

3^ο Φροντιστήριο Δομημένου Προγραμματισμού

Άσκηση 1 : Έλεγχος θετικών, αρνητικών και πολλαπλασίων του 5

```
#include <stdio.h>

int main() {
    int nums[5]; // πίνακας 5 ακεραίων

    // Είσοδος δεδομένων
    printf("Δώσε 5 αριθμούς:\n");
    for (int i = 0; i < 5; i++) {
        printf("Αριθμός %d: ", i + 1);
        scanf("%d", &nums[i]);
    }

    // Επεξεργασία και έξοδος
    printf("\nΑποτελέσματα:\n");
    for (int i = 0; i < 5; i++) {
        if (nums[i] > 0) {
            if (nums[i] % 5 == 0)
                printf("%d είναι θετικός και πολλαπλάσιο του 5\n", nums[i]);
            else
                printf("%d είναι θετικός\n", nums[i]);
        } else if (nums[i] < 0) {
            if (nums[i] % 5 == 0)
                printf("%d είναι αρνητικός και πολλαπλάσιο του 5\n", nums[i]);
            else
                printf("%d είναι αρνητικός\n", nums[i]);
        } else {
            printf("%d είναι μηδέν\n", nums[i]);
        }
    }

    return 0;
}
```

Ανάλυση τμημάτων

1. Δήλωση πίνακα

2. `int nums[5];`

Ορίζουμε έναν πίνακα 5 ακεραίων για αποθήκευση των αριθμών που θα δώσει ο χρήστης.

3. Είσοδος δεδομένων

4. `for (int i = 0; i < 5; i++) {`

5. `printf("Αριθμός %d: ", i + 1);`

6. `scanf("%d", &nums[i]);`

7. `}`

- Χρησιμοποιούμε for για να ζητήσουμε 5 αριθμούς.
 - Κάθε τιμή αποθηκεύεται στη θέση nums[i].
8. Έλεγχος και εμφάνιση
9. if (nums[i] > 0) { ... }
10. else if (nums[i] < 0) { ... }
11. else { ... }
- Ελέγχουμε για κάθε στοιχείο αν είναι θετικό, αρνητικό ή 0.
 - Μέσα σε κάθε περίπτωση υπάρχει εμφωλευμένος if που ελέγχει αν είναι πολλαπλάσιο του 5.
12. Έξοδος
- Εκτυπώνουμε περιγραφικό μήνυμα για κάθε στοιχείο του πίνακα.

Άσκηση 2 : Υπολογισμός μέσου όρου και μέγιστου με πίνακα

```
#include <stdio.h>

int main() {
    int n;
    printf("Πόσους αριθμούς θα εισάγεις; ");
    scanf("%d", &n);

    int arr[n]; // πίνακας με n στοιχεία

    // Είσοδος δεδομένων
    for (int i = 0; i < n; i++) {
        printf("arr[%d] = ", i);
        scanf("%d", &arr[i]);
    }

    // Υπολογισμοί
    int sum = 0;
    int max = arr[0];

    for (int i = 0; i < n; i++) {
        sum += arr[i];
        if (arr[i] > max)
            max = arr[i];
    }

    double average = (double)sum / n;

    // Έξοδος αποτελεσμάτων
    printf("\nΑθροισμα: %d\n", sum);
    printf("Μέγιστο: %d\n", max);
    printf("Μέσος όρος: %.2f\n", average);

    return 0;
}
```

Ανάλυση τμημάτων

1. Δυναμικό μέγεθος πίνακα
 - ο Ο χρήστης δίνει πόσους αριθμούς θα εισαγάγει.
 - ο Δημιουργούμε πίνακα arr[n] (Variable Length Array).
2. Εισαγωγή δεδομένων
 - ο Με for, γεμίζουμε τον πίνακα με τιμές από το πληκτρολόγιο.
3. Υπολογισμός αθροίσματος και μέγιστου
 - ο Σε κάθε επανάληψη προσθέτουμε το στοιχείο στο sum.
 - ο Με if ελέγχουμε αν είναι μεγαλύτερο από το τρέχον max.

4. Υπολογισμός μέσου όρου
 - Μετατρέπουμε το άθροισμα σε double για δεκαδικό αποτέλεσμα.
5. Εκτύπωση αποτελεσμάτων
 - Εμφανίζουμε τα συνολικά στατιστικά του πίνακα.

Άσκηση 3 : Αναζήτηση συγκεκριμένου στοιχείου σε πίνακα

```
#include <stdio.h>

int main() {
    int arr[10];
    int search, found = 0;

    printf("Δώσε 10 αριθμούς:\n");
    for (int i = 0; i < 10; i++)
        scanf("%d", &arr[i]);

    printf("Ποιον αριθμό ψάχνεις; ");
    scanf("%d", &search);

    for (int i = 0; i < 10; i++) {
        if (arr[i] == search) {
            printf("Ο αριθμός βρέθηκε στη θέση %d\n", i);
            found = 1;
        }
    }

    if (!found)
        printf("Ο αριθμός δεν βρέθηκε.\n");

    return 0;
}
```

Ανάλυση τμημάτων

1. Εισαγωγή πίνακα

Χρησιμοποιούμε for για να διαβάσουμε 10 αριθμούς.

2. Αναζήτηση στοιχείου

- Ο χρήστης δίνει ποιον αριθμό ψάχνει.
- Ο βρόχος συγκρίνει κάθε στοιχείο με το search.

3. Έλεγχος με εμφωλευμένο if

- Αν βρεθεί, εκτυπώνεται η θέση και θέτουμε found = 1.
- Αν δεν βρεθεί κανένα ταίριασμα, εκτυπώνουμε κατάλληλο μήνυμα.

Άσκηση 4 : Εμφάνιση ζυγών και περιττών αριθμών από πίνακα

```
#include <stdio.h>

int main() {
    int arr[8];
    printf("Δώσε 8 αριθμούς:\n");
    for (int i = 0; i < 8; i++)
        scanf("%d", &arr[i]);

    printf("\nΖυγοί αριθμοί:\n");
    for (int i = 0; i < 8; i++) {
        if (arr[i] % 2 == 0)
            printf("%d ", arr[i]);
    }

    printf("\nΠεριττοί αριθμοί:\n");
    for (int i = 0; i < 8; i++) {
        if (arr[i] % 2 != 0)
            printf("%d ", arr[i]);
    }

    return 0;
}
```

Ανάλυση τμημάτων

1. Δήλωση πίνακα και είσοδος
Ζητάμε 8 αριθμούς και τους αποθηκεύουμε.
2. Διάκριση ζυγών/περιττών
 - ο Χρησιμοποιούμε δύο βρόχους for.
 - ο Ο πρώτος ελέγχει $\text{arr}[i] \% 2 == 0$ για ζυγούς.
 - ο Ο δεύτερος $\text{arr}[i] \% 2 != 0$ για περιττούς.
3. Εμφάνιση αποτελεσμάτων
Εκτυπώνουμε σε δύο ξεχωριστές λίστες.

Άσκηση 5 — Αντιστροφή πίνακα

```
#include <stdio.h>

int main() {
    int arr[6];
    printf("Δώσε 6 αριθμούς:\n");
    for (int i = 0; i < 6; i++)
        scanf("%d", &arr[i]);

    printf("\nΑντίστροφος πίνακας:\n");
    for (int i = 5; i >= 0; i--)
        printf("%d ", arr[i]);

    return 0;
}
```

Ανάλυση τμημάτων

1. Είσοδος δεδομένων

Διαβάζουμε 6 τιμές και τις βάζουμε στον πίνακα.

2. Αντιστροφή

- Η εκτύπωση ξεκινά από το τέλος ($i = 5$) προς την αρχή ($i \geq 0$).
- Δεν αλλάζουμε τη μνήμη του πίνακα, απλώς τον εμφανίζουμε ανάποδα.

Άσκηση 6: Εύρεση δεύτερου μεγαλύτερου αριθμού

```
#include <stdio.h>

int main() {
    int arr[10];
    printf("Δώσε 10 αριθμούς:\n");
    for (int i = 0; i < 10; i++)
        scanf("%d", &arr[i]);

    int max1 = arr[0], max2 = arr[0];

    for (int i = 1; i < 10; i++) {
        if (arr[i] > max1) {
            max2 = max1;
            max1 = arr[i];
        } else if (arr[i] > max2 && arr[i] != max1) {
            max2 = arr[i];
        }
    }

    printf("Μεγαλύτερος αριθμός: %d\n", max1);
    printf("Δεύτερος μεγαλύτερος: %d\n", max2);

    return 0;
}
```

Ανάλυση τμημάτων

1. Εισαγωγή 10 τιμών
 - ο χρήστης δίνει τις τιμές του πίνακα.
2. Αρχικοποίηση μεταβλητών
 - max1: ο μεγαλύτερος αριθμός.
 - max2: ο δεύτερος μεγαλύτερος.
3. Εμφωλευμένη λογική ελέγχου
 - Αν arr[i] είναι μεγαλύτερος από max1, ο παλιός max1 γίνεται max2, και ο νέος μπαίνει ως max1.
 - Αν δεν είναι ο μεγαλύτερος αλλά είναι μεγαλύτερος από τον max2 και διαφορετικός του max1, τότε γίνεται νέος max2.
4. Εκτύπωση
 - Τυπώνουμε και τους δύο αριθμούς.