



104

Αλγόριθμοι, Προγραμματισμός και Δομές Δεδομένων

Εργαστήριο – Κεφάλαιο 3 & Κεφάλαιο 4 Έξοδος δεδομένων, Τελεστές

Διδάσκουσα:
Σοφία Αλμπάνη



Κεφάλαιο 3: Άσκηση 3.1

3.1) Να γραφτεί πρόγραμμα σε γλώσσα **C** το οποίο:

Ζητά από τον χρήστη να εισάγει **έναν χαρακτήρα**, **έναν ακέραιο αριθμό** και **έναν δεκαδικό αριθμό (float)**.

Διαβάζει τις τιμές αυτές από το πληκτρολόγιο χρησιμοποιώντας την εντολή `scanf()`.

Εμφανίζει τις τιμές που δόθηκαν στην οθόνη με την εντολή `printf()`, χρησιμοποιώντας κατάλληλους **μορφοδείκτες** (`%c`, `%d`, `%f`).

Στόχος

Εξάσκηση στη χρήση της εντολής **`scanf()`** για είσοδο δεδομένων διαφορετικών τύπων.

Εξοικείωση με τους **μορφοδείκτες**:



Ενδεικτική Λύση 3.1

```
#include <stdio.h>
int main()
{
    char ch;
    int i;
    float f;

    printf("Enter character, int and float: ");
    scanf("%c%d%f", &ch, &i, &f);

    printf("Char:%c \n", ch);
    printf("Int:%d \n", i);
    printf("Float:%f \n", f);

    //printf("\nC:%c\tI:%d\tF:%f\n", ch, i, f);
    return 0;
}
```



Άσκηση 3.2

3.2) Τροποποίησε την άσκηση 2.4 ώστε να παίρνει τιμές από το πληκτρολόγιο για τους 2 ακέραιους και να υπολογίζει το άθροισμα, τη διαφορά, το γινόμενο, το πηλίκο της διαίρεσής τους, και το υπόλοιπο της ακέραιας διαίρεσης .



Ενδεικτική Λύση 3.2

```
#include <stdio.h>
int main()
{
    int i, j;

    printf("Enter 2 integers (the second should not be 0):");
    scanf("%d%d", &i, &j);

    printf("Sum:%d Diff:%d Prod:%d Div:%f\n", i+j, i-j, i*j, (double)i/j);
    return 0;
}
```



Άσκηση 3.3

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

Ζητά από τον χρήστη να εισάγει **δύο ακέραιους αριθμούς**.

Ανταλλάσσει τις τιμές των δύο αριθμών μεταξύ τους.

Εμφανίζει στην οθόνη τις νέες τιμές των αριθμών μετά την ανταλλαγή.

Στόχος

Κατανόηση της διαδικασίας **ανταλλαγής τιμών** μεταξύ δύο μεταβλητών



Ενδεικτική Λύση 3.3

```
#include <stdio.h>
int main(void)
{
    int i, j, tmp;

    printf("Enter numbers: ");
    scanf("%d%d", &i, &j);
    tmp = i;
    i = j;
    j = tmp;
    printf("%d %d\n", i, j);
    return 0;
}
```



Άσκηση 3.4

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

Ζητά από τον χρήστη να εισάγει **έναν ακέραιο αριθμό μεταξύ 100 και 999** (τριψήφιο αριθμό).

Αντιστρέφει τη σειρά των ψηφίων του αριθμού που δόθηκε.

Εμφανίζει τον νέο αριθμό στην οθόνη.

Στόχος

Εξάσκηση στη χρήση του **τελεστή υπολοίπου %** και της **ακέραιας διαίρεσης /**.

Κατανόηση του τρόπου **αντιστροφής ψηφίων** ενός τριψήφιου αριθμού.



Ενδεικτική Λύση 3.4

```
#include <stdio.h> int main(void)
{
    int i;

    printf("Enter number between 100 and 999: ");
    scanf("%d", &i);

    i = 100*(i%10) + 10*(i%100/10) + i/100;
    printf("%d\n", i);
    return 0;
}
```

Άσκηση 4.1

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

Εισάγει από το πληκτρολόγιο δύο ακέραιες μεταβλητές, π.χ. $a = 3$ και $b = 5$, καθώς και μία μεταβλητή c .

Υπολογίζει νέες τιμές για τις μεταβλητές χρησιμοποιώντας **λογικές εκφράσεις (συνθήκες)**:

$a = (a > 3) + (b \leq 5);$

$b = (a == 3) + ((b - 2) \geq 3);$

$c = (b != 1);$

Εμφανίζει στην οθόνη τις τιμές των a , b και c με την εντολή `printf()`.

Στόχος

- Να κατανοήσετε τη λειτουργία των **λογικών τελεστών σύγκρισης** ($>$, $<$, $==$, $!=$, $\leq$, $\geq$) στη C.
- Να παρατηρήσετε ότι κάθε λογική έκφραση επιστρέφει:
 - 1 όταν η συνθήκη είναι **αληθής**,
 - 0 όταν είναι **ψευδής**.



Ενδεικτική Λύση 4.1

```
#include <stdio.h>
int main()
{
    int a, b, c;

    printf("Enter 2 integers:");
    scanf("%d %d", &a, &b);

    a = (a > 3) + (b <= 5);
    b = (a == 3) + ((b-2) >= 3);
    c = (b != 1);
    printf("%d %d %d\n", a, b, c);
    return 0;
}
```

Άσκηση 4.2

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

Δηλώνει και αρχικοποιεί τρεις ακέραιες μεταβλητές: `int a = 1, b = 4, c = 4;`

Υπολογίζει και εμφανίζει στην οθόνη τα αποτελέσματα δύο **λογικών εκφράσεων**:

1. `(a - 1) || (b-- == c)`
2. `!(a > b) && (a - 1 == b - 3) && (c % 2 || (a + b - c))`

Εμφανίζει τα αποτελέσματα των εκφράσεων ως **0 (ψευδές)** ή **1 (αληθές)**.

Στόχος

- Κατανόηση της λειτουργίας των **λογικών τελεστών**:
 - `||` (OR) → αληθές αν τουλάχιστον μία συνθήκη είναι αληθής
 - `&&` (AND) → αληθές μόνο αν όλες οι συνθήκες είναι αληθείς
 - `!` (NOT) → αντιστρέφει την τιμή της συνθήκης
- Εξάσκηση στη χρήση **σύνθετων λογικών εκφράσεων** και στη **σειρά εκτέλεσης** τους.
- Παρατήρηση του τρόπου που λειτουργεί το `b--` (μεταβολή της τιμής μετά τη χρήση).



Ενδεικτική Λύση 4.2

```
#include <stdio.h>
int main()
{
    int a = 1, b = 4, c = 4;

    printf("%d\n", (a-1) || (b-- == c));
    printf("%d", !(a > b) && (a-1 == b-3) && (c%2 || (a+b-c)));
    return 0;
}
```

Άσκηση 4.3

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

Ζητά από τον χρήστη να εισάγει το **εισόδημά του** (a).

Υπολογίζει τον **φόρο** (tax) σύμφωνα με τους παρακάτω κανόνες:

- Για εισόδημα έως **5000€**, **δεν επιβάλλεται φόρος**.
- Για εισόδημα από **5000€ έως 20000€**, ο φόρος είναι **15%** για το ποσό πάνω από τα 5000€.
- Για εισόδημα πάνω από **20000€**, ο φόρος είναι:
 - 15%** για το ποσό από 5000€ έως 20000€, και
 - 30%** για το ποσό πάνω από 20000€.

Εμφανίζει στην οθόνη τον υπολογισμένο φόρο με δύο δεκαδικά ψηφία.

Σημείωση

Ο υπολογισμός του φόρου υλοποιείται με **λογικές εκφράσεις (συνθήκες)**, π.χ.:

Για το b) $\text{tax} = (a > 5000 \ \&\& \ a \leq 20000) * (a - 5000) * 0.15$

Η συνθήκη $(a > 5000 \ \&\& \ a \leq 20000)$ επιστρέφει **1 αν είναι αληθής**, ή **0 αν είναι ψευδής**.

Στόχος

Εξάσκηση στη χρήση **λογικών τελεστών** (&&, >, <=) για αριθμητικούς υπολογισμούς.

Κατανόηση της έννοιας **συνθηκών μέσα σε εκφράσεις** (χωρίς if).



Ενδεικτική Λύση 4.3

```
#include <stdio.h>
int main()
{
    float tax, a;

    printf("Enter income: ");
    scanf("%f", &a);

    tax = (a > 5000 && a <= 20000) * (a - 5000) * 0.15 +
          (a > 20000) * ((a - 20000) * 0.3 + 15.000 * 0.15);
    printf("Tax = %.2f\n", tax);
    return 0;
}
```

Συνοπτικά

Κατηγορία	Συντακτικό	Σύντομη Περιγραφή
Είσοδος δεδομένων	<code>scanf ("%c%d%f", &ch, &i, &f);</code>	διαβάζει τιμές από το πληκτρολόγιο και τις αποθηκεύει σε μεταβλητές με χρήση % και &.
Τελεστές σύγκρισης	<code>>, <, >=, <=, ==, !=</code>	Επιστρέφουν 1 (αληθές) ή 0 (ψευδές).
Λογικοί τελεστές	<code>&&</code> (AND) <code> </code> (OR)	AND: Επιστρέφει 1 (αληθές) μόνο αν όλες οι συνθήκες είναι αληθείς. OR: Επιστρέφει 1 (αληθές) αν τουλάχιστον μία συνθήκη είναι αληθής.
Λογικές εκφράσεις σε πράξεις	<code>(a > 3) + (b <= 5)</code>	Οι λογικές εκφράσεις μπορούν να συμμετέχουν σε αριθμητικές πράξεις.
Τελεστής ανάθεσης	<code>=</code>	Αναθέτει τιμή σε μεταβλητή.
Δυαδικοί τελεστές (bitwise)	<code>&, </code>	Συγκρίνει bit προς bit δύο αριθμούς.
Χωρίς if – με λογικές εκφράσεις	<code>(a > 5000 && a <= 20000) * (a - 5000) * 0.15</code>	Υπολογισμός με συνθήκες χωρίς χρήση <code>if</code> .
Boolean αποτελέσματα	<code>αληθές → 1, ψευδές → 0</code>	Οι λογικές εκφράσεις επιστρέφουν αριθμητικές τιμές 0 ή 1.