

Ανάπτυξη εφαρμογών σε
προγραμματιστικό περιβάλλον
Γ' Λυκείου

ΚΕΦΑΛΑΙΟ 8

Επιλογή και Επανάληψη



Χρήστος Μουρατίδης - Έκδοση 2020

mouratx@yahoo.com

<http://users.sch.gr/mouratx>

Περιεχόμενα

ΕΙΣΑΓΩΓΗ	1
ΛΟΓΙΚΗ ΕΚΦΡΑΣΗ	1
ΣΥΝΘΕΤΕΣ ΛΟΓΙΚΕΣ ΕΚΦΡΑΣΕΙΣ	2
ΕΝΤΟΛΕΣ ΕΠΙΛΟΓΗΣ	3
Η ΕΝΤΟΛΗ ΑΝ	3
1 ^η μορφή: ΑΝ . ΤΟΤΕ	3
2 ^η μορφή: ΑΝ . ΤΟΤΕ . ΑΛΛΙΩΣ	4
Εμφωλευμένες εντολές ΑΝ	5
3 ^η μορφή: ΑΝ . ΤΟΤΕ . ΑΛΛΙΩΣ ΑΝ	8
ΕΝΤΟΛΗ ΕΠΙΛΕΞΕ	11
ΕΝΤΟΛΕΣ ΕΠΑΝΑΛΗΨΗΣ.....	14
ΕΝΤΟΛΗ ΟΣΟ . ΕΠΑΝΑΛΑΒΕ	14
ΠΙΝΑΚΑΣ ΤΙΜΩΝ	17
ΕΝΤΟΛΗ ΜΕΧΡΙΣ ΟΤΟΥ	18
ΕΝΤΟΛΗ ΓΙΑ . ΑΠΟ . ΜΕΧΡΙ	22
ΕΜΦΩΛΕΥΜΕΝΟΙ ΒΡΟΧΟΙ	25
ΕΡΩΤΗΣΕΙΣ ΚΑΤΑΝΟΗΣΗΣ.....	27

Εισαγωγή

Στο προηγούμενο Κεφάλαιο 7 είδαμε προγράμματα που επιλύουν απλά προβλήματα με τη διαδοχική (σειριακή) εκτέλεση των εντολών. Αυτή η δομή των εντολών γνωρίζουμε ότι είναι η ακολουθιακή. Όμως, στις περισσότερες περιπτώσεις, η επίλυση των προβλημάτων απαιτεί τον **έλεγχο καταστάσεων (λογικών συνθηκών) και ανάλογα εκτελείται ένα συγκεκριμένο τμήμα του προγράμματος (δομή επιλογής) ή να εκτελείται επαναληπτικά (δομή επανάληψης).**

Στο Κεφάλαιο αυτό θα δούμε με ποιές εντολές της ΓΛΩΣΣΑΣ υλοποιούνται η δομή επιλογής και η δομή επανάληψης. Πρώτα, όμως, θα εξετάσουμε ένα ουσιώδες χαρακτηριστικό των δομών αυτών: την δημιουργία λογικών εκφράσεων ή συνθηκών.

Λογική έκφραση

Η δημιουργία μίας λογικής έκφρασης μπορεί να εμπλέκει σταθερές, μεταβλητές, αριθμητικές παραστάσεις, συγκριτικούς και λογικούς τελεστές καθώς και παρενθέσεις. **Η αποτίμηση μίας λογικής έκφρασης είναι πάντα η λογική τιμή ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ.**

Στον παρακάτω πίνακα βλέπουμε τους **συγκριτικούς τελεστές** της ΓΛΩΣΣΑΣ :

Τελεστής	Περιγραφή	Παράδειγμα
=	Ίση με	$X = 0$, Επώνυμο = 'Μουρατίδης', $Y = 'N'$
<>	Διάφορη από	$X <> 0$, Επώνυμο <> 'Μουρατίδης'
<	Μικρότερη από	$X < 0$, $X + Y < (Z + 2) / 5$
<=	Μικρότερη ή ίση με	$X <= 0$, Επώνυμο < 'Μουρατίδης'
>	Μεγαλύτερη από	$X > 0$, $X ^ 2 > 20$
>=	Μεγαλύτερη ή ίση με	$X >= 0$

Στην περίπτωση που συγκρίνουμε αλφαριθμητικά, όπως στο παράδειγμα Επώνυμο < 'Μουρατίδης', η σύγκριση πραγματοποιείται αλφαβητικά χαρακτήρα προς χαρακτήρα ξεκινώντας από τα αριστερά. Έτσι, αν η μεταβλητή Επώνυμο έχει την τιμή 'Μαργαρίτης' τότε η λογική έκφραση αποτιμάται ως ΑΛΗΘΗΣ. Στον πρώτο χαρακτήρα έχουν ισότητα (Μ) αλλά στον δεύτερο χαρακτήρα το 'α' προηγείται του 'ο'.

Σύνθετες λογικές εκφράσεις

Ο συνδυασμός πολλών λογικών εκφράσεων με τη χρήση των λογικών τελεστών **ΚΑΙ**, **Η** και **ΟΧΙ** δημιουργεί σύνθετες λογικές εκφράσεις.

Μερικά παραδείγματα:

Μαθηματική έκφραση	Σύνθετη λογική έκφραση στη ΓΛΩΣΣΑ
$0 < X < 10$	$X > 0$ ΚΑΙ $X < 10$
$X=5$ ή 10 ή 15	$X = 5$ Η $X = 10$ Η $X = 15$
$X=5$ και Y διαφορετικό του 7	$X = 5$ ΚΑΙ $Y \neq 7$



Όταν σε μία έκφραση συνυπάρχουν αριθμητικοί και συγκριτικοί τελεστές, τότε προηγούνται οι πρώτοι, δηλαδή προηγούνται οι αριθμητικές πράξεις.

Π.χ. $Y + 5 > 12$, θα εκτελεστεί πρώτα η πρόσθεση και κατόπιν θα γίνει η σύγκριση.

Επίσης, οι λογικοί τελεστές (**ΚΑΙ**, **Η**, **ΟΧΙ**) έχουν την χαμηλότερη προτεραιότητα, ακόμα κι από τους συγκριτικούς.

Π.χ. Το X να μην είναι μεγαλύτερο του 12 μπορεί να εκφραστεί με 3 τρόπους:

$X < 12$
ΟΧΙ $X > 12$
ΟΧΙ ($X > 12$)

Στην τρίτη περίπτωση η παρένθεση είναι περιττή διότι ο συγκριτικός τελεστής (>) έχει υψηλότερη προτεραιότητα από τον λογικό **OXI**. Κάποιοι προτιμούν αυτόν τον τρόπο για λόγους σαφήνειας του κώδικα.

Ως γνωστόν, η ιεραρχία μπορεί να αλλάξει με τις παρενθέσεις.

Εντολές επιλογής

Μία από τις βασικότερες αλγοριθμικές δομές είναι η δομή επιλογής, όπως είδαμε στο Κεφάλαιο 2. Στις γλώσσες προγραμματισμού υλοποιείται με διάφορες εντολές επιλογής.

Στη ΓΛΩΣΣΑ, η **υλοποίηση της δομής επιλογής** γίνεται με:

- Την **εντολή AN** και τις παραλλαγές (μορφές) της.
- Την **εντολή ΕΠΙΛΕΞΕ**

Η εντολή AN

1^η μορφή: AN . . ΤΟΤΕ

Σύνταξη:

AN συνθήκη **ΤΟΤΕ**

εντολή-1

εντολή-2

...

εντολή-ν

ΤΕΛΟΣ AN

Πρόκειται για την πιο απλή μορφή. Η εντολή **AN** ελέγχει τη (λογική) συνθήκη και αν η αποτίμηση είναι **ΑΛΗΘΗΣ** τότε εκτελούνται οι εντολές στο εσωτερικό της.

Παράδειγμα:

Γράψτε ένα πρόγραμμα στη ΓΛΩΣΣΑ που διαβάζει από το πληκτρολόγιο τον μισθό ενός εργαζομένου κι αν αυτός είναι πάνω από 2000 τότε να εμφανίζει επί της οθόνης το μήνυμα «Έχετε υψηλό μισθό!»

ΠΡΟΓΡΑΜΜΑ Μισθός_εργαζομένου

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: Ποσό

ΑΡΧΗ

!Είσοδος δεδομένων, εδώ του ποσού.

ΓΡΑΨΕ 'Παρακαλώ, δώστε το ποσό του μισθού σας'

ΔΙΑΒΑΣΕ Ποσό

!Έλεγχος του ποσού, αν ξεπερνάει τις 2000.

AN Ποσό > 2000 **ΤΟΤΕ**

ΓΡΑΨΕ 'Έχετε υψηλό μισθό!'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

2^η μορφή: **ΑΝ .. ΤΟΤΕ .. ΑΛΛΙΩΣ**

Σύνταξη:

ΑΝ συνθήκη **ΤΟΤΕ**

εντολή-1

εντολή-2

...

εντολή-ν

ΑΛΛΙΩΣ

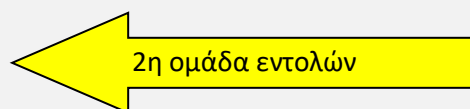
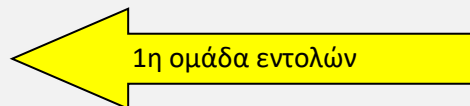
εντολή-1

εντολή-2

...

εντολή-ν

ΤΕΛΟΣ_ΑΝ



Πρόκειται για την επεκταμένη μορφή. Η εντολή **AN** ελέγχει τη (λογική) συνθήκη και αν η αποτίμηση είναι **ΑΛΗΘΗΣ** τότε εκτελούνται οι εντολές της πρώτης ομάδας. Αν, όμως, η αποτίμηση είναι **ΨΕΥΔΗΣ** τότε εκτελούνται οι εντολές της δεύτερης ομάδας.

Είναι προφανές ότι μόνο μία ομάδα εντολών θα εκτελεστεί ανάλογα με την τιμή της συνθήκης.

Παράδειγμα:

Γράψτε ένα πρόγραμμα στη **ΓΛΩΣΣΑ** που διαβάζει από το πληκτρολόγιο τον μισθό ενός εργαζομένου κι αν αυτός είναι πάνω από 2000 τότε να εμφανίζει επί της οθόνης το μήνυμα «Έχετε υψηλό μισθό!» αλλιώς το μήνυμα «Έχετε χαμηλό μισθό!».

ΠΡΟΓΡΑΜΜΑ Μισθός_εργαζομένου

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: Ποσό

ΑΡΧΗ

!Είσοδος δεδομένων, εδώ του ποσού.

ΓΡΑΨΕ 'Παρακαλώ, δώστε το ποσό του μισθού σας'

ΔΙΑΒΑΣΕ Ποσό

!Έλεγχος του ποσού, αν ξεπερνάει τις 2000.

AN Ποσό > 2000 **ΤΟΤΕ**

ΓΡΑΨΕ 'Έχετε υψηλό μισθό!'

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Έχετε χαμηλό μισθό!'

ΤΕΛΟΣ_AN

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Αν το ποσό που δώσει ο χρήστης είναι ακριβώς 2000 τότε θα εκτελεστεί η εντολή μετά το **ΑΛΛΙΩΣ**. Αν θέλαμε να συμπεριλάβουμε το 2000 στους υψηλόμισθους θα αλλάζαμε τη συνθήκη:

AN Ποσό >= 2000 **ΤΟΤΕ**

Εμφωλευμένες εντολές **AN**

Με τον όρο **εμφωλευμένες εντολές AN** εννοούμε ότι μία εντολή **AN** μπορεί να βρίσκεται στο εσωτερικό μίας άλλης εντολής **AN**. Η παρακάτω σύνταξη είναι ενδεικτική:

Σύνταξη:

```
AN συνθήκη1 ΤΟΤΕ
  AN συνθήκη2 ΤΟΤΕ
    εντολή-1
    εντολή-2
    ...
    εντολή-ν
  ΑΛΛΙΩΣ
    εντολή-1
    εντολή-2
    ...
    εντολή-ν
ΤΕΛΟΣ_ΑΝ
ΑΛΛΙΩΣ
  εντολή-1
  εντολή-2
  ...
  εντολή-ν
ΤΕΛΟΣ_ΑΝ
```

Εδώ έχει εμφωλευμένη εντολή **AN**

Εδώ όχι. Θα μπορούσε όμως.

Παράδειγμα:

Γράψτε ένα πρόγραμμα στη ΓΛΩΣΣΑ που διαβάζει από το πληκτρολόγιο τον μισθό ενός εργαζομένου και το τμήμα στο οποίο εργάζεται. Αν ο μισθός είναι πάνω από 2000 κι εργάζεται στο τμήμα Προγραμματισμού τότε να εμφανίζει επί της οθόνης το μήνυμα «Προγραμματιστής με υψηλό μισθό!» αλλιώς το μήνυμα «Εργαζόμενος με υψηλό μισθό!». Σε οποιοδήποτε άλλο τμήμα και να εργάζεται, αν ο μισθός είναι κάτω από 2000 να εμφανίζει το μήνυμα «Εργαζόμενος με χαμηλό μισθό».

ΠΡΟΓΡΑΜΜΑ Μισθός_και_Τμήμα_εργαζομένου

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: Ποσό

ΧΑΡΑΚΤΗΡΕΣ: Τμήμα

ΑΡΧΗ

!Είσοδος δεδομένων, εδώ του ποσού και του τμήματος.

ΓΡΑΨΕ 'Παρακαλώ, δώστε το ποσό του μισθού σας'

ΔΙΑΒΑΣΕ Ποσό

ΓΡΑΨΕ 'Παρακαλώ, δώστε το τμήμα που εργάζεστε'

ΔΙΑΒΑΣΕ Τμήμα


```
!Έλεγχος του ποσού και του τμήματος.  
AN Ποσό > 2000 ΤΟΤΕ  
    AN Τμήμα = 'Προγραμματισμού' ΤΟΤΕ  
        ΓΡΑΨΕ 'Προγραμματιστής με υψηλό μισθό!'  
    ΑΛΛΙΩΣ  
        ΓΡΑΨΕ 'Εργαζόμενος με υψηλό μισθό!'  
    ΤΕΛΟΣ_ΑΝ  
ΑΛΛΙΩΣ  
    ΓΡΑΨΕ 'Εργαζόμενος με χαμηλό μισθό!'  
ΤΕΛΟΣ_ΑΝ  
  
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Παρατηρήσεις:

- Η χρήση **εμφωλευμένων AN** ενδέχεται να οδηγήσει σε **πολύπλοκες δομές** με αποτέλεσμα να **αυξάνεται η πιθανότητα λογικού λάθους** και η **δυσκολία να κατανοήσει κάποιος άλλος προγραμματιστής το πρόγραμμά μας**.
- **Αντί** της χρήσης **εμφωλευμένων AN**, θα ήταν προτιμότερο να εξετάσουμε τη χρήση μίας ή περισσότερων **δομών AN με σύνθετη λογική συνθήκη** ή τη χρήση της **άλλης μορφής AN . . ΤΟΤΕ . . ΑΛΛΙΩΣ_ΑΝ** που θα δούμε παρακάτω.

Όσον αφορά με τη δεύτερη παρατήρηση, το παραπάνω τμήμα του προγράμματος με την **εμφωλευμένη δομή AN** θα μπορούσε να **γραφεί διαφορετικά με δομές AN με σύνθετη λογική συνθήκη**:

```
!Έλεγχος του ποσού και του τμήματος.  
AN Ποσό > 2000 ΚΑΙ Τμήμα = 'Προγραμματισμού' ΤΟΤΕ  
    ΓΡΑΨΕ 'Προγραμματιστής με υψηλό μισθό!'  
ΤΕΛΟΣ_ΑΝ  
  
AN Ποσό > 2000 ΚΑΙ Τμήμα <> 'Προγραμματισμού' ΤΟΤΕ  
    ΓΡΑΨΕ 'Εργαζόμενος με υψηλό μισθό!'  
ΤΕΛΟΣ_ΑΝ  
  
AN Ποσό <= 2000 ΤΟΤΕ  
    ΓΡΑΨΕ 'Εργαζόμενος με χαμηλό μισθό!'  
ΤΕΛΟΣ_ΑΝ
```

Ακόμα καλύτερη και πιο κατανοητή είναι η δόμηση με τη μορφή **AN . . ΤΟΤΕ . . ΑΛΛΙΩΣ_ΑΝ**

3η μορφή: **ΑΝ . . ΤΟΤΕ . . ΑΛΛΙΩΣ _ΑΝ**

Αυτή η 3^η μορφή είναι ένα είδος πολλαπλής επιλογής που προσφέρει καλύτερη κατανόηση του κώδικα αντί των εμφωλευμένων **ΑΝ**.

Σύνταξη:

ΑΝ συνθήκη1 **ΤΟΤΕ**

εντολή-1

εντολή-2

...

εντολή-ν

ΑΛΛΙΩΣ_ΑΝ συνθήκη2 **ΤΟΤΕ**

εντολή-1

εντολή-2

...

εντολή-ν

ΑΛΛΙΩΣ_ΑΝ συνθήκη3 **ΤΟΤΕ**

εντολή-1

εντολή-2

...

εντολή-ν

ΑΛΛΙΩΣ

εντολή-1

εντολή-2

...

εντολή-ν

ΤΕΛΟΣ_ΑΝ

Αν η συνθήκη1 είναι ΑΛΗΘΗΣ εκτελείται αυτή η ομάδα εντολών.

Αν η συνθήκη2 είναι ΑΛΗΘΗΣ εκτελείται αυτή η ομάδα εντολών.

Αν η συνθήκη3 είναι ΑΛΗΘΗΣ εκτελείται αυτή η ομάδα εντολών.

Αν καμία από τις παραπάνω συνθήκες δεν είναι ΑΛΗΘΗΣ εκτελείται αυτή η ομάδα εντολών.

Παρατηρήσεις:

- Όταν μία συνθήκη είναι ΑΛΗΘΗΣ τότε εκτελείται η αντίστοιχη ομάδα εντολών και κατόπιν η ροή εκτέλεσης του προγράμματος συνεχίζεται μετά το **ΤΕΛΟΣ_ΑΝ**.
- Αν καμία από τις συνθήκες δεν είναι ΑΛΗΘΗΣ τότε εκτελείται η ομάδα εντολών της **ΑΛΛΙΩΣ**.

Παράδειγμα 1

Τροποποιήστε το παράδειγμα της προηγούμενης υπο-ενότητας με τον εργαζόμενο και το μισθό με το τμήμα του ώστε από εμφωλευμένη δομή AN να γίνει σε δομή AN..ΤΟΤΕ..ΑΛΛΙΩΣ_ΑΝ.

ΠΡΟΓΡΑΜΜΑ Μισθός_και_Τμήμα_εργαζομένου
ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: Ποσό

ΧΑΡΑΚΤΗΡΕΣ: Τμήμα

ΑΡΧΗ

!Είσοδος δεδομένων, εδώ του ποσού και του τμήματος.

ΓΡΑΨΕ 'Παρακαλώ, δώστε το ποσό του μισθού σας'

ΔΙΑΒΑΣΕ Ποσό

ΓΡΑΨΕ 'Παρακαλώ, δώστε το τμήμα που εργάζεστε'

ΔΙΑΒΑΣΕ Τμήμα

!Έλεγχος του ποσού και του τμήματος.

ΑΝ Ποσό > 2000 **ΚΑΙ** Τμήμα = 'Προγραμματισμού' **ΤΟΤΕ**

ΓΡΑΨΕ 'Προγραμματιστής με υψηλό μισθό!'

ΑΛΛΙΩΣ_ΑΝ Ποσό > 2000 **ΚΑΙ** Τμήμα <> 'Προγραμματισμού'

ΓΡΑΨΕ 'Εργαζόμενος με υψηλό μισθό!'

ΑΛΛΙΩΣ_ΑΝ Ποσό <= 2000

ΓΡΑΨΕ 'Εργαζόμενος με χαμηλό μισθό!'

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Μάλλον δώσατε μη έγκυρο ποσό μισθού!'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Παράδειγμα 2

Ένας καταστηματάρχης έχει αποφασίσει να κάνει κάποιες εκπτώσεις στις πωλήσεις των προϊόντων του ανάλογα με τη συνολική αξία τους. Στον παρακάτω πίνακα, φαίνεται η συνολική αξία πώλησης (ή αγοράς από την πλευρά του καταναλωτή) και την αντίστοιχη έκπτωση.

Συνολική αξία (€)	Ποσοστό έκπτωσης
1-20	0%
21-50	5%
51-100	10%
>100	15%

Γράψτε ένα πρόγραμμα στη ΓΛΩΣΣΑ που διαβάζει από το πληκτρολόγιο την συνολική αξία πώλησης των προϊόντων και υπολογίζει και τυπώνει επί της οθόνης την αντίστοιχη έκπτωση καθώς και την συνολική αξία μετά την εφαρμογή της έκπτωσης.

Προφανώς, ο πίνακας παραθέτει τις αντίστοιχες περιπτώσεις που «μεταφράζονται» σε ισάριθμες επιλογές. Πρόκειται για περίπτωση πολλαπλής επιλογής:

ΠΡΟΓΡΑΜΜΑ Υπολογισμός_έκπτωσης_επί_της_αξίας_των_προϊόντων
ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: Συνολική_αξία, Έκπτωση, Τελική_συνολική_αξία
ΑΡΧΗ

!Είσοδος δεδομένων, εδώ της συνολικής αξίας των προϊόντων.

ΓΡΑΨΕ 'Παρακαλώ, δώστε τη συνολική αξία των προϊόντων'

ΔΙΑΒΑΣΕ Συνολική_αξία

!Υπολογισμοί έκπτωσης με βάση τον πίνακα των περιπτώσεων.

ΑΝ Συνολική_αξία <= 20 **ΤΟΤΕ**

Έκπτωση ← 0

ΑΛΛΙΩΣ_ΑΝ Συνολική_αξία <= 50

Έκπτωση ← Συνολική_αξία * 0.05

ΑΛΛΙΩΣ_ΑΝ Συνολική_αξία <= 100

Έκπτωση ← Συνολική_αξία * 0.10

ΑΛΛΙΩΣ

Έκπτωση ← Συνολική_αξία * 0.15

ΤΕΛΟΣ_ΑΝ

!Υπολογισμός τελικής αξίας μετά την έκπτωση.

Τελική_συνολική_αξία ← Συνολική_αξία - Έκπτωση

!Εκτύπωση επί της οθόνης της έκπτωσης και τελικής αξίας.

ΓΡΑΨΕ 'Η έκπτωση είναι ', Έκπτωση

ΓΡΑΨΕ 'Η τελική αξία μετά την έκπτωση είναι ',
& Τελική_συνολική_αξία

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

- ❖ Αν, για παράδειγμα, ο χρήστης δώσει ως Συνολική αξία το ποσό των **80€** τότε το πρόγραμμα θα διακλαδωθεί στην **τρίτη επιλογή** και θα υπολογίσει $80 * 0.10$.

- ❖ Αν δώσει **125€** θα διακλαδωθεί στην επιλογή του **ΑΛΛΙΩΣ** και θα υπολογίσει $125 * 0.15$ διότι όλες οι πιο πάνω συνθήκες αποτιμώνται ως ΨΕΥΔΗΣ.

Ο αναγνώστης ας δοκιμάσει να χρησιμοποιήσει στο τμήμα του υπολογισμού της έκπτωσης την αντίστοιχη εμφωλευμένη δομή **AN** και ας συγκρίνει τον κώδικα. *Ποιά δομή είναι καλύτερη από πλευράς κατανόησης και συντήρησης του κώδικα;*

Εντολή **ΕΠΙΛΕΞΕ**

Μία υλοποίηση της αλγοριθμικής δομής πολλαπλής επιλογής στη ΓΛΩΣΣΑ γίνεται με την εντολή **ΕΠΙΛΕΞΕ**. Η εντολή αυτή χρησιμοποιείται όταν έχουμε πολλές εναλλακτικές επιλογές όπου ακόμα και η χρήση της εντολής **AN . . ΤΟΤΕ . . ΑΛΛΙΩΣ _AN** κάνει πιο πολύπλοκο και δυσνόητο τον κώδικα.

Σύνταξη:

```
ΕΠΙΛΕΞΕ έκφραση
  ΠΕΡΙΠΤΩΣΗ λίστα-τιμών-1
    ομάδα_εντολών-1
  ΠΕΡΙΠΤΩΣΗ λίστα-τιμών-2
    ομάδα_εντολών-2
  . . .
  ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ
    ομάδα_εντολών-αλλιώς
ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
```

Η εντολή λειτουργεί ως εξής:

- Αρχικά, γίνεται **αποτίμηση της έκφρασης**. Συνήθως, εδώ γράφουμε το όνομα μίας μεταβλητής, την τιμή της οποίας επιθυμούμε να ελέγξουμε σε ποιά λίστα τιμών υπάγεται.
- Στη συνέχεια, **ελέγχεται αν η τιμή βρίσκεται στην πρώτη λίστα, δηλαδή της πρώτης περίπτωσης**. Αν ναι, τότε εκτελείται η πρώτη ομάδα εντολών και κατόπιν η ροή εκτέλεσης του προγράμματος μεταφέρεται μετά το **ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ**. Αν όχι, προχωράει να ελέγξει αν η τιμή βρίσκεται στη δεύτερη λίστα, δηλαδή της δεύτερης περίπτωσης κ.ο.κ.
- **Αν η τιμή δεν βρίσκεται σε καμία από τις πιο πάνω λίστες τότε εκτελείται η ομάδα εντολών της περίπτωσης **ΑΛΛΙΩΣ**.**

Παράδειγμα

Να γραφεί πρόγραμμα στη ΓΛΩΣΣΑ που διαβάζει τον αριθμό ενός μήνα (π.χ. 4) και εμφανίζει το πόσες ημέρες έχει (π.χ. 30 ημέρες).

Για λόγους απλότητας, θεωρούμε τον Φεβρουάριο ότι έχει 28 ημέρες. Επίσης, αν δοθεί αριθμός έξω από το διάστημα [1-12] να εμφανίζει σχετικό μήνυμα λάθους.

ΠΡΟΓΡΑΜΜΑ Ημέρες_ενός_μήνα

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Αριθμός_μήνα, Ημέρες_μήνα

ΑΡΧΗ

!Είσοδος δεδομένων, εδώ του αριθμού του μήνα.

ΓΡΑΨΕ 'Παρακαλώ, δώστε τον αριθμό του μήνα (1-12)'

ΔΙΑΒΑΣΕ Αριθμός_μήνα

!Έλεγχος του αριθμού μήνα και ανάλογα παίρνουμε
!τις ημέρες του.

ΕΠΙΛΕΞΕ Αριθμός_μήνα

ΠΕΡΙΠΤΩΣΗ 1, 3, 5, 7, 8, 10, 12

Ημέρες_μήνα ← 31

ΠΕΡΙΠΤΩΣΗ 4, 6, 9, 11

Ημέρες_μήνα ← 30

ΠΕΡΙΠΤΩΣΗ 2

Ημέρες_μήνα ← 28

ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ

Ημέρες_μήνα ← 0

ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ

!Εμφάνιση του αριθμού των ημερών

!εφόσον δεν υπήρξε λανθασμένη είσοδος δεδομένων.

ΑΝ Ημέρες_μήνα > 0 **ΤΟΤΕ**

ΓΡΑΨΕ 'Ο μήνας έχει ', Ημέρες_μήνα, ' ημέρες.'

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Δώσατε λανθασμένο αριθμό μήνα'

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

- ❖ Αν, για παράδειγμα, ο χρήστης δώσει τον **αριθμό 4** τότε το πρόγραμμα θα διακλαδωθεί στη **δεύτερη λίστα τιμών** (4, 6, 9, 11) και θα καταχωρήσει στη μεταβλητή Ημέρες_μήνα τον αριθμό 30. Στη συνέχεια, η ροή εκτέλεσης θα μεταφερθεί μετά το **ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ**. Εκεί, θα γίνει ο έλεγχος αν η τιμή της μεταβλητής Ημέρες_μήνα είναι πάνω από το 0. Θα αποτιμηθεί ως ΑΛΗΘΗΣ και θα εμφανιστεί το μήνυμα «Ο μήνας έχει 30 ημέρες».

- ❖ Αν ο χρήστης δώσει τον **αριθμό 13** τότε το πρόγραμμα θα διακλαδωθεί στην **ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ** και θα καταχωρήσει στη μεταβλητή `Ημέρες_μήνα` τον αριθμό 0. Στη συνέχεια, η ροή εκτέλεσης θα μεταφερθεί μετά το **ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ**. Εκεί, θα γίνει ο έλεγχος αν η τιμή της μεταβλητής `Ημέρες_μήνα` είναι πάνω από το 0. Θα αποτιμηθεί ως ΨΕΥΔΗΣ και θα εμφανιστεί το μήνυμα «Δώσατε λανθασμένο αριθμό μήνα».

Η εντολή **ΕΠΙΛΕΞΕ** προσφέρει μία συμπαγή δομή πολλαπλών επιλογών. Ας δούμε πώς θα υλοποιούνταν ισοδύναμα το τμήμα του προγράμματος της εντολής **ΕΠΙΛΕΞΕ** με τη χρήση της εντολής **ΑΝ...ΤΟΤΕ...ΑΛΛΙΩΣ_ΑΝ**:

!Έλεγχος του αριθμού μήνα και ανάλογα παίρνουμε το
!τις ημέρες του.
!Η χρήση των παρενθέσεων δεν είναι υποχρεωτική καθόσον
!δεν αλλάζει εδώ την προτεραιότητα των πράξεων αλλά τις
!τοποθετούμε για λόγους καλύτερης κατανόησης.

```
ΑΝ (Αριθμός_μήνα = 1) `Η (Αριθμός_μήνα = 3) `Η  
& (Αριθμός_μήνα = 5) `Η (Αριθμός_μήνα = 7) `Η  
& (Αριθμός_μήνα = 8) `Η (Αριθμός_μήνα = 10) `Η  
& (Αριθμός_μήνα = 12) ΤΟΤΕ
```

`Ημέρες_μήνα` ← 31

```
ΑΛΛΙΩΣ_ΑΝ (Αριθμός_μήνα = 4) `Η (Αριθμός_μήνα = 6) `Η  
& (Αριθμός_μήνα = 9) `Η (Αριθμός_μήνα = 11) ΤΟΤΕ
```

`Ημέρες_μήνα` ← 30

```
ΑΛΛΙΩΣ_ΑΝ (Αριθμός_μήνα = 2) ΤΟΤΕ  
Ημέρες_μήνα ← 28
```

```
ΑΛΛΙΩΣ  
Ημέρες_μήνα ← 0
```

```
ΤΕΛΟΣ_ΑΝ
```



Κάθε εντολή **ΑΝ...ΤΟΤΕ...ΑΛΛΙΩΣ_ΑΝ** μπορεί να μετατραπεί σε ισοδύναμη εντολή **ΕΠΙΛΕΞΕ** και το αντίστροφο.

Όμως, όταν έχουμε, τόσες πολλές εναλλακτικές περιπτώσεις επιλογών η εντολή **ΕΠΙΛΕΞΕ** προσφέρει ένα σαφές πλεονέκτημα συμπαγούς και ευκολοκατανόητου κώδικα.

Εντολές Επανάληψης

Οι εντολές επανάληψης υλοποιούν την αλγοριθμική δομή επανάληψης, όπου **μία ομάδα εντολών εκτελούνται συνεχώς μέχρι να ικανοποιηθεί μία συνθήκη ελέγχου**.

Στη ΓΛΩΣΣΑ είναι τρεις:

1. **ΟΣΟ . . ΕΠΑΝΑΛΑΒΕ**
2. **ΜΕΧΡΙΣ . . ΟΤΟΥ**
3. **ΓΙΑ . . ΑΠΟ . . ΜΕΧΡΙ**

Εντολή ΟΣΟ . . ΕΠΑΝΑΛΑΒΕ

Σύνταξη:

```
ΟΣΟ συνθήκη ΕΠΑΝΑΛΑΒΕ  
    εντολή-1  
    εντολή-2  
    ...  
    εντολή-ν  
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

Όσο η λογική συνθήκη είναι **ΑΛΗΘΗΣ** οι εντολές στο εσωτερικό της επαναλαμβάνονται. Όταν γίνει **ΨΕΥΔΗΣ** η ροή εκτέλεσης του προγράμματος μεταφέρεται μετά το **ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**

Παρατηρήσεις:

- Οι εντολές μπορούν να εκτελεστούν **0 η 1 ή περισσότερες φορές**. Αν η συνθήκη αποτιμηθεί εξαρχής ως **ΨΕΥΔΗΣ** τότε οι εντολές δεν θα εκτελεστούν καθόλου.

- Πρέπει κάπου μέσα στην επανάληψη να αλλάζει η τιμή της μεταβλητής που βρίσκεται στην συνθήκη ελέγχου ώστε κάποια στιγμή να τερματιστεί η επανάληψη. Αν δεν συμβεί αυτό τότε θα έχουμε ατέρμονα βρόχο, δηλαδή η επανάληψη θα εκτελείται συνεχώς.
- Είναι η καταλληλότερη εντολή όταν δεν γνωρίζουμε από πριν τον αριθμό των επαναλήψεων.
- Ενώ οι άλλες εντολές επανάληψης μπορούν να χρησιμοποιηθούν σε συγκεκριμένες περιπτώσεις, η **ΟΣΟ . . ΕΠΑΝΑΛΑΒΕ** μπορεί να χρησιμοποιηθεί **ΓΙΑ ΟΠΟΙΟΔΗΠΟΤΕ πρόβλημα που απαιτεί επαναληπτικότητα στην επίλυσή του (είτε γνωρίζουμε τον αριθμό των επαναλήψεων είτε όχι)**.

Παράδειγμα

Να γραφεί πρόγραμμα στη ΓΛΩΣΣΑ που διαβάσει μία ακολουθία ακέραιων αριθμών και υπολογίζει το άθροισμά τους. Η ακολουθία αριθμών τελειώνει με τον αριθμό 9999 (ο οποίος δεν συμμετέχει στο άθροισμα).

Όπως έχουμε δει και στους αλγορίθμους, η επίλυση αυτού του προβλήματος περιλαμβάνει μία επαναληπτικότητα. Διαβάσει έναν-έναν αριθμό και τον προσθέτει σε έναν συσσωρευτή (π.χ. Sum). Η συνεχής είσοδος αριθμών (δηλαδή, η επαναληπτικότητα) τελειώνει μόλις διαβαστεί ο συγκεκριμένος αριθμός 9999¹. Αυτό, υποδεικνύει και τη συνθήκη τερματισμού της επαναληπτικότητας της εισόδου των στοιχείων.

Προφανώς, το πρόβλημα δεν μας λέει πόσοι αριθμοί θα διαβαστούν, άρα δεν γνωρίζουμε το πλήθος τους. Το μόνο που μας λέει είναι ότι μόλις διαβαστεί ο αριθμός 9999 τότε η είσοδος των αριθμών τελειώνει. Η εντολή **ΟΣΟ..ΕΠΑΝΑΛΑΒΕ** είναι η πλέον κατάλληλη.

ΠΡΟΓΡΑΜΜΑ Άθροισμα_ακολουθίας_αριθμών

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: x, Sum

ΑΡΧΗ

!Αρχικοποίηση της μεταβλητής Sum.

!Ο συσσωρευτής Sum πρέπει να πάρει αρχική τιμή 0.

Sum ← 0

!Επαναληπτική είσοδος δεδομένων, εδώ ενός ακεραίου αριθμού. Καθένας που διαβάζεται προστίθεται στη Sum.

!Για να λειτουργήσει η εντολή **ΟΣΟ..ΕΠΑΝΑΛΑΒΕ** πρέπει να διαβαστεί ο πρώτος αριθμός πριν το ξεκίνημα της

¹ Τέτοια τιμή που χρησιμοποιείται για τον τερματισμό της εισόδου των δεδομένων αποκαλείται ως «τιμή φρουρός».

!επανάληψης.

ΓΡΑΨΕ 'Παρακαλώ, δώστε ακέραιο αριθμό (9999 για τέλος)'

ΔΙΑΒΑΣΕ x

ΟΣΟ x <> 9999 **ΕΠΑΝΑΛΑΒΕ**

!Πρόσθεσε τον αριθμό x που διάβασες στον συσσωρευτή Sum
Sum ← Sum + x

!Διάβασε τον επόμενο αριθμό.

ΓΡΑΨΕ 'Παρακαλώ, δώστε ακέραιο αριθμό (9999 για τέλος)'

ΔΙΑΒΑΣΕ x

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

!Η επανάληψη τελείωσε. Εμφάνισε την τιμή του συσσωρευτή
!Sum στην οθόνη (αποτέλεσμα).

ΓΡΑΨΕ 'Το άθροισμα των αριθμών που δώσατε είναι ', Sum

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

- ❖ Αν, για παράδειγμα, ο χρήστης δώσει τους αριθμούς 4, 5, 7 και 9999 τότε το αποτέλεσμα (Sum) είναι 16.
- ❖ Αν για παράδειγμα, ο χρήστης δώσει τους αριθμούς 2, 6, 7, -2, 4 και 9999 τότε το αποτέλεσμα (Sum) είναι 17.
- ❖ Αν, για παράδειγμα, ο χρήστης δώσει μόνο τον 9999 τί θα γίνει; Το αποτέλεσμα (Sum) είναι 0. Θα διαβαστεί ο αριθμός 9999 πριν την επανάληψη αλλά δεν θα μπει μέσα καθόσον η συνθήκη x <> 9999 είναι ΨΕΥΔΗΣ. Η μεταβλητή Sum ήδη έχει αρχικοποιηθεί με την τιμή 0 και αυτή η τιμή θα εμφανιστεί στο μήνυμα στο τέλος.

Άσκηση για τον αναγνώστη: Πώς θα μπορούσαμε να γνωρίζουμε το πλήθος των αριθμών (πλην του 9999) που διαβάστηκαν;

Πίνακας τιμών

Πολύ σημαντικό για την κατανόηση της επαναληπτικής διαδικασίας είναι να παρακολουθούμε πώς διαμορφώνονται οι τιμές των μεταβλητών σε κάθε επανάληψη. Σε αυτό βοηθάει ο λεγόμενος **πίνακας τιμών**.

Στην επικεφαλίδα του (πρώτη γραμμή) του πίνακα τοποθετούμε σε ξεχωριστές στήλες τα βήματα, τον αριθμό της επανάληψης, τα ονόματα των μεταβλητών και τη συνθήκη ελέγχου. Στις επόμενες γραμμές, συνήθως, τοποθετούμε τα βήματα της αρχικοποίησης των μεταβλητών (π.χ. συσσωρευτών, μετρητών κλπ.). Στη συνέχεια, παρακολουθούμε τη ροή εκτέλεσης σε κάθε επανάληψη, όπου βηματικά παρακολουθούμε τη ροή εκτέλεσης και διαμόρφωσης των μεταβλητών και της συνθήκης ελέγχου με τα δοκιμαστικά δεδομένα.

Στο προηγούμενο παράδειγμα της εντολής **ΟΣΟ . ΕΠΑΝΑΛΑΒΕ**, έχουμε μία επαναληπτική διαδικασία που διαβάζει μία ακολουθία αριθμών και υπολογίζει το άθροισμά τους. Ας υποθέσουμε ότι έχουμε τα εξής **δοκιμαστικά δεδομένα**:

4, 5, 7 και 9999

Σύμφωνα με το πρόγραμμα, ο πίνακας τιμών είναι ο εξής:

Βήμα	Επανάληψη	x	Συνθήκη ελέγχου x <> 9999	Sum
1				0
2		4		
3	1η		ΑΛΗΘΗΣ	
4	1η			0 + 4 = 4
5	1η	5		
6	2η		ΑΛΗΘΗΣ	
7	2η			4 + 5 = 9
8	2η	7		
9	3η		ΑΛΗΘΗΣ	
10	3η			9 + 7 = 16
11	3η	9999		
12	4η		ΨΕΥΔΗΣ	

ΑΣΚΗΣΗ για τον αναγνώστη: Δοκιμάστε να φτιάξετε τον πίνακα τιμών για τα δοκιμαστικά δεδομένα:

2, 6, 7, -2, 4, 9999

22, 9999

9999

Εντολή ΜΕΧΡΙΣ_ΟΤΟΥ

Σύνταξη:

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

εντολή-1

εντολή-2

...

εντολή-ν

ΜΕΧΡΙΣ_ΟΤΟΥ συνθήκη

Πρώτα εκτελούνται οι εντολές μέσα στο σώμα της επανάληψης και κατόπιν γίνεται ο έλεγχος της λογικής συνθήκης. Αν η λογική συνθήκη αποτιμηθεί σε ΨΕΥΔΗΣ τότε οι εντολές εκτελούνται ξανά. Αν, όμως, η λογική συνθήκη αποτιμηθεί σε ΑΛΗΘΗΣ τότε η επανάληψη σταματά.

Παρατηρήσεις:

- **Οι εντολές μπορούν να εκτελεστούν 1 ή περισσότερες φορές.** Αν η συνθήκη αποτιμηθεί εξαρχής ως ΨΕΥΔΗΣ τότε οι εντολές δεν θα εκτελεστούν καθόλου.
- **Πρέπει κάπου μέσα στην επανάληψη να αλλάζει η τιμή της μεταβλητής που βρίσκεται στην συνθήκη ελέγχου** ώστε κάποια στιγμή να τερματιστεί η επανάληψη. Αν δεν συμβεί αυτό τότε θα έχουμε *ατέρμονα βρόχο*, δηλαδή η επανάληψη θα εκτελείται συνεχώς.
- Την χρησιμοποιούμε κυρίως **όταν θέλουμε να ελέγξουμε την εγκυρότητα μίας τιμής που εισάγει ο χρήστης** καθώς και στον έλεγχο μίας από κάποιες προκαθορισμένες

απαντήσεις του χρήστη ή επιλογών από κάποιο μενού.

Παράδειγμα 1

Να γραφεί πρόγραμμα στη ΓΛΩΣΣΑ που διαβάζει τον μισθό ενός εργαζομένου, ο οποίος πρέπει να είναι θετικός αριθμός και κάτω από 3000 €. **Το πρόγραμμα πρέπει να ελέγχει αν ο μισθός είναι έγκυρος.**

ΠΡΟΓΡΑΜΜΑ Έλεγχος_εγκυρότητας_μισθού

ΜΕΤΑΒΛΗΤΕΣ

ΠΡΑΓΜΑΤΙΚΕΣ: Μισθός

ΑΡΧΗ

!Η επανάληψη θα συνεχίζεται όσο το ποσό που δίνει ο
!χρήστης δεν είναι έγκυρο.

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Παρακαλώ, δώστε το ποσό του μισθού: '

ΔΙΑΒΑΣΕ Μισθός

ΜΕΧΡΙΣ_ΟΤΟΥ Μισθός > 0 **ΚΑΙ** Μισθός < 3000

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Όταν η συνθήκη $\text{Μισθός} > 0$ **ΚΑΙ** $\text{Μισθός} < 3000$ αποτιμηθεί σε ΑΛΗΘΗΣ τότε το ποσό είναι έγκυρο και η επανάληψη σταματά.

Το παραπάνω τμήμα κώδικα αποτελεί ένα τυπικό δείγμα ελέγχου εγκυρότητας εισαγόμενων δεδομένων.

Παράδειγμα 2

Μία επιχείρηση επιθυμεί να δώσει ένα επίδομα 100 € σε όσους εργαζόμενους έχουν 3 ή περισσότερα παιδιά. Γράψτε ένα πρόγραμμα στη ΓΛΩΣΣΑ που διαβάζει το ονοματεπώνυμο και τον μισθό ενός εργαζομένου και αν είναι πολύτεκνος. Σε περίπτωση που η απάντηση είναι καταφατική τότε να προστίθεται το επίδομα στο μισθό και να τον εμφανίζει στην οθόνη.

ΠΡΟΓΡΑΜΜΑ Επίδομα_πολυτέκνου

ΣΤΑΘΕΡΕΣ

Επίδομα = 100

ΜΕΤΑΒΛΗΤΕΣ

ΧΑΡΑΚΤΗΡΕΣ: Ονοματεπώνυμο, Απάντηση

ΠΡΑΓΜΑΤΙΚΕΣ: Μισθός

ΛΟΓΙΚΕΣ: Είναι_πολύτεκνος

ΑΡΧΗ

!Είσοδος των δεδομένων.

ΓΡΑΨΕ 'Παρακαλώ, δώστε ονοματεπώνυμο του εργαζομένου: '
ΔΙΑΒΑΣΕ Ονοματεπώνυμο

ΓΡΑΨΕ 'Παρακαλώ, δώστε το ποσό του μισθού: '
ΔΙΑΒΑΣΕ Μισθός

!Ρώτησε εάν είναι πολύτεκνος. Δεν θα προχωρήσεις αν δεν
!πάρεις καταφατική ή αρνητική απάντηση (N/O).

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ 'Είναι πολύτεκνος; (N/O): '
ΔΙΑΒΑΣΕ Απάντηση

ΜΕΧΡΙΣ_ΟΤΟΥ Απάντηση = 'N' **Ή** Απάντηση = 'O'

!Βάλε στη λογική μεταβλητή την τιμή ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ
!ανάλογα με την απάντηση.

ΑΝ Απάντηση = 'N' **ΤΟΤΕ**
Είναι_πολύτεκνος = 'ΑΛΗΘΗΣ'

ΑΛΛΙΩΣ
Είναι_πολύτεκνος = 'ΨΕΥΔΗΣ'

ΤΕΛΟΣ_ΑΝ

!Έλεγξε την λογική μεταβλητή και ανάλογα πρόσθεσε ή όχι το
!επίδομα στον μισθό και εμφάνισε κατάλληλο μήνυμα στην
!οθόνη.

ΑΝ Είναι_πολύτεκνος **ΤΟΤΕ**

Μισθός ← Μισθός + Επίδομα

ΓΡΑΨΕ 'Ο εργαζόμενος ', Ονοματεπώνυμο, ' είναι
& πολύτεκνος και ο μισθός του διαμορφώνεται
& σε: ', Μισθός

ΑΛΛΙΩΣ

ΓΡΑΨΕ 'Ο εργαζόμενος ', Ονοματεπώνυμο, ' δεν είναι
& πολύτεκνος και ο μισθός του είναι: '
& , Μισθός

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Στο παραπάνω παράδειγμα, βλέπουμε την ιδανική χρήση της εντολής **ΜΕΧΡΙΣ_ΟΤΟΥ** σε μία ερώτηση που επαναλαμβάνεται συνεχώς μέχρι να πάρει την απάντηση 'N' ή 'O' από τον χρήστη.

Παράδειγμα 3

Σε ένα φουτουριστικό εστιατόριο οι πελάτες κάθονται σε τραπέζια τα οποία διαθέτουν μία οθόνη αφής μέσω της οποίας μπορούν να επιλέξουν το φαγητό τους από ένα μενού. Το μενού έχει τις εξής επιλογές:

1. Μακαρονάδα με σάλτσα
2. Μακαρονάδα με κιμά
3. Μουσακάς
4. Γεμιστά
5. Γάβρος στον φούρνο
6. Έξοδος, δεν θα παραγγείλω

Ο πελάτης θα πατάει τον αντίστοιχο αριθμό και αυτόματα η παραγγελία θα μεταβιβάζεται στον σεφ. Στον πελάτη, απλά, θα εμφανίζεται στην οθόνη ένα μήνυμα αποδοχής (ή και ευχαριστίας).

Γράψτε ένα πρόγραμμα στη ΓΛΩΣΣΑ που υλοποιεί την παραπάνω διαδικασία.

Ο σκοπός εδώ είναι να καταδείξουμε άλλη μία ιδανική χρήση της εντολής **ΜΕΧΡΙΣ_ΟΤΟΥ**. Στον χρήστη εμφανίζεται ένα μενού αριθμημένων επιλογών από το 1 μέχρι το 6, το οποίο θα εμφανίζεται συνεχώς μέχρι να κάνει μία έγκυρη επιλογή.

ΠΡΟΓΡΑΜΜΑ Παραγγελία

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: Επιλογή

ΑΡΧΗ

!Το μενού θα εμφανίζεται συνεχώς μέχρι ο πελάτης να δώσει
!έγκυρη επιλογή.

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

```
ΓΡΑΨΕ \           ΜΕΝΟΥ ΕΔΕΣΜΑΤΩΝ           '  
ΓΡΑΨΕ \====='  
ΓΡΑΨΕ `1. Μακαρονάδα με σάλτσα`  
ΓΡΑΨΕ `2. Μακαρονάδα με κιμά`  
ΓΡΑΨΕ `3. Μουσακάς`  
ΓΡΑΨΕ `4. Γεμιστά`  
ΓΡΑΨΕ `5. Γάβρος στον φούρνο`  
ΓΡΑΨΕ `6. Έξοδος - δεν θα παραγγείλω`  
ΓΡΑΨΕ \====='  
ΔΙΑΒΑΣΕ Επιλογή
```

ΜΕΧΡΙΣ_ΟΤΟΥ Επιλογή \geq 1 **ΚΑΙ** Επιλογή \leq 6

!Αν έδωσε έγκυρη επιλογή και διαφορετική του 6 να
!εμφανιστεί ένα μήνυμα ευχαριστίας.
!Αν έχει επιλεγεί το 6 να εμφανιστεί ένα κατάλληλο μήνυμα
!αποχαιρετισμού.

```
AN Επιλογή <> 6 ΤΟΤΕ
    ΓΡΑΨΕ 'Ευχαριστούμε για την επιλογή σας!'
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Θα χαρούμε πολύ να σας ξαναδούμε!'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Εντολή ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ

Η εντολή αυτή είναι **ιδανική όταν είναι γνωστός από πριν ο αριθμός των επαναλήψεων.**

Σύνταξη:

```
ΓΙΑ μεταβλητή ΑΠΟ αρχική_τιμή ΜΕΧΡΙ τελική_τιμή ΜΕ ΒΗΜΑ
    εντολή-1
    εντολή-2
    ...
    εντολή-ν
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
                                τιμή_βήματος
```

Παρατηρήσεις:

- Η εντολή χειρίζεται μία μεταβλητή που ονομάζεται **μεταβλητή ελέγχου**.
- Η μεταβλητή ελέγχου λαμβάνει τιμές ξεκινώντας από την αρχική_τιμή και μόλις ξεπεράσει την τελική_τιμή τότε ο βρόχος σταματάει (και η ροή της εκτέλεσης του προγράμματος πηγαίνει στην επόμενη εντολή μετά το **ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**).
- Μόλις ολοκληρωθεί για πρώτη φορά η εκτέλεση των εντολών τότε η τιμή της μεταβλητής ελέγχου αυξάνεται κατά την τιμή_βήματος.²
- Αν το βήμα είναι κατά 1 τότε το τμήμα ΜΕ ΒΗΜΑ μπορεί να παραληφθεί.
- Αν η μεταβλητή ξεκινάει από την τιμή 1 και αυξάνεται κατά 1 τότε θεωρείται και ως μετρητής επαναλήψεων.

² Αν το βήμα είναι αρνητικό (π.χ. -1) τότε η τιμή της μεταβλητής ελέγχου μειώνεται κατά την τιμή του βήματος.

- Κάθε εντολή **ΓΙΑ . . ΑΠΟ . . ΜΕΧΡΙ** μπορεί να μετατραπεί σε **ΟΣΟ . . ΕΠΑΝΑΛΑΒΕ** και **ΜΕΧΡΙΣ_ΟΤΟΥ**. Το αντίθετο δεν ισχύει (γιατί;).

Παράδειγμα 1

Να γραφεί πρόγραμμα στη ΓΛΩΣΣΑ που εκτυπώνει στην οθόνη όλους τους τριψήφιους αριθμούς που διαιρούνται ακριβώς με το 3. Επίσης, να μας εμφανίζει και πόσοι είναι τελικά αυτοί οι αριθμοί.

Προφανώς ο πρώτος τριψήφιος είναι ο 100 και ο τελευταίος είναι ο 999. Άρα, θα ξεκινήσουμε από τον 100 (αρχική τιμή) και θα φτάσουμε στο 999 (τελική τιμή). Σε κάθε επανάληψη θα ελέγχουμε αν ο τρέχον αριθμός (της μεταβλητής ελέγχου) διαιρείται ακριβώς με το 3. Αν ναι (υπόλοιπο 0), τότε τον τυπώνουμε αλλά και αυξάνουμε έναν μετρητή.

Με βάση αυτές τις σκέψεις ξεκινάμε τη γραφή του προγράμματος:

```
ΠΡΟΓΡΑΜΜΑ Εκτύπωση_τριψηφίων_που_διαιρούνται_με_το_3
```

```
ΜΕΤΑΒΛΗΤΕΣ
```

```
  ΑΚΕΡΑΙΕΣ: x, πόσοι_είναι
```

```
ΑΡΧΗ
```

```
  !Πρώτα, αρχικοποίηση του μετρητή.
```

```
  πόσοι_είναι ← 0
```

```
  !Ξεκινάει ο βρόχος.
```

```
  ΓΙΑ x ΑΠΟ 100 ΜΕΧΡΙ 999
```

```
    ΑΝ (x MOD 3) = 0 ΤΟΤΕ
```

```
      ΓΡΑΨΕ 'Ο αριθμός ', x, ' διαιρείται με το 3'
```

```
      πόσοι_είναι ← πόσοι_είναι + 1
```

```
    ΤΕΛΟΣ_ΑΝ
```

```
  ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
  !Ο βρόχος τελείωσε. Εμφάνισε την τιμή του μετρητή.
```

```
  ΓΡΑΨΕ 'Καταμετρήθηκαν ', πόσοι_είναι, ' τριψήφιοι αριθμοί  
    & που διαιρούνται ακριβώς με το 3'
```

```
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ
```

Επειδή η μεταβλητή ελέγχου (x) αυξάνεται κατά 1, το τμήμα **ΜΕ ΒΗΜΑ 1** της **ΓΙΑ . . ΑΠΟ . . ΜΕΧΡΙ** μπορεί να παραλειφθεί.

Επίσης, δεν ξεχνάμε ποτέ να δώσουμε αρχικές τιμές σε τυχόν μετρητές, συσσωρευτές κ.λπ. συνήθως στην αρχή του προγράμματος ή πριν από έναν βρόχο.

Παράδειγμα 2

Μία επιχείρηση διατηρεί 5 υποκαταστήματα σε όλη την Ελλάδα αλλά κυρίως την ενδιαφέρει το υποκατάστημα της Αττικής. Το συγκεκριμένο υποκατάστημα απασχολεί 8 πωλητές. Την επιχείρηση την ενδιαφέρει ποιός πωλητής έκανε τις μεγαλύτερες, σε χρηματική αξία, πωλήσεις κατά τον τρέχοντα μήνα.

Γράψτε ένα πρόγραμμα στη ΓΛΩΣΣΑ, που διαβάζει τα ονοματεπώνυμα των 8 πωλητών και την αξία των πωλήσεων του καθενός και υπολογίζει και εμφανίζει το ονοματεπώνυμο και την αξία του «καλύτερου» πωλητή.

Πρόκειται για μία υλοποίηση του αλγορίθμου της εύρεσης μέγιστης τιμής.

ΠΡΟΓΡΑΜΜΑ Ο_καλύτερος_πωλητής_στο_υποκατάστημα_Αττικής
ΜΕΤΑΒΛΗΤΕΣ

ΧΑΡΑΚΤΗΡΕΣ: Ονοματεπώνυμο, Ονοματεπώνυμο_καλύτερου

ΠΡΑΓΜΑΤΙΚΕΣ: Αξία_πωλήσεων, Αξία_πωλήσεων_καλύτερου

ΑΚΕΡΑΙΕΣ: i !Η μεταβλητή ελέγχου του βρόχου, μετρητής.

ΑΡΧΗ

!Αρχικοποίηση της μέγιστης αξίας.

Αξία_πωλήσεων_καλύτερου ← 0

!Ξεκινάει ο βρόχος, για τους 8 πωλητές.

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 8

!Είσοδος των δεδομένων του πωλητή.

ΓΡΑΨΕ 'Παρακαλώ, δώστε το ονοματεπώνυμο του πωλητή'

ΔΙΑΒΑΣΕ Ονοματεπώνυμο

ΓΡΑΨΕ 'Παρακαλώ, δώστε την αξία πωλήσεων του πωλητή'

ΔΙΑΒΑΣΕ Αξία_πωλήσεων

!Έλεγχξε αν ο τρέχων πωλητής έχει κάνει μεγαλύτερες
!πωλήσεις.

ΑΝ Αξία_πωλήσεων > Αξία_πωλήσεων_καλύτερου **ΤΟΤΕ**

Ονοματεπώνυμο_καλύτερου ← Ονοματεπώνυμο

Αξία_πωλήσεων_καλύτερου ← Αξία_πωλήσεων

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

!Ο βρόχος τελείωσε. Εμφάνισε τα στοιχεία του «καλύτερου».

ΓΡΑΨΕ 'Καλύτερος πωλητής : ', Ονοματεπώνυμο_καλύτερου

ΓΡΑΨΕ 'με αξία πωλήσεων : ', Αξία_πωλήσεων_καλύτερου

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Μία σημαντική παρατήρηση: Το πρόγραμμα δεν είναι ικανοποιητικό αν υπάρχουν δύο ή περισσότεροι πωλητές που πραγματοποίησαν ισάξιες πωλήσεις. Στην περίπτωση αυτή, το πρόγραμμα θα κρατήσει κι εμφανίσει το ονοματεπώνυμο του τελευταίου που εισάγαμε (κάτι άδικο!).

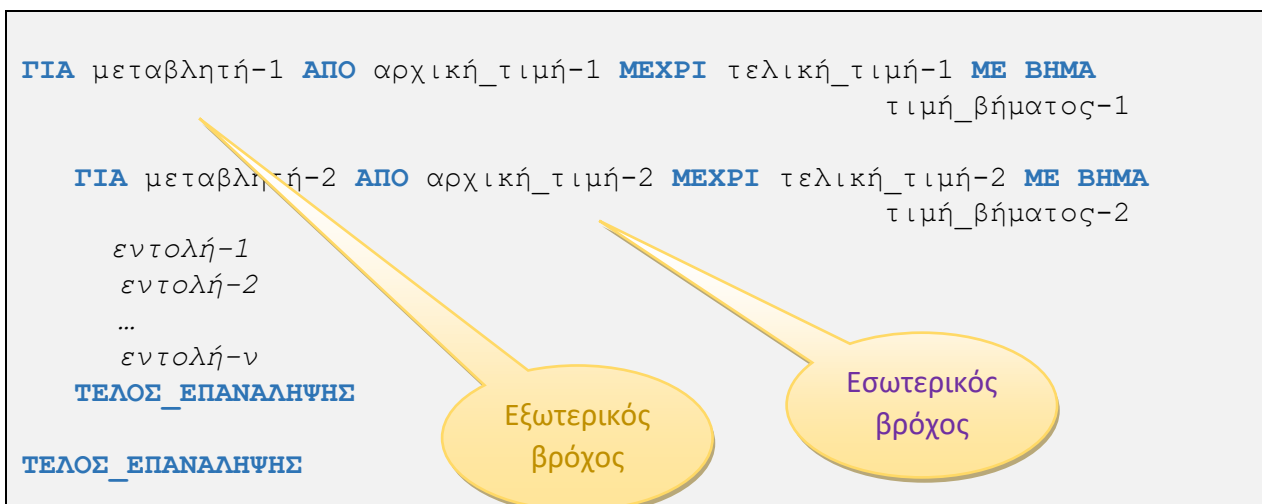
Μπορείτε να βελτιώσετε το πρόγραμμα ώστε να αντιμετωπιστεί αυτή η ατέλεια;

Εμφωλευμένοι βρόχοι

Όπως και με τις εμφωλευμένες εντολές **AN** έτσι κι εδώ, για πιο σύνθετα προβλήματα, μπορούμε να χρησιμοποιήσουμε εμφωλευμένες εντολές επανάληψης, δηλαδή έναν βρόχο μέσα σε έναν άλλο βρόχο.

Ένα παράδειγμα σύνταξης με τη χρήση δύο εντολών **ΓΙΑ . . ΑΠΟ . . ΜΕΧΡΙ** είναι το εξής:

Σύνταξη δύο εμφωλευμένων ΓΙΑ . . ΑΠΟ . . ΜΕΧΡΙ:



Η πρώτη εντολή **ΓΙΑ . . ΑΠΟ . . ΜΕΧΡΙ** συνιστά τον **εξωτερικό βρόχο** και η δεύτερη εντολή **ΓΙΑ . . ΑΠΟ . . ΜΕΧΡΙ** που βρίσκεται εντός της πρώτης συνιστά τον **εσωτερικό βρόχο**.

Παρατηρήσεις:

- Ο εσωτερικός βρόχος βρίσκεται ολόκληρος εντός του εξωτερικού.
- Η είσοδος σε κάθε βρόχο γίνεται από την αρχή του.
- Δεν μπορεί να χρησιμοποιηθεί η ίδια μεταβλητή ελέγχου ή μετρητής και στους δύο.

- Δεν είναι υποχρεωτικό οι βρόχοι να είναι του ίδιου τύπου, δηλαδή με τις ίδιες εντολές επανάληψης. Για παράδειγμα, μπορεί ο εξωτερικός βρόχος να υλοποιείται με μία εντολή **ΓΙΑ** .. **ΑΠΟ** .. **ΜΕΧΡΙ** και ο εσωτερικός με μία εντολή **ΟΣΟ** .. **ΕΠΑΝΑΛΑΒΕ**.

Παράδειγμα

Θα επεκτείνουμε το παράδειγμα 2 της προηγούμενης ενότητας με την επιχείρηση που έχει 5 υποκαταστήματα και αναζητά τον «καλύτερο» πωλητή. Κάθε υποκατάστημα απασχολεί σταθερά 8 πωλητές.

Τώρα, το πρόγραμμα θα διαβάσει τα ονοματεπώνυμα των 8 πωλητών και την αξία των πωλήσεων του καθενός από κάθε υποκατάστημα και θα υπολογίζει και εμφανίζει το ονοματεπώνυμο και την αξία του «καλύτερου» πωλητή.

Το πρόγραμμα είναι παρόμοιο με τη διαφορά ότι έχει εμφωλευμένο βρόχο.

ΠΡΟΓΡΑΜΜΑ Ο_καλύτερος_πωλητής_σε_όλα_τα_υποκαταστήματα
ΜΕΤΑΒΛΗΤΕΣ

ΧΑΡΑΚΤΗΡΕΣ: Ονοματεπώνυμο, Ονοματεπώνυμο_καλύτερου

ΠΡΑΓΜΑΤΙΚΕΣ: Αξία_πωλήσεων, Αξία_πωλήσεων_καλύτερου

ΑΚΕΡΑΙΕΣ: i, j !Η μεταβλητή ελέγχου του εξωτερικού και
!εσωτερικού βρόχου, αντίστοιχα.

ΑΡΧΗ

!Αρχικοποίηση της μέγιστης αξίας.

Αξία_πωλήσεων_καλύτερου ← 0

!Ξεκινάει ο εξωτερικός βρόχος, για τα 5 υποκαταστήματα.

ΓΙΑ i **ΑΠΟ** 1 **ΜΕΧΡΙ** 5

!Ξεκινάει ο εσωτερικός βρόχος, για τους 8 πωλητές
!του υποκαταστήματος.

ΓΙΑ j **ΑΠΟ** 1 **ΜΕΧΡΙ** 8

!Είσοδος των δεδομένων του πωλητή.

ΓΡΑΨΕ 'Παρακαλώ, δώστε το ονοματεπώνυμο του πωλητή'

ΔΙΑΒΑΣΕ Ονοματεπώνυμο

ΓΡΑΨΕ 'Παρακαλώ, δώστε την αξία πωλήσεων του πωλητή'

ΔΙΑΒΑΣΕ Αξία_πωλήσεων

!Έλεγχξε αν ο τρέχων πωλητής έχει κάνει μεγαλύτερες
!πωλήσεις.

ΑΝ Αξία_πωλήσεων > Αξία_πωλήσεων_καλύτερου **ΤΟΤΕ**

Ονοματεπώνυμο_καλύτερου ← Ονοματεπώνυμο

Αξία_πωλήσεων_καλύτερου ← Αξία_πωλήσεων

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ !εσωτερικού βρόχου

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ !εξωτερικού βρόχου

!Ο βρόχος τελείωσε. Εμφάνισε τα στοιχεία του «καλύτερου».

ΓΡΑΨΕ 'Καλύτερος πωλητής : ', Ονοματεπώνυμο_καλύτερου

ΓΡΑΨΕ 'με αξία πωλήσεων : ', Αξία_πωλήσεων_καλύτερου

ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

Σχετικά προβλήματα όπως ο καλύτερος πωλητής ανά υποκατάστημα ή συνολική αξία των πωλήσεων ανά υποκατάστημα κ.λπ. αντιμετωπίζονται καλύτερα με τη χρήση πινάκων, που θα εξετάσουμε στο επόμενο κεφάλαιο.

Ερωτήσεις κατανόησης

1. Ποιοί είναι οι συγκριτικοί τελεστές στη ΓΛΩΣΣΑ;
2. Τί τιμή μπορεί να δώσει μία λογική έκφραση;
3. Με ποιούς λογικούς τελεστές συνδέονται μεταξύ τους οι λογικές εκφράσεις;
4. Ο συμμαθητής σας ισχυρίζεται ότι οι παρακάτω δύο λογικές εκφράσεις είναι ισοδύναμες. Έχει δίκιο;
 $x <> 12$
ΌΧΙ ($x = 12$)
5. Παρομοίως για τις επόμενες δύο. Έχει δίκιο;
 $x >= 5$
ΌΧΙ ($x < 4$)
6. Αλλά για τις παρακάτω δύο λέει ότι δεν είναι ισοδύναμες. Έχει δίκιο;
 $x >= 5$ **ΚΑΙ** $x <= 20$
ΌΧΙ ($x < 4$ **Η** $x > 20$)

7. Ποιά η σύνταξη των εντολών **AN . . ΤΟΤΕ** και **AN . . ΤΟΤΕ . . ΑΛΛΙΩΣ**; Ποιά η διαφορά τους;
8. Τί είναι οι εμφωλευμένες εντολές **AN**;
9. Ο φίλος σας ισχυρίζεται ότι οι εμφωλευμένες εντολές **AN** μπορούν να οδηγήσουν σε δυσκολία κατανόησης του προγράμματος και ότι ενδεχομένως η χρήση σύνθετων λογικών συνθηκών σε απλές εντολές **AN** να πετύχει το ίδιο αποτέλεσμα. Συμφωνείτε;
10. Πότε θα χρησιμοποιήσετε μία εντολή **AN . . ΤΟΤΕ . . ΑΛΛΙΩΣ _AN**;
11. Πότε θα χρησιμοποιήσετε μία εντολή **ΕΠΙΛΕΞΕ**;
12. Δίνεται η παρακάτω εντολή:

AN $x = 0$ **H** $x = 2$ **H** $x = 4$ **H** $x = 6$ **H** $x = 8$ **ΤΟΤΕ**
ΓΡΑΨΕ 'Ο x είναι ζυγός'

ΑΛΛΙΩΣ _AN $x = 1$ **H** $x = 3$ **H** $x = 5$ **H** $x = 7$ **H** $x = 9$
ΓΡΑΨΕ 'Ο x είναι περιττός'

ΑΛΛΙΩΣ
ΓΡΑΨΕ 'Η τιμή του x είναι άλλη'

ΤΕΛΟΣ _AN

Μετατρέψτε την παραπάνω εντολή σε εντολή **ΕΠΙΛΕΞΕ**.

13. Ποιά είναι η σύνταξη της εντολής **ΟΣΟ . . ΕΠΑΝΑΛΑΒΕ**;
14. Ο φίλος σας ισχυρίζεται ότι η εντολή **ΟΣΟ . . ΕΠΑΝΑΛΑΒΕ**:
 - α) Χρησιμοποιείται κυρίως όταν είναι γνωστός από πριν (προκαθορισμένος) ο αριθμός των επαναλήψεων.
 - β) Μπορεί οι εντολές που περιέχει να μην εκτελεστούν καθόλου.Σε ποιά περίπτωση έχει δίκιο ή μήπως έχει σε όλα δίκιο;
15. Ποιά είναι η σύνταξη της εντολής **ΜΕΧΡΙΣ . . ΟΤΟΥ**;
16. Δίνεται το παρακάτω τμήμα κώδικα:

$k \leftarrow 10$

ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ

ΓΡΑΨΕ k

$k \leftarrow k - 2$

ΜΕΧΡΙΣ_ΟΤΟΥ $k = 1$

α) Ποιές τιμές του k θα εμφανιστούν στην οθόνη;

β) Πότε θα τερματιστεί ο βρόχος;

17. Αναφέρατε περιπτώσεις που ενδείκνυται να χρησιμοποιείται η εντολή

ΜΕΧΡΙΣ..ΟΤΟΥ.

18. Ποιά είναι η σύνταξη της εντολής **ΓΙΑ..ΑΠΟ..ΜΕΧΡΙ;**

19. Δίνεται το παρακάτω τμήμα κώδικα:

ΓΙΑ x **ΑΠΟ** 0 **ΜΕΧΡΙ** 10 **ΜΕ ΒΗΜΑ** 2

$y \leftarrow 2 * x + 1$

$z \leftarrow y \text{ MOD } 3$

ΓΡΑΨΕ x, y, z

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

α) Πόσες φορές θα εκτελεστεί η επανάληψη;

β) Δώστε τον πίνακα τιμών, δηλαδή πώς διαμορφώνονται οι τιμές των x, y, z κατά την ροή εκτέλεσης του προγράμματος.

γ) Μετατρέψτε τον παραπάνω βρόχο σε ισοδύναμο με τη χρήση **ΟΣΟ..ΕΠΑΝΑΛΑΒΕ.**

20. Δίνεται το παρακάτω τμήμα προγράμματος:

$x \leftarrow 5$

μετρητής $\leftarrow 0$

ΟΣΟ μετρητής ≤ 3 **ΕΠΑΝΑΛΑΒΕ**

ΑΝ $(x \text{ MOD } 3) = 0$ **ΤΟΤΕ**

μετρητής \leftarrow μετρητής + 1

ΤΕΛΟΣ_ΑΝ

$x \leftarrow x + 5$

ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

- α)** Περιγράψτε συνοπτικά τί κάνει το παραπάνω τμήμα προγράμματος.
- β)** Κατασκευάστε τον πίνακα των τιμών.
- γ)** Μετατρέψτε τον βρόχο σε ισοδύναμο με τη χρήση της **ΜΕΧΡΙΣ . . ΟΤΟΥ**.
- δ)** Θα μπορούσαμε να χρησιμοποιήσουμε την εντολή **ΓΙΑ . . ΑΠΟ . . ΜΕΧΡΙ**;
Δικαιολογήστε την απάντησή σας.

ΤΕΛΟΣ ΣΗΜΕΙΩΣΕΩΝ ΚΕΦΑΛΑΙΟΥ 8