

Η ΓΛΩΣΣΑ C

Μάθημα 9:

Εισαγωγή στην Είσοδο/Εξοδο Δεδομένων

Δημήτρης Ψούνης



www.psounis.gr



Περιεχόμενα Μαθήματος

A. Έξοδος Δεδομένων

1. Η εντολή printf

1. Τρόποι Χρήσης
2. Ειδικοί Χαρακτήρες
3. Προσδιοριστές για Βασικούς Τύπους Δεδομένων
4. Χρήση Προσδιοριστών για Δεκαδικούς Αριθμούς
5. Συμβουλές
6. Τυπικό Συντακτικό της printf

B. Είσοδος Δεδομένων

1. Η εντολή scanf

1. Συντακτικό
2. Τρόποι Χρήσης

Γ. Ασκήσεις



A. Έξοδος Δεδομένων

1. Γενικά

- Γενικά με τον όρο «**έξοδος δεδομένων**» εννοούμε την διαδικασία που το πρόγραμμα μας προβάλλει τα δεδομένα που έχουν οι μεταβλητές του.
 - Δύο τρόπους θα μάθουμε:
 - Την έξοδο στην οθόνη.
 - Έχουμε ήδη δει την βασική χρήση της **printf** και στο μάθημα αυτό θα την δούμε πιο αναλυτικά, με όλες τις λεπτομέρειες της σύνταξής της.
 - Σε επόμενο μάθημα θα δούμε και άλλες βοηθητικές συναρτήσεις που κάνουν πιο εξειδικευμένες ενέργειες όπως η puts, η putc, κ.λπ.
 - Την έξοδο σε αρχείο του υπολογιστή
 - Όπου θα μάθουμε να αποθηκεύουμε τα δεδομένα που έχουν οι μεταβλητές μας σε αρχεία που είναι αποθηκευμένα στον υπολογιστή. Θα δούμε αναλυτικά αυτές τις ενέργειες σε επόμενο μάθημα.



A. Έξοδος Δεδομένων

2. Η εντολή printf

- Η εντολή printf είναι μια συνάρτηση βιβλιοθήκης, η οποία προβάλλει το περιεχόμενο της στην οθόνη.
- Μερικά Παραδείγματα είναι τα εξής:

```
printf("Eisagete arithmo metaksi 1 kai 100:");  
  
printf("Eisagete allon ena arithmo metaksi 1 kai 100:");  
  
printf("To ginomeno tou %d kai tou %d einai: %d\n",a,b,c);  
  
printf("\n\nPatiste Enter gia eksodo");
```

- Η printf είναι μια πάρα πολύ σημαντική συνάρτηση, ακριβώς διότι προβάλλει στην οθόνη τα μηνύματα που θέλουμε.
 - Θα δούμε διαφορετικούς τρόπους με τους οποίους μπορούμε να την συντάξουμε.

A. Έξοδος Δεδομένων

2. Η εντολή printf

1. Τρόποι Χρήσης

Όπως κάθε συνάρτηση, έτσι και η printf παίρνει ορίσματα.

- Στην 1^η χρήση της βλέπουμε πως λειτουργεί με ένα όρισμα:

```
printf("Συμβολοσειρά");
```

- όπου συμβολοσειρά είναι οποιοδήποτε κείμενο θέλουμε να προβάλλουμε στην οθόνη.

- **ΠΡΟΣΟΧΗ!** Μία συμβολοσειρά χαρακτηρίζεται από τα διπλά εισαγωγικά "...". Αν ξεχάσουμε να βάλουμε τα εισαγωγικά είτε στην αρχή είτε στο τέλος, ο μεταγλωττιστής θα διαμαρτυρηθεί.

- Μία συμβολοσειρά αποτελείται από χαρακτήρες (δηλαδή ένα-ένα τα γράμματα που την αποτελούν)
- Έχουμε δικαίωμα να εισάγουμε και ειδικούς χαρακτήρες που κάνουν μια συγκεκριμένη δουλειά
 - Τέτοιοι είναι και ο '\n' που λέγεται χαρακτήρας αλλαγής γραμμής διότι η εκτύπωση του προκαλεί την αλλαγή γραμμής στην οθόνη.

A. Έξοδος Δεδομένων

2. Η εντολή printf

1. Τρόποι Χρήσης

- Στην 2^η χρήση της βλέπουμε πως λειτουργεί με δύο ορίσματα (σημειώστε ότι ο μεταγλωττιστής αντιλαμβάνεται τα ορίσματα διότι αυτά είναι χωρισμένα με κόμμα):

```
printf("Συμβολοσειρά που έχει μέσα της %d",ακέραια μεταβλητή);
```

- Το 1^ο όρισμα είναι η συμβολοσειρά που θα εκτυπωθεί στην οθόνη που περιέχει μέσα της το %d. Το %d καλείται προσδιοριστής και στο σημείο όπου βρίσκεται θα εκτυπωθεί η τιμή της ακέραιας μεταβλητής που έχουμε ως δεύτερο όρισμα.
- Για παράδειγμα το τμήμα κώδικα:

```
int x;
x=1;
printf(" Η metabliti exei timi %d ",x)
```

x					
				1	

- Τυπώνει στην οθόνη

```
Η metabliti exei timi 1
```

A. Έξοδος Δεδομένων

2. Η εντολή printf

1. Τρόποι Χρήσης

- Στην 3^η χρήση της βλέπουμε πως λειτουργεί με τρία ορίσματα (και πάλι πρέπει να τα χωρίσουμε με κόμμα για να αντιληφθεί ο μεταγλωττιστής ότι πρόκειται για τρία ορίσματα)

```
printf("Συμβολοσειρά που έχει μέσα της 2 %d",ακέραιος1, ακέραιος2);
```

- Μέσα στην συμβολοσειρά τώρα έχουμε δύο %d. Το 1^ο %d θα αντικατασταθεί με την τιμή της μεταβλητή1 (2^ο όρισμα) και το 2^ο %d θα αντικατασταθεί με την τιμή της μεταβλητή2 (3^ο όρισμα)
- Για παράδειγμα το τμήμα κώδικα:

```
int x,y;
x=1;
y=6;
printf(" Oi times einai: %d kai %d ",y,x)
```

x				y	
			1		

- Τυπώνει στην οθόνη

```
Oi times einai: 6 kai 1
```

A. Έξοδος Δεδομένων

2. Η εντολή printf

1. Τρόποι Χρήσης

- Ο τρόπος που είδαμε στην σύνταξη της **printf** γενικεύεται, δηλαδή:
 - Μπορούμε στην συμβολοσειρά να έχουμε **τρία %d** ακολουθούμενα από τρία ακόμη ορίσματα που θα είναι ακέραιες μεταβλητές.
 - Μπορούμε στην συμβολοσειρά να έχουμε **τέσσερα %d** ακολουθούμενα από τέσσερα ακόμη ορίσματα που θα είναι ακέραιες μεταβλητές.
 - κ.ο.κ.
- Στο σημερινό μάθημα θα δούμε πως μπορούμε με την printf να εκτυπώσουμε και άλλους τύπους δεδομένων όπως πραγματικές μεταβλητές, συμβολοσειρές κ.λ.π.

A. Έξοδος Δεδομένων

2. Η εντολή printf

2. Ειδικοί Χαρακτήρες

- Είδαμε ήδη ότι όταν βάζουμε το 'ln' μέσα στην συμβολοσειρά, αλλάζει γραμμή στην οθόνη. Ο χαρακτήρας αυτός είναι ένας ειδικός χαρακτήρας που σημαίνει ότι προκαλεί μια συμπεριφορά και δεν τυπώνεται στην οθόνη.
- Υπάρχουν και άλλοι ειδικοί χαρακτήρες που μπορούμε να χρησιμοποιήσουμε μέσα στην συμβολοσειρά. Κοινό χαρακτηριστικό τους είναι ότι ξεκινούν με \ (backslash). Ας τους δούμε:

Ειδικός Χαρακτήρας	Συμπεριφορά
\n	Αλλαγή Γραμμής
\t	Tab (Μετακίνηση μερικές θέσεις δεξιά)
\a	Beep (Ο υπολογιστής κάνει beep)
\\	Εκτύπωση του \ (χαρακτήρας backslash)
\"	Εκτύπωση του " (χαρακτήρας διπλο εισαγωγικό)
\'	Εκτύπωση του ' (χαρακτήρας μονο εισαγωγικό)

A. Έξοδος Δεδομένων

2. Η εντολή printf

2. Ειδικοί Χαρακτήρες

- Μελετήστε το πρόγραμμα για την χρήση των συνήθων ειδικών χαρακτήρων

[illegible]

και εδώ βλέπουμε την έξοδο:

ANAFORA	
Atomo	Amoibi
Giorgos	100
Kostas	200
Eleni	150

Beep.
Beep.
Beep.

Telos Programmatos.

A. Έξοδος Δεδομένων

2. Η εντολή printf

3. Προσδιοριστές Βασικών Τύπων Δεδομένων

- Το %d το χρησιμοποιήσαμε μέσα σε μια συμβολοσειρά ώστε να το αντικαταστήσουμε με την τιμή μιας ακέрайας μεταβλητής.
- Κάθε τύπος δεδομένων, συνοδεύεται από τον προσδιοριστή του. Που σημαίνει ότι ανάλογα με τον τύπο δεδομένων που θέλουμε να τυπώσουμε, χρησιμοποιούμε και τον ανάλογο προσδιοριστή.

Προσδιοριστής	Τύπος δεδομένων που αντιστοιχεί
%d	int
%f	float, double
%c	char
%s	Συμβολοσειρά (πίνακας χαρακτήρων)
%ld	long
%u	unsigned int

- Στην πράξη θα χρειαστούμε τους 4 πρώτους.

A. Έξοδος Δεδομένων

2. Η εντολή printf

3. Προσδιοριστές Βασικών Τύπων Δεδομένων

- Μελετήστε το πρόγραμμα για την χρήση των προσδιοριστών:

```

/* prosd.c: Anadeikniei tin xrisi twn prosdioristwn */

#include <stdio.h>

int main()
{
    int x,y;
    float a,b;

    x=1;
    y=2;
    a=2.2;
    b=3.3;
    printf("\nAkeraioi Arithmoi: (%d+%d)/2=%d",x,y,(x+y)/2);
    printf("\nPragmatoi Arithmoi: (%f+%f)/2=%f",a,b,(a+b)/2);
}

```

- Πειραματιστείτε με τον κώδικα: Αν χρησιμοποιήσουμε λάθος προσδιοριστή, τότε τα αποτελέσματα θα είναι μη αναμενόμενα!!

A. Έξοδος Δεδομένων

2. Η εντολή printf

4. Χρήση Προσδιοριστών για δεκαδικούς αριθμούς

- Όπως είδαμε στο προηγούμενο παράδειγμα, η εκτύπωση ενός πραγματικού αριθμού γίνεται με τον προσδιοριστή %f και εξ'ορισμού τυπώνεται με 6 δεκαδικά ψηφία.
- Μας δίνεται η δυνατότητα όμως να προσδιορίσουμε ακριβώς το μέγεθος του χώρου και το πλήθος των δεκαδικών ψηφίων που θέλουμε να εμφανιστούν.
- Το συντακτικό του προσδιοριστή %f γίνεται:

```
%Ψ.Af
```

- Σημαίνει ότι θα εκτυπωθούν Ψ θέσεις (αν είναι λιγότερες συμπληρώνονται με κενά) και θα τυπωθούν Α δεκαδικά ψηφία (ακρίβεια Α δεκαδικών ψηφίων)
- Για παράδειγμα η εντολή:

```
printf("%8.3f", 1/3);
```

- Θα τυπώσει στην οθόνη:

```
0.333
```

- (Προσοχή ότι τυπώθηκαν 3 κενά αριστερά του αριθμού, το 0, η υποδιαστολή, 3 δεκαδικά, άρα σύνολο 8 θέσεις)

- Αρκετά χρήσιμο είναι το πλήθος των θέσεων και για την εκτύπωση ακεραίων αριθμών για μορφοποιημένη έξοδο στην οθόνη.

- Π.χ. ο προσδιοριστής %4d θα εκτυπώσει τον ακέραιο που ακολουθεί με συμπλήρωση κενών.

A. Έξοδος Δεδομένων

2. Η εντολή printf

4. Χρήση Προσδιοριστών για δεκαδικούς αριθμούς

- Μελετήστε το πρόγραμμα για την χρήση των προσδιοριστών:

```
/*width_precision.c: Deixnei tin epirooi ston prosdioristi tw n width+precision */
#include <stdio.h>

main()
{
    float f=10.0/7.0;
    int i,num;

    printf("\nPragmatikoi Arithmoi: ");
    printf("\n%8.3f", f);
    printf("\n%8.4f", f);
    printf("\n%8.5f", f);
    printf("\n%8.6f", f);

    printf("\n\nAkerairoi Arithmoi: ");
    num=1;
    for (i=1; i<=8; i++)
    {
        printf("\n|%-10d|%-10d|", num,num);
        num*=10;
    }
}
```

A. Έξοδος Δεδομένων

2. Η εντολή printf

5. Συμβουλές

- Μερικές συμβουλές για την printf

- Αλλαγή γραμμής γίνεται σε δύο περιπτώσεις:

- Όταν χρησιμοποιούμε τον χαρακτήρα \n
- Όταν δεν χωράει πλέον στην γραμμή της οθόνης οπότε αλλάζει αυτομάτως γραμμή!

- Καλό θα είναι να υιοθετήσουμε μια σύμβαση για τις αλλαγές γραμμής.

- Μια καλή πολιτική είναι να γράφουμε μια ξεχωριστή printf όποτε θέλουμε να αλλάξουμε γραμμή στην οθόνη, βάζοντας μια \n στην αρχή της συμβολοσειράς (Αυτή η σύμβαση έχει υιοθετηθεί στις σημειώσεις)

- Έχετε ως αναφορά όταν γράφετε κώδικα, τους προσδιοριστές για κάθε τύπο δεδομένων. Αν χρησιμοποιήσετε λάθος προσδιοριστή το πρόγραμμα θα εκτυπώσει τα δεδομένα με λάθος τρόπο!

A. Έξοδος Δεδομένων

2. Η εντολή printf

6. Τυπικό Συντακτικό της printf

- Το συντακτικό της printf είναι:

```
int printf ( const char * format, ... );
```

- Το πρώτο όρισμα είναι η συμβολοσειρά που τυπώνεται στην οθόνη και αν αυτό περιέχει προσδιοριστές τότε αυτοί αντικαθίστανται με μία κατάλληλη μορφοποίηση από τα ορίσματα που ακολουθούν με βάση τη σειρά τους.
- Η συνάρτηση επιστρέφει το πλήθος των χαρακτήρων που γράφηκαν στην οθόνη.
- Το συντακτικό των προσδιοριστών είναι:

```
%[flags][width][.precision][length]specifier
```

- Όπου flags: Μορφοποιεί τον αριθμό σε στοίχιση, πρόσημο κ.α.
- width: ελάχιστος αριθμός χαρακτήρων που τυπώνονται
- precision: Ακρίβεια Δεκαδικών Ψηφίων
- length: μετατροπές του τύπου δεδομένων
- specifier: προσδιοριστής

Λεπτομέρειες του μεταγλωττιστή:

Στην υποενότητα αυτή ξεφεύγουμε και βλέπουμε πλήρως το συντακτικό της printf. Σίγουρα δεν αποτελούν απαραίτητες πληροφορίες για κάποιον που μαθαίνει προγραμματισμό, αλλά η ύπαρξη του χρησιμεύει ως εγχειρίδιο αναφοράς.



A. Έξοδος Δεδομένων

2. Η εντολή printf

6. Τυπικό Συντακτικό της printf

- Τυπικοί προσδιοριστές της printf:

Προσδιοριστής	Τύπος δεδομένων που αντιστοιχεί
d or i	Προσημασμένος ακέραιος
u	Μη Προσημασμένος ακέραιος
o	Μη Προσημασμένος Οκταδικός
x	Μη Προσημασμένος 16δικός
X	Μη Προσημασμένος 16δικός (Κεφαλαία)
f	Πραγματικός
F	Πραγματικός (Κεφαλαία)
e	Επιστημονική Μορφή
E	Επιστημονική Μορφή (Κεφαλαία)
a	Πραγματικός Δεκαεξαδικός
A	Πραγματικός Δεκαεξαδικός (Κεφαλαία)
c	Χαρακτήρας
s	Πίνακας Χαρακτήρων
p	Διεύθυνση Δείκτη
%	Το % ακολουθούμενο από το %, τυπώνει το %



A. Έξοδος Δεδομένων

2. Η εντολή printf

6. Τυπικό Συντακτικό της printf

- flags: Τροποποιούν την εμφάνιση του αριθμού:

Flags	Ενέργεια Εκτύπωσης
-	Πραγματοποιεί αριστερή στοίχιση στον αριθμό
+	Τυπώνει το πρόσημο και για τους θετικούς αριθμούς
#	Χρησιμοποιούμενο με το o, x και X η τιμή που τυπώνεται θα έχει μπροστά 0, 0x και 0X αντίστοιχα
0	Γεμίζει εξ' αριστερών τον αριθμό με μηδενικά, σε συνδυασμό με τον προσδιοριστή width.

- width: Ελάχιστο πλήθος χαρακτήρων που τυπώνονται:

Width	Ενέργεια Εκτύπωσης
Ακέραιος Αριθμός	Ελάχιστος αριθμός δεκαδικών ψηφίων που θα τυπωθούν. Σε περίπτωση που ο αριθμός είναι μικρότερος συμπληρώνεται με κενά και ακολουθείται δεξιά στοίχιση.
*	Το πλάτος δεν προσδιορίζεται με αριθμό, αλλά με όρισμα που ακολουθεί την συμβολοσειρά εκτύπωσης.

- precision: Ακρίβεια Δεκαδικών Ψηφίων

- Αντίστοιχα με το width, είτε βάζουμε αριθμό δεκαδικών ψηφίων, είτε βάζουμε αστέρι και το πλήθος των ψηφίων καθορίζονται με το επόμενο όρισμα.



A. Έξοδος Δεδομένων

2. Η εντολή printf

6. Τυπικό Συντακτικό της printf

- length: Αλλάζει τον τύπο δεδομένων της εκτύπωσης

	specifiers						
length	d i	u o x X	f F e E g G a A	c	s	p	n
(none)	int	unsigned int	double	int	char*	void*	int*
hh	signed char	unsigned char					signed char*
h	short int	unsigned short int					short int*
l	long int	unsigned long int		wint_t	twchar_t*		long int*
ll	long long int	unsigned long long int					long long int*
j	intmax_t	uintmax_t					intmax_t*
z	size_t	size_t					size_t*
t	ptrdiff_t	ptrdiff_t					ptrdiff_t*
L			long double				



A. Έξοδος Δεδομένων

2. Η εντολή printf

6. Τυπικό Συντακτικό της printf

- Και ένα διεστραμμένο παράδειγμα:

```
/* printf example */
#include <stdio.h>

int main()
{
    printf ("Simplirwsi me kena: |%10d| \n", 1980);
    printf ("Simplirwsi me midenika: |%010d| \n", 1980);
    printf ("Morfes arithmou: %d %x %o %#x %#o \n", 100, 100, 100, 100, 100);
    printf ("Pragmatikoi: %4.2f %+.0e %E \n", 3.1416, 3.1416, 3.1416);
    printf ("Me to asteraki: %*d \n", 5, 10);
}
```



Β. Είσοδος Δεδομένων

1. Γενικά

- Η εντολή scanf είναι η βασική συνάρτηση που δέχεται είσοδο από τον χρήστη.
- Είναι ορισμένη στο αρχείο κεφαλίδας stdio.h, γι' αυτό πρέπει πάντα να το κάνουμε #include αν θέλουμε να την χρησιμοποιήσουμε.
- Όπως και με την printf που είναι η πιο σημαντική συνάρτηση εξόδου δεδομένων, έτσι και η scanf είναι η πιο σημαντική συνάρτηση εισόδου δεδομένων.
- Ας θυμηθούμε από το Μάθημα 2, τους τρόπους γραφής της scanf και το συντακτικό τους.



Β. Είσοδος Δεδομένων

1. Η εντολή scanf

1. ΣΥΝΤΑΚΤΙΚΟ

- Η scanf συντάσσεται ως εξής:

```
scanf ("%x", &μεταβλητή);
```

- Όπως βλέπουμε παίρνει 2 ορίσματα:

- Το 1^ο όρισμα είναι ο προσδιοριστής του τύπου δεδομένων που θα διαβάσουμε
 - Έτσι στη θέση του %x (οποσδήποτε μέσα σε διπλά εισαγωγικά) μπορούμε να βάλουμε τον προσδιοριστή που θέλουμε
- Το 2^ο όρισμα είναι υποχρεωτικά το & ακολουθούμενο από το όνομα της μεταβλητής, στο οποίο θα αποθηκευτεί η τιμή που θα εισάγει ο χρήστης
 - Περισσότερες λεπτομέρειες για το υποχρεωτικό σύμβολο & που οφείλει να προηγείται από το όνομα της μεταβλητής θα δούμε σε επόμενο μάθημα που θα μελετήσουμε τους δείκτες.
- Πρακτικά όταν εκτελείται η εντολή scanf το πρόγραμμα «παγώνει» και αναμένει από τον χρήστη να πληκτρολογήσει την είσοδο και να πατήσει Enter. Μόλις γίνει αυτό, η τιμή που πληκτρολόγησε ο χρήστης αποθηκεύεται στην μεταβλητή, και το πρόγραμμα εκτελεί την επόμενη εντολή.

Σημαντική Λεπτομέρεια: Ο προσδιοριστής για το double δεν είναι %f, αλλά %lf. Συνεπώς για double:

- Στην printf ο προσδιοριστής είναι %f (ή %lf)
- Στην scanf ο προσδιοριστής είναι %lf



Β. Είσοδος Δεδομένων

1. Η εντολή scanf

2. Τρόποι Χρήσης

- Μπορούμε να εισάγουμε περισσότερες μεταβλητές με μία εντολή scanf χωρίζοντας με κενά τους προσδιοριστές.

```
int x;
float y;

scanf ("%d %f", &x, &y);
```

- Το πρόγραμμα θα περιμένει να βάλουμε 2 αριθμούς (στο παράδειγμα έναν ακέραιο και έναν πραγματικό)
 - Προσοχή! Στην εκτέλεση θα προσπεράσει τυχόν αλλαγές γραμμής και κενά μέχρι να εντοπίσει τους αριθμούς που του είπαμε να διαβάσει.
 - Έτσι π.χ. Θα δουλέψει αν ο χρήστης εισάγει: `2 4.25`
 - Ή ακόμη αν εισάγει: `8 5.333` (δηλ. Αν βάλει περιττά κενά)
 - Ή ακόμη και αν βάλει περιττές αλλαγές γραμμής ή tabs.

- Θεωρούμε καλή τακτική να έχουμε μία scanf για κάθε διάβασμα μεταβλητής.
- Επίσης καλό θα είναι να έχουμε και μηνύματα στον χρήστη για να ξέρει τι πρόκειται να πληκτρολογήσει κάθε φορά!



Και...

- Δεν έχουμε τελειώσει ακόμη με την είσοδο/έξοδο...
 - Υπάρχουν ακόμη πολλές συναρτήσεις για την είσοδο/έξοδο.
 - Θα δούμε και πώς να κάνουμε είσοδο/έξοδο σε αρχεία δίσκου.
- Ο στόχος του μαθήματος ήταν να μπορέσουμε να έχουμε είσοδο/έξοδο για όλους τους βασικούς τύπους δεδομένων.
- Τα υπόλοιπα σε επόμενο μάθημα!



Γ. Ασκήσεις

1. Ένα απλό πρόγραμμα

1. Γράψτε ένα ολοκληρωμένο πρόγραμμα που θα δέχεται και θα τυπώνει διαφορετικές εισόδους του χρήστη.
2. Συγκεκριμένα η main σας θα προτρέπει το χρήστη να εισάγει έναν αριθμό από το 1 έως το 4 (να γίνει έλεγχος ότι ο χρήστης έβαλε αριθμό από 1 έως 4)
 1. Αν ο χρήστης εισάγει 1, να διαβάζει μια ακέραια μεταβλητή και να τυπώνει το τετράγωνό της.
 2. Αν ο χρήστης εισάγει 2, να καλεί μια συνάρτηση χρήστη (ονομάστε την print) στην οποία να διαβάζει έναν ακέραιο αριθμό N και να τυπώνει N φορές τη λέξη Kalimera.
 3. Αν ο χρήστης εισάγει 3, να διαβάζει μια float μεταβλητή, και να τυπώνει το $\frac{1}{4}$ της (δηλαδή να την διαιρεί με το 4 και να τυπώνει το αποτέλεσμα) με 4 δεκαδικά ψηφία.
 4. Αν ο χρήστης εισάγει 4, να μεταβαίνει σε μία συνάρτηση χρήστη (ονομάστε την eisagogi_rinaka) που δηλώνει έναν πίνακα 5 double μεταβλητών, διαβάζει τις τιμές τους από το πληκτρολόγιο και τυπώνει τον μέσο όρο τους.



Γ. Ασκήσεις

2. Μία μορφοποιημένη εκτύπωση

1. Κατασκευάστε πρόγραμμα που τυπώνει το εξής τρίγωνο:

```

*           *
**        **
***      ***
****    ****
*****  *****

```



Γ. Ασκήσεις

3. Ακόμη μία μορφοποιημένη εκτύπωση

Και τώρα ώρα για εκτύπωση βουνών από αριθμούς::

```

123456432112345643211234564321
1234  43211234  43211234  4321
123   321123   321123   321
12    2112    2112    21
1     11     11     1

```

- (εννοείται ότι απαιτείται το τρίγωνο να κατασκευαστεί με κατάλληλη χρήση δομών επανάληψης και όχι με ξερές printf!)
- Το πρόγραμμα να δουλεύει για οποιαδήποτε N από 1 έως 9 (στο παράδειγμα έχουμε N=5)