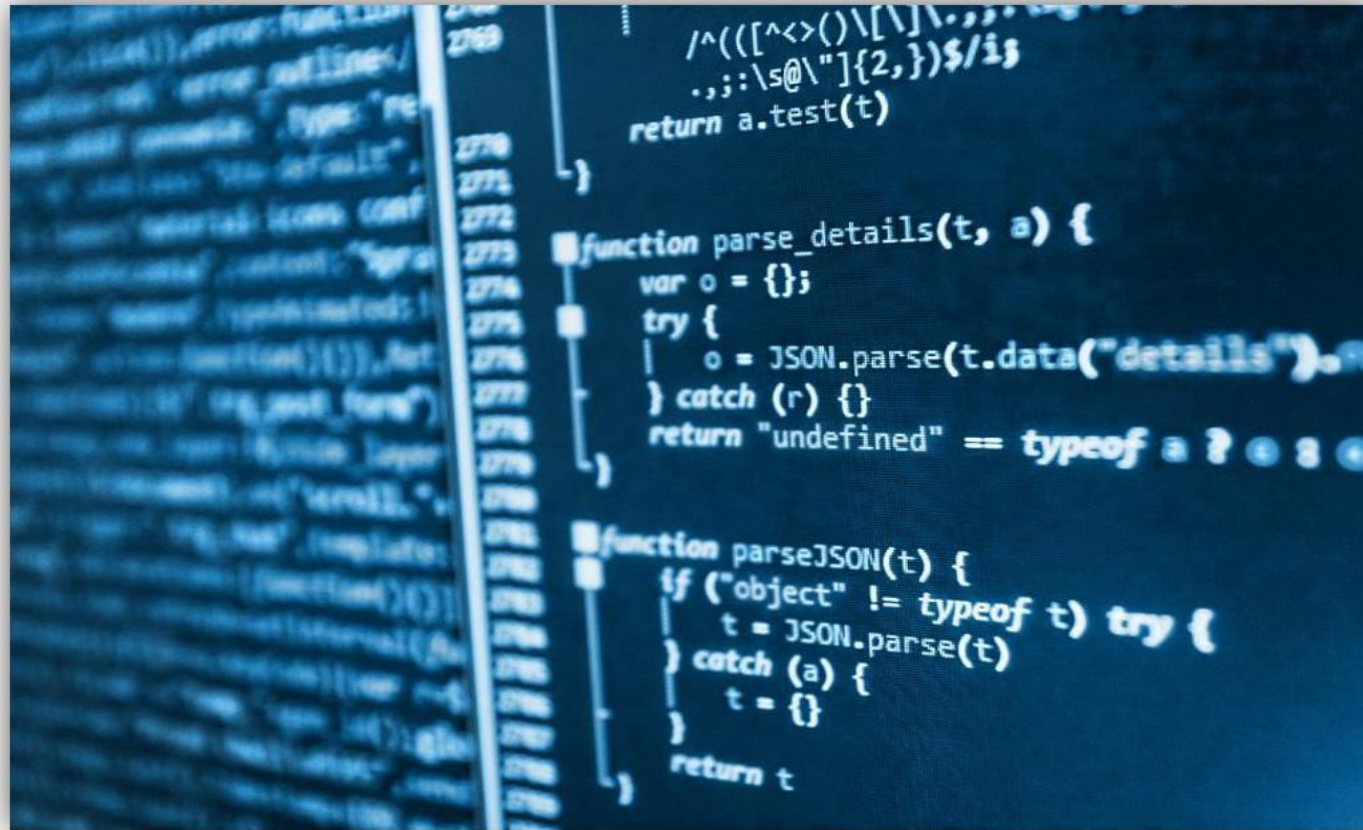


**ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΙΩΑΝΝΙΝΩΝ**



**ΤΜΗΜΑ  
ΠΛΗΡΟΦΟΡΙΚΗΣ &  
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ Ι



```
2769  / ^ ( ( [ ^ < > ( ) \ [ \ ] \ . , : ; \ s @ \ " ] { 2 , } ) $ / i ;  
2770  . , : ; \ s @ \ " ] { 2 , } ) $ / i ;  
2771  return a . test ( t )  
2772  }  
2773  function parse_details ( t , a ) {  
2774    var o = {} ;  
2775    try {  
2776      o = JSON . parse ( t . data ( " details " ) ) ;  
2777    } catch ( r ) {}  
2778    return " undefined " == typeof a ? a : o ;  
2779  }  
2780  function parseJSON ( t ) {  
2781    if ( " object " != typeof t ) try {  
2782      t = JSON . parse ( t )  
2783    } catch ( a ) {  
2784      t = {}  
2785    }  
2786    return t  
2787  }
```

# Ονόματα Μεταβλητών

- Κανόνες κατά τη δήλωση του ονόματος μίας μεταβλητής
  - Μπορεί να αποτελείται από:
    - Πεζά και κεφαλαία γράμματα του λατινικού αλφαβήτου
    - Ψηφία
    - Τον χαρακτήρα υπογράμμισης '\_' (underscore)
  - Ο πρώτος χαρακτήρας πρέπει να είναι γράμμα ή ο χαρακτήρας υπογράμμισης '\_'
  - Οι δεσμευμένες λέξεις της C απαγορεύεται να χρησιμοποιηθούν ως ονόματα μεταβλητών

auto	do	goto	signed	unsigned
break	double	if	sizeof	void
case	else	int	static	volatile
char	enum	long	struct	while
const	extern	register	switch	
continue	for	return	typedef	
default	float	short	union	

- Η γλώσσα C είναι case sensitive



# Δήλωση Μεταβλητών

- Για να χρησιμοποιήσετε μία μεταβλητή μέσα σε ένα πρόγραμμα πρέπει πρώτα να τη δηλώσετε
- Η δήλωση μίας μεταβλητής γίνεται με τον ακόλουθο τρόπο:

```
τύπος_δεδομένων όνομα_μεταβλητής;
```

- Το όνομα\_μεταβλητής είναι το τυχαίο όνομα που επιλέγει ο προγραμματιστής σύμφωνα με τους κανόνες
- Ο τύπος\_δεδομένων είναι ένας από τους αριθμητικούς τύπους δεδομένων που υποστηρίζει η γλώσσα C

# Τύποι Μεταβλητών

Τύπος	Συνηθισμένο μέγεθος (bytes)	Εύρος τιμών (min-max)	Ψηφία ακρίβειας
<code>char</code>	1	-128 ... 127	
<code>short</code>	2	-32.768 ... 32.767	
<code>int</code>	4	-2.147.483.648...2.147.483.647	
<code>long</code>	4	-2.147.483.648...2.147.483.647	
<code>float</code>	4	Χαμηλότερη θετική τιμή: $1.17 \cdot 10^{-38}$ Υψηλότερη θετική τιμή: $3.4 \cdot 10^{38}$	6
<code>double</code>	8	Χαμηλότερη θετική τιμή: $2.2 \cdot 10^{-308}$ Υψηλότερη θετική τιμή: $1.8 \cdot 10^{308}$	15
<code>unsigned char</code>	1	0 ... 255	
<code>unsigned short</code>	2	0 ... 65535	
<code>unsigned int</code>	4	0 ... 4.294.967.295	
<code>unsigned long</code>	4	0 ... 4.294.967.295	

# Παρατηρήσεις

- Πολλές μεταβλητές του ίδιου τύπου μπορούν να δηλωθούν στην ίδια γραμμή, αρκεί να διαχωρίζονται μεταξύ τους με κόμμα (,)

- Δηλαδή, αντί να δηλώσετε τις μεταβλητές a, b και c σε τρεις ξεχωριστές γραμμές:

```
int a;  
int b;  
int c;
```

μπορείτε να τις δηλώσετε σε μία γραμμή ως εξής:

```
int a, b, c;
```

- Η εκχώρηση μίας τιμής σε μία μεταβλητή γίνεται είτε μαζί με τη δήλωση της μεταβλητής είτε αργότερα
  - Π.χ. με την πρώτη εντολή δηλώνεται μία ακέραια μεταβλητή (`int`) με όνομα a και μετά της εκχωρείται η τιμή 100

```
int a;  
a = 100;
```

- Εναλλακτικά, θα μπορούσαμε να γράψουμε την εκχώρηση τιμής μαζί με τη δήλωση:

```
int a = 100;
```

# Παρατηρήσεις (II)

- Η τιμή που εκχωρείται σε μία μεταβλητή πρέπει να συμβαδίζει με τον τύπο της μεταβλητής

- Π.χ. με την εντολή:

```
int a = 10.9;
```

η τιμή της μεταβλητής `a` γίνεται `10`, γιατί η μεταβλητή `a` δηλώνεται σαν ακέραια μεταβλητή και όχι σαν πραγματική και το δεκαδικό μέρος αποκόπτεται (Προσοχή!! Δεν στρογγυλοποιείται)

- Η τιμή που εκχωρείται σε μία μεταβλητή πρέπει να είναι μέσα στο επιτρεπτό εύρος τιμών

- Π.χ. με την εντολή:

```
char ch = 130;
```

η τιμή της μεταβλητής `ch` δεν γίνεται `130`, γιατί το εύρος τιμών μίας μεταβλητής τύπου `char` είναι από `-128` έως `127`. Άρα, η τιμή `130` είναι μία τιμή εκτός των επιτρεπτών ορίων

# Παρατηρήσεις (III)

- Η τιμή μίας πραγματικής μεταβλητής μπορεί να είναι και ακέραια
  - Π.χ. επιτρέπεται να γράφουμε:  
`float a = 50;`  
γιατί είναι ισοδύναμο με:  
`float a = 50.0;`
- Η τιμή μίας πραγματικής μεταβλητής μπορεί να γραφεί και με επιστημονική σημειογραφία (συνήθως χρησιμοποιείται όταν η τιμή είναι πολύ μικρή ή πολύ μεγάλη)
  - Π.χ. αντί για  
`a = 0.085;`  
μπορούμε να γράφουμε  
`a = 85E-3;`
- Το γράμμα E ή e αναπαριστά το 10, ενώ ο αριθμός που το ακολουθεί είναι η θετική ή αρνητική δύναμη του 10.
- Δηλαδή, η έκφραση `85E-3` αντιστοιχεί στον αριθμό  $85 \cdot 10^{-3}$



# Σταθερές (I)

- Σταθερά ονομάζεται μία μεταβλητή που η τιμή της δεν μπορεί να αλλάξει μέσα στο πρόγραμμα
- Για να δηλωθεί μία μεταβλητή σαν σταθερά, πρέπει να προηγηθεί η λέξη `const` πριν από τον τύπο της μεταβλητής
- Επίσης, μαζί με τη δήλωση της σταθεράς, πρέπει να της εκχωρηθεί και μία αρχική τιμή, η οποία δεν θα μπορεί να αλλάξει μέσα στο πρόγραμμα

- Π.χ. με την επόμενη εντολή η ακέραια μεταβλητή `a` δηλώνεται σαν σταθερά και της εκχωρείται (μόνιμα) η τιμή 10

```
const int a = 10;
```

- Αν σε κάποιο σημείο του προγράμματος επιχειρήσουμε να της αλλάξουμε τιμή, π.χ. να γράψουμε:

```
a = 100;
```

τότε ο μεταγλωττιστής θα εμφανίσει μήνυμα λάθους για μη επιτρεπτή ενέργεια

## Σταθερές (II)

- Εναλλακτικός τρόπος για τη δήλωση μίας σταθεράς είναι η χρήση της οδηγίας `#define`, η οποία χρησιμοποιείται για τη δήλωση μακροεντολών
- Συνήθως, μία μακροεντολή αντιστοιχίζει ένα συμβολικό όνομα με κάποια αριθμητική τιμή
- Για τη δήλωση μακροεντολών, η οδηγία `#define` χρησιμοποιείται ως εξής:

```
#define όνομα_μακροεντολής τιμή
```

- Π.χ. η εντολή:

```
#define NUM 100
```

δηλώνει τη μακροεντολή με όνομα `NUM` και τιμή `100`

- Η `NUM` μπορεί να χρησιμοποιηθεί οπουδήποτε μέσα στο πρόγραμμα
- Ο μεταγλωττιστής όταν συναντάει τη `NUM` μέσα στο πρόγραμμα την αντικαθιστά με την τιμή `100`

# Παρατηρήσεις

- Οι δηλώσεις των μακροεντολών με την οδηγία `#define` είναι προτιμότερο να γίνονται πριν από τη συνάρτηση `main()`
- Τα ονόματα των μακροεντολών με την οδηγία `#define` είναι προτιμότερο να δηλώνονται με κεφαλαία γράμματα
- Στο τέλος της οδηγίας `#define` δεν μπαίνει ελληνικό ερωτηματικό (;)
- Π.χ.

```
#include <stdio.h>

#define NUM 100

int main()
{
    int a,b,c;
    a = 20 - NUM;
    b = 20 + NUM;
    c = 3*NUM;
    return 0;
}
```

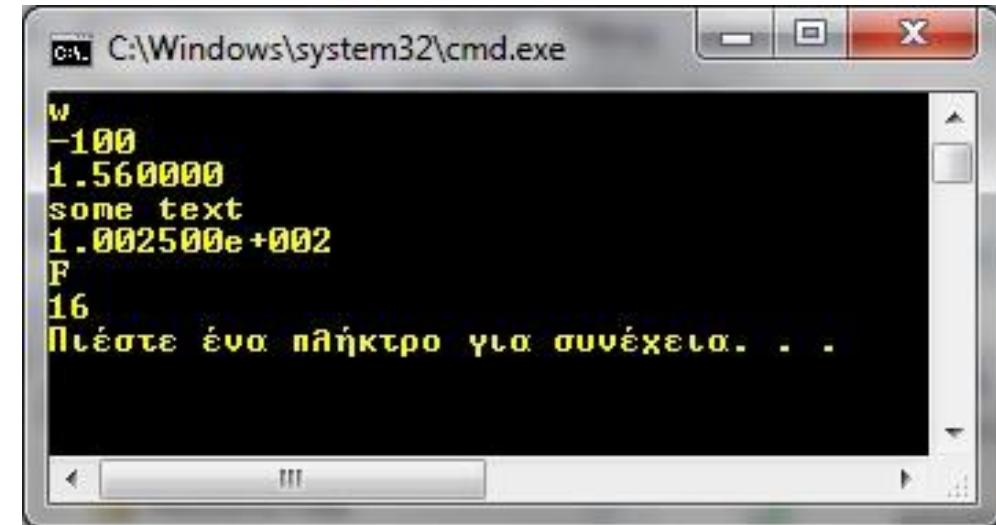
# Προσδιοριστικό Μετατροπής

- Ένα προσδιοριστικό μετατροπής (conversion specification) αρχίζει με τον χαρακτήρα % και ακολουθείται από έναν ή περισσότερους χαρακτήρες μετατροπής που έχουν ειδική σημασία

Χαρακτήρας μετατροπής	Σημασία
c	Χρησιμοποιείται για την εμφάνιση του χαρακτήρα που αντιστοιχεί σε μία ακέραια τιμή.
d ή i	Χρησιμοποιείται για την εμφάνιση ενός ακεραίου.
u	Χρησιμοποιείται για την εμφάνιση ενός μη-προσημασμένου ακεραίου.
f	Χρησιμοποιείται για την εμφάνιση ενός πραγματικού αριθμού. Η εξ' ορισμού ακρίβεια είναι έξι δεκαδικά ψηφία.
s	Χρησιμοποιείται για την εμφάνιση των χαρακτήρων ενός αλφαριθμητικού.
e ή E	Χρησιμοποιείται για την εμφάνιση ενός πραγματικού αριθμού σε επιστημονική μορφή. Ανάλογα με την επιλογή, εμφανίζεται το γράμμα e ή E πριν από τον εκθέτη.
g ή G	Χρησιμοποιείται για την εμφάνιση ενός πραγματικού αριθμού σε κανονική ή επιστημονική μορφή.
p	Χρησιμοποιείται για την εμφάνιση μίας διεύθυνσης μνήμης σε δεκαεξαδική μορφή.
x ή X	Χρησιμοποιείται για την εμφάνιση ενός μη-προσημασμένου ακεραίου σε δεκαεξαδική μορφή. Με το %x τα δεκαεξαδικά ψηφία (a-f) εμφανίζονται πεζά, ενώ με το %X εμφανίζονται ως κεφαλαία (A-F).
o	Χρησιμοποιείται για την εμφάνιση ενός μη-προσημασμένου ακεραίου σε οκταδική μορφή.
%	Χρησιμοποιείται για την εμφάνιση του χαρακτήρα %.

# Παραδείγματα (II)

```
#include <stdio.h>
int main()
{
    printf("%c\n", 'w');
    printf("%d\n", -100);
    printf("%f\n", 1.56);
    printf("%s\n", "some text");
    printf("%e\n", 100.25);
    printf("%X\n", 15);
    printf("%o\n", 14);
    return 0;
}
```

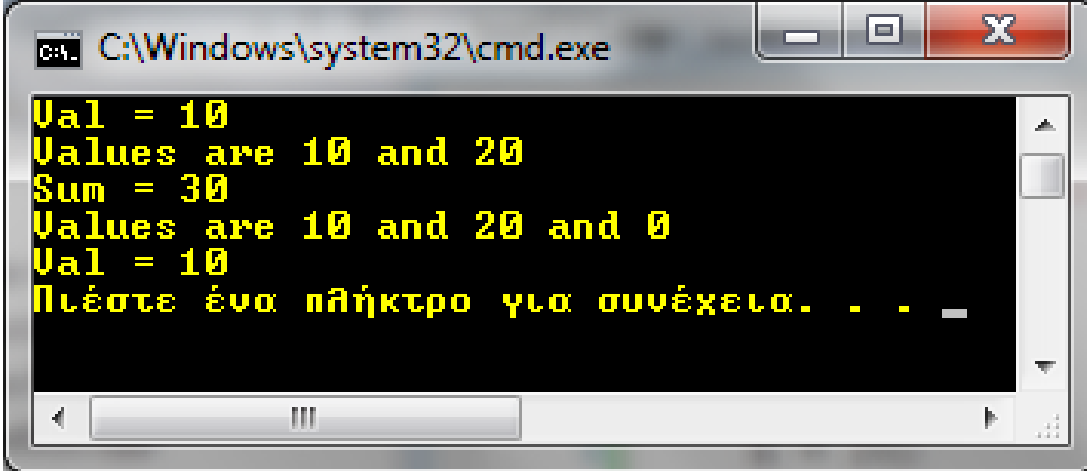


A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window has a black background with yellow text. The output of the program is displayed line by line: 'w', -100, 1.560000, some text, 1.002500e+002, F, 16, and Πιέστε ένα πλήκτρο για συνέχεια. . . . The window includes standard Windows window controls (minimize, maximize, close) in the top right corner and a scrollbar on the right side.



# Παράδειγμα

```
#include <stdio.h>
int main()
{
    int a,b;
    a = 10;
    b = 20;
    printf("Val = %d\n",a);
    printf("Values are %d and %d\n",a,b);
    printf("Sum = %d\n",a+b);
    printf("Values are %d and %d and %d\n",a,b);
    printf("Val = %d\n",a,b);
    return 0;
}
```



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window displays the output of the C program in yellow text on a black background. The output is as follows:

```
Val = 10
Values are 10 and 20
Sum = 30
Values are 10 and 20 and 0
Val = 10
Πιέστε ένα πλήκτρο για συνέχεια. . .
```

# Ασκήσεις

1. Βρείτε τα λάθη στο παρακάτω πρόγραμμα και να τα διορθώσετε, ώστε το πρόγραμμα να εκτελείται και να εμφανίζει την τιμή της μεταβλητής m.

```
include  
<stdio.h> int  
mein(void); (  
    int m;  
    a = 10  
    m = 2a+100  
    print("%f\n",  
    M); return 0;  
)
```

Λύση: [Solution](#)

## Ασκήσεις

2. Να κατασκευαστεί πρόγραμμα το οποίο θα αρχικοποιεί μία μεταβλητή με τον αριθμό 16.5 και να εκτυπώνει μήνυμα στην οθόνη (Program number is:16.5), χρησιμοποιώντας την μεταβλητή που είναι αποθηκευμένος ο αριθμός.

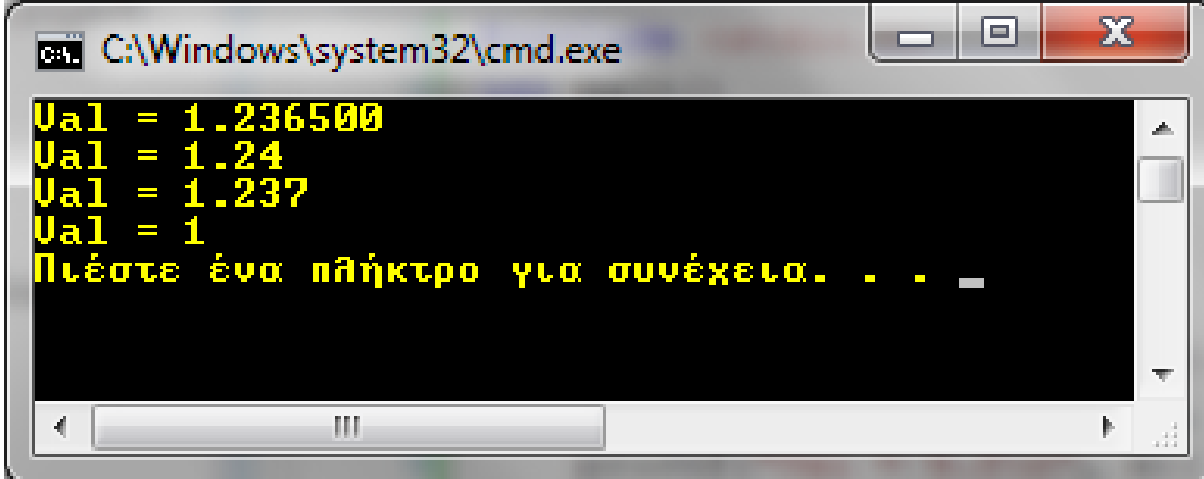
**Λύση:** [Solution](#)

3. Να κατασκευαστεί πρόγραμμα στο οποίο θα δηλώνονται 2 μεταβλητές,  $k=10$  και  $L=25$ . Το πρόγραμμα θα πρέπει να εμφανίζει μήνυμα στην οθόνη μήνυμα (To athroisma gia  $k=10$  kai  $L=25$  einai sum:35), χρησιμοποιώντας για τις πράξεις τις μεταβλητές που είναι αποθηκευμένοι οι αριθμοί. **Λύση:** [Solution](#)

# Παράδειγμα

```
#include <stdio.h>
int main()
{
    float a = 1.2365;

    printf("Val = %f\n", a);
    printf("Val = %.2f\n", a);
    printf("Val = %.3f\n", a);
    printf("Val = %.1f\n", a);
    return 0;
}
```



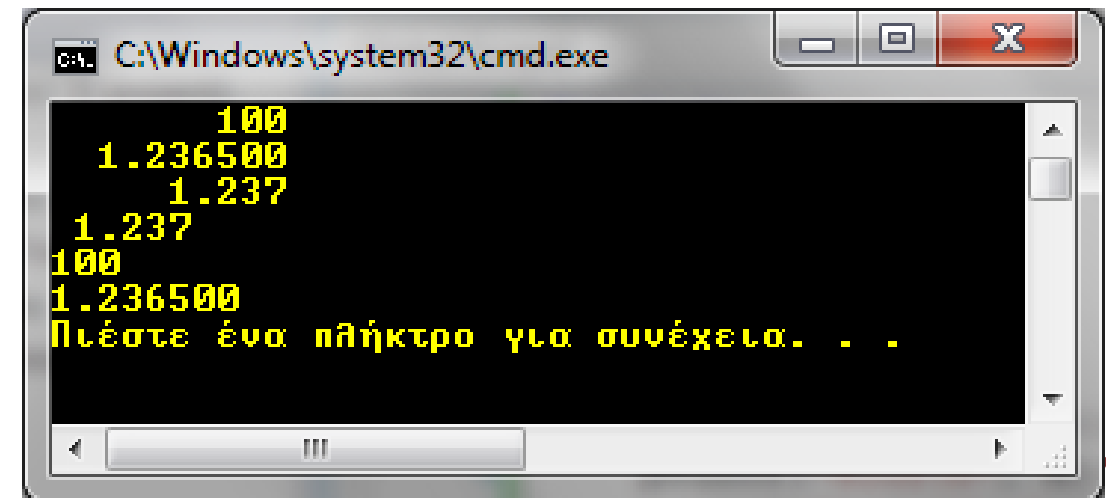
A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window has a black background with yellow text. The output of the program is displayed as follows:

```
Val = 1.236500
Val = 1.24
Val = 1.237
Val = 1
Πιέστε ένα πλήκτρο για συνέχεια. . .
```

# Παράδειγμα

```
#include <stdio.h>
int main()
{
    int a = 100;
    float b = 1.2365;

    printf("%10d\n", a);
    printf("%10f\n", b);
    printf("%10.3f\n", b);
    printf("%*.3f\n", 6, b);
    printf("%2d\n", a);
    printf("%6f\n", b);
    return 0;
}
```



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window has a black background with yellow text. The output of the program is displayed as follows:

```
          100
      1.236500
          1.237
      1.237
    100
    1.236500
Πιέστε ένα πλήκτρο για συνέχεια. . .
```