

1.

Επαναλήψεις- Βρόχοι

1.1. Δομές Επανάληψης ή Βρόχοι

Πολλές φορές πρέπει να εκτελεστεί μια ομάδα εντολών περισσότερες από μια φορές. Σ' αυτές τις περιπτώσεις χρησιμοποιούνται οι δομές επανάληψης. Υπάρχουν τρεις δομές επανάληψης και μπορούν να κατηγοριοποιηθούν σε δυο περιπτώσεις χρήσης:

- Όταν το πλήθος των επαναλήψεων να είναι γνωστό εξ αρχής:
 - for που εκτελεί τις εντολές, που περικλείει, προκαθορισμένες φορές,
- Όταν το πλήθος των επαναλήψεων ΔΕΝ να είναι γνωστό εξ αρχής:
 - while που εκτελεί τις εντολές που περικλείει όσο η συνθήκη είναι ΑΛΗΘΗΣ και αυτή γράφεται στην αρχή της εντολής,
 - do...while που εκτελεί τις εντολές, που περικλείει, όσο η συνθήκη είναι ΨΕΥΔΗΣ και αυτή γράφεται στο τέλος της εντολής.

Όλες οι δομές επανάληψης έχουν τρία πολύ σημαντικά χαρακτηριστικά, στην πλειοψηφία των περιπτώσεων χρήσης τους:

- Αρχική Τιμή (αρχικοποίηση)
- Τελική Τιμή (Τερματισμός και παράλληλα έλεγχος για τον τερματισμό)
- Βήμα Μεταβολής (Αύξησης/Μείωσης)

1.2. Δομή επανάληψης *for*

Η δομή επανάληψης *for* χρησιμοποιείται όταν είναι γνωστό το πλήθος των επαναλήψεων, π.χ. να βρούμε το ΜΟ 35 μαθητών. Έχει σχεδιαστεί για να απλοποιήσει τον έλεγχο των επαναλήψεων, αν και μπορούν όλα τα προβλήματα να υλοποιηθούν και με τις άλλες δύο δομές επανάληψης υπό συνθήκη. Η γενική της σύνταξη είναι:

```
for (αρχική_τιμή_μεταβλητής_ελέγχου;  
    τελική_τιμή_μεταβλητής_ελέγχου;  
    βήμα_μεταβολής_μεταβλητής_ελέγχου) {  
    .  
    . <εντολές>  
    .  
}
```

Η δομή αυτή χειρίζεται από μια μεταβλητή στην οποία καταχωρείται αρχικά η αρχική τιμή. Η τιμή που παίρνει η μεταβλητή συγκρίνεται με τη τελική τιμή και εφόσον δεν έχει φτάσει στην τιμή αυτή (θέλει προσοχή στο χειρισμό της σε περιπτώσεις που αποκλίνει σύμφωνα με τη μεταβολή), τότε εκτελούνται οι εντολές που βρίσκονται στο βρόχο. Στη συνέχεια η μεταβλητή αλλάζει κατά την τιμή του βήματος μεταβολής. Αν η νέα τιμή είναι μικρότερη της τελικής, τότε ο βρόχος εκτελείται ξανά. Η διαδικασία επαναλαμβάνεται μέχρι η μεταβλητή να γίνει ξεπεράσει (θετικά ή αρνητικά, ανάλογα με τον τρόπο μεταβολής) την τελική τιμή ελέγχου, αλλιώς η δομή τερματίζεται και ο έλεγχος του προγράμματος συνεχίζει με την επόμενη- μετά το βρόχο- εντολή.

Πρέπει να επισημανθεί ότι:

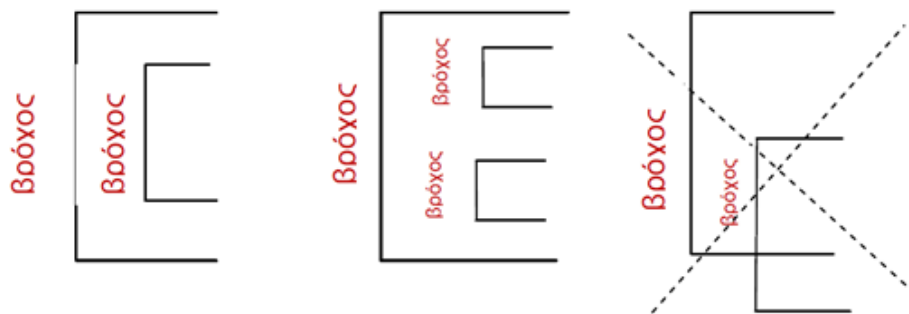
1. Μέσα στη δομή *for* **δεν επιτρέπεται να αλλάζετε την τιμή της μεταβλητής ελέγχου της δομής *for***, γιατί τότε δεν λειτουργεί σωστά η εντολή επανάληψης.
2. Το βήμα δεν είναι απαραίτητο να είναι ακέραιος αριθμός. Μπορεί να πάρει οποιαδήποτε πραγματική τιμή.

1.3. Εμφωλευμένοι Βρόχοι

Στην περίπτωση αυτή ο ένας βρόχος βρίσκεται μέσα στον άλλο. Για τη σωστή λειτουργία των εμφωλευμένων βρόχων πρέπει να ισχύουν συγκεκριμένοι κανόνες:

- Ο εσωτερικός βρόχος πρέπει να βρίσκεται ολόκληρος μέσα στον εξωτερικό
- Οι επαναλήψεις του εσωτερικού βρόχου πρέπει να ολοκληρώνονται για όλες τις τιμές της μεταβλητής του εσωτερικού και έπειτα να μεταβάλλεται το βήμα της εξωτερικής και πάλι να επαναλαμβάνεται ο εσωτερικός βρόχος για όλες τις

τιμές της εσωτερικής και στο τέλος να μεταβάλλεται πάλι το βήμα της εξωτερικής.



Η είσοδος σε βρόχο υποχρεωτικά γίνεται από την αρχή του.

- Δεν πρέπει να χρησιμοποιηθεί η ίδια μεταβλητή ως μετρητής δύο ή περισσότερων βρόχων που ο ένας βρίσκεται στο εσωτερικό του άλλου.

1.4. Δομή Επανάληψης *while*

Όταν δεν είναι καθορισμένο το πλήθος των επαναλήψεων εξ' αρχής τότε δεν μπορεί να χρησιμοποιηθεί η δομή επανάληψης *for*. Σε αυτήν την περίπτωση χρησιμοποιείται η δομή *while*, με την οποίες οι εντολές που περιλαμβάνει επαναλαμβάνονται όσο η συνθήκη που ελέγχει είναι ΑΛΗΘΗΣ και μόνο όταν γίνει ΨΕΥΔΗΣ σταματά να εκτελεί αυτές και εκτελεί την αμέσως επόμενη εντολή μετά της δομής επανάληψης *while*.

Στον τρόπο λειτουργίας της δομής πρέπει να επισημανθούν τα ακόλουθα χαρακτηριστικά:

- Επειδή η συνθήκη βρίσκεται στην αρχή της εντολής υπάρχει περίπτωση να βγει ΨΕΥΔΗΣ από την αρχή οπότε και να μην εκτελεστούν καμία φορά οι εντολές της επανάληψης.
- Αν η συνθήκη δεν βγει ΨΕΥΔΗΣ ποτέ, τότε οι εντολές της επανάληψης εκτελούνται συνεχώς και ο αλγόριθμος (ή το πρόγραμμα) δεν τελειώνει ποτέ. Τότε λέμε ότι έχουμε "ατέρμων βρόχο".

Κατά τη σύνταξη της:

- Δίνουμε αρχική τιμή στη μεταβλητή της συνθήκης, γιατί η συνθήκη βρίσκεται στην αρχή της *while*. και πρέπει η μεταβλητή να έχει αρχικοποιηθεί για να μπορεί να εκτελέσει ο υπολογιστής τη συνθήκη και να δώσει τη τιμή ΑΛΗΘΗΣ ή ΨΕΥΔΗΣ.

- Πριν τελειώσει η επανάληψη, δηλαδή πριν το τέλος της επανάληψης πρέπει να αλλάξουμε τη τιμή της μεταβλητής της συνθήκης, όπου αυτό χρειάζεται, για να μην έχουμε ατέρμων βρόχο.

Η γενική σύνταξη της είναι της μορφής:

```
while (συνθήκη) {
    .
    . <εντολές>
    .
}
```

Και ορισμένες από πιο συχνές μορφές εμφάνισης της, σε γενική περιγραφή είναι:

```
scanf(μεταβλητή);
while(έλεγχος_τελικής_τιμής_μεταβλητής) {
    .
    . <εντολές>
    .
    scanf(μεταβλητή);
}
```

ή

```
scanf(μεταβλητή);
while(έλεγχος_τελικής_τιμής_μεταβλητής) {
    .
    . <εντολές>
    .
    μεταβολή_μεταβλητής_ελέγχου; (εντός της δομής, σε
    όποιο σημείο έχει το επιθυμητό αποτέλεσμα)
}
```

ή

```
αρχικοποίηση μεταβλητής ελέγχου;
while(έλεγχος_τελικής_τιμής_μεταβλητής) {
    .
    . <εντολές>
    .
    μεταβολή_μεταβλητής_ελέγχου; (εντός της δομής, σε
    όποιο σημείο έχει το επιθυμητό αποτέλεσμα)
}
```

ή

```
παράσταση υπολογισμού μεταβλητής ελέγχου;
while(έλεγχος_τελικής_τιμής_μεταβλητής) {
    .
    . <εντολές>
    .
    μεταβολή_μεταβλητής_ελέγχου; (εντός της δομής, σε
    όποιο σημείο έχει το επιθυμητό αποτέλεσμα)
}
```

Ωστόσο, στη συνθήκη ελέγχου μπορεί αντί για μια μόνο μεταβλητή να υπάρχει παράσταση, κυρίως με τη συνδυαστική χρήση λογικών πράξεων, οπότε είτε η ανάγνωση, είτε η αρχικοποίηση είτε η παράσταση υπολογισμού αναφέρεται σε όλες τις μεταβλητές που εμπλέκονται στο έλεγχο.

1.5. Δομή Επανάληψης *do...while*

Συμπληρωματικά της δομής *while* χρησιμοποιείται η δομή *do...while*. Η σύνταξη της δομής είναι:

```
do {  
    .  
    . <εντολές>  
    .  
} while (συνθήκη);
```

Παρατηρούμε ότι σε αυτή τη δομή ο έλεγχος της συνθήκης πραγματοποιείται στο τέλος του βρόχου. Οι εντολές που περιλαμβάνει επαναλαμβάνονται μέχρις ότου η συνθήκη γίνει αληθής. Δηλαδή, όσο η συνθήκη είναι ΨΕΥΔΗΣ έχουμε επανάληψη, ενώ όταν η συνθήκη γίνει ΑΛΗΘΗΣ η επανάληψη σταματά και εκτελείται η αμέσως επόμενη εντολή μετά το βρόχο.

Μια πιο πρακτική μορφή χρήσης της είναι:

```
αρχικοποίηση μεταβλητής ελέγχου;  
do {  
    .  
    . <εντολές>  
    .  
    μεταβολή_μεταβλητής_ελέγχου; (εντός της δομής, σε  
    όποιο σημείο έχει το επιθυμητό αποτέλεσμα)  
}  
while (συνθήκη_έλεγχος_τελικής_τιμής_μεταβλητής_ελέγχου);
```

Και σε αυτήν την περίπτωση, στη συνθήκη ελέγχου μπορεί αντί για μια μόνο μεταβλητή να υπάρχει παράσταση με τη συνδυαστική χρήση λογικών πράξεων.

1.6. Σύγκριση των δομών επανάληψης *while* και *do...while*

Στη δομή *do...while* η συνθήκη που ελέγχεται είναι η αντίθετη από την *while*.

Επειδή η συνθήκη στην εντολή επανάληψης *do...while* βρίσκεται στο τέλος τη εντολής οι εντολές, που περιλαμβάνονται σ' αυτήν, θα εκτελεστούν τουλάχιστον μια φορά.

Χρησιμοποιείται:

- Στα προβλήματα που έχουν Μενού επιλογής.
- Στα προβλήματα που έχουμε Έλεγχο δεδομένων από το πληκτρολόγιο.
- Στα προβλήματα που θέλουμε να επαναλάβουμε μια ολόκληρη επεξεργασία (π.χ. Αναζήτηση δεδομένων) απαντώντας στην ερώτηση 'Θέλεις να συνεχίσεις (N/O)?'.

Στις παραπάνω τρεις περιπτώσεις, που αναφέρθηκαν για επιλογή της εντολής do...while, μπορεί να χρησιμοποιηθεί και η while αλλά οδηγεί σε υλοποιήσεις με περισσότερες εντολές.

1.7. Μετατροπές μεταξύ των δομών επανάληψης

Ένα ζήτημα που μπορεί να μας απασχολήσει είναι η μετατροπή από την μία δομή στην άλλη.

1. Μια δομή for πάντα μπορεί να μετατραπεί στις while και do...while, **το αντίθετο δεν ισχύει**.
2. Η δομή while μετατρέπεται πάντα στην do...while.
3. Ειδικές περιπτώσεις χρήσης της do...while (αλλά και της while) είναι ο έλεγχος εγκυρότητας εισόδου, δηλαδή να επιθυμούμε να ελέγχεται μια τιμή και εφόσον ο χρήστης του προγράμματος μας δε δίνει σωστή και σύμφωνη με το σχεδιασμό είσοδο, να επαναλαμβάνεται μια προτροπή για τον τύπο της αποδεκτής εισόδου.

1.8. Ασκήσεις

1. Γράψτε ένα πρόγραμμα το οποίο θα εκτυπώνει τους αριθμούς 1 έως 100 διατεταγμένους σε 10 στήλες. Κάθε αριθμός θα πρέπει να χωρίζεται από τον επόμενο του με ένα χαρακτήρα οριζόντιου στηλοθέτη (tab).
2. Να υλοποιηθεί πρόγραμμα σε C που θα εκτυπώνει τους ASCII χαρακτήρες και την αντίστοιχη δεκαδική τους αναπαράσταση. Τα μοναδικά δεδομένα που σας δίνονται είναι ότι η δεκαδική αναπαράσταση του χαρακτήρα μηδέν (0) είναι το 48 και ότι θα εκτυπώνει μέχρι τον χαρακτήρα που η δεκαδική του αναπαράσταση είναι το 126. Η εκτύπωση των χαρακτήρων και της δεκαδικής τους τιμής θα γίνει ως εξής: σε κάθε γραμμή θα εκτυπώνονται δέκα ASCII χαρακτήρες με την αντίστοιχη δεκαδική τους αναπαράσταση μέσα σε παρένθεση. Μεταξύ δύο χαρακτήρων θα παρεμβάλλεται ένας χαρακτήρας '\t', π.χ. 0 (48) 1(..) 2 (...) ... 9(..)
3. Δημιουργήστε ένα πρόγραμμα το οποίο δέχεται από το χρήστη το πληθυσμό μιας πόλης και τον ετήσιο ρυθμό αύξησής του και υπολογίστε και εμφανίστε στην οθόνη σε πόσα χρόνια θα έχει διπλασιαστεί.
4. Να γραφεί ένα πρόγραμμα το οποίο να διαβάσει έναν ακέραιο (π.χ. N) και να εμφανίζει το αποτέλεσμα της παράστασης: $3^2+6^2+9^2+ \dots +(3 \times N)^2$. Το πρόγραμμα να

υποχρεώνει τον χρήστη να εισάγει έναν θετικό ακέραιο μικρότερο του 10. Να χρησιμοποιήσετε έναν μόνο for βρόχο.

5. Να γράψετε πρόγραμμα που δέχεται ως είσοδο ένα αριθμό και τσεκάρει αν ο αριθμός είναι πρώτος ή όχι. Αν ΔΕΝ είναι πρώτος να εμφανίσει το πλήθος των αριθμών που τον διαιρούν ακριβώς;
6. Να γραφεί πρόγραμμα το οποίο διαβάζει μία ακολουθία από n αριθμούς και εμφανίζει τον αριθμό που εμφανίστηκε στην ακολουθία περισσότερες συνεχόμενες φορές.
7. Να γράψετε πρόγραμμα σε C, που θα διαβάζει έναν ακέραιο αριθμό N από το πληκτρολόγιο και θα εμφανίζει:
 - A. τους N πρώτους φυσικούς αριθμούς,
 - B. το άθροισμά τους,
 - Γ. την μέση τιμή τους,
 - Δ. μια νέα ακολουθία αριθμών, όπου κάθε στοιχείο της θα ισούται με το αντίστοιχο της πρώτης υψωμένο στο τετράγωνο.