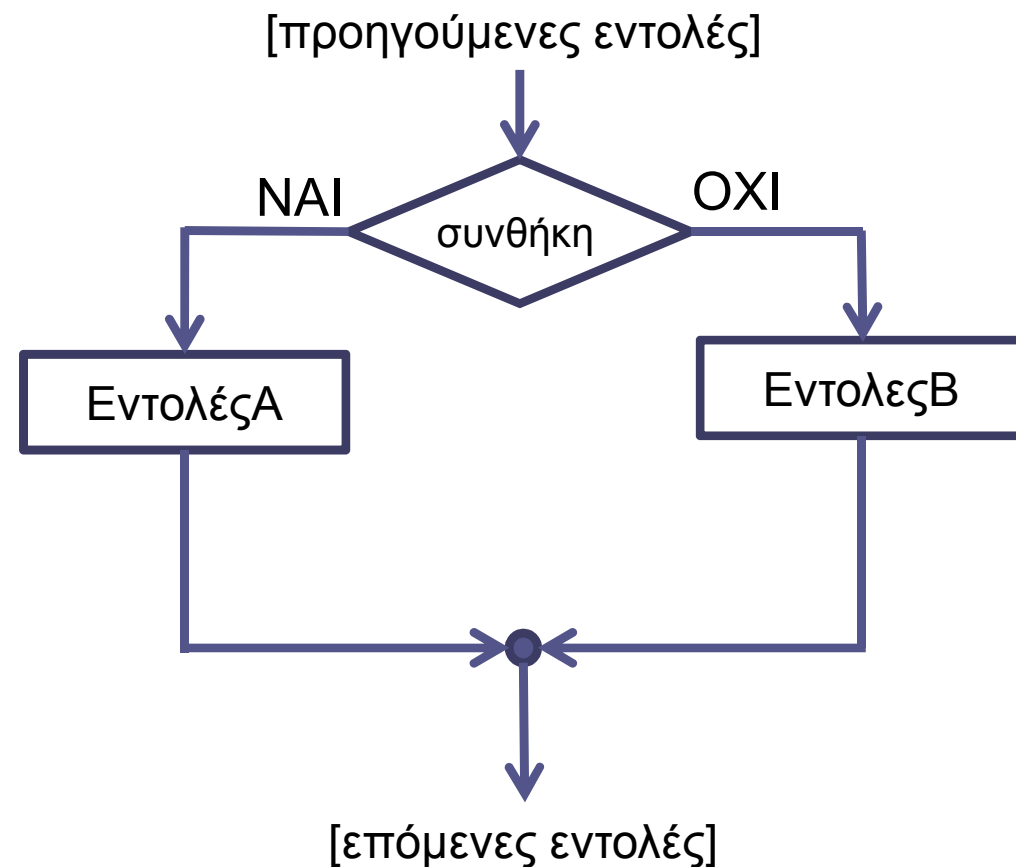
(ΕντολέςΒ)

**ΕΑΝ-ΤΕΛΟΣ**

[επόμενες εντολές]

- Αφού εκτελεστούν οι προηγούμενες εντολές
- Ελέγχεται η συνθήκη (συνήθως θα είναι μια παράσταση που θα χρησιμοποιεί κάποιον σχεσιακό/λογικό τελεστή)
  - Αν η συνθήκη είναι ΑΛΗΘΗΣ εκτελούνται οι Εντολές Α και έπειτα οι (επόμενες εντολές)
  - Αν η συνθήκη είναι ΨΕΥΔΗΣ εκτελούνται οι Εντολές Β και έπειτα οι (επόμενες εντολές)

### ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ





# A. Θεωρία

## 2. Η Εντολή Απόφασης

### 3. Δεύτερη Μορφή (ΕΑΝ...ΑΛΛΙΩΣ)

- Μεταγλωττίστε, εκτελέστε και μελετήστε το παρακάτω πρόγραμμα που δείχνει την 2<sup>η</sup> χρήση της εντολής απόφασης ΕΑΝ.
- Πόσες συγκρίσεις θα γίνουν κατά την εκτέλεση του προγράμματος;

**ΑΛΓΟΡΙΘΜΟΣ** if2

**ΔΕΔΟΜΕΝΑ**

X: INTEGER;

**ΑΡΧΗ**

ΤΥΠΩΣΕ ("ΔΩΣΕ ΤΟΝ ΑΚΕΡΑΙΟ X: ");

ΔΙΑΒΑΣΕ (X);

**ΕΑΝ** (X MOD 2 = 0) **ΤΟΤΕ**

ΤΥΠΩΣΕ ("Ο ", X, " ΕΙΝΑΙ ΑΡΤΙΟΣ!", EOLN);

**ΑΛΛΙΩΣ**

ΤΥΠΩΣΕ ("Ο ", X, " ΕΙΝΑΙ ΠΕΡΙΤΤΟΣ!", EOLN);

**ΕΑΝ-ΤΕΛΟΣ**

**ΤΕΛΟΣ**



# A. Θεωρία

## 2. Η Εντολή Απόφασης

### 4. Εμφωλιασμένες Εντολές Απόφασης

- Η δομή ελέγχου είναι απλά μία εντολή!  
Άρα μπορούμε να έχουμε π.χ. μία εντολή ΕΑΝ μέσα σε μία εντολή ΕΑΝ όσες φορές θέλουμε
  - Αυτό αναφέρεται και σαν **εμφωλιασμένες εντολές απόφασης**
- Μελετήστε την παραλλαγή προηγούμενου προγράμματος.
- Πόσες συγκρίσεις θα γίνουν στο πρόγραμμα αυτό;

**ΑΛΓΟΡΙΘΜΟΣ** if3

**ΔΕΔΟΜΕΝΑ**

X, Y: INTEGER;

**ΑΡΧΗ**

ΤΥΠΩΣΕ ("ΔΩΣΕ ΤΟΝ ΑΚΕΡΑΙΟ Χ: ");

ΔΙΑΒΑΣΕ (X);

ΤΥΠΩΣΕ ("ΔΩΣΕ ΤΟΝ ΑΚΕΡΑΙΟ Υ: ");

ΔΙΑΒΑΣΕ (Y);

**ΕΑΝ** (X<Y) **ΤΟΤΕ**

ΤΥΠΩΣΕ ("X<Y", EOLN);

**ΑΛΛΙΩΣ**

**ΕΑΝ** (X=Y) **ΤΟΤΕ**

ΤΥΠΩΣΕ ("X=Y", EOLN);

**ΑΛΛΙΩΣ**

ΤΥΠΩΣΕ ("X>Y", EOLN);

**ΕΑΝ-ΤΕΛΟΣ**

**ΕΑΝ-ΤΕΛΟΣ**

**ΤΕΛΟΣ**





# Β.Ασκήσεις

## Εφαρμογή 1: Ελάχιστος από δύο

Γράψτε ένα πρόγραμμα που ζητάει από τον χρήστη να εισάγει 2 ακέραιους αριθμούς και τυπώνει στην οθόνη τον ελάχιστο από τους 2 αριθμούς



# Β.Ασκήσεις

## Εφαρμογή 2: Μέγιστος από τρεις

- Γράψτε ένα πρόγραμμα που ζητάει από τον χρήστη να εισάγει 3 ακέραιους αριθμούς και τυπώνει στην οθόνη τον ελάχιστο από τους 3 αριθμούς



# Β.Ασκήσεις

## Εφαρμογή 3: Λύσεις Πρωτοβάθμιας Εξίσωσης

- Γράψτε ένα πρόγραμμα που ζητάει από τον χρήστη να εισάγει 2 αριθμούς  $a, b$  και υπολογίζει την λύση της εξίσωσης  $ax+b=0$  (Τα  $a$  και  $b$  είναι πραγματικοί αριθμοί)

Υπενθύμιση: Αν το  $a$  δεν είναι 0 η λύση της εξίσωσης είναι  $x=-b/a$ . Αν το  $a$  είναι 0 να εκτυπώνεται το μήνυμα «Η εξίσωση δεν έχει λύση»



# Β.Ασκήσεις

## Εφαρμογή 4: DIV και MOD

- Γράψτε ένα πρόγραμμα το οποίο
  - Να δέχεται από τον χρήστη σαν είσοδο έναν ακέραιο αριθμό, που θα απεικονίζει δευτερόλεπτα.
  - Να υπολογίζει πόσες ώρες, λεπτά και δευτερόλεπτα είναι η είσοδος του χρήστη.
- Η παρακάτω είναι η επιθυμητή έξοδος αν ο χρήστης εισάγει το 5000:

```
ΕΙΣΑΓΕΤΕ ΠΛΗΘΟΣ ΔΕΥΤΕΡΟΛΕΠΤΩΝ: 5000
ΩΡΕΣ: 1
ΛΕΠΤΑ: 23
ΔΕΥΤΕΡΟΛΕΠΤΑ: 20
```

- Υπόδειξη:
  - Μελετήστε πως μπορείτε να χρησιμοποιήσετε τους τελεστές DIV και MOD για να επιτύχετε το επιθυμητό αποτέλεσμα!



# Β.Ασκήσεις

## Εφαρμογή 5: Πιο σύνθετες συνθήκες

- Γράψτε ένα πρόγραμμα C το οποίο
  - Να δέχεται από τον χρήστη σαν είσοδο έναν ακέραιο αριθμό
  - Να εξετάζει αν ο αριθμός είναι στο κλειστό διάστημα 1..10.
  - Να εκτυπώνει κατάλληλο μήνυμα
- Παραδείγματα εκτέλεσης:

```
ΕΙΣΑΓΕΤΕ ΤΟ Χ: 3  
ΕΙΝΑΙ ΣΤΟ ΕΠΙΘΥΜΗΤΟ ΔΙΑΣΤΗΜΑ
```

```
ΕΙΣΑΓΕΤΕ ΤΟ Χ: 11  
ΔΕΝ ΕΙΝΑΙ ΣΤΟ ΕΠΙΘΥΜΗΤΟ ΔΙΑΣΤΗΜΑ
```

```
ΕΙΣΑΓΕΤΕ ΤΟ Χ: 1  
ΕΙΝΑΙ ΣΤΟ ΕΠΙΘΥΜΗΤΟ ΔΙΑΣΤΗΜΑ
```