



Κεφάλαιο 2.6:

Είσοδος / Έξοδος Δεδομένων, Μορφοποίηση Δεδομένων Εξόδου

(Διάλεξη 7)

Διδάσκων: Δημήτρης Ζεϊναλιπούρ

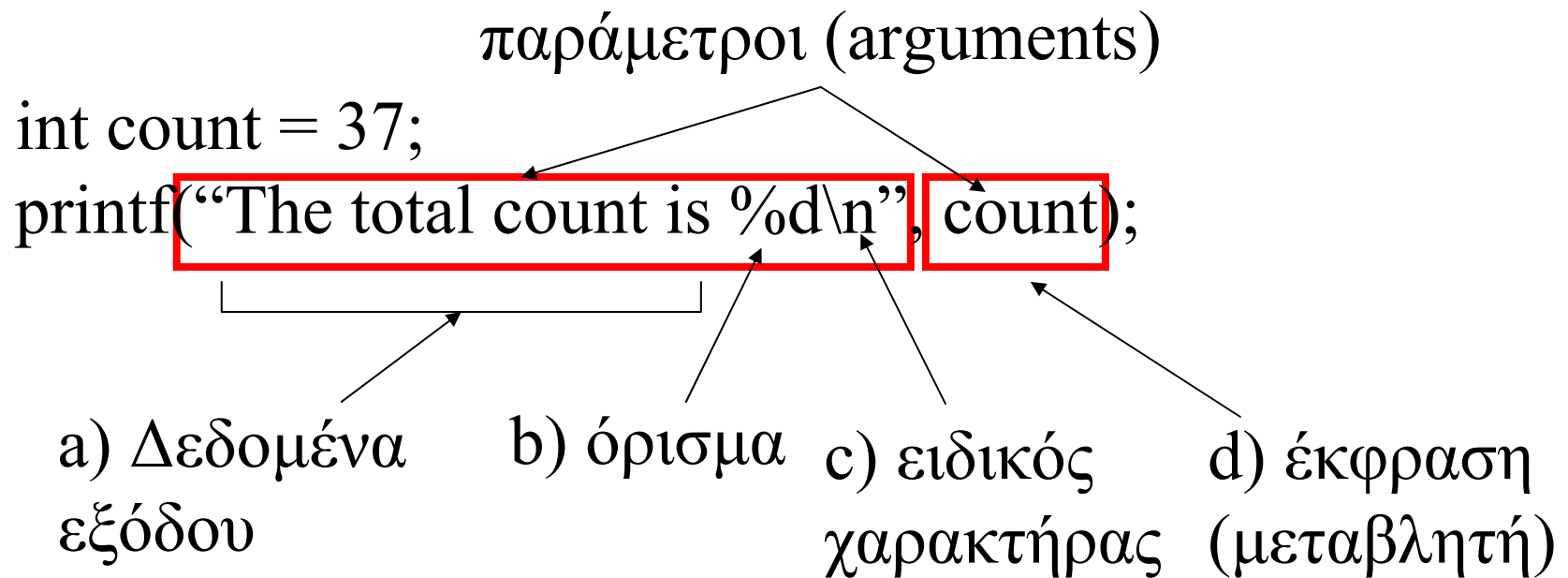
Είσοδος/ Έξοδος



- Σε σχεδόν όλα τα προγράμματα πρέπει να πάρουμε κάποια δεδομένα και να δώσουμε αποτέλεσμα
 - **Συνάρτηση εισόδου** – αντιγράφει δεδομένα από μονάδα εισόδου σε χώρο στην μνήμη, πχ scanf,
 - **Συνάρτηση εξόδου** – εκτυπώνει σε μονάδα εξόδου πληροφορίες αποθηκευμένες στην μνήμη, πχ printf,
 - Η βιβλιοθήκη **stdio** περιέχει βασικές συναρτήσεις εισόδου/εξόδου

Δεδομένα Εξόδου - Εντολή printf

- Σύνταξη:
 - printf(μορφή εξόδου);
 - printf(μορφή εξόδου, λίστα εκφράσεων);



a) Δεδομένα εξόδου



- Σύνταξη:
 - Ξεκινά με “ και τελειώνει με ”
 - Περιλαμβάνει: **strings** (σειρές από χαρακτήρες), ορίσματα, ειδικούς χαρακτήρες
- Σημασία:
 - τυπώνει ως έχουν τα strings
 - ορίσματα και ειδικοί χαρακτήρες...



b) Ορίσματα printf

- Σύνταξη: **%d %c %f %e %s**
- Σημασία: αντικατάστησε όρισμα με τιμή έκφρασης από τη λίστα εκφράσεων
 - %d ακέραια τιμή (int) (int => 4 byte => [-2G..2G])
 - %f κινητή υποδιαστολή μονής ακρίβειας (float)
 - %lf ή %e κινητή υποδιαστολή διπλής ακρίβειας (double)
 - %c ένας χαρακτήρας (char)
 - %s σειρά από χαρακτήρες (string)
- Αριθμός ορισμάτων ίδιος με το μέγεθος λίστας εκφράσεων και σειρά και τύποι να ταιριάζουν

a) Παραδείγματα Ορισμάτων 1



```
printf("to EPL033 exei %d foitites\n", 74);
```

```
int n_students = 100;
```

```
printf("to EPL033 exei %d foitites\n", n_students);
```

```
float average;
```

```
printf("o mesos oros einai %f\n", average);
```

```
printf("simeio = (%f, %f, %f)", x, y, z);
```

b) a) Παραδείγματα Ορισμάτων 2



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

```
int main() {  
    printf ("Characters: %c %c \n", 'a', 65);  
    printf ("Decimals: %d %d\n", 1977, 2000000000);  
    printf ("Some different radices: %d %x %o \n", 100, 100, 100);  
    printf ("floats: %f %.2f \n", 3.1416, 3.1416);  
    printf ("%s \n", "A string");  
    return 0;  
}
```

%o = octal (οκταδικό)

**%x = hexadecimal
(δεκαεξαδικό)**

Στην οθόνη θα εκτυπωθούν τα εξής δεδομένα

Characters: a A

Decimals: 1977 2000000000

Some different radices: 100 64 144

floats: 3.141600 3.14

A string

βάση

c) Ειδικοί Χαρακτήρες (escape sequences)

- Σημασία:
 - \n μετάφερε τον δρομέα στην επόμενη γραμμή
 - \t μετάφερε τον δρομέα στην επόμενη δεξιά στήλη
 - \' εκτύπωσε \'
 - \\ εκτύπωσε \
- Παραδείγματα
 - printf (“hello world”);
 - printf (“hello world\n”);
 - printf (“hello world\nhello world\n”);
 - printf (“hello world\tI’m alive\n”);

d) Μεταβλητές (Παραδείγματα)



Παραδείγματα

- **Πράξεις κατευθείαν στην έκφραση**

```
int a = 1, b = 2, c=4;
```

```
printf("%d, %d, %d, %d, %d", a, b, a+b, a/b, a*b+c-a);
```

⇒ 1, 2, 3, 0, 5

- **Αυτόματο casting**

```
int a = 1, b = 2;
```

```
printf("%d, %d, %f", a, b, (float)a/b);    => 1, 2, 0.5
```

- **Garbage in => Garbage out**

Κατά λάθος `int a = '1'` αντί `int a = 1;`

```
printf("%d", a);    => εκτυπώνει ascii:'1' δηλαδή 49
```

Μορφοποίηση δεδομένων εξόδου τύπου Integer (integer formatting)

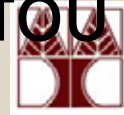


- Σύνταξη: **%nd**
- Σημασία:
 - χρησιμοποίηση **n** θέσεων για την εκτύπωση του ακέραιου αριθμού.
 - Εάν ο αριθμός έχει λιγότερα από **n** ψηφία, τοποθετούνται κενά στα αριστερά (δεξιά ευθυγράμμιση). Εάν ο αριθμός έχει περισσότερα ψηφία, εκτυπώνονται **όλα** **π.χ**

```
printf("\n%3d%3d\n\n%2d %3d", 1, 22, 33, -444);
```

| | | | | | | | |
|---|---|---|---|---|---|---|--|
| | | | | | | | |
| | | 1 | | 2 | 2 | | |
| | | | | | | | |
| 3 | 3 | | - | 4 | 4 | 4 | |

Μορφοποίηση δεδομένων εξόδου τύπου float (float formatting)



- Σύνταξη: **%n.mf**
- Σημασία:
 - χρησιμοποίηση τουλάχιστο **n** θέσεων για την εκτύπωση του αριθμού (συμπεριλαμβάνει την **'.'** και το **'–'**, αν χρειάζεστε)
 - Τα **m** ψηφία να είναι δεκαδικά
 - Εάν ο αριθμός έχει περισσότερα ψηφία, εκτυπώνονται **όλα**
- Σύνταξη: **%.mf** (m δεκαδικά ψηφία)
`printf("%4.2f %9.6f %3.2f", 4.4, 22.1, 466.00);`

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|--|---|---|---|---|---|---|---|---|---|--|---|---|---|---|---|---|
| 4 | . | 4 | 0 | | 2 | 2 | . | 1 | 0 | 0 | 0 | 0 | 0 | | 4 | 6 | 6 | . | 0 | 0 |
|---|---|---|---|--|---|---|---|---|---|---|---|---|---|--|---|---|---|---|---|---|

Προσοχή: Η στρογγυλοποίηση γίνεται κατευθείαν (π.χ. `%.2f` και `8.154` => `8.15`, ενώ `8.155` => `8.16`)

Είσοδος -- scanf



- Σύνταξη:
 - scanf(μορφή εισόδου, λίστα διευθύνσεων μεταβλητών);

```
int numbera, numberb;  
scanf("%d%d", &numbera, &numberb);
```

ορίσματα

διευθύνσεις
μεταβλητών

Μορφή Εισόδου



- Σύνταξη:
 - ξεκινά “ και τελειώνει με ”
 - Περιλαμβάνει: ορίσματα
- Σημασία:
 - τύπος ορισμάτων και σειρά τιμών που θα εισαχθούν
 - τύποι ανάλογοι με printf (πχ %d %f %e %c κτλ)
 - %d, %f και %e διαπερνούν white space

(δηλαδή αν ο χρήστης δώσει “ 023” τότε η μεταβλητή θα πετάξει τα κενά (whitespaces) => τιμή = 23.

Τελεστής διεύθυνσης και scanf



- Σύνταξη: `<όνομα μεταβλητής>`
- Σημασία: δίνει την διεύθυνση της μεταβλητής
- πχ `&numbera`

- `scanf` αποθηκεύει τιμές στις μεταβλητές χρησιμοποιώντας τις διευθύνσεις τους

- Όπως κάθε συνάρτηση στην C
 - Θα το δούμε σε λεπτομέρεια πιο μετά

Παράδειγμα scanf



```
int numbera, numberb;
```

```
printf("Enter two integer numbers: ");
```

```
scanf("%d%d",&numbera, &numberb);
```

```
printf("The two numbers entered are %d %d\n\n",  
       numbera, numberb);
```



Τύποι δεδομένων - Σύνοψη

| Τύπος δεδομένων | printf | scanf |
|-----------------|---------|-------|
| float | %f | %f |
| double | %f ή %e | %lf |
| int | %d | %d |
| char | %c | %c |

Περισσότερες Πληροφορίες

<http://www.cplusplus.com/ref/cstdio/printf.html>

<http://www.cplusplus.com/ref/cstdio/scanf.html>

Παράδειγμα



- Γράψετε πρόγραμμα που υπολογίζει και τυπώνει την περίμετρο ($2 \cdot \pi \cdot R$) και εμβαδό ($\pi \cdot R^2$) ενός κύκλου. Το μόνο δεδομένο είναι η ακτίνα (R) του κύκλου σε μέτρα.

Παράδειγμα



```
/* program that computes the perimeter and area of a circle*/  
#include <stdio.h>  
  
#define PI 3.14259    /*program constant*/  
int main()  
{  
    float radius;          /* variable for input data*/  
    float perimeter, area; /*variable for output data*/  
  
    /*GET INPUT DATA*/  
    printf("This program computes the area and perimeter of a circle\n");  
    printf("Enter radius of a circle in meters: ");  
    scanf("%f",&radius);  
  
    /*COMPUTE PERIMETER AND AREA*/  
    perimeter = 2 * PI * radius;  
    area = PI * radius * radius;  
  
    /*PRINT RESULTS*/  
    printf("A circle with radius %.2f m has perimeter %.2f m and area %.2f sq. m\n",  
           radius, perimeter, area);  
    return 0;  
}
```