



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

Τμήμα Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών
Πανεπιστημιούπολη Σερρών

Προγραμματισμός Ι (Θ)

Δρ. Δημήτρης Βαρσάμης
Αναπληρωτής Καθηγητής

Οκτώβριος 2019

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ Ι (Θ)

- 1 Μεταβλητές στην C
- 2 Έξοδος σε εφαρμογές κονσόλας
 - Συνάρτηση printf
 - Άλλες συναρτήσεις εξόδου (Output)
- 3 Είσοδος σε εφαρμογές κονσόλας
 - Συνάρτηση scanf
 - Άλλες συναρτήσεις εισόδου (Input)
- 4 Προβλήματα στη χρήση των εντολών I/O

- 1 Μεταβλητές στην C
- 2 Έξοδος σε εφαρμογές κονσόλας
 - Συνάρτηση printf
 - Άλλες συναρτήσεις εξόδου (Output)
- 3 Είσοδος σε εφαρμογές κονσόλας
 - Συνάρτηση scanf
 - Άλλες συναρτήσεις εισόδου (Input)
- 4 Προβλήματα στη χρήση των εντολών I/O

Μεταβλητές στην C

- Η μεταβλητή είναι μία θέση μνήμης για ένα δεδομένο. Η τιμή της μπορεί να είναι άγνωστη έως ότου εκτελεσθεί το πρόγραμμα.
- Για να ορίσουμε μια μεταβλητή στην C θα πρέπει να δηλώσουμε τον τύπο της μεταβλητής (`data_type`) και το όνομα της μεταβλητής αντίστοιχα.
- Η δήλωση των μεταβλητών γίνεται στην αρχή της συνάρτησης στην οποία χρησιμοποιούνται. Δηλαδή, στα πλαίσια του μαθήματος, στην αρχή της βασικής συνάρτησης `main()`.

Μεταβλητές στην C - Ονοματολογία

- Στην C τα ονόματα των μεταβλητών σχηματίζονται από:
 - ▶ τα γράμματα του λατινικού αλφαβήτου
 - ▶ τα ψηφία 0 έως 9
 - ▶ το χαρακτήρα υπογράμμισης (`_`), underscore.
- Το όνομα πρέπει να ξεκινά με γράμμα ή τον χαρακτήρα υπογράμμισης (στη δεύτερη περίπτωση ο επόμενος χαρακτήρας πρέπει να είναι μικρό γράμμα).
- Το όνομα δεν πρέπει να είναι ίδιο με δεσμευμένη λέξη.
- Σημαντικοί είναι μόνο οι πρώτοι 31 χαρακτήρες του ονόματος. Οι υπόλοιποι δε λαμβάνονται υπόψη.

Μεταβλητές στην C - Ονοματολογία

Ονόματα μεταβλητών

- Έγκυρα ονόματα μεταβλητών:
 - ▶ `totalArea`, `max_amount`, `counter1`
`Counter1`, `_temp_in_F`
- Μη έγκυρα ονόματα μεταβλητών:
 - ▶ `@product`, `total%`, `3rd`
- Απαράδεκτα ονόματα μεταβλητών:
 - ▶ `1`, `x2`, `maximum_number_of_students_in_my_class`

Μεταβλητές στην C - Τύποι Δεδομένων

- Υπάρχουν 4 βασικοί τύποι μεταβλητών στη γλώσσα C:

| Τύπος | Λέξη κλειδί στη C: |
|----------------|--------------------|
| Integer | int |
| Floating point | float |
| Double | double |
| Character | char |

- Η δήλωση των μεταβλητών γίνεται με πρόταση ορισμού (τελειώνει πάντοτε με ;)

Η δήλωση θα έχει τη μορφή:

```
|| data_type var1, var2, ... ;
```

Παράδειγμα:

```
|| int counter1, counter2;
```

Μεταβλητές στην C - Παράδειγμα

- Παράδειγμα δήλωσης μεταβλητών

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 |     int var1;
5 |     float var2;
6 |     system("PAUSE");
7 |     return 0;
8 | }
```

- Απλή δήλωση μεταβλητών, δεν έχει γίνει ανάθεση τιμών.

Μεταβλητές στην C - Παράδειγμα

- Παράδειγμα δήλωσης μεταβλητών

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 |     int var1;
5 |     float var2;
6 |     var1=10;
7 |     var2=3.14;
8 |     system("PAUSE");
9 |     return 0;
10| }
```

- Απλή δήλωση μεταβλητών, με χωριστή ανάθεση τιμών.

Μεταβλητές στην C - Παράδειγμα

- Παράδειγμα δήλωσης μεταβλητών

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 |     int var1=10;
5 |     float var2=3.14;
6 |     system("PAUSE");
7 |     return 0;
8 | }
```

- Απλή δήλωση μεταβλητών, με ταυτόχρονη ανάθεση τιμών.

Contents

- 1 Μεταβλητές στην C
- 2 Έξοδος σε εφαρμογές κονσόλας
 - Συνάρτηση printf
 - Άλλες συναρτήσεις εξόδου (Output)
- 3 Είσοδος σε εφαρμογές κονσόλας
 - Συνάρτηση scanf
 - Άλλες συναρτήσεις εισόδου (Input)
- 4 Προβλήματα στη χρήση των εντολών I/O

Έξοδος σε εφαρμογές κονσόλας

- Συναρτήσεις μη μορφοποιημένης εξόδου
 - ▶ `putchar`, `putc`, `puts`
- Συνάρτηση μορφοποιημένης εξόδου
 - ▶ `printf`

Συνάρτηση printf

Η συνάρτηση `printf` χρησιμοποιείται για να εμφανίζει **μορφοποιημένες** εκφράσεις ο υπολογιστής στην οθόνη.

- Στην πιο απλή μορφή συντάσσεται ως εξής `printf("printed area")` όπου `"printed area"` είναι η εκτυπώσιμη περιοχή η οποία μπορεί να είναι κάποιο αλφαριθμητικό (κείμενο).
- Στη πλήρη μορφή συντάσσεται ως εξής `printf("printed area", list of variables)` όπου `"printed area"` είναι η εκτυπώσιμη περιοχή και `list of variables` είναι η λίστα των μεταβλητών που θα εκτυπωθούν μέσα από την εκτυπώσιμη περιοχή.
 - ▶ Στην εκτυπώσιμη περιοχή θα πρέπει να υπάρχει ο χαρακτήρας `%` ο οποίος εκτυπώνει τιμή μεταβλητής.
 - ▶ Οι μεταβλητές στη `list of variables` χωρίζονται με κόμμα.

Συνάρτηση printf - Αλφαριθμητικά

- Απλό παράδειγμα εκτύπωσης απλού κειμένου

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 |     printf("Hello");
5 |     printf("  Hello");
6 |     printf(" Hello ");
7 |     printf("Hello");
8 |     system("PAUSE");
9 |     return 0;
10| }
```

Συνάρτηση printf - Αλφαριθμητικά

- ```
Hello Hello Hello Hello
```

- Προσοχή στα κενά, είναι εκτυπώσιμοι χαρακτήρες
- Η χρήση ξεχωριστής `printf` δεν σημαίνει αλλαγή γραμμής στην οθόνη (κονσόλα)

# Συνάρτηση printf - Ειδικοί Χαρακτήρες

- Οι ειδικοί χαρακτήρες που μπορούμε να χρησιμοποιήσουμε στην εκτυπώσιμη περιοχή, χωρίς να εκτυπώνονται οι ίδιοι, είναι

|                 |                        |
|-----------------|------------------------|
| <code>\n</code> | Αλλαγή γραμμής         |
| <code>\f</code> | Αλλαγή σελίδας         |
| <code>\t</code> | Οριζόντιος στηλοθέτης  |
| <code>\v</code> | Κατακόρυφος στηλοθέτης |
| <code>\a</code> | Καμπανάκι              |
| <code>\b</code> | Οπισθοχώρηση           |



# Συνάρτηση printf - Ειδικοί Χαρακτήρες \n

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | printf("Hello\n");
5 | printf(" Hello\n");
6 | printf(" Hello \n");
7 | printf("Hello\n");
8 | system("PAUSE");
9 | return 0;
10| }
```

```
Hello
 Hello
 Hello
Hello
```

# Συνάρτηση printf - Ειδικοί Χαρακτήρες \t

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | printf("Hello\t");
5 | printf("Hello\n");
6 | printf(" Hello\t");
7 | printf("Hello\n");
8 | system("PAUSE");
9 | return 0;
10| }
```

```
Hello Hello
Hello Hello
```

# Συνάρτηση printf - Ασκήσεις

- ❶ Να γίνει πρόγραμμα που να εμφανίζει το παρακάτω

```
*

*
*
```

- ❷ Να γίνει πρόγραμμα που να εμφανίζει το παρακάτω (χωρίς την χρήση κενών)

```
1 2 3
 4
5 6
```

# Προτεινόμενη λύση της άσκησης

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | printf(" *\\n");
5 | printf(" ***\\n");
6 | printf("*****\\n");
7 | printf(" *\\n");
8 | printf(" *\\n");
9 | system("PAUSE");
10 | return 0;
11 | }
```

# Προτεινόμενη λύση της άσκησης

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | printf("1 \t 2 \t 3\n");
5 | printf(" \t 4 \t \n");
6 | printf("5 \t \t 6\n");
7 | system("PAUSE");
8 | return 0;
9 | }
```

# Συνάρτηση printf - Μεταβλητές

Για να εκτυπωθεί η τιμή (ή το περιεχόμενο) μιας μεταβλητής με τη χρήση της συνάρτησης `printf` θα πρέπει να χρησιμοποιήσουμε το σύμβολο `%` μέσα στην εκτυπώσιμη περιοχή και να τοποθετήσουμε την μεταβλητή στη λίστα των μεταβλητών.

# Συνάρτηση printf - Μεταβλητές

Γενικά η σύνταξη είναι η ακόλουθη

```
%[width][.precision]type
```

όπου

- `[width]` είναι το πλήθος των χαρακτήρων που θα εκτυπωθούν. Αν η τιμή της μεταβλητής έχει λιγότερους χαρακτήρες τότε θα συμπληρωθούν με κενά. Αν η τιμή της μεταβλητής έχει περισσότερους χαρακτήρες τότε αγνοείται αυτή η ιδιότητα.
- `.[precision]` είναι το πλήθος των δεκαδικών ψηφίων που θα εκτυπωθούν.
- `type` είναι ο τύπος σύμφωνα με τον οποίο θα εκτυπωθεί η τιμή της μεταβλητής.

Οι ιδιότητες οι οποίες βρίσκονται μέσα σε αγκύλες ([]) είναι προαιρετικές

# Συνάρτηση printf - Μεταβλητές

Οι ειδικοί χαρακτήρες με τους οποίους καθορίζουμε τον τύπο σύμφωνα με τον οποίο θα εκτυπωθεί η τιμή μιας μεταβλητής και τους χρησιμοποιούμε στην εκτυπώσιμη περιοχή, **χωρίς να εκτυπώνονται οι ίδιοι**, είναι

|    |                              |
|----|------------------------------|
| %d | Δεκαδικό σύστημα             |
| %b | Δυαδικό σύστημα              |
| %o | Οκταδικό σύστημα             |
| %x | Δεκαεξαδικό σύστημα          |
| %i | Ακέραιος                     |
| %f | Κινητής υποδιαστολής         |
| %e | Εκθετικής μορφής             |
| %c | Χαρακτήρας                   |
| %s | Αλφαριθμητικό (συμβολοσειρά) |



# Συνάρτηση printf - Μεταβλητές

Εκτύπωση Τιμών Μεταβλητών - Χρήση του %

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | char x;
5 | int y;
6 | float z;
7 | x='a';
8 | y=6;
9 | z=7.8;
10 | printf("%c\n", x);
11 | printf("%d\n", y);
12 | printf("%f\n", z);
13 | system("PAUSE");
14 | return 0;
15 | }
```

# Συνάρτηση printf - Μεταβλητές

Εκτύπωση Τιμών Μεταβλητών - Χρήση του %

```
a
6
7.800000
```

# Συνάρτηση printf - Μεταβλητές

Εκτύπωση Τιμών Μεταβλητών με καθορισμένο πλάτος

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | int x, y, z;
5 | x=12345678;
6 | y=1;
7 | z=1234;
8 | printf("x=%10d\n", x);
9 | printf("y=%10d\n", y);
10 | printf("z=%10d\n", z);
11 | system("PAUSE");
12 | return 0;
13 | }
```

# Συνάρτηση printf - Μεταβλητές

Εκτύπωση Τιμών Μεταβλητών με καθορισμένο πλάτος

```
x= 12345678
y= 1
z= 1234
```

# Συνάρτηση printf - Μεταβλητές

Εκτύπωση Τιμών Μεταβλητών τύπου **float** με καθορισμένο αριθμό δεκαδικών ψηφίων

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main () {
4 | float x,y;
5 | x=1.23456789;
6 | y=12.345;
7 | printf ("x= %.4f\n", x);
8 | printf ("y= %.4f\n", y);
9 | system ("PAUSE");
10 | return 0;
11 | }
```

# Συνάρτηση printf - Μεταβλητές

Εκτύπωση Τιμών Μεταβλητών τύπου **float** με καθορισμένο αριθμό δεκαδικών ψηφίων

```
x= 1.2346
y= 12.3450
```

# Συνάρτηση printf - Μεταβλητές

Εκτύπωση Τιμών Μεταβλητών τύπου **float** με καθορισμένο πλάτος και με καθορισμένο αριθμό δεκαδικών ψηφίων

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main () {
4 | float x,y;
5 | x=1.23456789;
6 | y=12.345;
7 | printf ("x= %15.8f\n", x);
8 | printf ("y= %15.2f\n", y);
9 | system ("PAUSE");
10 | return 0;
11 | }
```

# Συνάρτηση printf - Μεταβλητές

Εκτύπωση Τιμών Μεταβλητών τύπου **float** με καθορισμένο πλάτος και με καθορισμένο αριθμό δεκαδικών ψηφίων

```
x= 1.23456788
y= 12.35
```



# Συνάρτηση printf - Μεταβλητές

Εκτύπωση Τιμών Μεταβλητών τύπου **char**

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | char x,y;
5 | x='a';
6 | y='A';
7 | printf("x= %c\n",x);
8 | printf("x= %d\n",x);
9 | printf("y= %c\n",y);
10 | printf("y= %d\n",y);
11 | system("PAUSE");
12 | return 0;
13 | }
```

# Συνάρτηση printf - Μεταβλητές

Εκτύπωση Τιμών Μεταβλητών τύπου **char**

```
x= a
```

```
x= 97
```

```
y= A
```

```
y= 65
```

# Συνάρτηση putchar

- Η συνάρτηση `putchar()` εμφανίζει στην οθόνη (`stdout`) τον χαρακτήρα που έχει ως όρισμα (π.χ. `c`), στην τρέχουσα θέση του δρομέα.
- Το πρωτότυπο της `putchar()` είναι το ακόλουθο:  
`int putchar(int c);`
- Το αρχείο κεφαλίδας της συνάρτησης `putchar()` βρίσκεται στο `stdio.h`.

# Συνάρτηση `putc`

- Η συνάρτηση `putc()` εμφανίζει στην έξοδο οριζόμενη από τον προγραμματιστή τον χαρακτήρα που έχει ως όρισμα (π.χ. `c`), στην τρέχουσα θέση του δρομέα.
- Το πρωτότυπο της `putc()` είναι το ακόλουθο:  
`int putc(int c, FILE *stream);`
- Η `putc(c, stdout)` είναι ίδια με την `putchar(c)`.
- Το αρχείο κεφαλίδας της συνάρτησης `putc()` βρίσκεται στο `stdio.h`.

# Contents

- 1 Μεταβλητές στην C
- 2 Έξοδος σε εφαρμογές κονσόλας
  - Συνάρτηση printf
  - Άλλες συναρτήσεις εξόδου (Output)
- 3 Είσοδος σε εφαρμογές κονσόλας
  - Συνάρτηση scanf
  - Άλλες συναρτήσεις εισόδου (Input)
- 4 Προβλήματα στη χρήση των εντολών I/O

# Είσοδος σε εφαρμογές κονσόλας

- Συναρτήσεις μη μορφοποιημένης εισόδου
  - ▶ `getchar`, `getc`, `gets`
- Συνάρτηση μορφοποιημένης εισόδου
  - ▶ `scanf`

# Συνάρτηση scanf

Η συνάρτηση `scanf` χρησιμοποιείται για να διαβάζει *μορφοποιημένες* εκφράσεις ο υπολογιστής από το πληκτρολόγιο και να τις τιμές σε μεταβλητές.

- Στη πλήρη μορφή συντάσσεται ως εξής `scanf("read area", list of variables)` όπου *"read area"* είναι η περιοχή που διαβάζει ο υπολογιστής και *list of variables* είναι η λίστα των μεταβλητών στις οποίες θα καταχωρηθούν οι τιμές που θα διαβάσει ο υπολογιστής από το πληκτρολόγιο
- Στην *"read area"* ορίζουμε την μορφοποίηση της εισόδου. Θα πρέπει να υπάρχει ο χαρακτήρας `%` ο οποίος διαβάζει την τιμή από το πληκτρολόγιο με συγκεκριμένο τύπο.
- Μπροστά από την μεταβλητή πρέπει να έχουμε τον τελεστή `&`.

# Συνάρτηση scanf - Αριθμοί

Απλή είσοδος αριθμού

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | int a;
5 | scanf("%d", &a);
6 | printf("Number = %d\n", a);
7 | system("PAUSE");
8 | return 0;
9 | }
```



# Συνάρτηση scanf - Αριθμοί

Απλή είσοδος αριθμού με μήνυμα

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | int a;
5 | printf("Give a Number : ");
6 | scanf("%d", &a);
7 | printf("Number = %d\n", a);
8 | system("PAUSE");
9 | return 0;
10| }
```

# Συνάρτηση scanf - Αριθμοί

Είσοδος πολλών αριθμών με μήνυμα

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | int a;
5 | float b;
6 | printf("Give the first Number : ");
7 | scanf("%d",&a);
8 | printf("Give the second Number : ");
9 | scanf("%f",&b);
10 | printf("First Number = %d\n",a);
11 | printf("Second Number = %f\n",b);
12 | system("PAUSE");
13 | return 0;
14 | }
```

# Συνάρτηση scanf - Αριθμοί

Είσοδος πολλών αριθμών με μήνυμα σε μια scanf με διαχωριστικό το κενό

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | int a;
5 | float b;
6 | printf("Give two Numbers : ");
7 | scanf("%d %f", &a, &b);
8 | printf("First Number = %d\n", a);
9 | printf("Second Number = %f\n", b);
10 | system("PAUSE");
11 | return 0;
12 | }
```

# Συνάρτηση scanf - Αριθμοί

Είσοδος πολλών αριθμών με μήνυμα σε μια scanf με διαχωριστικό το (,)

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | int a;
5 | float b;
6 | printf("Give two Numbers : ");
7 | scanf("%d,%f", &a, &b);
8 | printf("First Number = %d\n", a);
9 | printf("Second Number = %f\n", b);
10 | system("PAUSE");
11 | return 0;
12 | }
```

# Συνάρτηση scanf - Αριθμοί

Είσοδος πολλών αριθμών με μήνυμα σε μια scanf με διαχωριστικό το (;)

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | int a;
5 | float b;
6 | printf("Give two Numbers : ");
7 | scanf("%d;%f", &a, &b);
8 | printf("First Number = %d\n", a);
9 | printf("Second Number = %f\n", b);
10 | system("PAUSE");
11 | return 0;
12 | }
```

# Συνάρτηση scanf - Χαρακτήρες

Είσοδος χαρακτήρων

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | char ch;
5 | printf("Give a Character : ");
6 | scanf("%c", &ch);
7 | printf("Character = %c\n", ch);
8 | system("PAUSE");
9 | return 0;
10| }
```

# Συνάρτηση getchar

- Η συνάρτηση `getchar()` διαβάζει ένα χαρακτήρα από την κύρια είσοδο (`stdin`) και τον επιστρέφει στο πρόγραμμα.
- Το πρωτότυπο της `getchar()` είναι το ακόλουθο:  
`int getchar(void);`
- Το αρχείο κεφαλίδας της συνάρτησης `getchar()` βρίσκεται στο `stdio.h`.

# Συνάρτηση getchar

Είσοδος χαρακτήρων

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | char ch;
5 | printf("Give a Character : ");
6 | ch=getchar();
7 | printf("Character = %c\n",ch);
8 | system("PAUSE");
9 | return 0;
10| }
```



# Συνάρτηση `getc`

- Η συνάρτηση `getc()` διαβάζει ένα χαρακτήρα από μια οριζόμενη από τον προγραμματιστή είσοδο και τον επιστρέφει στο πρόγραμμα..
- Το πρωτότυπο της `getc()` είναι το ακόλουθο:  
`int getc(FILE *stream);`
- Η `getc(stdin)` είναι ίδια με την `getchar()`.
- Το αρχείο κεφαλίδας της συνάρτησης `getc()` βρίσκεται στο `stdio.h`.

# Contents

- 1 Μεταβλητές στην C
- 2 Έξοδος σε εφαρμογές κονσόλας
  - Συνάρτηση printf
  - Άλλες συναρτήσεις εξόδου (Output)
- 3 Είσοδος σε εφαρμογές κονσόλας
  - Συνάρτηση scanf
  - Άλλες συναρτήσεις εισόδου (Input)
- 4 Προβλήματα στη χρήση των εντολών I/O

# Προβλήματα στη χρήση των εντολών I/O

## Προβλήματα στην είσοδο (1)

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | char ch1, ch2;
5 | printf("Give a Character : ");
6 | scanf("%c", &ch1);
7 | printf("Give a Character : ");
8 | scanf("%c", &ch2);
9 | printf("Character = %c\n", ch1);
10 | printf("Character = %c\n", ch2);
11 | system("PAUSE");
12 | return 0;
13 | }
```

# Προβλήματα στη χρήση των εντολών I/O

## Προβλήματα στην είσοδο (2)

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | int num1, num2;
5 | char ch1, ch2;
6 | ch1=getchar();
7 | ch2=getchar();
8 | scanf("%d", &num1);
9 | scanf("%d", &num2);
10 | printf(" ch1=%c\n ch2=%c\n num1=%d\n num2=%d\n
 | n", ch1, ch2, num1, num2);
11 | system("PAUSE");
12 | return 0;
13 | }
```

# Προβλήματα στη χρήση των εντολών I/O

## Προβλήματα στην είσοδο (3)

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | int num1, num2;
5 | char ch1, ch2;
6 | scanf ("%d", &num1);
7 | scanf ("%d", &num2);
8 | ch1=getchar();
9 | ch2=getchar();
10 | printf (" num1=%d\n num2=%d\n ch1=%c\n ch2=%c\n
 | n", num1, num2, ch1, ch2);
11 | system ("PAUSE");
12 | return 0;
13 | }
```

# Προβλήματα στη χρήση των εντολών I/O

## Προβλήματα στην είσοδο (4)

```
1 | #include <stdio.h>
2 | #include <stdlib.h>
3 | int main() {
4 | char ch1, ch2, ch3, ch4;
5 | scanf ("%c", &ch1);
6 | scanf ("%c", &ch2);
7 | ch3=getchar();
8 | ch4=getchar();
9 | printf (" ch1=%c\n ch2=%c\n ch3=%c\n ch4=%c\n"
10 | , ch1, ch2, ch3, ch4);
11 | system ("PAUSE");
12 | return 0;
 | }
```