

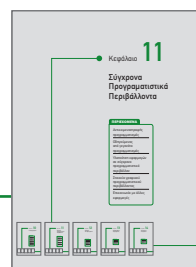
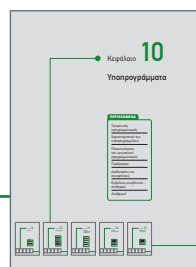
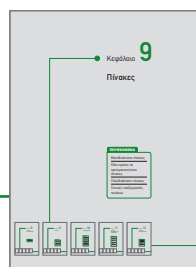
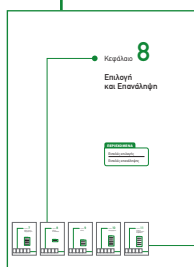
Κεφάλαιο 8

Επιλογή και Επανάληψη

ΠΕΡΙΕΧΟΜΕΝΑ

Εντολές επιλογής

Εντολές επανάληψης



8.1 | Εντολές 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 μιας 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.

ΣΥΓΚΡΙΤΙΚΟΙ ΤΕΛΕΣΤΕΣ		
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

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.



Όταν αριθμητικοί και συγκριτικοί τελεστές συνδυάζονται σε μια έκφραση, οι αριθμητικές πράξεις εκτελούνται πρώτες. Ακόμη, οι λογικοί τελεστές έχουν χαμηλότερη ιεραρχία από τους συγκριτικούς.

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 **ΑΝ** καλύπτει την επιλογή μιας από δύο εναλλακτικές περιπτώσεις.

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

ΟΡΙΣΜΟΣ

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 ... ΤΟΤΕ ... 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

```
AN 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**

ΠΑΡΑΔΕΙΓΜΑ 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	580
51-100	520
101-200	470
πάνω από 200	440

Ο υπολογισμός με χρήση εμφωλευμένων εντολών ΑΝ είναι:

```
ΑΝ Ποσότητα<=50 ΤΟΤΕ
    Κόστος <- Ποσότητα*580
ΑΛΛΙΩΣ
    ΑΝ Ποσότητα<=100 ΤΟΤΕ
        Κόστος <- Ποσότητα*520
    ΑΛΛΙΩΣ
        ΑΝ Ποσότητα<=200 ΤΟΤΕ
            Κόστος <- Ποσότητα*470
        ΑΛΛΙΩΣ
            Κόστος <- Ποσότητα*440
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
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.



Στην πρώτη εντολή ΑΛΛΙΩΣ_ΑΝ ο έλεγχος της συνθήκης Ποσότητα>50 είναι περιττός.

Το ίδιο πρόγραμμα με τη χρήση της εντολής ΑΝ...ΤΟΤΕ...ΑΛΛΙΩΣ_ΑΝ έχει ως εξής:

```
ΑΝ Ποσότητα<=50 ΤΟΤΕ
    Κόστος <- Ποσότητα*580
ΑΛΛΙΩΣ_ΑΝ Ποσότητα<=100 ΤΟΤΕ
    Κόστος <- Ποσότητα*520
ΑΛΛΙΩΣ_ΑΝ Ποσότητα<=200 ΤΟΤΕ
    Κόστος <- Ποσότητα*470
ΑΛΛΙΩΣ
    Κόστος <- Ποσότητα*440
ΤΕΛΟΣ_ΑΝ
```

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

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

```
ΑΝ Ποσότητα<=50 ΤΟΤΕ
    Κόστος <- Ποσότητα*580
ΑΛΛΙΩΣ_ΑΝ Ποσότητα>50 ΚΑΙ Ποσότητα <=100 ΤΟΤΕ
    Κόστος <- Ποσότητα*520
ΑΛΛΙΩΣ_ΑΝ Ποσότητα>100 ΚΑΙ Ποσότητα <=200 ΤΟΤΕ
    Κόστος <- Ποσότητα*470
ΑΛΛΙΩΣ
    Κόστος <- Ποσότητα*440
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 φορά. 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.

ΠΑΡΑΔΕΙΓΜΑ 2

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

```
ΠΡΟΓΡΑΜΜΑ Αθροισμα  
ΜΕΤΑΒΛΗΤΕΣ  
    ΑΚΕΡΑΙΕΣ:Χ, Αθροισμα, Πλήθος  
    ΠΡΑΓΜΑΤΙΚΕΣ:ΜΟ  
ΑΡΧΗ  
    Πλήθος < -0  
    Αθροισμα <- 0  
    ΓΡΑΨΕ 'Δώσε Αριθμό'  
    ΔΙΑΒΑΣΕ Χ  
    ΟΣΟ Χ<>0 ΕΠΑΝΑΛΑΒΕ  
        Αθροισμα <- Αθροισμα+Χ  
        Πλήθος <- Πλήθος+1  
        ΓΡΑΨΕ 'Δώσε Αριθμό'  
        ΔΙΑΒΑΣΕ Χ  
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```

ΑΝ Πλήθος > 0 ΤΟΤΕ
    ΜΟ <- Αθροισμα/Πλήθος
    ΓΡΑΨΕ 'Το Αθροισμα είναι : ', Αθροισμα
    ΓΡΑΨΕ 'Ο Μέσος όρος είναι : ', ΜΟ
ΑΛΛΙΩΣ
    ΓΡΑΨΕ 'Δεν δόθηκαν στοιχεία'
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΠΡΟΓΡΑΜΜΑΤΟΣ

```

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.

ΠΑΡΑΔΕΙΓΜΑ 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 εισαγάγουμε άλλο. Η διαδικασία θα τελειώσει, όταν η απάντηση θα είναι Όχι (ο ή Ο).

ΠΡΟΓΡΑΜΜΑ Άθροισμα2

ΜΕΤΑΒΛΗΤΕΣ

ΑΚΕΡΑΙΕΣ: X, Άθροισμα, Πλήθος

ΠΡΑΓΜΑΤΙΚΕΣ: ΜΟ

ΧΑΡΑΚΤΗΡΕΣ: Επιλογή

ΑΡΧΗ

Πλήθος <- 0

Άθροισμα <- 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

Άθροισμα <- Άθροισμα+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 <- Άθροισμα/Πλήθος

ΓΡΑΨΕ 'Άθροισμα =', Άθροισμα

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

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 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 παραλειφθεί.



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

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
```

Λειτουργία

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.

ΠΑΡΑΔΕΙΓΜΑ 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

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 τη 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.



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

1. Τι χρησιμοποιούμε για την σύνταξη μιας λογικής έκφρασης;

Οι λογικές εκφράσεις είναι συνδυασμοί τελεστών και τελεστών κατά τέτοιο τρόπο ώστε όταν γίνουν οι υπολογισμοί να προκύψει λογική τιμή (ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ).

Για να σχηματίσουμε (συντάξουμε) μια λογική έκφραση μπορούμε να χρησιμοποιήσουμε κάθε είδους μεταβλητές, σταθερές, συναρτήσεις, τελεστές, παρενθέσεις.

2. (κ) Οι λογικές εκφράσεις είναι παραστάσεις που υπολογίζονται ή είναι έλεγχοι;

Οι λογικές εκφράσεις είναι παραστάσεις που υπολογίζονται, όπως και οι αριθμητικές εκφράσεις. Η διαφορά τους βρίσκεται στον τύπο του αποτελέσματος. Οι λογικές εκφράσεις έχουν τιμή ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ, ενώ η τιμή των αριθμητικών εκφράσεων είναι αριθμός.

Επομένως, οι λογικές εκφράσεις μπορούν να χρησιμοποιηθούν σε μια εντολή εκχώρησης (όπως οι αριθμητικές), πχ $X \leftarrow 3 >= 5$ οπότε η μεταβλητή X θα πάρει την τιμή ΨΕΥΔΗΣ.

Μπορούν όμως να χρησιμοποιηθούν και σε ελέγχους (σαν συνθήκες σε εντολές ΑΝ, ΟΣΟ, ΜΕΧΡΙΣ_ΟΤΟΥ, ΓΙΑ).

3. Πως λειτουργούν οι συγκριτικοί τελεστές;

Η σύγκριση μεταξύ αριθμών γίνεται με τον τρόπο που γνωρίζουμε από τα μαθηματικά.

Η σύγκριση λογικών τιμών (ΑΛΗΘΗΣ, ΨΕΥΔΗΣ) γίνεται μόνο με τους τελεστές = και <>.

Η σύγκριση μεμονωμένων χαρακτήρων βασίζεται στην αλφαβητική σειρά και συγκεκριμένα 'α' < 'β' < 'γ' < ...

Η σύγκριση συμβολοσειρών (αλφαριθμητικών) βασίζεται στη σύγκριση χαρακτήρα προς χαρακτήρα σε κάθε θέση, μέχρις ότου να βρεθεί ο πρώτος διαφορετικός χαρακτήρας, και τότε εφαρμόζεται η αλφαβητική σειρά.

4. Πως δημιουργούνται οι σύνθετες λογικές εκφράσεις στην ΓΛΩΣΣΑ;

Σύνθετες εκφράσεις δημιουργούνται με τη χρήση των λογικών τελεστών ΟΧΙ, ΚΑΙ, Η.

5. Ποια είναι η προτεραιότητα (ιεραρχία) των τελεστών;

Η προτεραιότητα των αριθμητικών τελεστών είναι γνωστή από τα μαθηματικά.

Οι συγκριτικοί τελεστές έχουν όλοι την ίδια προτεραιότητα.

Οι λογικοί τελεστές έχουν την εξής προτεραιότητα: ΟΧΙ, ΚΑΙ, Η.

Όταν σε μια έκφραση υπάρχουν τελεστές από όλες τις κατηγορίες, πρώτα γίνονται οι αριθμητικοί τελεστές, στη συνέχεια οι συγκριτικοί, και στο τέλος οι λογικοί.

Ο μόνος λόγος για να αλλάξει η σειρά εκτέλεσης των πράξεων είναι η χρήση παρενθέσεων.

6. Πως συντάσσεται και πως λειτουργεί η εντολή AN (3 μορφές) κ η εντολή ΕΠΙΛΕΞΕ.

Τα είδαμε στο 2ο κεφάλαιο.

7. Τι είναι τα εμφωλευμένα AN; Αποτελούν πλεονέκτημα ή μειονέκτημα για τα προγράμματα;

Εμφωλευμένα AN είναι εντολές AN που περιέχονται η μία μέσα στην άλλη.

Η χρήση εμφωλευμένων AN οδηγεί σε πολύπλοκα προγράμματα, αυξάνει την πιθανότητα λάθους και δυσκολεύει την κατανόηση του προγράμματος. Μπορούμε να αντικαταστήσουμε τα εμφωλευμένα AN με χρήση σύνθετων συνθηκών και με χρήση δομών πολλαπλής επιλογής.

Σημείωση: ότι τα εμφωλευμένα AN οδηγούν σε σύνθετα προγράμματα είναι κατά τη γνώμη μου ηλίθια παρατήρηση, αλλά μπορεί να το βάλουν σε Σ/Λ οπότε ας συμφωνήσουμε...

8. Σχολιάστε την χρήση περιττών συνθηκών σε ένα πρόγραμμα.

Η χρήση περιττών συνθηκών, ιδίως σε δομές πολλαπλής επιλογής, είναι ένα **συχνό λάθος**. Οι επιπλέον έλεγχοι αυξάνουν την πολυπλοκότητα του προγράμματος.

9. Σχολιάστε τη χρήση της εντολής ΕΠΙΛΕΞΕ.

Η εντολή ΕΠΙΛΕΞΕ λόγω της συμπαγούς δομής της, προσφέρει πλεονεκτήματα στον προγρ/μό.

10. Οι επαναλήψεις ελέγχονται πάντα από συνθήκη που καθορίζει την έξοδο από το βρόχο;

Ναι. Στις εντολές ΟΣΟ και ΜΕΧΡΙΣ_ΟΤΟΥ η συνθήκη είναι εμφανής. Στην εντολή ΓΙΑ η συνθήκη είναι η σύγκριση του μετρητή της ΓΙΑ με την τιμή ΜΕΧΡΙ.

Η ΓΙΑ και η ΟΣΟ βγαίνουν από την επανάληψη όταν η συνθήκη πάψει η ισχύει. Η ΜΕΧΡΙΣ_ΟΤΟΥ βγαίνει από την επανάληψη όταν η συνθήκη επαληθευτεί.

11. Σύνταξη και λειτουργία των εντολών ΟΣΟ, ΜΕΧΡΙΣ_ΟΤΟΥ, ΓΙΑ.

Τα είδαμε στο 2ο κεφάλαιο.

12. Τι πρέπει να περιλαμβάνει η εντολή ΟΣΟ (και η ΜΕΧΡΙΣ_ΟΤΟΥ) ώστε να μην μετατραπεί σε ατέρμονα βρόχο;

Εφόσον κάθε φορά (στην αρχή του βρόχου για την ΟΣΟ, στο τέλος του βρόχου για τη ΜΕΧΡΙΣ_ΟΤΟΥ) ελέγχεται η συνθήκη της επανάληψης, θα πρέπει μέσα στις εντολές που επαναλαμβάνονται να υπάρχει εντολή που να μεταβάλλει (ανανεώνει) την τιμή των μεταβλητών που ελέγχονται στη συνθήκη.

13. Τι είναι η “τιμή φρουρός”;

Είναι μία τιμή που επιλέγεται από τον προγραμματιστή και η οποία χρησιμοποιείται για τον τερματισμό μιας επαναληπτικής διαδικασίας.

14. Συγκρίνετε τις εντολές ΓΙΑ, ΟΣΟ και ΜΕΧΡΙΣ_ΟΤΟΥ (από πλευράς γενικότητας και ευκολίας στην χρήση).

Οι εντολές ΟΣΟ και ΜΕΧΡΙΣ_ΟΤΟΥ είναι γενικής χρήσης καθώς με αυτές μπορούμε να υλοποιήσουμε οποιαδήποτε επανάληψη. Η εντολή ΓΙΑ έχει τον περιορισμό ότι μπορεί να χρησιμοποιηθεί μόνο όταν γνωρίζουμε εκ των προτέρων το πλήθος των επαναλήψεων.

Η χρήση της εντολής ΜΕΧΡΙΣ_ΟΤΟΥ σε σχέση με την ΟΣΟ μπορεί να οδηγήσει σε απλούστερα προγράμματα (πχ στον έλεγχο εγκυρότητας ή στο μενού επιλογών).

Γενικά, όταν απαιτείται η εκτέλεση της επανάληψης τουλάχιστον μία φορά, είναι βολικότερη η χρήση της ΜΕΧΡΙΣ_ΟΤΟΥ.

15. Τι είναι εμφωλευμένοι βρόχοι;

Εμφωλευμένοι βρόχοι ονομάζονται δύο εντολές επανάληψης όταν ο ένας βρόχος περιέχεται μέσα στον άλλον.

16. Αναφέρατε τους κανόνες σωστής χρήσης των εμφωλευμένων βρόχων.

- Η είσοδος σε κάθε βρόχο γίνεται υποχρεωτικά από την αρχή του.
- Ο εσωτερικός βρόχος πρέπει να βρίσκεται ολόκληρος μέσα στον εξωτερικό, έτσι ώστε ο βρόχος που ξεκινάει τελευταίος να ολοκληρώνεται πρώτος.
- Εμφωλευμένοι βρόχοι δεν μπορούν να χρησιμοποιούν την ίδια μεταβλητή για μετρητή.