

Κεφάλαια 02-08

Δομές Επανάληψης

Εντολές επανάληψης

- Επιτρέπουν την εκτέλεση εντολών περισσότερες από μία φορά
- Οι επαναλήψεις ελέγχονται πάντοτε από κάποια συνθήκη η οποία καθορίζει την έξοδο από το βρόχο
- Βρόχος αποκαλείται το τμήμα του αλγόριθμου που επαναλαμβάνεται

Τρεις εντολές επανάληψης

- **ΟΣΟ...ΕΠΑΝΑΛΑΒΕ**
 - Η επανάληψη ελέγχεται από μία λογική έκφραση στην αρχή και εκτελείται συνεχώς όσο η συνθήκη είναι Αληθής
- **ΜΕΧΡΙΣ_ΟΤΟΥ**
 - Η συνθήκη βρίσκεται στο τέλος του βρόχου και εκτελείται συνεχώς μέχρις ότου η συνθήκη αυτή γίνει Αληθής
- **ΓΙΑ**
 - Ο βρόχος επαναλαμβάνεται για προκαθορισμένο αριθμό φορών.

Επιλογή εντολής επανάληψης

- **ΟΣΟ...ΕΠΑΝΑΛΑΒΕ**
 - Όταν δεν ξέρουμε πόσες επαναλήψεις θα κάνουμε
 - Ο βρόχος **μπορεί** να μην εκτελεστεί **καμία** φορά
- **ΜΕΧΡΙΣ_ΟΤΟΥ**
 - Όταν δεν ξέρουμε πόσες επαναλήψεις θα κάνουμε
 - Ο βρόχος θα εκτελεστεί **τουλάχιστον μία** φορά
- **ΓΙΑ**
 - Όταν ξέρουμε πόσες επαναλήψεις θα κάνουμε

ΟΣΟ...ΕΠΑΝΑΛΑΒΕ

- 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 στη ΓΛΩΣΣΑ

Σύνταξη

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

Παράδειγμα

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

- 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**.

Σημαντικό

Σύνταξη

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

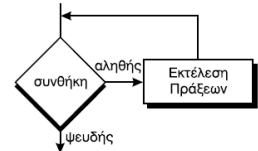
Παράδειγμα

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

- 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.
- Άρα θα παραβιάζεται το κριτήριο της ;

Αλγόριθμος & Διάγραμμα Ροής

```
Όσο συνθήκη επανάλαβε
  εντολές
Τέλος_επανάληψης
```



Παράδειγμα 7

Να γραφεί αλγόριθμος που να εμφανίζει τους αριθμούς από 1 έως 100.

```
Αλγόριθμος Παράδειγμα_7
i ← 1
Όσο i ≤ 100 επανάλαβε
  Εμφάνισε i
  i ← i + 1
Τέλος_επανάληψης
Τέλος Παράδειγμα_7
```

ΜΕΧΡΙΣ_ΟΤΟΥ

- Οι εντολές του βρόχου εκτελούνται μέχρις ότου ικανοποιηθεί κάποια συνθήκη η οποία ελέγχεται στο τέλος της επανάληψης.
- Πολύ συχνά η ίδια επαναληπτική διαδικασία μπορεί να γραφεί εξίσου σωστά είτε με τη δομή ΟΣΟ...ΕΠΑΝΑΛΑΒΕ είτε με τη δομή

ΜΕΧΡΙΣ_ΟΤΟΥ

- 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.

Σύνταξη

Σύνταξη

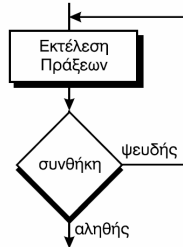
```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  εντολή-1
  εντολή-2
  ...
  εντολή-ν
ΜΕΧΡΙΣ_ΟΤΟΥ λογική-έκφραση
```

Παράδειγμα

```
ΑΡΧΗ_ΕΠΑΝΑΛΗΨΗΣ
  ΔΙΑΒΑΣΕ Α
  Αθροισμα <- Αθροισμα + Α
ΜΕΧΡΙΣ_ΟΤΟΥ Αθροισμα >= 1000
```

- 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.

Διάγραμμα Ροής



Παράδειγμα 9

Να διαβάζονται και να εκτυπώνονται όσοι θετικοί αριθμοί δίνονται από το πληκτρολόγιο. Ο αλγόριθμος τελειώνει, όταν δοθεί ένας αρνητικός αριθμός.

```
Αλγόριθμος Παράδειγμα_9
Αρχή_επανάληψης
  Διάβασε x
  Εμφάνισε x
Μέχρις_ότου x < 0
Τέλος Παράδειγμα_9
```

ΓΙΑ...ΑΠΟ...ΜΕΧΡΙ

- Χρησιμοποιείται στην περίπτωση που πρέπει να επαναληφθεί η εκτέλεση κάποιων εντολών για προκαθορισμένο αριθμό επαναλήψεων.

Σύνταξη

Σύνταξη

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

Παράδειγμα

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

Λειτουργία

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

- Η εντολή αυτή χειρίζεται μια **μεταβλητή**, στην οποία αρχικά εκχωρείται η **τιμή1**.
- Η τιμή της μεταβλητής συγκρίνεται με την **τιμή2** 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** (**τιμή3**).

Λειτουργία

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

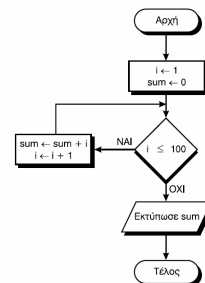
- Αν η νέα τιμή είναι μικρότερη της τελικής, τότε ο βρόχος εκτελείται ξανά.
- Η διαδικασία αυτή επαναλαμβάνεται συνεχώς, έως ότου η τιμή ελέγχου γίνει μεγαλύτερη της τελικής τιμής, οπότε η τερματίζεται η επανάληψη και το πρόγραμμα συνεχίζει με την εντολή που ακολουθεί το **ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ**.
- Αν η **τιμή3** του βήματος είναι 1, τότε το **ΜΕ ΒΗΜΑ** μπορεί να παραληφθεί.

Παράδειγμα 10

Να βρεθεί και να εκτυπωθεί το άθροισμα των 100 ακεραίων από το 1 μέχρι το 100.

```
Αλγόριθμος Παράδειγμα_10
Sum ← 0
Για i από 1 μέχρι 100
    Sum ← Sum + i
Τέλος_επανάληψης
Εκτύπωσε Sum
Τέλος Παράδειγμα_10
```

Παράδειγμα 10 - Διάγραμμα Ροής



Εμφωλευμένοι βρόχοι

- Πολύ συχνά για την επίλυση των προβλημάτων απαιτείται η χρήση εμφωλευμένων βρόχων.
- Σε αυτή την περίπτωση ο ένας βρόχος βρίσκεται μέσα στον άλλο.
- 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 2 – σελ. 174

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.

Παράδειγμα 3 – σελ. 177

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).

Παράδειγμα 4 – σελ. 178

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

Παράδειγμα 5 – σελ. 180

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

Ολίσθηση

- Στον υπολογιστή τα δεδομένα αποθηκεύονται με δυαδική μορφή, δηλαδή 0 και 1
- $17_{(10)} \Leftrightarrow 00010001_{(2)}$ (ένα byte)
- $00100010_{(2)} = 34_{(10)}$
- $00001000_{(10)} = 8_{(10)}$
- Η ολίσθηση προς τα αριστερά ισοδυναμεί με πολλαπλασιασμό επί δύο
- Η ολίσθηση προς τα δεξιά ισοδυναμεί με την ακέραια διαίρεση διά δύο.

Πολλαπλασιασμός αλά ρωσικά

- Η μέθοδος αυτή χρησιμοποιείται πρακτικά στους υπολογιστές, γιατί υλοποιείται πολύ πιο απλά απ' ό,τι ο γνωστός μας χειρωνακτικός τρόπος πολλαπλασιασμού.
- Απαιτεί πολλαπλασιασμό επί δύο, διαίρεση διά δύο και πρόσθεση.
- Η γνωστή διαδικασία πολλαπλασιασμού απαιτεί πολλαπλασιασμό με οποιοδήποτε ακέραιο και πρόσθεση.

Πολλαπλασιασμός αλά ρωσικά

- Ο λόγος που ο πολλαπλασιασμός αλά ρωσικά είναι προτιμότερος απ' ό,τι ο χειρωνακτικός τρόπος πολλαπλασιασμού δύο ακεραίων είναι ότι σε επίπεδο κυκλωμάτων υπολογιστή ο πολλαπλασιασμός επί δύο και η διαίρεση διά δύο μπορούν να υλοποιηθούν ταχύτατα με μία απλή εντολή ολίσθησης (shift), σε αντίθεση με τον πολλαπλασιασμό με οποιοδήποτε ακέραιο που θεωρείται πιο χρονοβόρα διαδικασία

Πολλαπλασιασμός αλά ρωσικά

Αλγόριθμος: Πολλαπλασιασμός δύο θετικών ακεραίων (αλά ρωσικά)	
Είσοδος:	Δύο ακέραιοι M1 και M2, όπου $M1, M2 \geq 1$
Έξοδος:	Το γινόμενο $P = M1 * M2$
Βήμα 1	Θέσε $P = 0$
Βήμα 2	Αν $M2 > 0$, τότε πήγαινε στο Βήμα 3, αλλιώς πήγαινε στο Βήμα 7
Βήμα 3	Αν ο M2 είναι περιττός, τότε θέσε $P = P + M1$
Βήμα 4	Θέσε $M1 = M1 * 2$
Βήμα 5	Θέσε $M2 = M2 / 2$ (θεώρησε μόνο το ακέραιο μέρος)
Βήμα 6	Πήγαινε στο Βήμα 2
Βήμα 7	Τύπωσε τον P.

Πολλαπλασιασμός αλά ρωσικά

```
Αλγόριθμος Πολλαπλασιασμός_αλά_ρωσικά
Δεδομένα // M1, M2 ακέραιοι //
P ← 0
Όσο M2 > 0 επανάλαβε
    Αν M2 mod 2 = 1 τότε P ← P+M1
    M1 ← M1*2
    M2 ← M2 div 2
Τέλος_επανάληψης
Αποτελέσματα // P, το γινόμενο των ακεραίων M1, M2 //
Τέλος Πολλαπλασιασμός_αλά_ρωσικά
```