

Η ΓΛΩΣΣΑ C

Μάθημα 2:

Βασικά Στοιχεία ενός προγράμματος C

Δημήτρης Ψούνης



www.psounis.gr



Περιεχόμενα Μαθήματος

A. Θεωρία

1. Στοιχεία ενός προγράμματος C
2. Η συνάρτηση main()
3. Η οδηγία #include
4. Μεταβλητές
5. Εντολές
 1. Η εντολή printf
 2. Η εντολή scanf
6. Συναρτήσεις Χρήστη
 1. Το πρωτότυπο μιας συνάρτησης
 2. Το σώμα μιας συνάρτησης
 3. Κλήση μιας συνάρτησης
7. Σχόλια Προγράμματος

B. Ασκήσεις



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

1. Ένα συνηθισμένο πρόγραμμα C

Στο σημερινό μάθημα, θα μελετήσουμε αναλυτικά την δομή ενός προγράμματος C

- Τα Δομικά Στοιχεία κάθε προγράμματος C είναι ίδια, άρα θα πρέπει να γίνει κατανοητή η εικόνα που θα έχει κάθε πρόγραμμα που θα γράφουμε
- Στο στάδιο αυτό δεν μας ενδιαφέρουν ιδιαίτερα οι τεχνικές λεπτομέρειες διότι θα τις μάθουμε αναλυτικά στα επόμενα μαθήματα



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

1. Ένα συνηθισμένο πρόγραμμα C

- Ωστόσο πρέπει να γνωρίζουμε τι είναι μια **συνάρτηση** στην C. Μια συνάρτηση είναι ένα ανεξάρτητο τμήμα προγράμματος που κάνει μια συγκεκριμένη δουλειά και έχει ένα όνομα. Αναφέροντας το όνομα της συνάρτησης, το πρόγραμμα μπορεί να εκτελέσει τον κώδικα της συνάρτησης. Επίσης:
 - Έχει την δυνατότητα να λάβει πληροφορίες από το πρόγραμμα, οι οποίες λέγονται ορίσματα
 - Μπορεί να επιστρέψει πληροφορίες στο πρόγραμμα που είναι οι επιστρεφόμενες τιμές.
- Στην C υπάρχουν δύο είδη συναρτήσεων:
 - Οι **συναρτήσεις βιβλιοθήκης**, οι οποίες έχουν γραφεί στον μεταγλωττιστή και είναι έτοιμες για να τις χρησιμοποιήσουμε
 - Οι **συναρτήσεις χρήστη**, οι οποίες γράφονται από εμάς, ώστε να τις χρησιμοποιήσουμε στο πρόγραμμά μας.
- Μεταγλωττίστε και εκτελέστε το πρόγραμμα της επόμενης διαφάνειας.



```
/* Multiply.c: Programma gia ton ipologismo tou ginomenou dio arithmwn */
#include <stdio.h>

int a,b,c;

int ginomeno(int x, int y);

main()
{
    /* Eisodos toy 1ou arithmou */
    printf("Eisagete arithmo metaksi 1 kai 100:");
    scanf("%d", &a);

    /* Eisodos tou 2ou arithmou */
    printf("Eisagete allon ena arithmo metaksi 1 kai 100:");
    scanf("%d", &b);

    c=ginomeno(a,b);
    printf("To ginomeno tou %d kai tou %d einai: %d\n",a,b,c);
}

/* H synartisi epistrefei to ginomeno twn dio orismatwn tis */
int ginomeno(int x, int y)
{
    return (x*y);
}
```



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

2. Η συνάρτηση main()

- Η συνάρτηση main() είναι απαραίτητη σε κάθε πρόγραμμα C και είναι το σημείο-αφετηρία του προγράμματος μας.
- Η απλούστερη μορφή της είναι το όνομα της, ακολουθούμενη από ένα ζευγάρι κενών παρενθέσεων () και ένα ζευγάρι άγκιστρα:

```
main( )  
{  
  
  
}
```

- Μέσα στα άγκιστρα υπάρχουν προτάσεις που αποτελούν το σώμα του προγράμματος και η εκτέλεση ξεκινά από την πρώτη πρόταση και τελειώνει στην τελευταία πρόταση.

Σημείωση 1: Στην συνέχεια όταν θα αναφερόμαστε στο «πρόγραμμα» εννοούμε πάντα την συνάρτηση main().

Σημείωση 2: Στο προηγούμενο μάθημα είδαμε μια παραλλαγή όπου η συνάρτηση main() γραφόταν int main() και επέστρεφε μία τιμή στο λειτουργικό σύστημα με την εντολή return 0. Η πιο απλή χρήση της main είναι να μην επιστρέφει τιμή στο λειτουργικό σύστημα.



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

3. Η οδηγία #include

- Στην αρχή του προγράμματος μας έχουμε την οδηγία:

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

- Η οδηγία αυτή επιβάλλει στον μεταγλωττιστή να εισάγει στο πρόγραμμά μας το **αρχείο κεφαλίδας** (ή **αρχείο βιβλιοθήκης**) `stdio.h`
- Τα αρχεία κεφαλίδας είναι αυτόνομα αρχεία, αποθηκευμένα στο σκληρό μας δίσκο τα οποία περιέχουν πληροφορίες που χρειάζονται για τον μεταγλωττιστή μας.
 - Στο συγκεκριμένο πρόγραμμα εισάγουμε το αρχείο κεφαλίδας `stdio.h` γιατί εκεί έχουν οριστεί οι συναρτήσεις βιβλιοθήκης `printf` και `scanf` που χρησιμοποιούμε στο πρόγραμμά μας
 - Ποτέ δεν θα χρειαστεί να τροποποιήσουμε τα αρχεία αυτά, γι' αυτό και βρίσκονται αποθηκευμένα σε ξεχωριστό χώρο στον δίσκο μας.

Παρατήρηση: Κάθε συνάρτηση βιβλιοθήκης που θέλουμε να χρησιμοποιήσουμε στο πρόγραμμά μας απαιτεί την εισαγωγή μέσω της οδηγίας `#include` του αντίστοιχου αρχείου κεφαλίδας στο οποίο έχει οριστεί (π.χ. Το `stdio.h` έχει συναρτήσεις εισόδου-εξόδου, το `stdlib.h` έχει συναρτήσεις διαχείρισης μνήμης, το `string.h` έχει συναρτήσεις διαχείρισης συμβολοσειρών κ.ο.κ.).

Συνεπώς πρέπει να ξέρουμε (ή να έχουμε σημειωμένες) ποια βιβλιοθήκη αντιστοιχεί σε κάθε συνάρτηση



A. Θεωρία

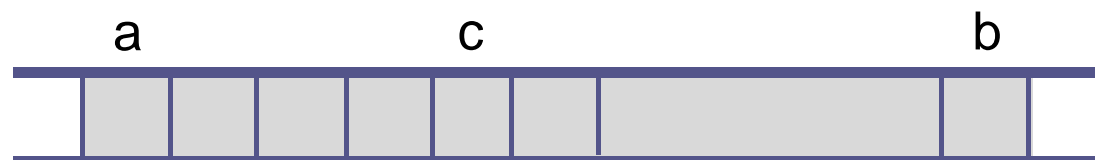
1. Στοιχεία ενός προγράμματος C

4. Μεταβλητές

- Κάθε μεταβλητή έχει ένα **όνομα** και αποθηκεύεται σε μια **θέση αποθήκευσης** στην μνήμη και έχει μία **τιμή**.
- Στην C προτού χρησιμοποιήσουμε μια μεταβλητή πρέπει να την δηλώσουμε δηλαδή να ορίσουμε στον μεταγλωττιστή τι τύπου δεδομένα μπορούμε να αποθηκεύσουμε στην μεταβλητή.
- Στο πρόγραμμα μας η γραμμή:

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

- Αποτελεί την δήλωση 3 ακεραίων μεταβλητών.
- Η λέξη-κλειδί int ορίζει ότι πρόκειται να δηλώσουμε ακέραιες μεταβλητές (υπάρχουν και άλλοι τύποι δέδομένων όπως π.χ. float που αποθηκεύει πραγματικές μεταβλητές, που θα δούμε αναλυτικά στο «μάθημα 3: Μεταβλητές και Σταθερές»)
- Ακολουθούν τα ονόματα των μεταβλητών που δηλώνουμε χωρισμένα με κόμματα.
- Η δήλωση τελειώνει με ερωτηματικό.
- Εικόνα της μνήμης μετά τη δήλωση των τριών μεταβλητών (έχει δεσμευτεί χώρος στη μνήμη για τις μεταβλητές ώστε να αποθηκεύσουμε τιμές σε αυτές).





A. Θεωρία

1. Στοιχεία ενός προγράμματος C

5. Εντολές

- Η πραγματική εργασία ενός προγράμματος γίνεται από τις **εντολές** που είναι γραμμένες ανάμεσα στα άγκιστρα της main.
- Οι **εντολές** της C:
 - Προβάλλουν πληροφορίες στην οθόνη
 - Εκτελούν μαθηματικές λειτουργίες
 - Καλούν συναρτήσεις
 - Διαβάζουν αρχεία από τον δίσκο
 - Διάβάζουν τιμές από το πληκτρολόγιο
 - Και άλλες λειτουργίες που θα μάθουμε στην διάρκεια των μαθημάτων.
- **Συμβουλή:** Γράφουμε μία εντολή ανά γραμμή και οι εντολές τελειώνουν πάντα με ερωτηματικό. (είναι συντακτικό λάθος αν ξεχάσουμε να βάλουμε ερωτηματικό και ο μεταγλωττιστής θα διαμαρτυρηθεί)



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

5. Εντολές (A. Η εντολή printf)

- Η εντολή **printf** είναι μια συνάρτηση βιβλιοθήκης, η οποία προβάλλει το περιεχόμενο της στην οθόνη.
- Την έχουμε χρησιμοποιήσει στην main σε 4 σημεία στο πρόγραμμά μας:

```
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.
 - Έχει πολλές διαφορετικές χρήσεις που θα αναλύσουμε στο μάθημα «Είσοδος και Έξοδος»
 - Εδώ θα δούμε 3 διαφορετικές της χρήσεις που θα χρειαστούμε για τα αμέσως επόμενα μαθήματα (απλή είσοδος και έξοδος ακεραίων αριθμών).



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

5. Εντολές (A. Η εντολή printf)

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
 - Τέτοιος είναι και ο ‘\n’ που λέγεται χαρακτήρας αλλαγής γραμμής διότι η εκτύπωση του προκαλεί την αλλαγή γραμμής στην οθόνη.



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

5. Εντολές (Α. Η εντολή printf)

- 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)
```



- Τυπώνει στην οθόνη

```
Η metabliti exei timi 1
```



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

5. Εντολές (A. Η εντολή printf)

- 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)
```



- Τυπώνει στην οθόνη

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



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

5. Εντολές (A. Η εντολή printf)

- 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.
- Σε επόμενο μάθημα θα δούμε πως μπορούμε με την printf να εκτυπώσουμε και άλλους τύπους δεδομένων όπως πραγματικές μεταβλητές, συμβολοσειρές κ.λ.π.



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

5. Εντολές (B. Η εντολή scanf)

- Η εντολή scanf είναι μια συνάρτηση βιβλιοθήκης (ορισμένη στο αρχείο κεφαλίδας stdio.h), η οποία διαβάζει από το πληκτρολόγιο δεδομένα.
- Την έχουμε χρησιμοποιήσει στην main σε 2 σημεία στο πρόγραμμά μας:

```
scanf ( "%d" , &a ) ;
```

```
scanf ( "%d" , &b ) ;
```

- Η scanf είναι επίσης πάρα πολύ σημαντική συνάρτηση, διότι διαβάζει δεδομένα από τον χρήστη.
 - Έχει πολλές διαφορετικές χρήσεις που θα αναλύσουμε σε επόμενο μάθημα (π.χ διάβασμα πραγματικών αριθμών, συμβολοσειρών κ.λπ.)
 - Εδώ θα δούμε μία μόνο χρήση της, για το πως να διαβάζουμε τιμές για ακέραιες μεταβλητές.



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

5. Εντολές (B. Η εντολή scanf)

- Η scanf συντάσσεται ως εξής:

```
scanf ( "%d" , &μεταβλητή ) ;
```

- Όπως βλέπουμε παίρνει 2 ορίσματα:

- Το 1^ο όρισμα είναι υποχρεωτικά η συμβολοσειρά "%d" που καθορίζει ότι θα διαβάσουμε μια ακέραια μεταβλητή

- Το %d είναι ο προσδιοριστής ακεραίου

- 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 συμβολο & που οφείλει να προηγείται από το όνομα της μεταβλητής θα δούμε στο «Μάθημα 9: Δείκτες»

- Πρακτικά όταν εκτελείται η εντολή scanf το πρόγραμμα «παγώνει» και αναμένει από τον χρήστη να πληκτρολογήσει τον ακέραιο αριθμό και να πατήσει Enter. Μόλις γίνει αυτό, η τιμή που πληκτρολόγησε ο χρήστης αποθηκεύεται στην μεταβλητή, και το πρόγραμμα εκτελεί την επόμενη εντολή.



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

6. Συναρτήσεις Χρήστη

- Στο πρόγραμμα αυτό, ο χρήστης έχει εισάγει μια δική του συνάρτηση, την `ginomeno`, η οποία υπολογίζει το γινόμενο δύο ακέραιων αριθμών που δέχεται ως ορίσματα και τα επιστρέφει.
- Βλέπουμε τα 3 σημεία στο πρόγραμμα που αναφέρονται στην συνάρτηση: Το πρωτότυπό της, το σώμα της και την κλήση της στην `main` ως εντολή.

.....

`int ginomeno(int x, int y);` **<- Αυτό είναι το πρωτότυπο της συνάρτησης**

```
main()  
{  
    ....  
    c=ginomeno(a,b); <- Εδώ καλούμε την συνάρτηση στην main,  
    .... σαν μία ακόμη εντολή του προγράμματος  
}
```

```
int ginomeno(int x, int y) <- Αυτό είναι το σώμα της συνάρτησης  
{  
    return (x*y);  
}
```



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

6. Συναρτήσεις Χρήστη (A. Το πρωτότυπο της συνάρτησης)

- Κάθε συνάρτηση έχει την εξής έννοια
 - Παίρνει ως είσοδο κάποια δεδομένα, μέσω των ορισμάτων της
 - Κάνει έναν υπολογισμό
 - Επιστρέφει το αποτέλεσμα του υπολογισμού της
- ΠΑΝΤΑ πριν από την main καταγράφουμε τα πρωτότυπα των συναρτήσεων που θα ορίσουμε. Το πρωτότυπο είναι μια περιγραφή μόνο των ορισμάτων της συνάρτησης και της επιστρεφόμενης τιμής (και όχι του υπολογισμού). Το συντακτικό είναι:

```
Τύπος_Επιστρεφόμενης_Τιμής ΟΝΟΜΑ_ΣΥΝΑΡΤΗΣΗΣ (Ορισμα1, Ορισμα2, ... ) ;
```

- Όπως στην συνάρτηση μάς:

```
int                ginomeno                (int x,    int y);
```

- όπου περιγράφουμε ότι πρόκειται να ορίσουμε μια συνάρτηση με όνομα ginomeno: που παίρνει δύο ακέραιες μεταβλητές ως ορίσματα και επιστρέφει μια ακέραια μεταβλητή



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

6. Συναρτήσεις Χρήστη (B. Το σώμα της Συνάρτησης)

- Το σώμα της συνάρτησης αποτελεί την περιγραφή των εντολών που εκτελεί η συνάρτηση. Πάντα θα είναι META την main και οι εντολές της θα βρίσκονται ανάμεσα σε άγκιστρα

```
int ginomeno(int x, int y)
{
    return (x*y);
}
```

- Παρατηρούμε ότι η 1^η γραμμή είναι ακριβώς ίδια με το πρωτότυπο (αλλά δεν έχει ερωτηματικό)
- Έπειτα μέσα στα άγκιστρα έχουμε το δικαίωμα να γράψουμε οποιαδήποτε εντολή.
- Η εντολή return επιστρέφει σε αυτόν που κάλεσε την συνάρτηση το όρισμά της. Άρα η συνάρτηση επιστρέφει το γινόμενο του αριθμού x με το y.
 - (Το * είναι τελεστής που εκτελεί τον πολλαπλασιασμό των μεταβλητών. Περισσότερους τελεστές θα δούμε σε επόμενο μάθημα)



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

6. Συναρτήσεις Χρήστη (Γ. Κλήση μιας συνάρτησης)

- ΑΦΟΥ έχουμε ορίσει μια συνάρτηση (έχοντας γράψει το πρωτότυπό της πριν την main και το σώμα της μετά την main), μπορούμε να την χρησιμοποιήσουμε σε οποιοδήποτε σημείο του προγράμματός μας, καλώντας την σαν μια οποιαδήποτε εντολή.

```
main( )  
{  
    ....  
    c=ginomeno(a,b);  
    ....  
}
```

- Όπου γίνεται κλήση της συνάρτησης με ορίσματα τις τιμές των a,b που έχει εισάγει ο χρήστης νωρίτερα.
- Ο έλεγχος μεταβαίνει στην συνάρτηση ginomeno όπου υπολογίζεται και επιστρέφεται το γινόμενο των δύο αριθμών. Το αποτέλεσμα αποθηκεύεται στην μεταβλητή c μέσω του τελεστή καταχώρησης (=).
 - Περισσότερα για τον τελεστή καταχώρησης (=) θα δούμε στο επόμενο μάθημα.



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

6. Συναρτήσεις Χρήστη (Γ. Κλήση μιας συνάρτησης)

- Οι συναρτήσεις είναι το A και το Ω της γλώσσας C. Πολλές φορές αναφέρεται για την C ότι είναι η πιο δημοφιλής **διαδικαστική γλώσσα** δηλαδή γλώσσα που λειτουργεί με διαδικασίες (συναρτήσεις)
- Αν και δεν υπάρχει περίπτωση να ορίσουμε μια συνάρτηση που εκτελεί μία τόσο απλή ενέργεια, όσο ένας πολλαπλασιασμός (όπως στο πρόγραμμά μας), είδαμε πως μπορούμε να **συντάξουμε** μια συνάρτηση χρήστη:
 - Ορίζουμε το πρωτότυπο της
 - Το σώμα της
 - Και την καλούμε ως εντολή στην συνέχεια.
- **Συμβουλή:** Ορίζουμε μία συνάρτηση χρήστη όταν θέλουμε να κάνουμε **πολλές φορές την ίδια δουλειά**, έτσι ώστε να γράψουμε το πρωτότυπο και το σώμα, και έπειτα απλά να καλούμε την συνάρτηση σαν μια εντολή, κάθε φορά που θέλουμε να ξαναγίνει η ίδια ενέργεια.



A. Θεωρία

1. Στοιχεία ενός προγράμματος C

7. Σχόλια Προγράμματος

- Είναι πολύ σημαντικό να τεκμηριώνουμε το πρόγραμμα μας με σχόλια που εξηγούν σημεία του κώδικα.
- Συνήθως βάζουμε σχόλια σε τμήματα κώδικα τα οποία είναι περίπλοκα και βοηθάνε είτε κάποιον άλλον που διαβάζει τον κώδικά μας, είτε κι εμάς τους ίδιους αν πρόκειται να ξαναδιαβάσουμε τον κώδικα μετα από καιρό.
- Υπάρχουν δύο ειδών σχόλια στην C:
 - Τα σχόλια μίας γραμμής (ξεκινάνε με //):

```
// Εδώ βάζουμε σχόλια που θα τελειώσουν με την αλλαγή γραμμής
```

- Τα σχόλια πολλών γραμμών (ξεκινάνε με /* και τελειώνουν με */))

```
/* Αρχή σχολίων  
Μία Δεύτερη Γραμμή Σχολίων  
Τέλος Σχολίων */
```



Β. Ασκήσεις

Εφαρμογή 1

Μεταγλωττίστε και εκτελέστε το ακόλουθο πρόγραμμα:

1. Εντοπίστε εντολές στο πρόγραμμα.
2. Εντοπίστε δηλώσεις μεταβλητών.
3. Εντοπίστε τη δήλωση πρωτοτύπου συνάρτησης
4. Εντοπίστε τις γραμμές σώματος συναρτησης
5. Ποιες γραμμές περιέχουν σχόλια;

```
/* print.c: Programma gia ektiposi grammwn */
#include <stdio.h>

int x;

void grammi();

main()
{
    // Ektyposeis...
    grammi();
    printf("\nSe posa mathimata thelete na mathete
tin C? ");
    scanf("%d", &x);
    grammi();

    printf("\nTha apaitithoun 20 mathimata!");
    grammi();
    printf("\nApantisate %d",x);
    grammi();
}

/* H synartisi ektiponei mia grammi */
void grammi()
{
    printf("\n=====");
}
```



B. Ασκήσεις

Εφαρμογή 2

Μεταγλωττίστε και εκτελέστε το ακόλουθο πρόγραμμα. Στη συνέχεια τροποποιήστε το πρόγραμμα ώστε να τυπώνει τον κύβο κάθε αριθμού.

```
/* numbers.c */
#include <stdio.h>

main()
{
    int i,N,y;

    printf("Eisagete enan akeraio arithmo: ");
    scanf("%d",&N);

    for (i=1; i<=N; i++)
    {
        y=i*i;
        printf("\n%d sto tetragono = %d",i,y);
    }

    printf("\n\nTelos Programmatos");
}
```