



104

Αλγόριθμοι, Προγραμματισμός και Δομές Δεδομένων

Εργαστήριο – Κεφάλαιο 6β & 7α

Δομές επανάληψης (while) & Μονοδιάστατοι πίνακες

Διδάσκουσα:

Σοφία Αλμπάνη



Άσκηση 6.1

6.1) Να γραφεί πρόγραμμα σε γλώσσα **C** το οποίο να ζητά από τον χρήστη έναν αριθμό, να υπολογίζει το άθροισμα **τετραγώνων διαδοχικών θετικών ακεραίων** και να βρίσκει ποιος είναι ο **τελευταίος αριθμός** που μπορεί να προστεθεί πριν το άθροισμα ξεπεράσει μία δοσμένη τιμή.

Παράδειγμα Εκτέλεσης

Enter number: 50

The last number is = 4

(Γιατί $1^2 + 2^2 + 3^2 + 4^2 = 30 < 50$, ενώ $+ 5^2 = 55 \geq 50 \rightarrow$ σταματά πριν το 5)

Στόχος:

- Χρήση άπειρης επανάληψης while(1) με έξοδο μέσω break.
- Εξάσκηση στη συσσώρευση αθροισμάτων.
- Εφαρμογή αριθμητικών υπολογισμών μέσα σε επανάληψη.

Άσκηση 6.2

6.2) Να γραφεί πρόγραμμα σε γλώσσα **C** το οποίο να ζητά από τον χρήστη επαναλαμβανόμενα να εισάγει έναν ακέραιο αριθμό (num).

- Αν ο αριθμός είναι **αρνητικός**, η επανάληψη **σταματά** (τερματισμός με break).
- Αν ο αριθμός είναι **μη αρνητικός**, τότε:
 - Εκτελείται ένας **εσωτερικός βρόχος while**, ο οποίος τυπώνει τη λέξη: «Hello» **όσες φορές** δείχνει η τιμή του αριθμού.
 - Ο αριθμός αυτός προστίθεται στο μετρητή times, ο οποίος κρατά το **σύνολο** των επαναλήψεων.
- Μετά τον τερματισμό, εμφανίζεται ο συνολικός αριθμός εμφανίσεων της λέξης "Hello".

Στόχος:

- Χρήση άπειρης επανάληψης: while(1)
- Διακοπή επανάληψης με break
- Χρήση εσωτερικού while για καθορισμένο αριθμό επαναλήψεων
- Χρήση μεταβλητής συσσώρευσης (times)

Άσκηση 6.3

6.3) Να γραφεί πρόγραμμα σε C που να διαβάζει επαναλαμβανόμενα τη βαθμολογία **ασκήσεων** και **εξέτασης** μαθητών.

- Η εισαγωγή τερματίζει όταν δοθεί **-1** σε έναν από τους δύο βαθμούς.
- Αν κάποιος βαθμός δεν είναι στο διάστημα **[0–10]**, να εμφανίζεται μήνυμα λάθους και να ζητείται ξανά.
- Αν **και οι δύο βαθμοί ≥ 5** , ο τελικός βαθμός να υπολογίζεται ως:

$$0.3 \times \text{βαθμός ασκήσεων} + 0.7 \times \text{βαθμός εξέτασης}$$

- Διαφορετικά, ο τελικός βαθμός να είναι ο **μικρότερος** από τους δύο βαθμούς.
- Στο τέλος, να εμφανίζεται ο **μέσος όρος των τελικών βαθμών** όλων των μαθητών.

Στόχος:

- Χρήση άπειρης επανάληψης `while(1)`, έλεγχος δεδομένων εισόδου και χρήση της εντολής `continue`, και όροι με λογικούς τελεστές,

Άσκηση 6.4

6.4) Να γραφεί πρόγραμμα σε C που να πραγματοποιεί **εξάσκηση στον πολλαπλασιασμό** με τυχαίους αριθμούς. Το πρόγραμμα:

- Παράγει τυχαία δύο αριθμούς μεταξύ **1 και 10**.
- Εμφανίζει την πράξη πολλαπλασιασμού και ζητά από τον χρήστη την απάντηση.
- Αν η απάντηση είναι σωστή, αυξάνει τον μετρητή **σωστών** (wins), αλλιώς αυξάνει τον μετρητή **λανθασμένων** (fails) και εμφανίζει τη σωστή λύση.
- Η διαδικασία συνεχίζεται **μέχρι ο χρήστης να εισάγει -1**.
- Στο τέλος, εμφανίζει το πλήθος των σωστών και λανθασμένων απαντήσεων.

Στόχος

Χρήση της δομής do-while για επαναλαμβανόμενη εισαγωγή δεδομένων

Χρήση της βιβλιοθήκης stdlib.h για παραγωγή τυχαίων αριθμών (rand()), αρχικοποίηση του γεννήτορα τυχειότητας με srand(time(NULL)).

Άσκηση 6.5

6.5) Να γραφεί πρόγραμμα σε C που να υπολογίζει μία **προσεγγιστική τιμή του π** χρησιμοποιώντας το ανάπτυγμα της σειράς του Leibniz:

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \dots$$

Το πρόγραμμα πρέπει να:

- χρησιμοποιεί **δομή επανάληψης do-while**,
- εναλλάσσει το πρόσημο του όρου σε κάθε βήμα,
- σταματά την επανάληψη όταν ο όρος $\frac{1}{i}$ γίνει μικρότερος από **0.0001**,
- και να εμφανίζει την τελική προσεγγιστική τιμή του π με **10 δεκαδικά ψηφία**.

Στόχος:

Χρήση της δομής do-while



Άσκηση 7.1

7.1) Να γραφεί πρόγραμμα σε C το οποίο να διαβάζει **10 ακέραιους αριθμούς** και να τους αποθηκεύει σε έναν πίνακα. Στη συνέχεια, το πρόγραμμα πρέπει να ελέγχει αν ο πίνακας είναι **συμμετρικός**, δηλαδή αν ισχύει:

$$a[0] = a[9], a[1] = a[8], a[2] = a[7], \dots$$

- Αν βρεθεί ζεύγος στοιχείων που **δεν είναι ίσο**, να εμφανίζεται το μήνυμα: «Non symmetric array» και το πρόγραμμα να τερματίζει.
- Αν όλα τα αντίστοιχα στοιχεία ταιριάζουν, να εμφανίζεται: «Symmetric array»

Στόχος:

Χρήση μονοδιάστατου πίνακα και έλεγχο συμμετρίας τους με χρήση επαναληπτικής δομής.



Άσκηση 7.2

7.2) Να γραφεί πρόγραμμα σε C το οποίο να διαβάζει **10 ακέραιες τιμές** για δύο πίνακες ίδιου μεγέθους (πίνακας 1 και πίνακας 2). Στη συνέχεια, το πρόγραμμα πρέπει να ελέγχει και να εντοπίζει **τα κοινά στοιχεία** των δύο πινάκων.

- Για κάθε κοινό στοιχείο που βρεθεί, να εμφανίζεται:
Common = <τιμή> (Pos_1 = <θέση στον πίνακα 1> , Pos_2 = <θέση στον πίνακα 2>)
- Αν δεν βρεθεί κανένα κοινό στοιχείο, να εμφανίζεται: No common elements were found

Στόχος:

Κατανόηση συσχέτισης δύο πινάκων με χρήση εμφωλευμένων επαναλήψεων (for μέσα σε for).



Άσκηση 7.3

7.3) Να γραφεί πρόγραμμα σε C το οποίο:

Δημιουργεί έναν πίνακα **10** θέσεων και τον γεμίζει με **τυχαίους ακέραιους αριθμούς** στο διάστημα **[5, 20]**.

Στη συνέχεια, να **διαγράψει τα διπλότυπα στοιχεία** του πίνακα:

- Κάθε φορά που εντοπίζεται δεύτερη εμφάνιση ενός αριθμού,
- το στοιχείο αυτό να αφαιρείται,
- και τα επόμενα στοιχεία να **μετακινούνται μία θέση αριστερά**.

Μετά τη διαδικασία αφαίρεσης, οι **κενές θέσεις** που απομένουν στο τέλος του πίνακα να συμπληρώνονται με την τιμή **-1**. Τέλος, να εμφανίζεται ο πίνακας **μετά την αφαίρεση των διπλοτύπων**.

Στόχος:

Εξάσκηση σε: χρήση τυχαίων αριθμών (rand() και srand()), εργασία με πίνακες και διαγραφή στοιχείων, εμφωλευμένους βρόχους (for + while), και διατήρηση μεγέθους πίνακα με τη μεταβλητή size.

Συνοπτικά

Κατηγορία	Συντακτικό / Μορφή	Περιγραφή – Χρήση
Επανάληψη while	<code>while (συνθήκη) { ... }</code>	Εκτελείται όσο η συνθήκη είναι αληθής. Η συνθήκη ελέγχεται πριν από κάθε επανάληψη.
Επανάληψη do-while	<code>do { ... } while (συνθήκη);</code>	Εκτελείται τουλάχιστον μία φορά , γιατί η συνθήκη ελέγχεται μετά το σώμα της επανάληψης.
Εμφωλευμένες επαναλήψεις	<code>for(...) { for(...) { ... } }</code>	Μία επανάληψη μέσα σε άλλη. Χρησιμοποιείται για πίνακες και σχήματα.
Έλεγχος τερματισμού με break	<code>if (συνθήκη) break;</code>	Διακόπτει την επανάληψη και συνεχίζει το πρόγραμμα μετά τον βρόχο.
Παράλειψη βήματος με continue	<code>if (συνθήκη) continue;</code>	Παραλείπει τις υπόλοιπες εντολές του τρέχοντος βρόχου και περνά στο επόμενο βήμα.
Δήλωση πίνακα	<code>int arr[SIZE];</code>	Δέσμευση συνεχόμενης μνήμης για αποθήκευση πολλών τιμών του ίδιου τύπου.
Προσπέλαση στοιχείου	<code>arr[i]</code>	Πρόσβαση σε συγκεκριμένη θέση του πίνακα (i-οστό στοιχείο).
Διάσχιση πίνακα	<code>for(i=0; i<SIZE; i++) printf("%d", arr[i]);</code>	Περνάμε όλα τα στοιχεία με βρόχο.