

8.

Επιλογή και επανάληψη



### Εισαγωγή

Στο προηγούμενο κεφάλαιο αναπτύξαμε προγράμματα, τα οποία ήταν πολύ απλά και οι εντολές των οποίων εκτελούνται η μία μετά την άλλη. Αυτή η σειριακή εκτέλεση των εντολών είναι κατάλληλη όμως μόνο για πολύ απλά προγράμματα, τα οποία εισάγουν δεδομένα, τα επεξεργάζονται και τυπώνουν το αποτέλεσμα, χωρίς να υπάρχει η δυνατότητα της επιλεκτικής εκτέλεσης τμημάτων του προγράμματος, σύμφωνα με την τιμή κάποιων δεδομένων ή την επανάληψη τμημάτων του προγράμματος. Όπως έχουμε αναφέρει οι τρεις βασικές δομές, είναι η δομή της ακολουθίας, της επιλογής και της επανάληψης. Οι δομές αυτές αποτελούν τη βάση του δομημένου προγραμματισμού και με τη χρήση αυτών μπορούν να υλοποιηθούν όλα τα προγράμματα υπολογιστών. Στο κεφάλαιο αυτό θα ασχοληθούμε με τις δύο αυτές βασικές δομές της επιλογής και της επανάληψης που θα μας επιτρέψουν την συγγραφή πληρέστερων και πιο πολύπλοκων προγραμμάτων.



### Διδακτικοί στόχοι

Να είναι σε θέση ο μαθητής:

- ⇒ Να σχηματίζει λογικές εκφράσεις, απλές και σύνθετες.
- ⇒ Να διατυπώνει τις μορφές της εντολής ελέγχου (επιλογής) ΑΝ.
- ⇒ Να διακρίνει τις διαφορές των μορφών της εντολής ΑΝ.
- ⇒ Να επιλέγει την καλύτερη μορφή της εντολής ΑΝ για το κάθε πρόγραμμα.
- ⇒ Να διατυπώνει τις εντολές επανάληψης.
- ⇒ Να επιλέγει την καλύτερη δομή επανάληψης και να χρησιμοποιεί την κατάλληλη εντολή.
- ⇒ Να συντάσσει προγράμματα τα οποία χρησιμοποιούν και τις τρεις βασικές δομές: της ακολουθίας, της επιλογής και της επανάληψης.



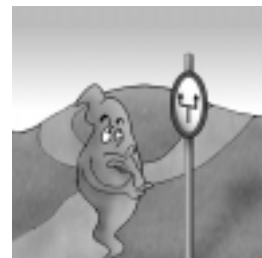
### Προερωτήσεις

- ✓ Γιατί χρησιμοποιούνται οι αλγοριθμικές δομές;
- ✓ Νομίζεις ότι όλες οι αλγοριθμικές δομές έχουν τις αντίστοιχες εντολές σε μία γλώσσα προγραμματισμού;
- ✓ Η δομή της επιλογής είναι σημαντική για την επίλυση προβλημάτων;
- ✓ Αρκεί μία εντολή για να εκφράσει την δομή της επανάληψης;
- ✓ Πώς μπορεί να ελέγχεται ο τερματισμός μίας επανάληψης;

## 8.1 Εντοχές Επιλογής

Μία από τις βασικότερες δομές που εμφανίζονται σε ένα πρόγραμμα, είναι η επιλογή. Σχεδόν σε όλα τα προβλήματα περιλαμβάνονται κάποιοι έλεγχοι και ανάλογα με το αποτέλεσμα αυτών των ελέγχων επιλέγονται οι ενέργειες που θα ακολουθήσουν.

Ας θεωρήσουμε το πολύ απλό πρόβλημα της καταμέτρησης των θετικών και των αρνητικών αριθμών. Πρέπει λοιπόν να γράψουμε ένα πρόγραμμα, το οποίο εισάγει αριθμούς και μετράει πόσοι από αυτούς είναι θετικοί και πόσοι αρνητικοί. Για να αποφασίσουμε, αν ένας αριθμός είναι θετικός ή αρνητικός, πρέπει να τον συγκρίνουμε με το 0. Το αποτέλεσμα αυτής της σύγκρισης καθορίζει το είδος του αριθμού, αν είναι μεγαλύτερος από το 0, τότε ο αριθμός είναι θετικός, ενώ αντίθετα αν είναι μικρότερος από το 0, είναι αρνητικός.



### Λογική Έκφραση

Για τη σύνταξη μιας λογικής έκφρασης ή συνθήκης χρησιμοποιούνται σταθερές, μεταβλητές, αριθμητικές παραστάσεις, συγκριτικοί και λογικοί τελεστές, καθώς και παρενθέσεις. Στις λογικές εκφράσεις γίνεται σύγκριση της τιμής μίας έκφρασης, που βρίσκεται αριστερά από το συγκριτικό τελεστή με την τιμή μιας άλλης έκφρασης που βρίσκεται δεξιά. Το αποτέλεσμα είναι μία λογική τιμή **ΑΛΗΘΗΣ** ή **ΨΕΥΔΗΣ**.

Οι χρησιμοποιούμενοι συγκριτικοί τελεστές παρουσιάζονται στον επόμενο πίνακα.

Συγκριτικοί τελεστές		
Τελεστής	Ελεγχόμενη σχέση	Παράδειγμα
=	Ισότητα	Αριθμός=0
<>	Ανισότητα	Όνομα1 <> 'Κώστας'
>	Μεγαλύτερο από	Τιμή>10000
>=	Μεγαλύτερο ή ίσο	$X+Y \geq (A+B)/\Gamma$
<	Μικρότερο από	$B^2-4*A*\Gamma < 0$
<=	Μικρότερο ή ίσο	Βάρος <= 500



Όταν αριθμητικοί και συγκριτικοί τελεστές συνδυάζονται σε μια έκφραση, οι αριθμητικές πράξεις εκτελούνται πρώτες.

Οι συγκρίσεις γίνονται σε δεδομένα αριθμητικά, αλφαριθμητικά και λογικά.

Η σύγκριση μεταξύ δύο αριθμών γίνεται με προφανή τρόπο. Στην περίπτωση των πραγματικών αριθμών θεωρούμε ότι οι αριθμοί μπορούν να έχουν άπειρο αριθμό ψηφίων.

Η σύγκριση ατομικών χαρακτήρων στηρίζεται στην αλφαβητική σειρά, για παράδειγμα το 'α' θεωρείται μικρότερο από το 'β'.

Η σύγκριση αλφαριθμητικών δεδομένων βασίζεται στη σύγκριση χαρακτήρα προς χαρακτήρα σε κάθε θέση μέχρις ότου βρεθεί κάποια διαφορά, για παράδειγμα η λέξη 'κακός' θεωρείται μικρότερη από τη λέξη 'καλός' αφού το γράμμα κ προηγείται του γράμματος λ.

Η σύγκριση λογικών έχει έννοια μόνο στην περίπτωση του ίσου (=) και του διάφορου (<>), αφού οι τιμές που μπορούν να έχουν είναι ΑΛΗΘΗΣ και ΨΕΥΔΗΣ.

### Σύνθετες Εκφράσεις

Σε πολλά προβλήματα οι επιλογές δεν αρκεί να γίνονται με απλές λογικές παραστάσεις όπως αυτές οι οποίες αναφέρθηκαν, αλλά χρειάζεται να συνδυαστούν μία ή περισσότερες λογικές παραστάσεις. Αυτό επιτυγχάνεται με τη χρήση των τριών βασικών λογικών τελεστών ΟΧΙ, ΚΑΙ, 'Η.

#### Παραδείγματα

$0 < X < 5$                        $X > 0$     ΚΑΙ     $X < 5$   
 $X = 1$  ή 2 ή 3               $X = 1$     'Η     $X = 2$     'Η     $X = 3$

Η ιεραρχία των λογικών τελεστών είναι μικρότερη των αριθμητικών.

#### 8.1.1 Εντολή ΑΝ

Η δομή επιλογής υλοποιείται στη ΓΛΩΣΣΑ με την εντολή **ΑΝ**. Η εντολή **ΑΝ** εμφανίζεται με τρεις διαφορετικές μορφές. Την απλή εντολή **ΑΝ...ΤΟΤΕ**, την εντολή **ΑΝ...ΤΟΤΕ...ΑΛΛΙΩΣ** και τέλος την εντολή **ΑΝ...ΤΟΤΕ...ΑΛΛΙΩΣ ΑΝ**. Κάθε εντολή **ΑΝ** πρέπει να κλείνει με **ΤΕΛΟΣ\_ΑΝ**.

Στην απλούστερη μορφή της η εντολή **ΑΝ** ελέγχει τη συνθήκη και αν αυτή ισχύει (είναι αληθής), τότε εκτελούνται οι εντολές που περιλαμβάνονται μεταξύ των λέξεων **ΤΟΤΕ** και **ΤΕΛΟΣ\_ΑΝ**.

Αν για παράδειγμα θέλουμε να υπολογίσουμε τη τετραγωνική ρίζα των αριθμών που διαβάζουμε από το πληκτρολόγιο, τότε το αντίστοιχο τμήμα προγράμματος είναι

```
ΔΙΑΒΑΣΕ α
ΑΝ α >=0 ΤΟΤΕ
    Ρίζα ← T_P(α)
ΤΕΛΟΣ_ΑΝ
```

Η γενική μορφή της εντολής ΑΝ είναι η εξής:

### Σύνταξη

```
ΑΝ συνθήκη ΤΟΤΕ
    εντολή-1
    εντολή-2
    ...
    εντολή-ν
ΤΕΛΟΣ_ΑΝ
```

### Παράδειγμα

```
ΑΝ αριθμός > 0 ΤΟΤΕ
    ΓΡΑΨΕ 'Ο αριθμός είναι θετικός'
    Πλήθος_θετικών <- Πλήθος_θετικών+1
ΤΕΛΟΣ_ΑΝ
```

### Λειτουργία

Αν η συνθήκη ισχύει, τότε εκτελούνται οι εντολές που βρίσκονται μεταξύ των λέξεων **ΤΟΤΕ** και **ΤΕΛΟΣ\_ΑΝ**, σε αντίθετη περίπτωση οι εντολές αυτές αγνοούνται. Η εκτέλεση του προγράμματος συνεχίζεται με την εντολή που ακολουθεί τη δήλωση **ΤΕΛΟΣ\_ΑΝ**.

Συχνά η εντολή **ΑΝ** εκτός από το τμήμα των εντολών, που εκτελούνται όταν η λογική έκφραση είναι Αληθής, περιέχει και το τμήμα των εντολών που εκτελούνται, αν δεν ισχύει η συνθήκη (είναι Ψευδής).

Η μορφή αυτής της εντολής ονομάζεται **ΑΝ...ΤΟΤΕ...ΑΛΛΙΩΣ**.

Στο παράδειγμα του υπολογισμού της τετραγωνικής ρίζας έχουμε

```

ΔΙΑΒΑΣΕ α
ΑΝ α >=0 ΤΟΤΕ
    Ρίζα <- T_P(α)
ΑΛΛΙΩΣ
    ΓΡΑΨΕ ` Η τετρ. ρίζα αρνητικού αριθμού δεν ορίζεται´
ΤΕΛΟΣ_ΑΝ

```

Η γενική μορφή της εντολής **ΑΝ...ΤΟΤΕ...ΑΛΛΙΩΣ** έχει ως εξής:

### Σύνταξη

```

ΑΝ συνθήκη ΤΟΤΕ
    εντολή-1
    εντολή-2
    ...
    εντολή-ν
ΑΛΛΙΩΣ
    εντολή-1
    εντολή-2
    ...
    εντολή-ν
ΤΕΛΟΣ_ΑΝ

```

### Παράδειγμα

```

ΑΝ αριθμός > 0 ΤΟΤΕ
    ΓΡΑΨΕ `Ο αριθμός είναι θετικός´
    Πλήθος_θετικών <- Πλήθος_θετικών+1
ΑΛΛΙΩΣ
    ΓΡΑΨΕ `Ο αριθμός είναι αρνητικός ή 0´
    Πλήθος_μη_θετικών <- Πλήθος_μη_θετικών +1
ΤΕΛΟΣ_ΑΝ

```

### Λειτουργία

Αν η συνθήκη ισχύει, τότε εκτελούνται οι εντολές που βρίσκονται μεταξύ των λέξεων **ΤΟΤΕ** και **ΑΛΛΙΩΣ**, διαφορετικά εκτελούνται οι εντολές μεταξύ **ΑΛΛΙΩΣ** και **ΤΕΛΟΣ\_ΑΝ**. Η εκτέλεση του προγράμματος συνεχίζεται με την εντολή που ακολουθεί τη δήλωση **ΤΕΛΟΣ\_ΑΝ**.

Η γενική μορφή της εντολής **AN** καλύπτει την επιλογή μιας από δύο εναλλακτικές περιπτώσεις.

Όταν οι εναλλακτικές περιπτώσεις είναι περισσότερες από τις δύο, τότε μπορούν να χρησιμοποιηθούν πολλές εντολές **AN** ή μία μέσα στην άλλη, οι εμφωλευμένες εντολές **AN**, όπως ονομάζονται.

Εμφωλευμένα **AN** ονομάζονται δύο ή περισσότερες εντολές της μορφής **AN...ΤΟΤΕ...ΑΛΛΙΩΣ** που περιέχονται η μία μέσα στην άλλη.



Για παράδειγμα οι παρακάτω εντολές προγράμματος

```

////////
ΔΙΑΒΑΣΕ Βάρος, Ύψος
AN Βάρος < 80 ΤΟΤΕ
  AN Ύψος < 1.70 ΤΟΤΕ
    ΓΡΑΨΕ `Ελαφρύς, κοντός´
  ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
////////

```

Η χρήση εμφωλευμένων εντολών **AN** οδηγεί συνήθως σε πολύπλοκες δομές που αυξάνουν την πιθανότητα του λάθους καθώς και τη δυσκολία κατανόησης του προγράμματος.

Πολύ συχνά οι εντολές που έχουν γραφεί με εμφωλευμένα **AN**, μπορούν να γραφούν πιο απλά χρησιμοποιώντας σύνθετες εκφράσεις ή την εντολή επιλογής **AN\_ΑΛΛΙΩΣ\_ΑΝ**, που θα παρουσιαστεί στη συνέχεια.

Το προηγούμενο τμήμα προγράμματος μπορεί να γραφεί ως εξής

```

////////
ΔΙΑΒΑΣΕ Βάρος, Ύψος
AN Βάρος < 80 ΚΑΙ Ύψος < 1.70 ΤΟΤΕ
  ΓΡΑΨΕ `Ελαφρύς, κοντός´
ΤΕΛΟΣ_ΑΝ
////////

```

Μία άλλη μορφή επιλογής είναι η εντολή **AN...ΤΟΤΕ...ΑΛΛΙΩΣ\_ΑΝ**

**Σύνταξη**

```

ΑΝ συνθήκη-1 ΤΟΤΕ
    εντολή-1
    εντολή-2
    ...
    εντολή-ν
ΑΛΛΙΩΣ_ΑΝ συνθήκη-2 ΤΟΤΕ
    εντολή-1
    εντολή-2
    ...
    εντολή-ν
...
ΑΛΛΙΩΣ
    εντολή-1
    εντολή-2
    ...
    εντολή-ν
ΤΕΛΟΣ_ΑΝ

```

**Παράδειγμα**

```

ΑΝ αριθμός > 0 ΤΟΤΕ
    ΓΡΑΨΕ `Ο αριθμός είναι θετικός`
    Πλήθος_θετικών <- Πλήθος_θετικών+1
ΑΛΛΙΩΣ_ΑΝ αριθμός <0 ΤΟΤΕ
    ΓΡΑΨΕ `Ο αριθμός είναι αρνητικός `
    Πλήθος_αρνητικών <- Πλήθος_ αρνητικών +1
ΑΛΛΙΩΣ
    ΓΡΑΨΕ `Ο αριθμός είναι 0`
    Πλήθος_0 <- Πλήθος_0 +1
ΤΕΛΟΣ_ΑΝ

```

**Λειτουργία**

Εκτελούνται οι εντολές που βρίσκονται στο αντίστοιχο τμήμα, όταν η συνθήκη είναι αληθής.

Η εκτέλεση του προγράμματος συνεχίζεται με την εντολή που ακολουθεί τη δήλωση **ΤΕΛΟΣ\_ΑΝ**



## Παράδειγμα 1

Στο πρόγραμμα του προηγούμενου κεφαλαίου (πωλήσεις υπολογιστών) υποθέτουμε ότι η τιμή των υπολογιστών εξαρτάται από την ποσότητα παραγγελίας. Συγκεκριμένα ισχύουν οι παρακάτω τιμές αγοράς υπολογιστών.

ΠΟΣΟΤΗΤΑ	ΤΙΜΗ ΜΟΝΑΔΑΣ
1-50	200,000
51-100	180,000
101-200	160,000
πάνω από 200	150,000

Ο υπολογισμός με χρήση εμφωλευμένων εντολών **ΑΝ** είναι:

```

ΑΝ Ποσότητα=<50 ΤΟΤΕ
    Κόστος <- Ποσότητα*200000
ΑΛΛΙΩΣ
    ΑΝ Ποσότητα =< 100 ΤΟΤΕ
        Κόστος <- Ποσότητα*180000
    ΑΛΛΙΩΣ
        ΑΝ Ποσότητα =< 200 ΤΟΤΕ
            Κόστος <- Ποσότητα*160000
        ΑΛΛΙΩΣ
            Κόστος <- Ποσότητα*150000
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ

```

Το ίδιο πρόγραμμα με τη χρήση της εντολής **ΑΝ...ΤΟΤΕ...ΑΛΛΙΩΣ\_ΑΝ** έχει ως εξής:

```

ΑΝ Ποσότητα=<50 ΤΟΤΕ
    Κόστος <- Ποσότητα*200000
ΑΛΛΙΩΣ_ΑΝ Ποσότητα =<100 ΤΟΤΕ
    Κόστος <- Ποσότητα*180000
ΑΛΛΙΩΣ_ΑΝ Ποσότητα =<200 ΤΟΤΕ
    Κόστος <- Ποσότητα*160000
ΑΛΛΙΩΣ
    Κόστος <- Ποσότητα*150000
ΤΕΛΟΣ_ΑΝ

```



Να αποφεύγεται, αν είναι δυνατόν, η χρήση των εμφωλευμένων **ΑΝ**, και στη θέση τους να χρησιμοποιούνται απλούστερες δομές που διευκολύνουν την ανάγνωση και την κατανόηση του προγράμματος.

Ένα συχνό λάθος που παρατηρείται στα προγράμματα είναι ο έλεγχος περιπτώσεων συνθηκών. Οι επιπλέον έλεγχοι αυξάνουν την πολυπλοκότητα του προγράμματος.

Στο προηγούμενο παράδειγμα για το οποίο θεωρούμε ότι η ποσότητα είναι θετικός αριθμός, ένα παράδειγμα περιπτώσεων ελέγχων είναι το ακόλουθο:



Στην πρώτη εντολή ΑΛΛΙΩΣ\_ΑΝ ο έλεγχος της συνθήκης Ποσότητα > 50 είναι περιττός

```

ΑΝ Ποσότητα<=50 ΤΟΤΕ
    Κόστος <- Ποσότητα*200000
ΑΛΛΙΩΣ_ΑΝ Ποσότητα>50 ΚΑΙ Ποσότητα <=100 ΤΟΤΕ
    Κόστος <- Ποσότητα*180000
ΑΛΛΙΩΣ_ΑΝ Ποσότητα>100 ΚΑΙ Ποσότητα <=200 ΤΟΤΕ
    Κόστος <- Ποσότητα*160000
ΑΛΛΙΩΣ
    Κόστος <- Ποσότητα*150000
ΤΕΛΟΣ_ΑΝ
  
```

### 8.1.2 Εντολή ΕΠΙΛΕΞΕ

Αν οι εναλλακτικές περιπτώσεις επιλογής είναι πολλές, μπορεί να χρησιμοποιηθεί η εντολή ΕΠΙΛΕΞΕ, η γενική μορφή της οποίας είναι:

#### Σύνταξη

```

ΕΠΙΛΕΞΕ έκφραση
    ΠΕΡΙΠΤΩΣΗ λίστα_τιμών_1
        εντολές_1
    ΠΕΡΙΠΤΩΣΗ λίστα_τιμών_2
        εντολές_2
    .....
    ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ
        εντολές_αλλιώς
ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ
  
```

#### Παράδειγμα

```

ΔΙΑΒΑΣΕ αριθμός
ΕΠΙΛΕΞΕ αριθμός
    ΠΕΡΙΠΤΩΣΗ 0
        ΓΡΑΨΕ `Μηδέν`
  
```

```

ΠΕΡΙΠΤΩΣΗ 1,3,5,7,9
ΓΡΑΨΕ `Μονός αριθμός`
ΠΕΡΙΠΤΩΣΗ 2,4,6,8
ΓΡΑΨΕ `Ζυγός`
ΠΕΡΙΠΤΩΣΗ ΑΛΛΙΩΣ
ΓΡΑΨΕ `Αριθμός < 0 ή >9 ή όχι ακέραιος`
ΤΕΛΟΣ_ΕΠΙΛΟΓΩΝ

```

### Λειτουργία

Υπολογίζεται η τιμή της έκφρασης και εκτελούνται οι εντολές που ανήκουν στην αντίστοιχη περίπτωση τιμών. Αν η τιμή της έκφρασης δεν αντιστοιχεί σε καμία περίπτωση, τότε εκτελούνται οι εντολές αλλιώς.

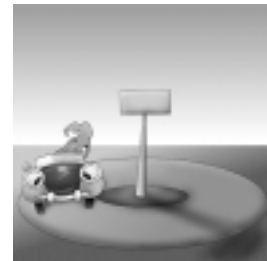
Στην εντολή αυτή οι λίστες τιμών που συνοδεύουν κάθε περίπτωση μπορούν να περιλαμβάνουν μία ή περισσότερες τιμές ή περιοχή τιμών από-έως.

Η χρήση της εντολής **ΕΠΙΛΕΞΕ** λόγω της συμπαγούς δομής της προσφέρει σημαντικά πλεονεκτήματα στον προγραμματισμό.

## 8.2 Εντολές επανάληψης

Η τρίτη βασική δομή είναι η δομή επανάληψης, ο βρόχος, η οποία επιτρέπει την εκτέλεση εντολών περισσότερες από μία φορές. Οι επαναλήψεις ελέγχονται πάντοτε από κάποια συνθήκη, η οποία καθορίζει την έξοδο από το βρόχο.

Η **ΓΛΩΣΣΑ** υποστηρίζει τρεις εντολές επανάληψης, την εντολή **ΟΣΟ** όπου η επανάληψη ελέγχεται από μία λογική έκφραση στην αρχή και εκτελείται συνεχώς όσο η συνθήκη είναι Αληθής, την εντολή **ΜΕΧΡΙΣ\_ΟΤΟΥ** όπου η συνθήκη βρίσκεται στο τέλος του βρόχου και εκτελείται συνεχώς μέχρις ότου η συνθήκη αυτή γίνει Αληθής και τέλος την εντολή **ΓΙΑ**, με την οποία ο βρόχος επαναλαμβάνεται για προκαθορισμένο αριθμό φορών.



### 8.2.1 Εντολή ΟΣΟ...ΕΠΑΝΑΛΑΒΕ

Η γενικότερη δομή επανάληψης υλοποιείται στη **ΓΛΩΣΣΑ** με την εντολή **ΟΣΟ... ΕΠΑΝΑΛΑΒΕ**. Σε αυτή, η συνθήκη που ελέγχει την επανάληψη βρίσκεται στην αρχή της επανάληψης και ο βρόχος επαναλαμβάνεται συνεχώς, όσο η συνθήκη αυτή ισχύει. Με τη δομή αυτή μπορούν να εκφραστούν

όλες οι επαναλήψεις και γι αυτό η εντολή **ΟΣΟ... ΕΠΑΝΑΛΑΒΕ** είναι η σημαντικότερη από όλες τις εντολές επανάληψης. Χαρακτηριστικό της επανάληψης αυτής είναι ότι ο αριθμός των επαναλήψεων δεν είναι γνωστός, ούτε μπορεί να υπολογιστεί πριν από την εκτέλεση του προγράμματος.

### Σύνταξη

```

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

```

### Παράδειγμα

```

Αθροισμα<-0
ΟΣΟ Αθροισμα<1000 ΕΠΑΝΑΛΑΒΕ
    ΔΙΑΒΑΣΕ Α
    Αθροισμα<- Αθροισμα+Α
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

### Λειτουργία

Ελέγχεται η συνθήκη και αν είναι Αληθής, εκτελούνται οι εντολές που βρίσκονται ανάμεσα στις **ΟΣΟ\_ΕΠΑΝΑΛΑΒΕ** και **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**. Στη συνέχεια ελέγχεται πάλι η συνθήκη και αν ισχύει, εκτελούνται πάλι οι ίδιες εντολές. Όταν η λογική έκφραση γίνει Ψευδής, τότε σταματάει η επανάληψη και εκτελείται η εντολή μετά το **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**.

Εφόσον μετά από κάθε επανάληψη ελέγχεται εκ νέου η συνθήκη, πρέπει υποχρεωτικά μέσα στο βρόχο να υπάρχει μία εντολή, η οποία να μεταβάλλει την τιμή της μεταβλητής που ελέγχεται με τη συνθήκη. Σε αντίθετη περίπτωση η επανάληψη δε θα τερματίζεται και θα εκτελείται συνεχώς.

### Παράδειγμα 2

Να γραφεί πρόγραμμα το οποίο διαβάζει από το πληκτρολόγιο μία σειρά μετρήσεων, ακεραίων μη μηδενικών αριθμών, υπολογίζει και τυπώνει το άθροισμα τους καθώς και το μέσο τους όρο. Ως τέλος της διαδικασίας εισαγωγής στοιχείων χρησιμοποιείται η τιμή 0.

```
ΠΡΟΓΡΑΜΜΑ Άθροισμα
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ:Χ, Άθροισμα, Πλήθος
    ΠΡΑΓΜΑΤΙΚΕΣ:ΜΟ
ΑΡΧΗ
    Πλήθος <- 0
    Άθροισμα <- 0
    ΓΡΑΨΕ `Δώσε Αριθμό´
    ΔΙΑΒΑΣΕ Χ
    ΟΣΟ Χ<>0 ΕΠΑΝΑΛΑΒΕ
        Άθροισμα <- Άθροισμα+Χ
        Πλήθος <- Πλήθος+1
        ΓΡΑΨΕ `Δώσε Αριθμό´
        ΔΙΑΒΑΣΕ Χ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΝ Πλήθος >0 ΤΟΤΕ
    ΜΟ <- Άθροισμα/Πλήθος
    ΓΡΑΨΕ `Το Άθροισμα είναι      : `, Άθροισμα
    ΓΡΑΨΕ `Ο Μέσος όρος είναι : `, ΜΟ
ΑΛΛΙΩΣ
    ΓΡΑΨΕ `Δεν δόθηκαν στοιχεία´
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ-ΠΡΟΓΡΑΜΜΑΤΟΣ Άθροισμα
```

### Παρατηρήσεις

Η χρήση τιμών για τον τερματισμό μίας επαναληπτικής διαδικασίας, όπως στο παράδειγμα η αυθαίρετη επιλογή του 0, είναι συνήθης στον προγραμματισμό.

Η τιμή αυτή ορίζεται από τον προγραμματιστή και αποτελεί μια σύμβαση για το τέλος του προγράμματος. Η τιμή αυτή είναι τέτοια, ώστε να μην είναι λογικά σωστή για το πρόβλημα, για παράδειγμα η τιμή 0 αποκλείεται από τις μετρήσεις σύμφωνα με την εκφώνηση του παραδείγματος. Η τιμή αυτή συχνά αποκαλείται “τιμή φρουρός”.

### 8.2.2 Εντολή ΜΕΧΡΙΣ\_ΟΤΟΥ

Η δεύτερη εντολή επανάληψης που χρησιμοποιεί η ΓΛΩΣΣΑ είναι η εντολή ΜΕΧΡΙΣ\_ΟΤΟΥ. Σε αυτή οι εντολές του βρόχου εκτελούνται μέχρις ότου ικανοποιηθεί κάποια συνθήκη η οποία ελέγχεται στο τέλος της επανάληψης.

### Σύνταξη

#### ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

εντολή-1

εντολή-2

...

εντολή-ν

**ΜΕΧΡΙΣ\_ΟΤΟΥ** λογική-έκφραση

### Παράδειγμα

#### ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ

**ΔΙΑΒΑΣΕ** Α

Αθροισμα <- Αθροισμα + Α

**ΜΕΧΡΙΣ\_ΟΤΟΥ** Αθροισμα >= 1000

### Λειτουργία

Εκτελούνται οι εντολές μεταξύ των **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** και **ΜΕΧΡΙΣ\_ΟΤΟΥ**. Στη συνέχεια ελέγχεται η λογική έκφραση και αν δεν ισχύει (είναι ψευδής), τότε οι εντολές που βρίσκονται ανάμεσα στις **ΑΡΧΗ\_ΕΠΑΝΑΛΗΨΗΣ** και **ΜΕΧΡΙΣ\_ΟΤΟΥ**, εκτελούνται πάλι. Ελέγχεται ξανά η λογική έκφραση και αν δεν ισχύει, επαναλαμβάνεται η εκτέλεση των ίδιων εντολών.

Όταν η λογική έκφραση γίνει Αληθής τότε σταματάει η επανάληψη και εκτελείται η εντολή μετά από την **ΜΕΧΡΙΣ\_ΟΤΟΥ**.

Πολύ συχνά η ίδια επαναληπτική διαδικασία μπορεί να γραφεί εξίσου σωστά χρησιμοποιώντας είτε τη δομή **ΟΣΟ...ΕΠΑΝΑΛΑΒΕ** είτε τη δομή **ΜΕΧΡΙΣ\_ΟΤΟΥ** και είναι προσωπική επιλογή του προγραμματιστή ποια από τις δυο θα χρησιμοποιήσει. Υπάρχουν όμως περιπτώσεις όπου η χρήση της εντολής **ΜΕΧΡΙΣ\_ΟΤΟΥ** οδηγεί σε απλούστερα και πιο ευκολονόητα προγράμματα. Γενικά σε περιπτώσεις όπου η επανάληψη θα συμβεί υποχρεωτικά μία φορά, είναι προτιμότερη η χρήση της **ΜΕΧΡΙΣ\_ΟΤΟΥ**.

Χαρακτηριστική περίπτωση όπου προτιμάται η εντολή **ΜΕΧΡΙΣ\_ΟΤΟΥ** είναι στον έλεγχο αποδεκτών τιμών καθώς και στην επιλογή από προκαθορισμένες απαντήσεις ή μενού.

### Παράδειγμα 3

Στο προηγούμενο παράδειγμα ας υποθέσουμε ότι οι μετρήσεις είναι υποχρεωτικά θετικοί αριθμοί και ότι μετά την εισαγωγή κάθε αριθμού υπάρχει η ερώτηση, αν θα εισάγουμε άλλο. Η διαδικασία θα τελειώσει, όταν η απάντηση θα είναι Όχι (ο ή Ο).

```

ΠΡΟΓΡΑΜΜΑ Αθροισμα2
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: X, Αθροισμα, Πλήθος
    ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ
    ΧΑΡΑΚΤΗΡΕΣ: Επιλογή
Πλήθος <- 0
Αθροισμα <- 0
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
    ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
        ! Έλεγχος δεδομένων
        ΓΡΑΨΕ `Δώσε Αριθμό`
        ΔΙΑΒΑΣΕ X
        ΑΝ X=<0 ΤΟΤΕ
            ΓΡΑΨΕ `Λάθος Αριθμός, Παρακαλώ δώστε ξανά...`
        ΤΕΛΟΣ_ΑΝ
        ! Αν το X δεν είναι θετικό εισάγουμε νέο αριθμό
        ΜΕΧΡΙΣ_ΟΤΟΥ X>0
        Αθροισμα <- Αθροισμα+X
        Πλήθος <- Πλήθος+1
        ΓΡΑΨΕ `Νέα μέτρηση ;`
        ΔΙΑΒΑΣΕ Επιλογή
        ! Αν η επιλογή είναι Ο ή ο τότε σταματάει η επανάληψη
        ΜΕΧΡΙΣ_ΟΤΟΥ Επιλογή='Ο' Η Επιλογή='ο'
    ΜΟ <- Αθροισμα/Πλήθος
    ΓΡΑΨΕ `Αθροισμα =`, Αθροισμα
    ΓΡΑΨΕ `Μέσος όρος =`, ΜΟ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```



Η εντολή επανάληψης **ΜΕΧΡΙΣ\_ΟΤΟΥ** εκτελείται υποχρεωτικά τουλάχιστον μία φορά

### Προγραμματιστικό περιβάλλον Pascal

```

PROGRAM athroisma2;
VAR
    x,sum,count:INTEGER;
    avg:REAL;
    choice:CHAR;

```

```

BEGIN
    count:=0;sum:=0;
    REPEAT
        REPEAT
            write(`Δώσε αριθμό:');
            readln (x);
            IF x<=0 THEN
                writeln (`Λάθος αριθμός, Δώσε ξανά..');
            UNTIL x>0;
            sum:=sum+x;
            count:=count+1;
            write(`Νέα μέτρηση:');
            readln(choice);
        UNTIL (choice='ο') OR (choice='Ο');
        avg:=sum/count;
        writeln(`Αθροισμα: `, sum:5);
        writeln(`Μέσος όρος: `,avg:6:2);
    END.

```

### 8.2.3 Εντολή ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ



Η εντολή ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ χρησιμοποιείται στην περίπτωση που πρέπει να επαναληφθεί η εκτέλεση κάποιων εντολών για προκαθορισμένο αριθμό επαναλήψεων.

Πολύ συχνά ο αριθμός των επαναλήψεων που πρέπει να εκτελεστούν, είναι γνωστός από την αρχή. Αν και αυτού του είδους οι επαναλήψεις μπορούν να αντιμετωπιστούν με τη χρήση των προηγούμενων εντολών επανάληψης, η **ΓΛΩΣΣΑ** διαθέτει και την εντολή **ΓΙΑ**. Η εντολή αυτή χειρίζεται μια μεταβλητή, στην οποία αρχικά εκχωρείται η αρχική τιμή. Η τιμή της μεταβλητής συγκρίνεται με την τελική τιμή και εφόσον είναι μικρότερη από αυτή, τότε εκτελούνται οι εντολές που βρίσκονται στο βρόχο (ανάμεσα στις εντολές **ΓΙΑ** και **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**). Στη συνέχεια η μεταβλητή ελέγχου αυξάνεται κατά την τιμή που ορίζει το **ΒΗΜΑ**. Αν η νέα τιμή είναι μικρότερη της τελικής, τότε ο βρόχος εκτελείται ξανά. Η διαδικασία αυτή επαναλαμβάνεται συνεχώς, έως ότου η τιμή ελέγχου γίνει μεγαλύτερη της τελικής τιμής, οπότε η τερματίζεται η επανάληψη και το πρόγραμμα συνεχίζει με την εντολή που ακολουθεί το **ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**.

Ας σημειωθεί ότι, αν η τιμή του βήματος είναι 1, τότε μπορεί να παραληφθεί.



**Σύνταξη**

```

ΓΙΑ μεταβλητή ΑΠΟ τιμή1 ΜΕΧΡΙ τιμή2 ΜΕ ΒΗΜΑ τιμή3
    εντολή-1
    εντολή-2
    ...
    εντολή-ν
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

**Παράδειγμα**

```

ΓΙΑ Αριθμό ΑΠΟ 1 ΜΕΧΡΙ 100 ΜΕ ΒΗΜΑ 2
    Άθροισμα <- Άθροισμα+Αριθμό
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ

```

**ΛΕΙΤΟΥΡΓΙΑ**

Οι εντολές του βρόχου εκτελούνται για όλες τις τιμές της μεταβλητής από την αρχική τιμή μέχρι την τελική τιμή, αυξανόμενες με την τιμή του βήματος. Αν το βήμα είναι ίσο με 1, τότε παραλείπεται.

**Παράδειγμα 4**

Το παρακάτω πρόγραμμα υπολογίζει το άθροισμα των περιττών αριθμών που είναι μικρότεροι από το 100.

```

ΠΡΟΓΡΑΜΜΑ Περιττοί
ΜΕΤΑΒΛΗΤΕΣ
    ΑΚΕΡΑΙΕΣ: Άθροισμα, Αριθμός
ΑΡΧΗ
    Άθροισμα <- 0
    ΓΙΑ Αριθμός ΑΠΟ 1 ΜΕΧΡΙ 100 ΜΕ ΒΗΜΑ 2
        Άθροισμα <- Άθροισμα + Αριθμός
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΓΡΑΨΕ `Άθροισμα περιττών αριθμών είναι: `, Άθροισμα
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```



Κάθε επανάληψη που εκτελείται με μία εντολή **ΓΙΑ..ΑΠΟ..ΜΕΧΡΙ**, μπορεί να υπολοποιηθεί και με τη χρήση των βασικών εντολών επανάληψης **ΟΣΟ..ΕΠΑΝΑΛΑΒΕ** και **ΜΕΧΡΙΣ..ΟΣΟΥ**.

Πολύ συχνά για την επίλυση των προβλημάτων απαιτείται η χρήση εμφωλευμένων βρόχων. Σε αυτή την περίπτωση ο ένας βρόχος βρίσκεται μέσα στον άλλο.

Στη χρήση των εμφωλευμένων βρόχων ισχύουν συγκεκριμένοι κανόνες που πρέπει να ακολουθούνται αυστηρά για την σωστή λειτουργία των προγραμμάτων.

Συγκεκριμένα:

- ⇒ Ο εσωτερικός βρόχος πρέπει να βρίσκεται ολόκληρος μέσα στον εξωτερικό. Ο βρόχος που ξεκινάει τελευταίος, πρέπει να ολοκληρώνεται πρώτος.
- ⇒ Η είσοδος σε κάθε βρόχο υποχρεωτικά γίνεται από την αρχή του.
- ⇒ Δεν μπορεί να χρησιμοποιηθεί η ίδια μεταβλητή ως μετρητής δύο ή περισσότερων βρόχων που ο ένας βρίσκεται στο εσωτερικό του άλλου.

### Παράδειγμα 5

**Να γραφεί πρόγραμμα το οποίο να εκτυπώνει τη προπαίδεια του πολλαπλασιασμού.**

**ΠΡΟΓΡΑΜΜΑ** Προπαίδεια

!Πρόγραμμα εκτύπωσης της προπαίδειας των αριθμών 1 έως 10

**ΜΕΤΑΒΛΗΤΕΣ**

**ΑΚΕΡΑΙΕΣ** : A, B, Γ

!A: Πολλαπλασιαστής

!B: Πολλαπλασιαστής

!Γ: Γινόμενο

**ΑΡΧΗ**

**ΓΙΑ** A **ΑΠΟ** 1 **ΜΕΧΡΙ** 10

**ΓΙΑ** B **ΑΠΟ** 1 **ΜΕΧΡΙ** 10

Γ ← A\*B

**ΓΡΑΨΕ** A, 'X', B, '=', Γ

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΓΡΑΨΕ** !Εισαγωγή κενής γραμμής στην εκτύπωση

**ΤΕΛΟΣ\_ΕΠΑΝΑΛΗΨΗΣ**

**ΤΕΛΟΣ\_ΠΡΟΓΡΑΜΜΑΤΟΣ**

### Προγραμματιστικό περιβάλλον Basic

` Προπαίδεια

DEFINT A-Z

FOR a = 1 TO 10

FOR b = 1 TO 10

c = a \* b

PRINT a; "x"; b; "="; c

NEXT b

PRINT  
NEXT a  
END

## Ανακεφαλαίωση

Στο κεφάλαιο αυτό παρουσιάστηκαν οι εντολές που χρησιμοποιεί η ΓΛΩΣΣΑ για να υλοποιήσει τις βασικές δομές της επιλογής και της επανάληψης.

Αρχικά παρουσιάζονται οι λογικές εκφράσεις καθώς και ο τρόπος που διατυπώνονται σύνθετες λογικές εκφράσεις με τη χρήση των λογικών τελεστών ΟΧΙ, Ή, ΚΑΙ. Η εντολή ΑΝ...ΤΟΤΕ υλοποιεί τη δομή της επιλογής.

Η εντολή αυτή εμφανίζεται με πιο σύνθετες μορφές: την εντολή ΑΝ...ΤΟΤΕ...ΑΛΛΙΩΣ καθώς και την εντολή ΑΝ...ΤΟΤΕ... ΑΛΛΙΩΣ\_ΑΝ. Μία άλλη εντολή επιλογής που υπάρχει είναι η εντολή ΕΠΙΛΕΞΕ.

Οι εντολές επανάληψης είναι τρεις. Η εντολή ΟΣΟ...ΕΠΑΝΑΛΑΒΕ, η εντολή ΜΕΧΡΙΣ\_ΟΤΟΥ και τέλος η εντολή ΓΙΑ. Η εντολή ΓΙΑ χρησιμοποιείται για καθορισμένο αριθμό επαναλήψεων, ενώ ο αριθμός επαναλήψεων των άλλων δύο δεν είναι γνωστός εκ των προτέρων και εξαρτάται από τις συνθήκες που τις ελέγχουν. Η εντολή ΟΣΟ...ΕΠΑΝΑΛΑΒΕ ελέγχει τη συνθήκη στην αρχή της επανάληψης, ενώ η εντολή ΜΕΧΡΙΣ\_ΟΤΟΥ κάνει τον έλεγχο στο τέλος της επανάληψης.



## Λέξεις κλειδιά

Λογική έκφραση, Επιλογή, Επανάληψη, Βρόχος



## Ερωτήσεις - Θέματα για συζήτηση

1. Ποιες είναι οι τιμές που μπορεί να πάρει μία λογική έκφραση;
2. Ποιοι είναι οι βασικοί λογικοί τελεστές; Αναφέρατε δύο παραδείγματα για τη χρήση του καθενός;
3. Ποια είναι η σύνταξη της εντολής ΑΝ;
4. Ποια είναι η διαφορά της εντολής ΑΝ- ΑΛΛΙΩΣ και της ΑΝ- ΑΛΛΙΩΣ\_ΑΝ;
5. Τι είναι τα εμφωλευμένα ΑΝ;
6. Πότε χρησιμοποιείται η εντολή ΕΠΙΛΕΞΕ;



7. Ποιες οι εντολές επανάληψης;
8. Ποιες οι διαφορές της εντολής ΟΣΟ και της εντολής ΜΕΧΡΙΣ\_ΟΤΟΥ;
9. Πώς συντάσσεται η εντολή ΓΙΑ;
10. Ποια η βασική διαφορά της εντολής ΓΙΑ από τις άλλες δύο εντολές επανάληψης;

### Βιβλιογραφία

1. Ι. Κάβουρας, *Δομημένος προγραμματισμός με Pascal*, Κλειδάριθμος, Αθήνα, 1997.
2. Κ. Γιαλούρης-Κ. Σταθόπουλος, *Προγραμματισμός σε Turbo Pascal*, Αθήνα, 1996.
3. Χρ. Κοίλιας, *Η QuickBasic και οι εφαρμογές της*, Εκδόσεις Νέων Τεχνολογιών, Αθήνα, 1992.
4. R. Shackelford, *Introduction to Computing and Algorithms*, Addison-Wesley, USA, 1998.
5. S. Leestma-L.Nyhoff, *Turbo Pascal, Programming and Solving*, McMillan, New York, 1990.
6. N. Wirth, *Systematic Programming: An introduction*, Prentice Hall, 1973.

### Διευθύνσεις Διαδικτύου

- ⇒ <http://www.swcp.com/~dodrill/>
- ⇒ <http://www.progsources.com>
- ⇒ [www.cit.ac.nz/smac/pascal/default.htm](http://www.cit.ac.nz/smac/pascal/default.htm)
- ⇒ <http://www.cs.vu.nl/~jprins/tp.html>
- ⇒ <http://qbasic.com/>
- ⇒ [www.basicguru.com](http://www.basicguru.com)