



104

Αλγόριθμοι, Προγραμματισμός και Δομές Δεδομένων

Εργαστήριο – Κεφάλαιο 6 **Δομές επανάληψης (for)**

Διδάσκουσα:
Σοφία Αλμπάνη

Άσκηση 6.1

6.1) Να γραφεί πρόγραμμα σε γλώσσα **C** το οποίο χρησιμοποιεί τη δομή **επανάληψης for** με **πολλαπλές μεταβλητές ελέγχου**.

Το πρόγραμμα πρέπει:

- Να ξεκινά με δύο ακέραιες μεταβλητές $i=10$ και $j=2$.
- Σε κάθε επανάληψη:
 - Η i να μειώνεται κατά 2.
 - Η j να αυξάνεται κατά 2.
 - Να εκτυπώνονται οι τρέχουσες τιμές τους.
- Η επανάληψη να συνεχίζεται **όσο οι τιμές τους δεν είναι ίσες** ($i \neq j$).

Στόχος:

Κατανόηση της δομής **for** με **πολλαπλές αρχικοποιήσεις, συνθήκες και μεταβολές** σε ένα βήμα.

Άσκηση 6.2

6.2) Να γραφεί πρόγραμμα σε γλώσσα **C** που να υπολογίζει το **παραγοντικό** ενός αριθμού ($n!$) χρησιμοποιώντας τη δομή **επανάληψης for**.

Να διαβάσει έναν ακέραιο αριθμό `num` από το πληκτρολόγιο, ο οποίος πρέπει να ανήκει στο διάστημα **[0, 170]**.
Αν η τιμή είναι εντός ορίων:

Να υπολογίζει το **παραγοντικό του αριθμού** χρησιμοποιώντας την επαναληπτική δομή **for**.

Να εμφανίζει το αποτέλεσμα σε **εκθετική μορφή (%e)**.

Αν η τιμή είναι εκτός ορίων, να εμφανίζει μήνυμα σφάλματος.

Υπενθύμιση

Το παραγοντικό ενός αριθμού n ορίζεται ως: $n! = 1 \times 2 \times 3 \times \dots \times n$ και ισχύει ότι $0! = 1$.

Στόχος:

Εξάσκηση στη χρήση της δομής **for** και στην επαναληπτική επίλυση μαθηματικών προβλημάτων.

Άσκηση 6.3

6.3) Να γραφεί πρόγραμμα σε γλώσσα **C** το οποίο:

- Να ζητά από τον χρήστη έναν ακέραιο αριθμό **μεταξύ 1 και 10**.
- Η είσοδος να γίνεται μέσα σε **άπειρη επανάληψη for(;;)**, ώστε να ζητείται ξανά ο αριθμός **μέχρι να δοθεί έγκυρη τιμή**.
- Όταν δοθεί έγκυρη τιμή ($1 \leq \text{num} \leq 10$), η επανάληψη να τερματίζεται με `break`.
- Στη συνέχεια, να εμφανίζεται ο **πίνακας πολλαπλασιασμού** του αριθμού αυτού από 1 έως 10.

Στόχος:

Χρήση της άπειρης επανάληψης `for(;;)`, της διακοπής επανάληψης με `break`, και εμφάνιση αποτελεσμάτων μέσα από επαναληπτική δομή.



Άσκηση 6.4

6.4) Να γραφεί πρόγραμμα σε γλώσσα **C** το οποίο χρησιμοποιεί **εμφωλευμένους βρόχους for** για να εμφανίζει όλους τους πίνακες πολλαπλασιασμού από το 1 έως το 10. Το πρόγραμμα πρέπει:

Να χρησιμοποιεί δύο βρόχους for:

- Ο **εξωτερικός βρόχος** (i) να επαναλαμβάνεται από 1 έως 10, αντιπροσωπεύοντας τον αριθμό του πίνακα πολλαπλασιασμού.
- Ο **εσωτερικός βρόχος** (j) να επαναλαμβάνεται επίσης από 1 έως 10, και να υπολογίζει κάθε γραμμή του πίνακα.

Να εμφανίζει τα αποτελέσματα στη μορφή: $i * j = (i*j)$

Στόχος

Κατανόηση εμφωλευμένων επαναλήψεων (for μέσα σε for) και εξάσκηση στην παραγωγή πολλαπλών υπολογισμών σε μορφή πινάκων.

Άσκηση 6.5

6.5) Να γραφεί πρόγραμμα σε γλώσσα **C** το οποίο χρησιμοποιεί **τριπλή εμφωλευμένη επανάληψη for** για την εμφάνιση ενός σύνθετου μοτίβου από σύμβολα στην οθόνη. Το πρόγραμμα πρέπει:

Να χρησιμοποιεί **τρεις εμφωλευμένους βρόχους for** με μετρητές i , j , και k .

- Ο εξωτερικός βρόχος (i) να εκτελείται όσο $i < 2$.
- Ο μεσαίος βρόχος (j) να ξεκινά από $i + 1$ και να μειώνεται μέχρι το 0.
- Ο εσωτερικός βρόχος (k) να εκτελείται από i έως $j + 1$.

Σε κάθε βήμα να εκτυπώνονται τα σύμβολα `?`, `!`, και `*` με κατάλληλη σειρά.

Παράδειγμα εκτέλεσης: `? ! * * * ! * * ? ! * * * ! * *`

(η έξοδος μπορεί να εμφανιστεί σε μία γραμμή, ανάλογα με τη ρύθμιση της `printf`)

Στόχος

Κατανόηση της έννοιας εμφωλευμένων επαναλήψεων (nested for) και της σειράς εκτέλεσης των εντολών σε πολλαπλούς βρόχους.



Άσκηση 6.6

6.6) Να γραφεί πρόγραμμα σε γλώσσα **C** που να εμφανίζει ένα **τριγωνικό σχήμα αστεριών (*)** στην οθόνη.

Το πρόγραμμα πρέπει:

- Να ζητά από τον χρήστη έναν ακέραιο αριθμό lines, ο οποίος θα καθορίζει τον **αριθμό γραμμών** του τριγώνου.
- Για κάθε γραμμή i (ξεκινώντας από 0), να εκτυπώνει **όσα αστεράκια (*) είναι ίσα με τον αριθμό της γραμμής + 1**.
- Μετά την εκτύπωση κάθε γραμμής, να γίνεται αλλαγή γραμμής με `\n`.

Παράδειγμα

Enter lines: 4

*

* *

* * *

* * * *

Στόχος

Εξάσκηση στη χρήση εμφωλευμένων επαναλήψεων (for μέσα σε for) και ενίσχυση της κατανόησης της σχέσης μεταξύ δείκτη επανάληψης και πλήθους εκτυπώσεων.

Άσκηση 6.7

6.7) Να γραφεί πρόγραμμα σε γλώσσα **C** που να υπολογίζει τους **μέσους όρους βαθμολογίας μαθητών** σε διάφορα μαθήματα. Το πρόγραμμα πρέπει:

Να ορίζει δύο σταθερές:

```
#define LESSONS 3 (ο αριθμός μαθημάτων)
```

```
#define STUDENTS 5 (ο αριθμός μαθητών)
```

Για κάθε μαθητή:

- Να εμφανίζει μήνυμα: ***** Student_X
- Να ζητά βαθμό για κάθε μάθημα (lesson 1, lesson 2, lesson 3).
- Να υπολογίζει τον **μέσο όρο** του συγκεκριμένου μαθητή και να τον εμφανίζει.

Μετά το τέλος όλων των μαθητών, να υπολογίζει και να εμφανίζει τον **συνολικό μέσο όρο** όλων των βαθμών.

Στόχος

Εξάσκηση στη χρήση σταθερών (#define), σε εμφωλευμένες επαναληπτικές δομές, και στον υπολογισμό μέσων όρων μέσω αθροισμάτων και μετρητών.

Συνοπτικά

Κατηγορία	Συντακτικό / Μορφή	Περιγραφή – Χρήση
Δομή επανάληψης for	<code>for (αρχικοποίηση; συνθήκη; μεταβολή) { ... }</code>	Εκτελεί το σώμα όσες φορές η συνθήκη είναι αληθής . Ιδανική όταν γνωρίζουμε εκ των προτέρων τον αριθμό επαναλήψεων.
Άπειρη επανάληψη	<code>for (; ;)</code>	Δημιουργεί βρόχο χωρίς συνθήκη λήξης . Τερματίζεται με <code>break</code> . Χρήσιμη για επαναλαμβανόμενη εισαγωγή δεδομένων.
Εμφωλευμένες επαναλήψεις (nested loops)	<code>for (...) { for (...) { ... } }</code>	Μία επανάληψη μέσα σε άλλη. Χρησιμοποιείται σε πίνακες, σχήματα, πινάκες πολλαπλασιασμού κ.ά.
Έλεγχος επανάληψης με break	<code>if (συνθήκη) break;</code>	Διακόπτει άμεσα την εκτέλεση της τρέχουσας επανάληψης και βγαίνει από τον βρόχο.
Μεταβλητές ελέγχου βρόχου	<code>π.χ. int i;</code> <code>for (i=0; i<10; i++)</code>	Χρησιμοποιούνται για να καθορίζουν το πλήθος των επαναλήψεων. Συνήθως ακέραιες.