



104

Αλγόριθμοι, Προγραμματισμός και Δομές Δεδομένων

Εργαστήριο – Κεφάλαιο 2

Τύποι δεδομένων, Δήλωση μεταβλητών, Έξοδος δεδομένων

Διδάσκουσα:

Σοφία Αλμπάνη



Άσκηση 2.1

2.1) Γράψτε ένα πρόγραμμα σε C που κάνει τα εξής:

- Δημιουργεί μια ακέραια μεταβλητή i και δίνει σε αυτήν την τιμή 100.
- Αλλάζει την τιμή της μεταβλητής i προσθέτοντάς την στον εαυτό της.
- Εκτυπώνει στην οθόνη:
 - Το διπλάσιο της νέας τιμής ($i + i$)
 - Την τρέχουσα τιμή της i

Η εκτύπωση πρέπει να γίνει με τη μορφή:

V1:... V2:...

Στόχος: Να κατανοήσετε πώς αλλάζει η τιμή μιας μεταβλητής και πώς να την εμφανίζετε στην οθόνη.



Ενδεικτική Λύση 2.1

```
#include <stdio.h>
int main()
{
    int i = 100;

    i = i+i;
    printf("V1:%d V2:%d\n", i+i, i);
    return 0;
}
```



Άσκηση 2.2

2.2) Γράψτε ένα πρόγραμμα σε C που:

- Δηλώνει δύο μεταβλητές:
 - Μια τύπου int με τιμή 30
 - Μια τύπου float με τιμή 10.65
- Εκτυπώσει και τις δύο μεταβλητές χρησιμοποιώντας την εντολή:

```
printf("%f %d\n", i, j);
```

Παρατηρήστε και εξηγήστε τι εκτυπώνεται στην οθόνη και γιατί.

Στόχος: Να κατανοήσετε τη σημασία των τύπων δεδομένων και τη χρήση των σωστών μορφοδεικτών (format specifiers) στο printf.



Ενδεικτική Λύση 2.2

```
#include <stdio.h>

int main()
{
    int i = 30;
    float j = 10.65;

    printf("%f %d\n", i, j);
    return 0;
}
```



Άσκηση 2.3

2.3) Να γραφτεί πρόγραμμα το οποίο ορίζεται μια ακτίνα (πραγματικός αριθμός) και να υπολογίζει την περίμετρο και το εμβαδόν του κύκλου.

Παράδειγμα χρήσης:

```
double a = 4 * M_PI;  
printf("%f", a);
```

Σημείωση: Η μαθηματική σταθερά $\pi=3.141592...$ στη C είναι ορισμένη στο αρχείο επικεφαλίδας (header file) *math.h*. Το όνομα της σταθεράς είναι *M_PI*. Στο ίδιο αρχείο είναι ορισμένες πολλές μαθηματικές συναρτήσεις. Για να μπορούμε να χρησιμοποιούμε πρέπει στην αρχή του κώδικα να συμπεριλάβουμε την εντολή *#include <math.h>*.

Στόχος: Χρήση της βιβλιοθήκης *<math.h>*.



Ενδεικτική Λύση 2.3

```
#include <stdio.h>
#include <math.h>

main() {

    double aktina = 2.5, perimetros, emvadon;

    perimetros = 2 * M_PI * aktina;
    emvadon = M_PI * aktina * aktina;

    //isodynamia to emvadon
    //emvadon = M_PI * pow(aktina, 2);
    //h synarthsh pow() ypologizei dunameis

    printf("H perimetros einai %f kai to emvadon %f", perimetros, emvadon);

}
```

Άσκηση 2.4

2.4) Γράψτε ένα πρόγραμμα σε C που:

Δηλώνει δύο ακέραιες μεταβλητές $i = 50$ και $j = 20$. Υπολογίζει και εκτυπώνει τα εξής:

- Το άθροισμα ($i + j$)
- Τη διαφορά ($i - j$)
- Το γινόμενο ($i * j$)
- Τη διαίρεση (i / j) ως δεκαδικό αριθμό (float ή double)
- Το υπόλοιπο της ακεράιης διαίρεσης ($i \% j$)

Χρησιμοποιήστε τη συνάρτηση `printf` και τους κατάλληλους μορφοδείκτες.

Στόχος: Να εξασκηθείτε στις βασικές αριθμητικές πράξεις σε C και στη χρήση σωστών τύπων δεδομένων και μορφοδεικτών για εκτύπωση.



Ενδεικτική Λύση 2.4

```
#include <stdio.h>
int main()
{
    int i = 50, j = 20;

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



Άσκηση 2.5

2.5) Να γραφτεί πρόγραμμα το οποίο υπολογίζει το εμβαδόν τριγώνου όταν δίνονται η βάση και το ύψος, καθώς και πρόγραμμα που υπολογίζει την υποτείνουσα ορθογωνίου τριγώνου όταν δίνονται οι κάθετες πλευρές του.

Σημείωση: Η τετραγωνική ρίζα υλοποιείται στη C με τη μαθηματική συνάρτηση `sqrt()`. Για να γίνει χρήση της συνάρτησης αυτής πρέπει στην αρχή του κώδικα να συμπεριλάβουμε την εντολή `#include <math.h>`.

Παράδειγμα χρήσης:

```
double a = sqrt(6);  
printf("%f", a);
```

Στόχος: Χρήση μαθηματικής συνάρτησης `sqrt()`.



Ενδεικτική Λύση 2.5

```
#include <stdio.h>
#include <math.h>    // Για τη συνάρτηση sqrt()

int main() {
    double base = 8.0;
    double height = 5.0; double area;

    area = (base * height) / 2;
    printf("Εμβαδον με basi %f kai upsos %f: %f\n", base, height, area);

    double a = 3.0;    // πρώτη κάθετη πλευρά
    double b = 4.0;    // δεύτερη κάθετη πλευρά
    double hypotenuse;

    hypotenuse = sqrt(a*a + b*b);
    printf("Υποτινuse trigwnou me pleures %f kai %f: %f\n", a, b, hypotenuse);
    return 0;
}
```

Άσκηση 2.6

2.6) Να γραφεί ένα πρόγραμμα σε γλώσσα **C** που:

- Δηλώνει δύο μεταβλητές κινητής υποδιαστολής (float) με τιμές $i = 10.9$ και $j = 20.3$ και μια ακέραια μεταβλητή k .
- Υπολογίζει την τιμή της παράστασης $(int)i + j$ και την αποθηκεύει στη μεταβλητή k .

Εμφανίζει στην οθόνη:

- την τιμή του k
- και την τιμή της παράστασης $(int)(i + (int)j)$

Χρησιμοποιήστε τη συνάρτηση `printf` με τον κατάλληλο μορφοδείκτη και να εξηγήσετε τη διαφορά μεταξύ των δύο αποτελεσμάτων που εμφανίζονται.

Σκοπός:

Να κατανοήσετε πώς λειτουργεί η **μετατροπή τύπων (type casting)** και πώς επηρεάζει το αποτέλεσμα αριθμητικών πράξεων στη C.



Ενδεικτική Λύση 2.6

```
#include <stdio.h>

int main()
{
    int k;
    float i = 10.9, j = 20.3;

    k = (int)i + j;
    printf("%d %d\n", k, (int)(i + (int)j));
    return 0;
}
```



Άσκηση 2.7

2.7) Να γραφεί ένα πρόγραμμα σε γλώσσα **C** που:

Ορίζει έναν ακέραιο αριθμό των 2 ψηφίων και να υπολογίζει το άθροισμα των ψηφίων του.

Παράδειγμα: αν ο αριθμός είναι το 35 τότε το αποτέλεσμα θα είναι 8 (3+5).

Σκοπός:

Να εξασκηθείτε στη χρήση ακέραιας διαίρεσης και υπολοίπου, καθώς και στον τρόπο εξαγωγής των ψηφίων ενός ακεραίου αριθμού στη C.



Ενδεικτική Λύση 2.7

```
#include <stdio.h>

int main()
{
    int i, j, k;

    i = 35;
    j = i/10;
    k = i - (10*j);
    printf("Sum = %d\n", j+k);
    return 0;
}
```



Άσκηση 2.8

2.8) Να γραφεί ένα πρόγραμμα σε γλώσσα **C** που:

Δηλώνει δύο μεταβλητές κινητής υποδιαστολής i και j .

Αλλάζει το δεκαδικό τους μέρος.

Εκτυπώνει στην οθόνη τις νέες τιμές των i και j .

Παράδειγμα: Αν οι αριθμοί είναι $i = 3.45$ και $j = 6.78$ θα τυπώσει 6.45 και 3.78.

Βοήθεια: Δηλώστε μία ακέραια μεταβλητή k και μία βοηθητική μεταβλητή tmp τύπου `float`.

Σκοπός:

Να κατανοήσετε τη διάσπαση ενός δεκαδικού αριθμού σε **ακέραιο και δεκαδικό μέρος**, καθώς και τον τρόπο μετατροπής τύπων δεδομένων (casting) στη C.



Ενδεικτική Λύση 2.8

```
#include <stdio.h>
int main()
{
    int k;
    float i = 3.45, j = 6.78, tmp;

    tmp = i - (int)i; /* Βρίσκουμε το δεκαδικό μέρος. */
    k = (int)i; /* Αποθηκεύουμε το υπάρχον ακέραιο μέρος, πριν αλλάξει η
                τιμή του. */
    i = (int)j + tmp;

    tmp = j - (int)j;
    j = k + tmp;
    printf("%f %f\n", i, j);
    return 0;
}
```

Συνοπτικά

Κατηγορία	Συντακτικό / Εντολή	Σύντομη Περιγραφή
Εκτύπωση	<code>printf("");</code>	Εκτυπώνει τιμές στην οθόνη.
Προσδιοριστικό μετατροπής	<code>printf("%d, %f", i, j);</code>	%d για ακέραιους (int), %f για δεκαδικούς (float/double)
Μεταβλητές	<code>int i; float j;</code>	Δήλωση ακέραιων (int) και δεκαδικών (float/ double) μεταβλητών
Αρχικοποίηση	<code>i = 10; j = 3.5;</code>	Ανάθεση αρχικών τιμών
Αριθμητικές πράξεις	<code>+ - * / %</code>	Πρόσθεση, αφαίρεση, πολλαπλασιασμός, διαίρεση, υπόλοιπο (μόνο για ακέραιους)
Μετατροπή τύπων	<code>(int) i, (double) j</code>	Casting για αλλαγή τύπου μεταβλητής
Αρχείο επικεφαλίδας (header file) math.h	<code>#include <math.h></code>	Τετραγωνική ρίζα: <code>sqrt()</code> Αριθμός π: <code>M_PI</code> Δυνάμεις: <code>pow()</code>