

ΤΕΙ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΠΟΛΥΜΕΣΩΝ

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ
ΕΡΓΑΣΤΗΡΙΟ

ΗΡΑΚΛΕΙΟ 2011

ΕΙΣΑΓΩΓΗ

(ΓΕΝΙΚΑ ΓΙΑ ΤΟ ΕΡΓΑΣΤΗΡΙΟ – ΜΑΘΗΜΑΤΑ – ΕΞΕΤΑΣΗ)

Το φυλλάδιο αυτό αφορά το εργαστήριο του μαθήματος «Προγραμματισμός» και περιλαμβάνει ασκήσεις, οι οποίες πρέπει να εκτελεστούν στη διάρκεια του εργαστηρίου, υποδείξεις επί των ασκήσεων, καθώς και προτεινόμενες ασκήσεις για περαιτέρω άσκηση.

Προσοχή! Στα εργαστήρια δεν επαναλαμβάνεται η θεωρία! Γίνεται μόνο υπενθύμιση ορισμένων σημείων. Αυτό σημαίνει ότι η προετοιμασία και η παρακολούθηση του μαθήματος της θεωρίας είναι απαραίτητη. Σημεία της θεωρίας θα θίγονται μόνο όταν υπάρχει κάποια χρονική «ασυμφωνία» στην διδασκόμενη ύλη, όταν δηλαδή κάποιο θέμα δεν έχει ήδη καλυφθεί στο μάθημα.

Στη διάρκεια του εξαμήνου γίνονται (τουλάχιστον) 12 εργαστήρια.

Στα εργαστήρια 3, 6, 9 και 12 διεξάγονται τα διαγωνίσματα (τεστ) T1, T2, T3 και T4, διάρκειας περίπου 15 λεπτών (ή όσων κρίνει ο διδάσκων), με ποσοστό συμμετοχής στον τελικό βαθμό 10%, 15%, 25% και 50% αντίστοιχα. Ο τελικός βαθμός στο εργαστήριο θα διαμορφώνεται δηλαδή σύμφωνα με τον εξής τύπο:

$$\text{Τελικός Βαθμός} = 0.1 \times T1 + 0.15 \times T2 + 0.25 \times T3 + 0.5 \times T4.$$

Για την ολοκλήρωση και παρουσίαση αυτού του φυλλαδίου συνεργάστηκαν ο Καθηγητής του ΕΠΠ Ι. Ξεζωνάκης και ο Επιστημονικός Συνεργάτης του ΕΠΠ Α. Καμπουρέλης

ΕΡΓΑΣΤΗΡΙΟ 1

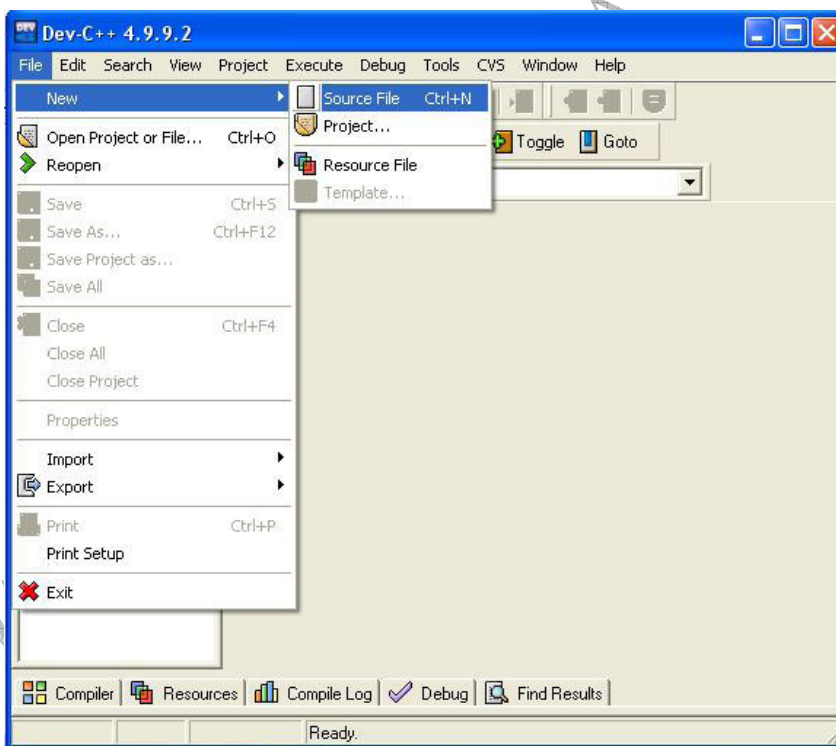
Α'. ΕΞΟΙΚΕΙΩΣΗ ΜΕ ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΕΡΓΑΣΙΑΣ

A.1. Dev C++ IDE

IDE: Integrated Development Environment, δηλαδή «Ολοκληρωμένο περιβάλλον ανάπτυξης προγραμμάτων»

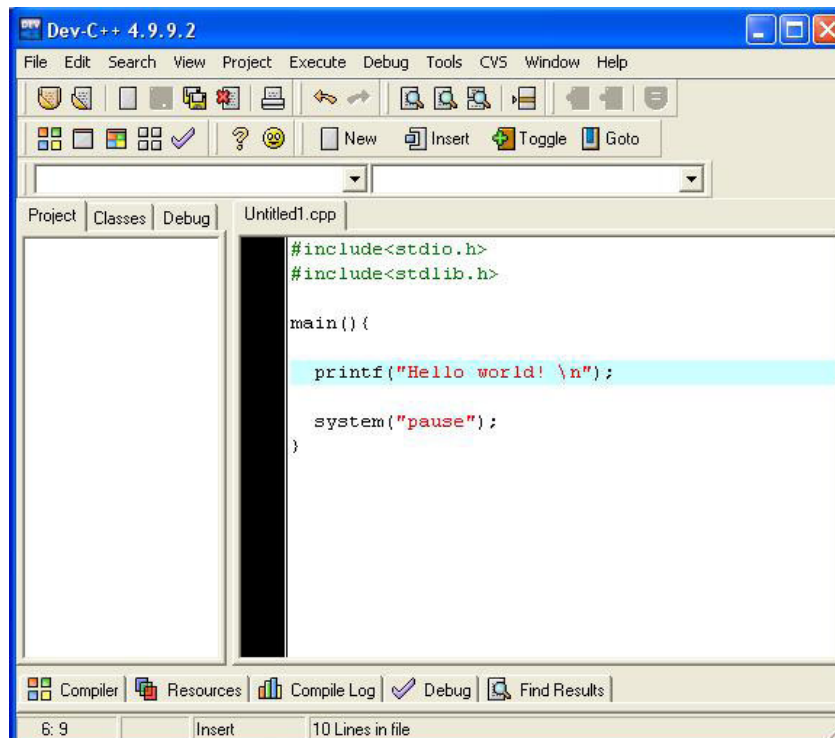
1. Δημιουργία του νέου προγράμματος

File/New/Source File, ή Ctrl-N



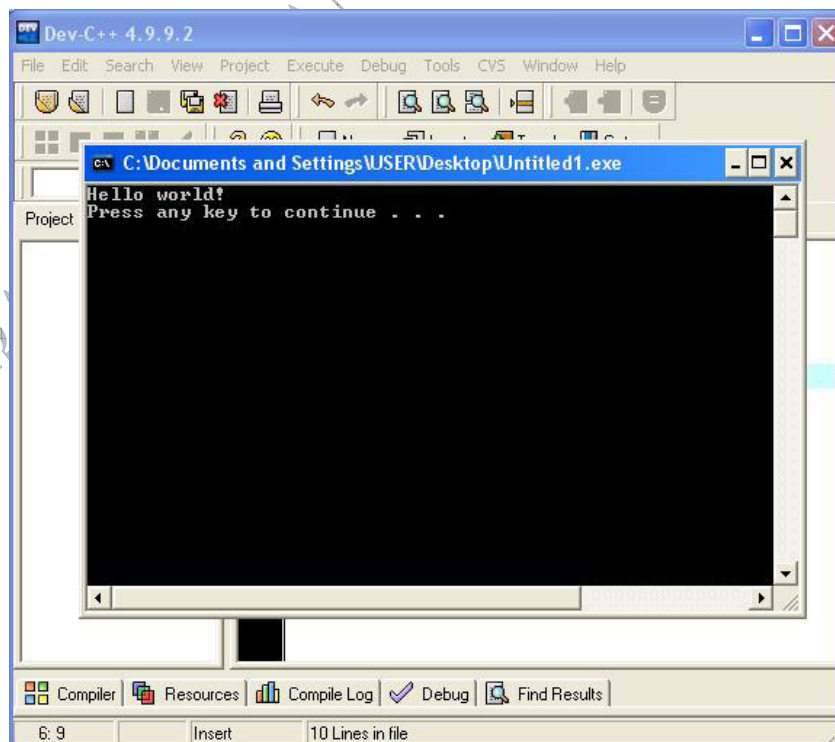
Εικ. 1

2. Σύνταξη του προγράμματος στον κειμενογράφο του Dev C++



ΕΙΚ. 2

3. Εκτέλεση του προγράμματος



ΕΙΚ. 3

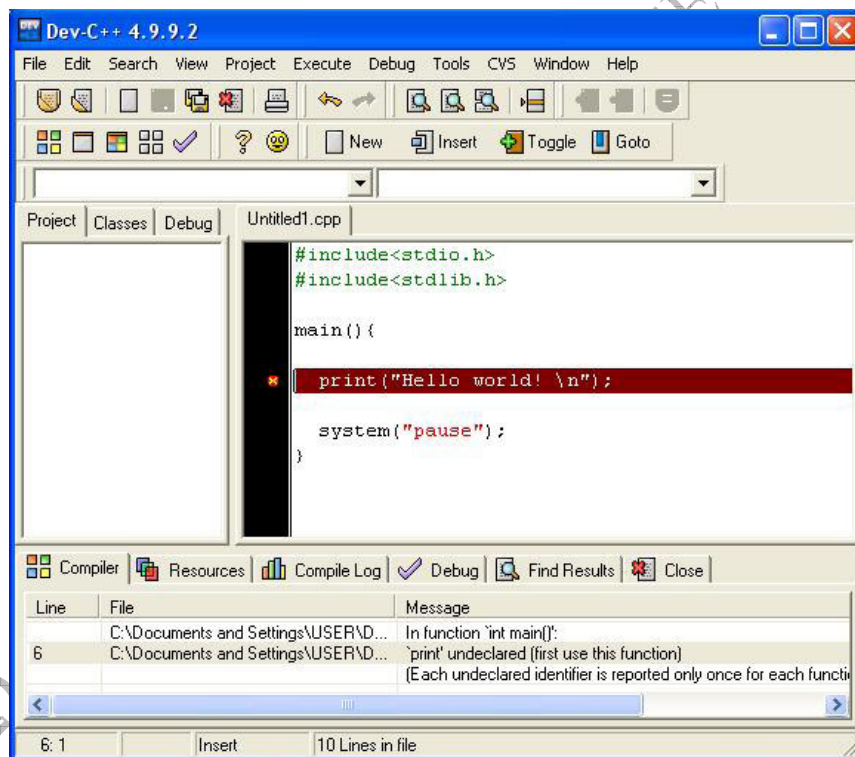
Πλήκτρο **F9**: *Compile and Run program.*

Μετά την εκτέλεση του προγράμματος που φαίνεται στην εικ. 2 θα δείτε στην οθόνη σας την εικ. 3. Το μαύρο πλαίσιο είναι το «παράθυρο αποτελεσμάτων», ο χώρος δηλαδή στον οποίο παρουσιάζονται τα αποτελέσματα του προγράμματός σας.

4. Διόρθωση του προγράμματος

Αν κάνατε λάθος, π.χ. γράψατε `print` αντί για `printf`, ο μεταγλωττιστής εμφανίζει ένα μήνυμα λάθους. Διαβάστε προσεκτικά το μήνυμα του μεταγλωττιστή, διορθώστε το λάθος και ξανατρέξτε το πρόγραμμα.

Η κόκκινη γραμμή στο παράθυρο όπου έχετε γράψει το πρόγραμμά σας υποδεικνύει την θέση του λάθους.



Εικ. 4

5. Αρχεία που δημιουργήθηκαν

Αν δεν έχετε καθορίσει εσείς κάποιο όνομα, τότε δημιουργούνται τα αρχεία:



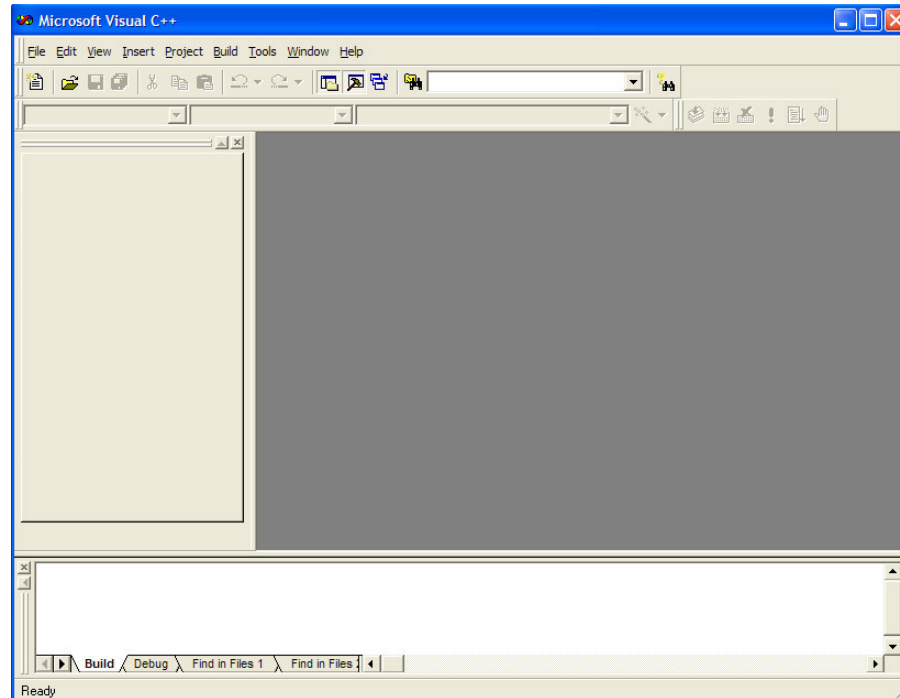
.cpp : Ο πηγαίος κώδικάς σας (source file)

.exe : Εκτελέσιμο πρόγραμμα (executable program file)

A.2. Microsoft Visual C++ IDE

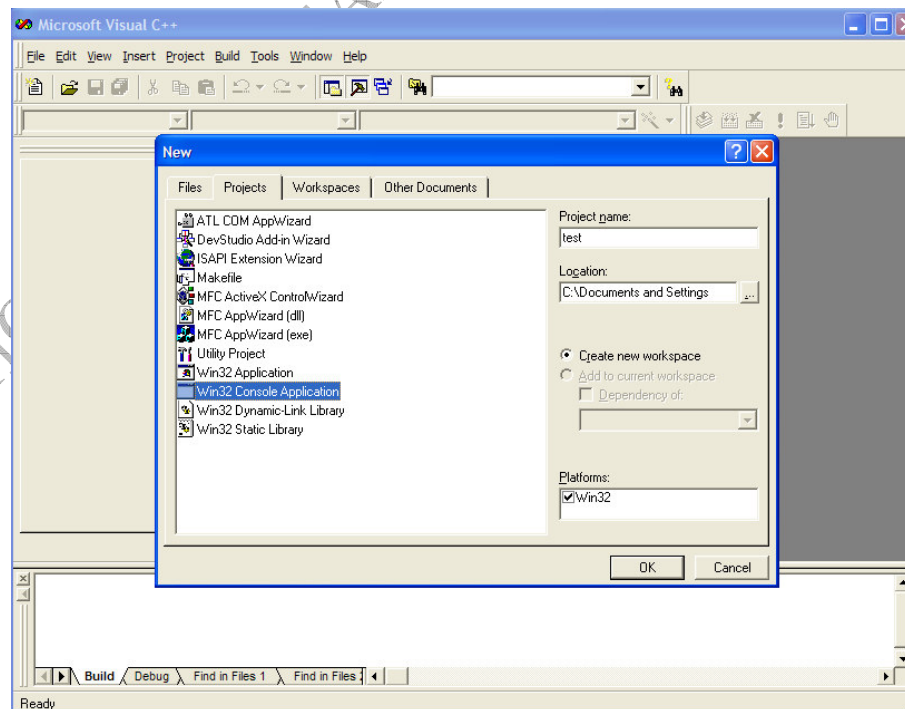
1. Δημιουργία του νέου προγράμματος

File/New



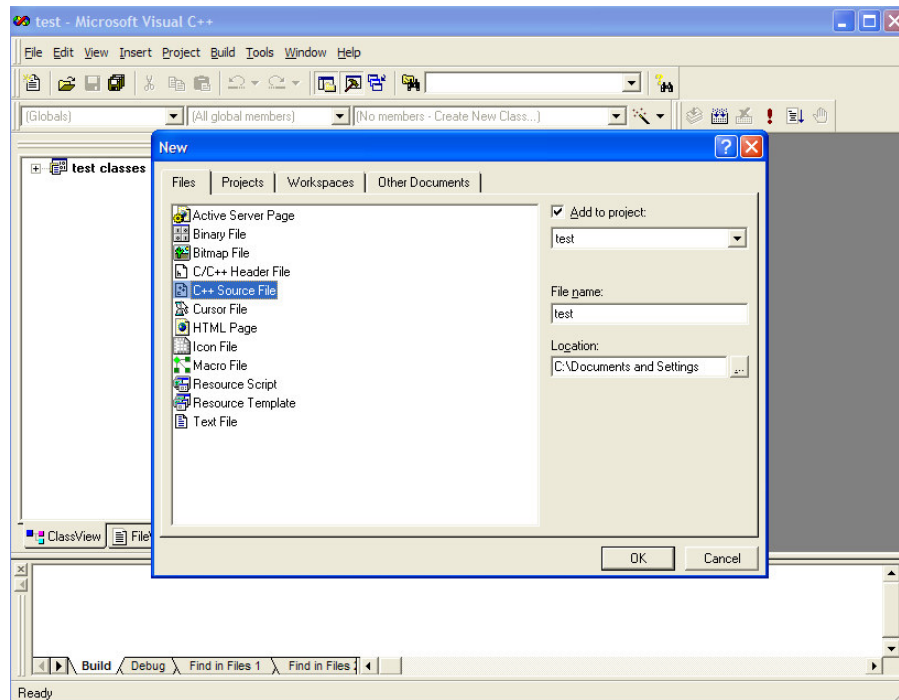
Εικ. 1

Win32 Console Application και καθορισμός Project Name



Εικ. 2

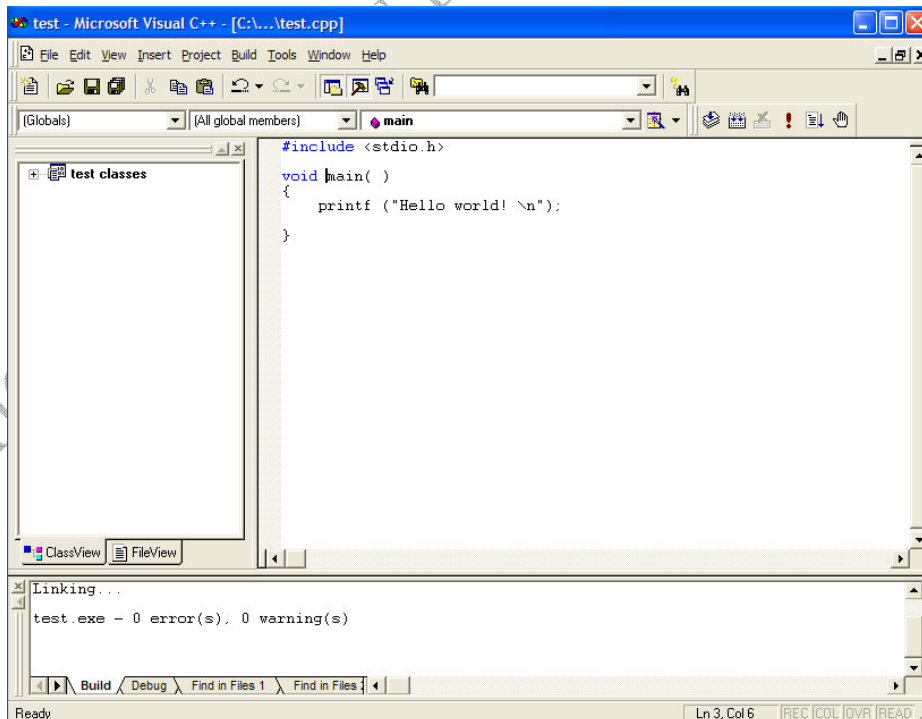
File/New/C++ Source File και καθορισμός File name



Εικ. 3

2. Σύνταξη του προγράμματος

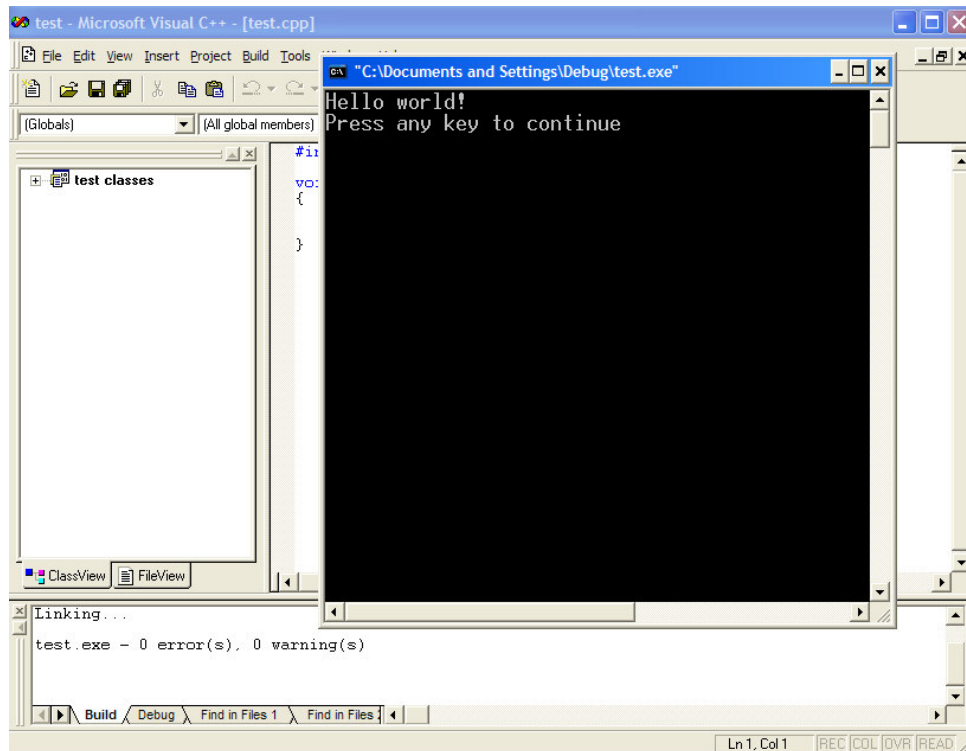
Μεταγλώττιση με: Build/Rebuild All



Εικ. 4

3. Εκτέλεση του προγράμματος

Build/Execute



Εικ. 5

4. Αρχεία που δημιουργήθηκαν

Ένα αρχείο με προέκταση cpp (περιέχει το πηγαίο πρόγραμμα), ένας κατάλογος με το όνομα Debug και πέντε άλλα αρχεία με προεκτάσεις dsp, dsw, ncb, opt και rlg, τα οποία δεν θα μας απασχολήσουν εδώ.

Β'. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 1 καλύπτονται τα παρακάτω θέματα:

- Εισαγωγή στον προγραμματισμό.
- Μορφή προγράμματος της C.
- Μεταβλητές.
- Μια πρώτη ματιά στην printf().
- Ειδικοί χαρακτήρες: \n \b \r \t \a \\ \' \" %%
- Τύποι int και long. Προσδιοριστές %d, %x, %o και χρήση της printf() με αυτούς.
- Τύπος char. Προσδιοριστής %c. Χρήση της printf() με αυτόν.

Γ'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

1. Να πληκτρολογηθεί το πιο κάτω πρόγραμμα και μετά να εκτελεστεί. Τι θα εμφανίσει στην οθόνη;

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

main( )
{
    printf ("Hello world! \n");
    system ("pause");
}
```

Επεξηγήσεις - υπενθυμίσεις:

```
#include <stdio.h> : Χρήση της βιβλιοθήκης stdio.h
main( ) {           : Κύρια συνάρτηση, αρχή του προγράμματος
printf ("Hello world! \n"); : Εμφάνιση στην οθόνη του Η/Υ
system ("pause");   : Περιμένει το πάτημα ενός πλήκτρου για την
                     συνέχεια
```

2. Να πληκτρολογηθεί και μετά να εκτελεστεί το πιο κάτω πρόγραμμα. Τι θα εμφανίσει στην οθόνη;

```
/* Χρήση της printf */
#include <stdio.h>
#include <stdlib.h>

main( )
{
    printf ("\nΤΕΙ ΚΡΗΤΗΣ\n");
    printf ("ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ\n ΕΦΑΡΜΟΓΩΝ\n");
    system ("pause");
}
```

3. Τι θα εμφανίσει στην οθόνη το πιο κάτω πρόγραμμα; Προσπαθείστε να βρείτε τι θα εμφανιστεί, πριν το εκτελέσετε.

```
/* Χρήση ειδικών χαρακτήρων */
#include <stdio.h>
#include <stdlib.h>
main( )
{
    printf ("\n\n Game over! \n\n");
    printf ("\t That's all folks! \n");
    printf ("\a Beep! \a Beep! \n");
    printf (" \" in double quotes \" \n");
    printf ("file c:\\new\\melody.mp3 \n");
    printf ("ena \r"); printf("dyo \n");
    printf ("xxx \b\b\b yyy \n");
    printf ("visible \0 invisible ");
    printf ("Students are 99%% good \n");
    system("pause");
}
```

Επεξηγήσεις - υπενθυμίσεις:

Τα παρακάτω ζευγάρια χαρακτήρων έχουν ειδική χρήση στην C και συγκεκριμένα:

\n	αλλαγή γραμμής (new line)
\b	backspace, χαρακτήρας οπισθοχώρησης
\r	επιστροφή στην αρχή της γραμμής
\t	χαρακτήρας tab (8 κενά)
\a	beep, κουδούνι, ηχείο του Η/Υ
\\	κάθετος γραμμή
\'	μονά εισαγωγικά (απόστροφος)
\"	διπλά εισαγωγικά
%%	ποσοστό

4. Τι θα εμφανίσει στην οθόνη το πιο κάτω πρόγραμμα;

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

main( )
{
    int num, art;          /* Δηλώσεις μεταβλητών */

    num = 1;
    printf("Ο ΑΡΙΘΜΟΣ ΙΣΟΥΤΑΙ ΜΕ %d\n", num);
    art = num + 1;
    printf("ΑΝ ΠΡΟΤΗΘΕΙΣ 1 ΠΑΙΡΝΟΥΜΕ %d\n", art);
    printf("ΑΤΗΡΟΙΣΜΑ = %d. ΔΙΑΦΟΡΑ = %d\n", num+art, num-art);
    system("pause");
}
```

Επεξηγήσεις - υπενθυμίσεις:

Προσδιοριστής %d. Εκεί που εμφανίζεται θα γραφτεί ένας ακέραιος.

5. Να πληκτρολογηθεί το πιο κάτω πρόγραμμα και μετά να εκτελεστεί. Τι θα εμφανίσει στην οθόνη;

```
/*Αλλα αριθμητικά συστήματα*/
#include <stdio.h>
#include <stdlib.h>

main( )
{
    int dek = 395;

    printf("Ο αριθμός στο δεκαδικό είναι %d\n", dek);
    printf("Ο ίδιος στο οκταδικό είναι ο %o\n", dek);
    printf("και στο δεκαεξαδικό είναι %x\n", dek);
    system("pause");
}
```

Επεξηγήσεις - υπενθυμίσεις:

- **Προσδιοριστής %d.** Εκεί που εμφανίζεται θα γραφτεί ένας ακέραιος του δεκαδικού αριθμητικού συστήματος.
- **Προσδιοριστής %o.** Εκεί που εμφανίζεται θα γραφτεί ένας ακέραιος του οκταδικού αριθμητικού συστήματος.
- **Προσδιοριστής %x.** Εκεί που εμφανίζεται θα γραφτεί ένας ακέραιος του δεκαεξαδικού αριθμητικού συστήματος.

6. Να γραφεί ένα πρόγραμμα που θα κάνει τα εξής:

- Θα δέχεται δύο ακέραιες μεταβλητές, τις `nik` και `nak`, με τιμές 3 και 5 αντίστοιχα.
- Να υπολογίσετε το άθροισμα των μεταβλητών (που θα το πείτε `atr`) και το γινόμενό τους (που θα το πείτε `gin`).
- Να γράφεται στην οθόνη η λέξη `VALUES` και από κάτω 5 αστεράκια. Στην από κάτω γραμμή θα γραφεί η τιμή της `nik`, και στην πιο κάτω η τιμή της `nak`.
- Θα αφήσετε τρεις κενές γραμμές στην οθόνη και από κάτω θα γραφεί:
Αθροισμα = xxxxx Γινόμενο = xxxxx

Στη θέση των xxxxx θα γραφεί η τιμή του αθροίσματος και η τιμή του γινομένου.

Επεξηγήσεις - υπενθυμίσεις:

- **Δηλώσεις μεταβλητών** στη C (κάποιες από αυτές θα συζητηθούν στα επόμενα εργαστήρια) :

```
int i;  
float x;  
long z;  
char ch;
```

- Τα ονόματα των μεταβλητών αποτελούνται από γράμματα (αγγλικά): A έως Z (κεφαλαία), και a έως z (πεζά), σύμβολο υπογράμμισης `_` (underscore) και ψηφία 0 έως 9. Ένα όνομα πρέπει να αρχίζει από γράμμα ή σύμβολο υπογράμμισης. Π. χ. τα:

`abc95`, `_2006`, `ABC`, `Very_long_name` είναι σωστά, ενώ τα

`9abc`, `αβγ` είναι λάθος

Προσοχή! Στη C τα κεφαλαία γράμματα ΔΕΝ ταυτίζονται με τα μικρά!

Τα ακόλουθα ονόματα είναι όλα διαφορετικά στη C: `abc`, `Abc`, `aBc`, κλπ.

- **Εντολή εκχώρησης** (assignment statement) :

```
i = 5;  
i = i + 10;  
x = 3.1415;
```

7. Το πιο κάτω πρόγραμμα εμφανίζει στην οθόνη τις τιμές τριών ακεραίων. Ποιές τιμές και γιατί; Να εκτελέσετε το πρόγραμμα.

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

```
main( )  
{  
    int ak = 2147483647;  
  
    printf("%d %d %d\n", ak, ak+1, ak+2);  
    system("pause");  
}
```

Επεξηγήσεις - υπενθυμίσεις:

- Η τήρηση των ορίων των τιμών των μεταβλητών είναι ευθύνη του προγραμματιστή.
- Στα 32-bit συστήματα τύπος **int** ταυτίζεται με **long** (4 bytes).

8. Γράψτε και εκτελέστε το πιο κάτω πρόγραμμα. Τι θα εμφανίσει στην οθόνη;

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

main( )
{
    char c, d, ch = 'd';

    c = '!';
    d = c;
    printf("\n");
    printf("%c\n", ch);
    printf("%c\n", c);
    printf("%c\n", d);
    system("pause");
}
```

Επεξηγήσεις - υπενθυμίσεις:

Μεταβλητές τύπου **χαρακτήρα (char)**: ένας (ακριβώς) χαρακτήρας, π.χ. 'A', 'B', 'C', '2', '3', '#', '␣' κλπ

Δ'. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

1. Να γράψετε ένα πρόγραμμα, στο οποίο να δίνονται τιμές σε δύο ακέραιες μεταβλητές, τις οποίες θα εμφανίζετε και στην οθόνη. Στη συνέχεια να εναλλάσσονται οι τιμές των μεταβλητών και οι εναλλαγμένες τιμές να εμφανίζονται και πάλι στην οθόνη.
2. Στα παλαιά χρόνια (όταν η μνήμη του Η/Υ ήταν πολύτιμη), κάποιος βρήκε την εξής "αριθμητική" λύση για την αντιμετάθεση τιμών, στην οποία αναφέρεται η πιο πάνω άσκηση: **a=a+b; b=a-b; a=a-b;** Είναι σωστή;

ΕΡΓΑΣΤΗΡΙΟ 2

Α'. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 2 καλύπτονται τα παρακάτω θέματα:

- Τύποι float και double. Προσδιοριστές %f και %lf. Χρήση της printf().
- Η συνάρτηση scanf().
- Σταθερές.
- Πίνακες (εισαγωγικά).

Β'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

float:

1. Να πληκτρολογηθεί το πιο κάτω πρόγραμμα και μετά να εκτελεστεί. Τι θα εμφανίσει στην οθόνη;

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

main( )
{
    float x = 67.1256;

    x = 67.1256;
    printf ("x=%12.4f\n",x);
    printf ("x=%9.3f\n",x);
    printf ("x=%8.5f\n",x);
    printf ("x=%0.2f\n",x);
    printf ("x=%0.0f\n",x);
    system ("pause");
}
```

Επεξηγήσεις - υπενθυμίσεις:

- Οι τύποι **float** και **double** έχουν κλασματικό μέρος (δεκαδικά).
- Η διαφορά είναι στην κατανάλωση της μνήμης, στον χρόνο εκτέλεσης των πράξεων και στην printf.
- Στα 32-bit συστήματα ένας float καταλαμβάνει 4 byte, ενώ double 8 byte.

- Βασικά, το θέμα είναι ότι double αποθηκεύει περισσότερα δεκαδικά. Στα 32-bit συστήματα: float = 8 δεκαδικά, ενώ double = 16 δεκαδικά.
- Στο παραπάνω πρόγραμμα εμφανίστηκε πρόσθετη μορφοποίηση %t.d, πχ. %12.4f, ο αριθμός t παριστάνει θέσεις και d τα δεκαδικά. Δηλαδή, ο αριθμός τυπώνεται σε t θέσεις (με στοίχιση προς τα δεξιά) από τις οποίες οι d για τα δεκαδικά.

scanf():

2. Να πληκτρολογηθεί το πιο κάτω πρόγραμμα και μετά να εκτελεστεί. Τι θα εμφανίσει στην οθόνη;

```
#include <stdio.h> // printf, scanf
#include <stdlib.h> // system

main( )
{
    int i;

    printf ("give number -> ");
    scanf ("%d", &i); // προσέξτε το & (εμπορικό "και")
    printf ("your number is %d\n", i); // όχι το & εδώ
    system ("pause");
}
```

Επεξηγήσεις - υπενθυμίσεις:

- Η συνάρτηση **scanf** διαβάζει από το πληκτρολόγιο.

3. Να πληκτρολογηθεί το πιο κάτω πρόγραμμα και μετά να εκτελεστεί. Τι θα εμφανίσει στην οθόνη;

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

main( )
{
    float salary;

    printf ("Τι μισθό θέλετε;");
    printf ("Συμπληρώστε (δρχ.) _____\b\b\b\b\b\b\b\b\b\b");
    scanf ("%f", &salary);
    printf ("\n%5.0f δρχ. το μήνα είναι %7.0f δρχ. τον χρόνο\n",
                                                salary, 12*salary );
}
```

4. Να γραφεί ένα πρόγραμμα, το οποίο θα κάνει τα εξής:

- Γράφει στην οθόνη: Δώστε μια θερμοκρασία σε Φαρενάιτ.
- Περιμένει δίπλα να διαβάσει μιαν ακέραια τιμή, την οποία αποδίδει στη μεταβλητή ftemp.
- Υπολογίζει την αντίστοιχη θερμοκρασία σε βαθμούς Κελσίου. Ο τύπος της μετατροπής είναι $C = (F - 32) * 5/9$.
- Γράφει στην οθόνη: Η θερμοκρασία είναι ----- βαθμοί Κελσίου.
- (Αντί για τις παύλες να γράψετε τους βαθμούς Κελσίου).

Σταθερές:

5. Παρακολουθείστε στο παρακάτω πρόγραμμα την χρήση σταθερών:

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

#define PI 3.14159
#define EURO 340.75

const int TEN = 10;
const float E = 2.71828;

main( )
{
    printf ("pi = %f \n",PI);
    printf ("euro = %f drx.\n", EURO);
    printf ("10 = %d \n", TEN);
    printf ("Eulers constant = %f \n", E);
    system ("pause");
}
```

Επεξηγήσεις - υπενθυμίσεις:

- Οι **σταθερές** σε ένα πρόγραμμα ορίζονται συνήθως με #define (ή και με const).
- Οι τιμές τους δεν αλλάζουν σε όλη τη διάρκεια εκτέλεσης του προγράμματος.

6. Να γραφεί ένα πρόγραμμα που θα διαβάζει από το πληκτρολόγιο 4 ακέραιους αριθμούς (τις τιμές δηλαδή 4 ακεραίων μεταβλητών), οι οποίοι αντιπροσωπεύουν το κόστος μιας τηλεόρασης, ενός βίντεο, ενός πλυντηρίου και ενός ψυγείου. Στη συνέχεια θα υπολογίζει τις τιμές με ΦΠΑ, θα αυξάνει δηλαδή τις αρχικές κατά 23% και θα γράφει τα αποτελέσματα στην οθόνη.

Να κάνετε το πρόγραμμά σας πιο λειτουργικό ορίζοντας την τιμή του ΦΠΑ ως σταθερά στην αρχή του προγράμματος, με το όνομα FPA.

Πίνακες:

7. Να πληκτρολογηθεί το πιο κάτω πρόγραμμα και να εκτελεστεί. Παρακολουθείστε τι θα εμφανιστεί στην οθόνη;

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

main( )
{
    int pin[3];

    pin[0] = 22;
    pin[1] = 33;
    pin[2] = 18;
    pin[3] = 55;           // Δεν πρέπει
    printf ("give pin2 -> ");
    scanf ("%d", &pin[2]);
    printf ("pin2 = %d \n", pin[2]);
    system ("pause");
}
```

Επεξηγήσεις - υπενθυμίσεις:

- Οι πίνακες είναι ακολουθίες μεταβλητών του ίδιου τύπου, οι οποίες έχουν αποθηκευτεί σε διαδοχικές θέσεις μνήμης.
 - int pin[100]; // 100 ακέραιοι
 - int math[200]; // 200 float
 - char name[80]; // 80 χαρακτήρες
- Δήλωση μεταβλητής πίνακα: τύπος όνομα[μήκος];
- Οι αναφορές στα στοιχεία ενός πίνακα ξεκινάνε από το μηδέν!
 - pin[0] : πρώτο στοιχείο
 - pin[99] : τελευταίο στοιχείο

Δ'. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

1. Να γραφεί ένα πρόγραμμα, το οποίο θα κάνει τα εξής:

- Γράφει στην οθόνη: Δώστε μια θερμοκρασία σε Φαρενάιτ.
- Περιμένει δίπλα να διαβάσει μιαν ακέραια τιμή, την οποία αποδίδει στη μεταβλητή `ftemp`.
- Υπολογίζει την αντίστοιχη θερμοκρασία σε βαθμούς Κελσίου. Ο τύπος της μετατροπής είναι $C = (F - 32) * 5/9$.
- Γράφει στην οθόνη: Η θερμοκρασία είναι ----- βαθμοί Κελσίου.

(Αντί για τις παύλες να γράψετε τους βαθμούς Κελσίου).

2. Να γραφεί ένα πρόγραμμα, το οποίο θα υπολογίζει το φόρο που πληρώνει ένας μισθωτός, ως εξής:

- Γράφει στην οθόνη: Δώστε το μισθό.
- Περιμένει δίπλα να διαβάσει μια `float` τιμή, την `mis`, η οποία αντιπροσωπεύει το μισθό του μισθωτού.
- Υπολογίζει το συνολικό μισθό για ένα έτος.
- Γράφει στην οθόνη: Δώστε τα έξοδα.
- Περιμένει δίπλα να διαβάσει μια `float` τιμή, την `exd`, η οποία αντιπροσωπεύει τα έξοδα που κάνει ο μισθωτός σε ένα μήνα.
- Υπολογίζει τα συνολικά έξοδα για ένα έτος.
- Υπολογίζει το φορολογητέο ποσόν `fr`, που ισούται με το συνολικό μισθό μείον τα συνολικά έξοδα.
- Τέλος, υπολογίζει τον φόρο, ο οποίος ισούται με το 10% του φορολογητέου ποσού και τον γράφει στην οθόνη σε χώρο 10 διαστημάτων, με 2 κλασματικά ψηφία.

ΕΡΓΑΣΤΗΡΙΟ 3

Α'. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 3 καλύπτονται τα παρακάτω θέματα:

- Τελεστές: =, +, -, *, /, %, ++, --, +=, -=, *=, /=, %= .
- Προτεραιότητα τελεστών.
- Μετατροπές τύπων: αυτόματες και casting.
- Συμβολοσειρές γενικά.
- Χρήση του %s στην scanf () και στην printf ().
- Τιμή επιστροφής συνάρτησης (γενικά) και μήκος συμβολοσειράς, συνάρτηση strlen ().

Β'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

Τελεστές γενικά:

1. Να πληκτρολογηθεί και να εκτελεστεί το πιο κάτω πρόγραμμα. Τι θα δώσει στην οθόνη;

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

main( )
{
    printf("5/4 δίνει %d\n", 5/4);
    printf("25/7 δίνει %d\n", 25/7);
    printf("3/6 δίνει %d\n", 3/6);
    printf("7./4. δίνει %8.2f\n", 7./4.);
    printf("7./4 δίνει %6.2f\n", 7./4);
    printf("Το υπόλοιπο της ακέραιας διαίρεσης"
           "39/6 δίνει %d\n", 39%6);
    system ("pause");
}
```

2. Να γραφεί ένα πρόγραμμα, το οποίο θα κάνει τα εξής:
 - Θα γράφει στην οθόνη "Δώστε αριθμό δευτερολέπτων"
 - Θα περιμένει να διαβάσει ένα ακέραιο, τον sec, ο οποίος αντιστοιχεί σε αριθμό δευτερολέπτων.

- Το πρόγραμμα στη συνέχεια θα υπολογίζει από πόσα λεπτά (συμβολίστε τα με min) αποτελείται ο sec και πόσα δευτερόλεπτα περισσεύουν (συμβολίστε τα με left). Αν για παράδειγμα ο sec ισούται με 389, τότε το min θα είναι 6 και το left θα είναι 29.
- Το πρόγραμμα να γράφει στην οθόνη τις τιμές των min και left, την κάθε μια σε χώρο 3 διαστημάτων.

Προτεραιότητα τελεστών:

3. Να εκτελεστεί το πιο κάτω πρόγραμμα. Τι θα δώσει στην οθόνη;

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

main()
{
    int top, score;

    top = score = -(2 + 5) * 6 + (4 + 3 * (2 + 3));
    printf("top = <%8d>\n", top);
    printf("score = <%-7d>\n", score);
    system ("pause");
}
```

Επεξηγήσεις - υπενθυμίσεις:

- Ακέραιος σε χώρο 8 διαστημάτων (στοιχημένος δεξιά): με τον προσδιοριστή %8d. Για στοίχιση αριστερά: προσδιοριστής %-8d.
- Προτεραιότητα τελεστών (από μεγαλύτερη προς μικρότερη):
 - ⇒ () παρενθέσεις
 - ⇒ + - (ως πρόσημα), ++ --
 - ⇒ * / %
 - ⇒ + -
 - ⇒ = *= /= %= += -=

Σε μια παράσταση, οι πράξεις με την ίδια προτεραιότητα υπολογίζονται από αριστερά προς τα δεξιά με τη σειρά που τους συναντούμε.

Μετατροπές τύπων:

4. Να εκτελεστεί το πιο κάτω πρόγραμμα. Τι θα εμφανίσει στην οθόνη;

```
/* Μετατροπές τύπων */
```

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

```
#include <stdlib.h>
```

```
main( )
```

```
{
```

```
    float fl;
```

```
    int ak;
```

```
    printf("Give a float: ");
```

```
    scanf("%f", &fl);
```

```
    ak = (int) fl;
```

```
    printf("%5d\n", ak);
```

```
    system ("pause");
```

```
}
```

Αν αντικαταστήσουμε τον προσδιοριστή %d με %c στο πιο πάνω πρόγραμμα, τι θα πάρουμε στην οθόνη και γιατί;

Επεξηγήσεις - υπενθυμίσεις:

- Οι μετατροπές τύπων είναι είτε **αυτόματες**, π.χ.:

```
float x;
```

```
int n;
```

```
n = 5;
```

```
x = n; // int to float
```

είτε «εξαναγκασμένες» (**casting**):

```
float x;
```

```
int n;
```

```
x = 1.5;
```

```
n = (int) x; // int casting of float, n is 1.
```

- Συμπληρωματικά, υπολογίστε τιμές μεταβλητών :

```
int a; float b;
```

```
a = (12-3)/12 - 3;
```

```
a = 9%5%3%2*5;
```

```
a = 3/4/2;
```

```
a = 3/(4/2);
```

```
a = 2*3/4*5/3;
```

```
b = (int)5.99 + 6.2*2;
```

```
b = 2/4*10.0 + 2%4 + 10;
```

```
b = 3/4;
```

```
b = 3.0/4;
```

```
b = 3./4;
```

```
b = 3/4.0;
```

```
b = 3.0/4.0;
```

Συμβολοσειρές (γενικά):

5. Να εκτελεστεί το πιο κάτω πρόγραμμα. Τι θα εμφανίσει στην οθόνη;

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

main( )
{
    char name[80];

    printf("your name -> ");
    scanf("%s", name);           // Δεν υπάρχει το &
    printf("Hello %s \n", name);
    system ("pause");
}
```

6. Να εκτελεστεί το πιο κάτω πρόγραμμα. Τι θα εμφανίσει στην οθόνη;

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

#define GOOD "EXEIS WRAIO ONOMA"

main( )
{
    char onoma[80];

    printf("PWS SE LENE; ");
    scanf("%s", onoma);
    printf("GEIA SOY %s, %s\n", onoma, GOOD);
    system ("pause");
}
```

Επεξηγήσεις - υπενθυμίσεις:

- Προσδιοριστής **%s** για διάβασμα και εμφάνιση στην οθόνη συμβολοσειρών.
- Εκχώρηση: **char name[8] = "Maria"**; επιτρέπεται μόνο στην δήλωση συμβολοσειράς. Δεν επιτρέπεται ως εντολή : **name = "Maria"**;
- Σταθερές συμβολοσειρές: Γράφονται μέσα σε διπλά εισαγωγικά.

Μήκος συμβολοσειράς:

7. Να εκτελεστεί το πιο κάτω πρόγραμμα. Τι θα εμφανίσει στην οθόνη;

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

```
main( )
{
    char name[80];
    int m;

    printf("what is your name? ");
    scanf("%s", name);
    m = strlen (name);
    printf ("Your name has %d letters \n", m);
    system ("pause");
}
```

8. Γράψτε ένα πρόγραμμα, το οποίο να ζητά από τον χρήστη το όνομα και το επώνυμό του. Στη συνέχεια να εμφανίζει στη μια γραμμή τα ονόματα και στην επόμενη γραμμή, κάτω από το τελευταίο γράμμα του καθενός, τον αριθμό των γραμμάτων του ονόματος και του επωνύμου. Π.χ.:

Νικόλαος Αποστόλου
8 9

Στη συνέχεια να εμφανίζει ξανά τα ίδια, αλλά με τον αριθμό των γραμμάτων κάτω από το αρχικό γράμμα κάθε ονόματος. Π.χ.:

Νικόλαος Αποστόλου
8 9

9. Γράψτε ένα πρόγραμμα, το οποίο να ζητά από τον χρήστη να διαβάσει δύο συμβολοσειρές, τις `pin` και `mat`. Να τις εμφανίζει στην οθόνη. Στη συνέχεια να αντιγράψει την μικρότερη στην μεγαλύτερη και να εμφανίζει και πάλι τις συμβολοσειρές στην οθόνη.

Επεξηγήσεις - υπενθυμίσεις:

- Η συνάρτηση `strlen(s)` **επιστρέφει το μήκος** της συμβολοσειράς `s`.
- Προσοχή στην ύπαρξη του `<string.h>`.
- Με την εντολή `printf ("%d", a, b);` γράφεται στην οθόνη ο ακέραιος `a` σε χώρο `b` διαστημάτων.
- Μια συμβολοσειρά **αντιγράφεται** σε μια άλλη μόνο με τη χρήση της `strcpy (s,t)`; (αναλυτικότερα γι' αυτήν στα επόμενα μαθήματα).

ΕΡΓΑΣΤΗΡΙΟ 4

Α'. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 4 καλύπτονται τα παρακάτω θέματα:

- Συναρτήσεις getchar(), getch(), getche(), putchar().
- Αληθές - ψευδές στην C
- Σχεσιακοί τελεστές: < > <= >= == !=
- Εντολή if και παραλλαγές: if-else, πολλαπλές if, πολλαπλές if-else.
- Απλές και σύνθετες εντολές.

Β'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

Συναρτήσεις getchar(), getch(), getche(), putchar().

1. Να εκτελεστεί το πιο κάτω πρόγραμμα. Τι θα εμφανίσει στην οθόνη;

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

main( )
{
    char ch;

    printf ("Type some letter and press <enter> -> ");
    ch = getchar( );
    printf ("You typed %c \n", ch);
    putchar ('B');
    putchar ('Y');
    putchar ('E');
    putchar ("\n");
    system ("pause");
}
```


2. Να εκτελεστεί το πιο κάτω πρόγραμμα. Τι θα εμφανίσει στην οθόνη;

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

main( )
{
    char ch;

    printf("Type some letter -> ");
    ch = getche( );
    printf("\n");
    printf("You typed %c \n", ch);
    system ("pause");
}
```

3. Να γράψετε ένα πρόγραμμα, το οποίο θα ζητά από τον χρήστη να δώσει τον κωδικό του για κάποια εργασία. Ο κωδικός υποτίθεται ότι αποτελείται από τρεις χαρακτήρες. Το πρόγραμμα θα διαβάζει ένα-ένα χαρακτήρα, ο οποίος όμως δεν θα εμφανίζεται στην οθόνη, αλλά αντί γι' αυτόν θα εμφανίζεται ένα αστεράκι. Αφού διαβαστούν οι τρεις χαρακτήρες να γραφεί στην οθόνη: "YOUR PIN IS" και δίπλα να γραφεί ο κωδικός που δόθηκε

Επεξηγήσεις - υπενθυμίσεις:

- **ch = getchar();** διαβάζει χαρακτήρα ch με πλήκτρο <enter>. Είναι ισοδύναμη με scanf ("%c", &ch);
- **putchar(ch)** γράφει τον χαρακτήρα ch. Είναι ισοδύναμη με printf ("%c", ch);
- Βιβλιοθήκη <conio.h>.
- **ch = getche();** διαβάζει χαρακτήρα ch χωρίς πλήκτρο <enter>.
- **ch = getch();** διαβάζει χαρακτήρα ch χωρίς πλήκτρο <enter> αλλά ΔΕΝ τον εμφανίζει στην οθόνη.

Εντολή if και παραλλαγές της.

4. Να γραφεί ένα πρόγραμμα που θα κάνει τα εξής:
Θα διαβάζει από το πληκτρολόγιο ένα ακέραιο, τον the, ο οποίος αντιπροσωπεύει μια θερμοκρασία.
Εάν ο the είναι έξω από την περιοχή τιμών -30 έως 50, τότε θα γράφεται στην οθόνη "ERROR"
Εάν ο the είναι μεταξύ 40 και 50, τότε θα γράφεται στην οθόνη "VERY HOT".
Εάν ο the είναι μεταξύ 10 και 40, τότε θα γράφεται στην οθόνη "GOOD WEATHER"

Εάν ο the είναι κάτω από 10 (αλλά προφανώς πάνω από -30), τότε θα γράφεται στην οθόνη "COLD".

5. Να γραφεί ένα πρόγραμμα που θα διερευνά εάν ένα τρίγωνο είναι ορθογώνιο ως εξής:

Θα διαβάζει από το πληκτρολόγιο τρεις ακεραίους, τους a, b και c, οι οποίοι αντιπροσωπεύουν τις πλευρές του τριγώνου.

Θα ελέγχει εάν το τετράγωνο κάποιας πλευράς ισούται με το άθροισμα των τετραγώνων των άλλων, οπότε το τρίγωνο είναι ορθογώνιο.

Εάν διαπιστωθεί ότι το τρίγωνο είναι ορθογώνιο, τότε στην οθόνη θα γράφονται:

"ORTHOGWNIO. KATHETES PLEYRES OI"

και δίπλα ποιές είναι οι κάθετες πλευρές του τριγώνου.

6. Να γραφεί ένα πρόγραμμα που θα υπολογίζει βαθμούς ενός σπουδαστή ως εξής:

- Το πρόγραμμα αρχικά θα γράφει στην οθόνη: "DWSTE VATHMOYS ERGASTHRIΟΥ KAI EKSETASTIKHS" και θα περιμένει να διαβάσει δυό float, τους pr και tel, οι οποίοι αντιπροσωπεύουν τους βαθμούς προόδου και εξεταστικής.
- Για να υπολογίσουμε τον ολικό βαθμό της θεωρίας (που λέγεται bm) αθροίζουμε το 40% του pr με το 60% του tel.
- Στη συνέχεια το πρόγραμμα θα γράφει στην οθόνη: "EGINE EKSETASH ERGASTHRIΟΥ?" και περιμένει να διαβάσει ένα χαρακτήρα, τον ch.
- Αν το ch είναι ίσο με N (εννοούμε δηλαδή ΝΑΙ), τότε το πρόγραμμα κάνει τα εξής:
 - ⇒ Περιμένει να διαβάσει ένα float, τον erg, ο οποίος αντιπροσωπεύει το βαθμό εργαστηρίου.
 - ⇒ Υπολογίζει τον ολικό βαθμό του μαθήματος (που λέγεται olikos) και ισούται με το 50% του erg, συν το 50% του bm.
 - ⇒ Γράφει στην οθόνη: "OLIKOS VATHMOS MATHIMATOS" και δίπλα την τιμή του olikos, σε χώρο 5 διαστημάτων με 1 δεκαδικό.
- Αν το ch δεν είναι ίσο με N, τότε το πρόγραμμα γράφει στην οθόνη: "VATHMOS THEWRIAS" και δίπλα την τιμή του bm, σε χώρο 5 διαστημάτων με 1 δεκαδικό.

Επεξηγήσεις - υπενθυμίσεις:

- Στη C οι λογικές συνθήκες έχουν αριθμητικές τιμές: **0** (μηδέν) αν είναι **ψευδής** και **1** αν είναι **αληθής**.
- Η C όμως δέχεται και οποιεσδήποτε αριθμητικές παραστάσεις ως συνθήκες. Μια τέτοια παράσταση **ισχύει** αν έχει **μη μηδενική τιμή** και **δεν ισχύει** μόνο αν η **τιμή της είναι μηδέν**.
- Όταν το πρόγραμμα φτάνει στην εντολή **if**, υπολογίζει την λογική τιμή της συνθήκης. Αν η συνθήκη ισχύει (έχει μη μηδενική τιμή), τότε εκτελείται ομάδα εντολών. Αν η συνθήκη δεν ισχύει, δεν γίνεται τίποτα. Μετά (και στις δύο περιπτώσεις), το πρόγραμμα προχωράει στην επόμενη εντολή του προγράμματος.

Γ'. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

1. Να γραφεί ένα πρόγραμμα που θα κάνει τα εξής:

Θα διαβάζει από το πληκτρολόγιο ένα ακέραιο, τον gwn , ο οποίος αντιπροσωπεύει το μέγεθος μιας γωνίας σε μοίρες. Το πρόγραμμα θα κάνει τα εξής:

- Εάν η gwn είναι μεγαλύτερη από 90, στην οθόνη θα γράφεται "AMVLEIA".
- Εάν η gwn είναι μικρότερη από 90, στην οθόνη θα γράφεται "OKSEIA".
- Εάν η gwn είναι ίση με 90, στην οθόνη θα γράφεται "ORTHI".

Αφού φτιάξετε το πρόγραμμα, να το τροποποιήσετε λαμβάνοντας υπ' όψη σας τα εξής: Αν η gwn που διαβάσατε είναι για παράδειγμα 750, το πρόγραμμα θα γράψει στην οθόνη «Αμβλεία γωνία». Όμως η γωνία αυτή είναι στην πραγματικότητα οξεία, αφού $750=2*360+30$, ισούται δηλαδή στην πράξη με 30 μοίρες. Το πρόγραμμα να αφαιρεί από τη γωνία που δώσατε τα ακέραια πολλαπλάσια του 360 και να βγάζει σωστό αποτέλεσμα.

2. Να γραφεί ένα πρόγραμμα που θα λύνει και θα διερευνά ένα τριώνυμο ως εξής:

- Αρχικά θα γράφεται στην οθόνη: "THE EQUATION IS $ax^2+bx+c=0$ "
- Μετά θα γράφεται: "GIVE a, b AND c" και θα περιμένει να διαβάσει τρεις float τιμές για τα a, b και c
- Εάν ισχύει ότι το a είναι μηδέν και το b είναι μηδέν, τότε θα γράφεται στην οθόνη: "NO EQUATION"
- Εάν ισχύει ότι το a είναι μηδέν και το b είναι διάφορο του μηδενός, τότε θα γράφεται στην οθόνη: "SINGLE ROOT" και δίπλα η τιμή της (η τιμή αυτή είναι $-c/b$).
- Εάν ισχύει ότι το a είναι διάφορο του μηδενός, το b είναι διάφορο του μηδενός και το c είναι μηδέν, τότε θα γράφεται στην οθόνη: "THE ROOTS ARE" και δίπλα οι τιμές τους (οι τιμές αυτές είναι 0 και $-b/a$).
- Εάν ισχύει ότι το a είναι διάφορο του μηδενός, το b είναι διάφορο του μηδενός, το c είναι διάφορο του μηδενός και η διακρίνουσα είναι θετική, τότε θα υπολογίσει τις ρίζες, θα γράψει στην οθόνη:

"TWO REAL ROOTS"

και από κάτω τις τιμές τους, ενώ εάν η διακρίνουσα είναι αρνητική θα γράψει στην οθόνη:

"COMPLEX ROOTS"

και από κάτω τις τιμές τους. (Για τις τιμές αυτές θα υπολογιστεί χωριστά το πραγματικό και το φανταστικό μέρος. Μετά αυτά θα γραφτούν δίπλα-δίπλα, έχοντας ανάμεσά τους το +j ή το -j).

3. Να γραφεί ένα πρόγραμμα που θα λύνει και θα υπολογίζει τον φόρο που θα πληρώσει ένας μισθωτός ως εξής:

Αρχικά διαβάζει από το πληκτρολόγιο ένα float, τον mis , ο οποίος αντιπροσωπεύει το μισθό ενός υπαλλήλου.

- Μετά διαβάζει ένα float, τον en , ο οποίος αντιπροσωπεύει το ενοίκιο που πληρώνει ο υπάλληλος.
- Στη συνέχεια διαβάζει ένα float, τον ex , ο οποίος αντιπροσωπεύει τα έξοδα διαβίωσης του υπαλλήλου.

- Το φορολογητέο ποσόν (fr) υπολογίζεται εάν από τους 14 μισθούς ενός έτους αφαιρέσουμε 12 ενοίκια και τα έξοδα διαβίωσης. Το fr δεν θα επιτρέπεται να είναι αρνητικό. Εάν είναι, τότε τίθεται ίσο με μηδέν.
- Ο φόρος υπολογίζεται από το fr ως εξής:
 - ⇒ Εάν το fr είναι μέχρι 10000, τότε ο φόρος θα είναι ίσος με το 10% του fr.
 - ⇒ Εάν το fr είναι μεταξύ 10000 και 20000, τότε: οι πρώτες 10000 φορολογούνται με 10%, το δε υπόλοιπο ποσόν (το πάνω από 10000) φορολογείται με 15%
 - ⇒ Εάν το fr είναι πάνω από 20000, τότε: οι πρώτες 10000 φορολογούνται με 10%, οι δεύτερες 10000 φορολογούνται με 15%, το δε υπόλοιπο ποσόν (το πάνω από 20000) φορολογείται με 20%
- Εάν ο φόρος που υπολογίζεται είναι έξω από τα όρια 1000 έως 30000, τότε στην οθόνη γράφεται: "AKRAIA EISODHMATA"
- Εάν εξ άλλου ο φόρος που υπολογίζεται είναι μηδέν, τότε στην οθόνη γράφεται: "ANEILIKRINHS DHLWSH". Στην περίπτωση αυτή ο φόρος τίθεται ίσος με 7 μισθούς (σαν ποινή για την ψεύτικη δήλωση).
- Σε κάθε περίπτωση, ο φόρος γράφεται στην οθόνη σε χώρο 10 διαστημάτων με 2 δεκαδικά.

ΕΡΓΑΣΤΗΡΙΟ 5

Α'. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 5 καλύπτονται τα παρακάτω θέματα:

- Εντολή **switch**.
- Επαναληπτικές εντολές: **for**

Β'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

Εντολή switch.

1. Στο παρακάτω πρόγραμμα υλοποιείται ένα μενού με την χρήση της switch και της getch(). Να το εκτελέσετε:

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>           // Για την getch( )

main( )
{
    char ch;

    printf("Menu : \n");
    printf("1 – Play \n");
    printf("2 – Work \n");
    printf("3 – EXIT \n");
    printf("\n Your choice -> "); ch = getch();
    printf("\n\n");
    switch (ch)
    {
        case '1' : { printf("Now playing ... \n"); break; }
        case '2' : { printf("Now working ... \n"); break; }
        case '3' : { printf("Bye ... \n"); break; }
        default : { printf("Bad choice\n"); break; }
    }
    system ("pause");
}
```

Επεξηγήσεις - υπενθυμίσεις:

- Όταν χρειάζονται πολλές διακλαδώσεις στο πρόγραμμα, η χρήση των if else γίνεται δύσκολη. Η εντολή switch οργανώνει καλύτερα τέτοιες περιπτώσεις.
- Η εντολή switch χρησιμοποιείται ως εξής :

```
switch (c)
{
    case c1 : Ομάδα Εντολών 1
    case c2 : Ομάδα Εντολών 2
    ...
    case cN : Ομάδα Εντολών N
    default : Ομάδα Εντολών του default
}
```

- Η **break** προκαλεί άμεσο τερματισμό της switch.

Επαναληπτικές εντολές: for.

2. Να γραφεί ένα πρόγραμμα που θα εμφανίζει στην οθόνη τους ακέραιους αριθμούς από το 1 έως το 100, τον καθένα σε χώρο 5 διαστημάτων.
3. Να γραφεί ένα πρόγραμμα που θα αθροίζει τους ακέραιους αριθμούς από το 1 έως το 100, και θα εμφανίζει το αποτέλεσμα στην οθόνη.
4. Να γραφεί ένα πρόγραμμα, το οποίο θα εμφανίζει στην οθόνη όλους τους χαρακτήρες του κώδικα ASCII με τη σειρά.
5. Να γραφεί ένα πρόγραμμα, το οποίο θα γράφει στην οθόνη τους αύξοντες αριθμούς των χαρακτήρων του κώδικα ASCII από τον χαρακτήρα z μέχρι τον χαρακτήρα #. Ο κάθε αριθμός να γραφεί στην οθόνη σε χώρο 5 διαστημάτων.
6. Να γραφεί ένα πρόγραμμα, το οποίο θα διαβάζει 30 float αριθμούς από το πληκτρολόγιο. Το πρόγραμμα θα μετρά πόσοι από τους αριθμούς αυτούς είναι θετικοί, θα τους αθροίζει (τους θετικούς) και θα γράφει το αποτέλεσμα στην οθόνη.
7. Θεωρείστε τις παρακάτω σειρές:

$$1.0 + 1.0/2.0 + 1.0/3.0 + 1.0/4.0 + \dots$$

$$1.0 - 1.0/2.0 + 1.0/3.0 - 1.0/4.0 + \dots$$

Γράψτε ένα πρόγραμμα, το οποίο να υπολογίζει το συνολικό άθροισμα των δύο σειρών μέχρι κάποιο αριθμό όρων. Το πρόγραμμα να ζητάει και να διαβάζει αυτό τον αριθμό από το πληκτρολόγιο. Εξετάστε τα συνολικά αθροίσματα μετά από 20 όρους, 100 όρους και 500 όρους. Φαίνεται να συγκλίνουν αυτές οι σειρές σε κάποια τιμή;

8. Γράψτε ένα πρόγραμμα, το οποίο να διαβάζει μια συμβολοσειρά και μετά να την εμφανίζει στην οθόνη ανάποδα, δηλαδή από το τέλος προς την αρχή. Π.χ. αν διαβαστεί η λέξη ΓΙΑΝΝΗΣ, στην οθόνη θα εμφανιστεί ΣΗΝΝΑΙΓ.
9. Γράψτε ένα πρόγραμμα, το οποίο να αθροίζει μεταξύ τους τα στοιχεία ενός πίνακα ακεραίων 10 θέσεων, ο οποίος λέγεται mat. Το πρόγραμμα επίσης να μετρά πόσα από τα στοιχεία του πίνακα είναι θετικοί ακέραιοι. Να εμφανίζονται τα αποτελέσματα στην οθόνη.

Επεξηγήσεις - υπενθυμίσεις:

- Η **for** έχει την εξής μορφή:

for (Start; Συνθήκη; Next) Ομάδα Εντολών

Π.χ. **for (i=1; i<=3; i++) printf("%3d",i);** // τυπώνει 001002003

Στην πιο γενική μορφή, η **Start** ορίζει την αρχική τιμή της μεταβλητής (ή των μεταβλητών) και εκτελείται μόνο μια φορά. Η **Συνθήκη** ελέγχει τον τερματισμό της επανάληψης και η **Next** περιγράφει πως προχωράμε στο επόμενο βήμα της επανάληψης.

- Αν για παράδειγμα δηλώσουμε τη συμβολοσειρά char nom[30] τότε το κάθε nom[k] είναι ένας χαρακτήρας.
- Το μήκος μιας συμβολοσειράς δίνεται από την συνάρτηση strlen().

Γ'. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

1. Να γραφεί ένα πρόγραμμα, το οποίο θα διαβάζει 30 float αριθμούς από το πληκτρολόγιο. Το πρόγραμμα θα υπολογίζει την τυπική απόκλιση (standard deviation) αυτών των αριθμών. Η τυπική απόκλιση είναι η τετραγωνική ρίζα της εξής διαφοράς: μέση τιμή των τετραγώνων των αριθμών μείον το τετράγωνο της μέσης τιμής. Το αποτέλεσμα να γράφεται στην οθόνη.
Για την εύρεση της τετραγωνικής ρίζας να χρησιμοποιήσετε τη συνάρτηση sqrt(), Πρέπει να υπάρχει το αρχείο επικεφαλίδας math.h.
2. Γράψτε ένα πρόγραμμα, το οποίο θα κάνει τα εξής:
Θα διαβάζει ακέραιους από το πληκτρολόγιο και θα γεμίζει ένα πίνακα ακεραίων 20 θέσεων. Στη συνέχεια θα βρίσκει και θα γράφει στην οθόνη τον μέγιστο και τον ελάχιστο ακέραιο του πίνακα.

ΕΡΓΑΣΤΗΡΙΟ 6

Α'. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 6 καλύπτονται τα παρακάτω θέματα:

- Επαναληπτικές εντολές **while**, **do-while**.
- Προτάσεις **break** και **continue**.
- **Εμφωλευμένες** επαναληπτικές εντολές (**for μέσα σε for** κλπ).

Β'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

Επαναληπτικές εντολές: while, do-while.

Να γράψετε τα παρακάτω προγράμματα χρησιμοποιώντας είτε τη μία είτε την άλλη επαναληπτική εντολή είτε και τις δύο.

1. Να γραφεί ένα πρόγραμμα, το οποίο να διαβάζει μια συμβολοσειρά (π.χ. την wrd) από το πληκτρολόγιο. Στη συνέχεια, το πρόγραμμα θα μετρά το πλήθος των χαρακτήρων της wrd (χωρίς τη χρήση της strlen()). Το πρόγραμμα δηλαδή θα ελέγχει όλους τους χαρακτήρες της wrd. Όσο δεν συναντά το \x0, θα αυξάνει ένα μετρητή. Μόλις συναντήσει το \x0, το πρόγραμμα σταματά αφού γράψει στην οθόνη την τιμή του μετρητή.
2. Να γραφεί ένα πρόγραμμα, το οποίο να διαβάζει συνεχώς χαρακτήρες από το πληκτρολόγιο, μέχρι να δοθεί ο χαρακτήρας A (με την getch() ή την getche()). Το πρόγραμμα μετρά τους χαρακτήρες που διαβάζει αυξάνοντας ένα μετρητή. Αφού ολοκληρωθεί το διάβασμα, γράφει στην οθόνη την τιμή του μετρητή.
3. Να γραφεί πρόγραμμα, το οποίο θα δέχεται μια σταθερά, την MHN με τιμή JOB IN PROCESS. Θα εμφανίζει στην οθόνη την τιμή της MHN και μετά την φράση:
CONTINUE? (Y/N)
Το πρόγραμμα θα περιμένει να διαβάσει ένα χαρακτήρα από το πληκτρολόγιο (συνάρτηση getch()) και θα συνεχίζεται όσο ο χαρακτήρας που διαβάστηκε δεν είναι το N. Σε αυτή την περίπτωση:
 - Εάν ο χαρακτήρας που πατήθηκε δεν είναι Y, τότε γράφεται στην οθόνη:
ERROR. GIVE ANOTHER CHARACTER
 - Εάν ο χαρακτήρας που πατήθηκε είναι Y, γράφεται στην οθόνη η τιμή του MHN, από κάτω η φράση:
CONTINUE? (Y/N)
και αυξάνεται ένας μετρητής (ας πούμε ότι λέγεται count).
Αφού ολοκληρωθούν οι επαναλήψεις, στην οθόνη γράφεται η τιμή του count, δηλαδή ένας ακέραιος, ο οποίος μας δείχνει πόσες φορές δώσαμε το Y.

Επεξηγήσεις - υπενθυμίσεις:

- Η while συντάσσεται: **while (Συνθήκη) Ομάδα Εντολών**
- Όταν το πρόγραμμα φτάνει στην while υπολογίζει **πρώτα** την λογική τιμή της συνθήκης. Αν η συνθήκη ισχύει, τότε εκτελείται η ομάδα

εντολών. Μετά η τιμή της συνθήκης υπολογίζεται ξανά. Αν ισχύει, τότε εκτελείται ξανά η ομάδα εντολών κ.ο.κ. Αν η συνθήκη ΔΕΝ ισχύει, τότε η επανάληψη τερματίζεται και το πρόγραμμα προχωράει στην επόμενη εντολή του προγράμματος.

- Η do-while εκτελεί την ομάδα εντολών και **μετά** υπολογίζει την τιμή της συνθήκης. Συντάσσεται: **do Ομάδα Εντολών while (Συνθήκη);**
- Στη while η συνθήκη μπορεί να είναι ψευδής αμέσως στην αρχή, με συνέπεια να μην εκτελεστεί η ομάδα εντολών της while καθόλου!
- Επειδή στη do while η συνθήκη υπολογίζεται μετά, η ομάδα εντολών της do while **εκτελείται τουλάχιστον μία φορά.**

Προτάσεις break και continue.

4. Εκτελέστε το παρακάτω πρόγραμμα:

```
#include <stdio.h>
#include <stdlib.h>
main( )
{
    int i;
    i = 5;
    while (i<5) { i++; }           // ΔΕΝ εκτελείται
    i = 5;
    do { i++; } while(i<5);       // ΜΙΑ φορά μόνο
    for (i=1; i<=10; i++)         // 1,2,3,4,5 μόνο
    {
        printf("%d\n", i);
        if (i==5) break;
    }
    for (i=-5; i<=5; i++)         // ΟΧΙ διαίρεση με το μηδέν
    {
        if (i==0) continue;
        printf("%f\n", 1.0/i);
    }
    for (i=1; i<10; i+=2)
        printf("%d\n", i);       // Τι τυπώνεται;
    for (i=10; i>0; i-=3)
        printf("%d\n", i);       // Τι τυπώνεται;
    system("pause");
}
```

5. Γράψτε ένα πρόγραμμα, το οποίο θα κάνει τα εξής:
Θα διαβάζει 20 χαρακτήρες από το πληκτρολόγιο, χρησιμοποιώντας τη συνάρτηση `getch( )` ή την `getche( )`. Σε κάθε επανάληψη θα γράφει:

DWSTE TON XARAKTHRA

Στη θέση των κενών την πρώτη φορά που θα διαβάζει θα γράφει «1ο», τη δεύτερη φορά θα γράφει «2ο», την τρίτη φορά «3ο» κλπ. Το πρόγραμμα θα μετρά χωριστά τους κεφαλαίους χαρακτήρες που δώσαμε και χωριστά τους μικρούς. Το διάβασμα θα διακόπτεται πριν την ολοκλήρωση των επαναλήψεων, εάν κάποια φορά δοθεί ο χαρακτήρας τελεία (.).

Μετά την ολοκλήρωση των επαναλήψεων θα γράφεται στην οθόνη το πλήθος των κεφαλαίων και το πλήθος των μικρών χαρακτήρων που διαβάστηκαν.

6. Γράψτε ένα πρόγραμμα, το οποίο με την χρήση της `continue` θα κάνει ό,τι και η παραπάνω άσκηση 6, αλλά δεν θα μετράει τους χαρακτήρες B και b.

Επεξηγήσεις - υπενθυμίσεις:

- Η **break** διακόπτει οριστικά την εκτέλεση μιας επαναληπτικής εντολής (`for`, `while`, `do-while`).
- Όταν σε μια επανάληψη συναντήσουμε την **continue** δεν εκτελούνται οι εντολές από αυτήν μέχρι το τέλος της επαναληπτικής εντολής (πηγαίνουμε ξανά στην αρχή της επαναληπτικής εντολής).

Εμφωλευμένες επαναληπτικές εντολές (for μέσα σε for κλπ).

7. Γράψτε ένα πρόγραμμα, το οποίο να γεμίζει με γκρι χρώμα μια περιοχή της οθόνης 20 γραμμές επί 40 στήλες.

8. α) Γράψτε ένα πρόγραμμα, το οποίο να εμφανίζει στην οθόνη το εξής:

123456789
123456789
123456789
123456789
123456789

β) Να τροποποιηθεί το παραπάνω για να εμφανίσει:

123456789
23456789
3456789
456789
56789

9. Γράψτε ένα πρόγραμμα, το οποίο να εμφανίζει στην οθόνη το εξής:

1
12
123
1234
12345

Επεξηγήσεις - υπενθυμίσεις:

- Σε κάθε κύκλο της εξωτερικής επανάληψης εκτελούνται όλες οι επαναλήψεις του εσωτερικού βρόχου.
- Ο δεκαεξαδικός αριθμός DB, όταν γραφεί στην οθόνη, "γκριζάρει" χώρο ενός διαστήματος. Η ακολουθία \x δηλώνει σαν δεκαεξαδικό τον αριθμό που ακολουθεί αμέσως μετά. Π.χ. \x25, σημαίνει "ο δεκαεξαδικός 25".

Γ'. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

1. Να γραφεί ένα πρόγραμμα, το οποίο να κάνει τα εξής:
Θα δημιουργεί ένα τυχαίο ακέραιο αριθμό μεταξύ 0 και 100. Για να το κάνει αυτό χρησιμοποιεί τη συνάρτηση rand(), η οποία δημιουργεί ένα τυχαίο ακέραιο. Η rand() χρειάζεται ένα "σπόρο" για να ξεκινήσει την παραγωγή τυχαίων αριθμών. Αυτό το "σπόρο" τον παίρνει από το ρολόι του υπολογιστή μια άλλη συνάρτηση, η srand() με παράμετρο time(NULL). Οι δυο αυτές συναρτήσεις χρειάζονται τα αρχεία επικεφαλίδας (header) time.h και stdlib.h.
Αφού παραχθεί ο τυχαίος ακέραιος (π.χ. ο num), να διαβάζονται ακέραιοι από το πληκτρολόγιο μέχρι να βρεθεί ο num. Όσο δηλαδή ο αριθμός που διαβάζεται (π.χ. ο gs) δεν είναι ίσος με num, γίνονται τα εξής:
 - Αυξάνεται ένας μετρητής, ο cnt.
 - Εάν ο gs είναι μεγαλύτερος από τον num, γράφεται στην οθόνη DWSE MIKROTERO.
 - Εάν ο gs είναι μικρότερος από τον num, γράφεται στην οθόνη DWSE MEGALYTERO.
 - Διαβάζεται καινούργιος ακέραιος από το πληκτρολόγιο.Αφού βρεθεί ο τυχαίος ακέραιος num, να γραφεί στην οθόνη το πλήθος των προσπαθειών που έγιναν, καθώς και ο num.
2. Γράψτε ένα πρόγραμμα, το οποίο, με τη χρήση της while, να αθροίζει μεταξύ τους τα στοιχεία ενός πίνακα ακεραίων 10 θέσεων, ο οποίος λέγεται mat. Το πρόγραμμα επίσης να μετρά πόσα από τα στοιχεία του πίνακα είναι θετικοί ακέραιοι. Να εμφανίζονται τα αποτελέσματα στην οθόνη.
3. Γράψτε ένα πρόγραμμα, το οποίο, να διαβάζει ένα θετικό ακέραιο και στη συνέχεια να τον παρουσιάζει στην οθόνη στο δυαδικό αριθμητικό σύστημα.
4. Παραλλαγή της πιο πάνω άσκησης 7: Να ζητάει το πρόγραμμα τη στήλη από όπου θα αρχίσει το "γκριζάρισμα", καθώς και το μέγεθος της γκριζας περιοχής (πλήθος γραμμών και στηλών).

5. Γράψτε πρόγραμμα το οποίο εμφανίζει τρίγωνα μορφής:

**

*

*

**

**

*

ΟΧΙ ΓΙΑ ΠΩΛΗΣΗ. ΔΩΡΕΑΝ ΒΟΗΘΗΜΑ ΣΕ ΦΟΙΤΗΤΕΣ ΕΠΠ

ΕΡΓΑΣΤΗΡΙΟ 7

Α'. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 7 καλύπτονται τα παρακάτω θέματα:

- Συναρτήσεις με ορίσματα μεταβλητές, τιμή επιστροφής συνάρτησης.
- Αναδρομικές συναρτήσεις.

Β'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

Συναρτήσεις με ορίσματα μεταβλητές, τιμή επιστροφής συνάρτησης.

1. Να παρατηρήσετε το παρακάτω πρόγραμμα και να δείτε τι κάνει πριν το εκτελέσετε (υπάρχουν σχετικά σχόλια). Στη συνέχεια να τοποθετήσετε τον ορισμό των συναρτήσεων πριν την main() και να το εκτελέσετε ξανά.

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

void BadNews( );
void pri (int);
int myread( );
int myfunc (int);
float func2 (int, float);

main( )
{
    int m;
    BadNews( );           // Κλήση της BadNews
    BadNews( );           // Κλήση της BadNews
    pri(3);                // Κλήση της pri με σταθερά
    m=5;
    pri(m);                // Κλήση της pri με μεταβλητή
    pri( 4*m+8 );          // Κλήση της pri με έκφραση
    m = myread( );         // Κλήση της myread, αποτελέσματα στο m
    m = myfunc(10);        // Κλήση με σταθερά, μεταβλητή ή έκφραση
    pri(m);                // Εμφάνιση αποτελέσματος
    printf("func2 = %f \n", func2(4,3.14) );    // Εμφανίζει 7.14
    system("pause");
}
```

```

void BadNews( )                // void, χωρίς παραμέτρους
{
    printf(" You have a VIRUS \n");
}

void pri (int n)                // void, μία παράμετρος
{
    printf("Number is %d\n", n );
}

int myread( )                  // Επιστρέφει int. Χωρίς παραμέτρους.
{                               // Καλεί την pri(), άρα πρέπει να
    int n;                     // βρίσκεται META την pri
    printf("give number -> ");
    scanf("%d",&n);
    pri(n);
    return n;
}

int myfunc (int n)             // Επιστρέφει int. Μία παράμετρος.
{
    int a;
    a = 7*n+8;
    return a;
}

float func2 (int n, float x)   // Επιστρέφει float. Δύο παράμετροι.
{
    return x+n;
}

```

2. Να γραφεί μία συνάρτηση, η οποία λέγεται computeraki(). Η συνάρτηση θα δέχεται ως ορίσματα δύο ακραίους, τους a και b και ένα χαρακτήρα, τον ch. Η συνάρτηση θα κάνει τα εξής:

- Εάν ο ch είναι +, θα υπολογίζει και θα επιστρέφει στην main() το a + b.
- Εάν ο ch είναι -, θα υπολογίζει και θα επιστρέφει στην main() το a - b.
- Εάν ο ch είναι *, θα υπολογίζει και θα επιστρέφει στην main() το a * b.
- Εάν ο ch είναι /, θα υπολογίζει και θα επιστρέφει στην main() το a / b.

Στην main() θα διαβάζονται δύο ακέραιοι και θα καλείται η συνάρτηση για να κάνει κάποια από τις πιο πάνω πράξεις.

3. Να γραφεί μία συνάρτηση, η οποία λέγεται power. Η συνάρτηση θα δέχεται ως όρισμα ένα ακέραιο (τον ak) και ένα float (τον bs). Η συνάρτηση θα υπολογίζει και θα επιστρέφει στη main() το bs υψωμένο στη δύναμη ak. Το ak θα μπορεί να είναι οποιοσδήποτε ακέραιος, θετικός, αρνητικός ή μηδέν.

Να γράψετε ένα πρόγραμμα, το οποίο θα ζητάει και θα διαβάζει από το πληκτρολόγιο ένα ακέραιο και ένα float και χρησιμοποιώντας τη συνάρτηση θα υπολογίζει τον ακέραιο υψωμένο στη float δύναμη.

4. Να γράψετε ένα πρόγραμμα, το οποίο θα ζητά και θα διαβάζει από το πληκτρολόγιο τρεις ακεραίους, τους a , b και c , οι οποίοι υποτίθεται ότι αντιπροσωπεύουν τις τρεις πλευρές ενός τριγώνου. Για να υπάρχει τρίγωνο με αυτά τα μήκη πλευρών, πρέπει οποιαδήποτε από αυτές να είναι μικρότερη από το άθροισμα των άλλων δύο και μεγαλύτερη από τη διαφορά τους. Τα μήκη θα περνούν σαν ορίσματα σε μια συνάρτηση, την `embadon()`. Εάν μπορούν να αποτελέσουν πλευρές τριγώνου, τότε η συνάρτηση θα επιστρέφει το εμβαδόν του τριγώνου. Το εμβαδόν βρίσκεται από τον τύπο:

Τετραγωνική ρίζα ($t(t-a)(t-b)(t-c)$)

όπου t η ημιπερίμετρος του τριγώνου. Εάν οι τρεις ακέραιοι δεν αποτελούν τρίγωνο, τότε η συνάρτηση `embadon()` να επιστρέφει τιμή 0.

Το πρόγραμμα να συνεχίζεται, όσο και οι τρεις ακέραιοι που διαβάζονται δεν ξεπερνούν το 100. Κάθε φορά στην οθόνη θα γράφεται το εμβαδόν του τριγώνου που υπολογίστηκε ή η φράση:

Οι ακέραιοι δεν αποτελούν τρίγωνο

(στη θέση των κενών διαστημάτων θα γράφονται κατά σειρά οι τρεις ακέραιοι που διαβάστηκαν).

Επεξηγήσεις - υπενθυμίσεις:

- Για τον υπολογισμό δύναμης, η C διαθέτει την συνάρτηση `pow()` στο `math.h`. Για περισσότερη εξάσκησή σας να μη την χρησιμοποιήσετε.
- Για τον υπολογισμό της τετραγωνικής ρίζας, η C διαθέτει την συνάρτηση `sqrt()` στο `math.h`.

Αναδρομικές συναρτήσεις.

5. Να παρατηρήσετε το παρακάτω πρόγραμμα και να δείτε τι κάνει πριν το εκτελέσετε

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

main()
{
    test(5);                // «Πυροδοτεί» 6 κλήσεις της test()
    system("pause");
}

void test (int n)
{
    printf ("Starting test with n = %d \n", n);
    if (n>0)
        test( n-1 ); // restricted recursion
    printf("Ending test with n = %d \n", n);
}
```

6. Προσπαθείστε να γράψετε ένα πρόγραμμα, στο οποίο θα διαβάσετε ένα ακέραιο, έστω τον n και στη συνέχεια να υπολογίζετε και να εμφανίζετε στην οθόνη το $n!$. Ο υπολογισμός μπορεί να γίνει είτε με την χρήση επαναληπτικής εντολής είτε με την χρήση αναδρομικής συνάρτησης.
7. Να γράψετε ένα πρόγραμμα, το οποίο υπολογίζει τους n πρώτους αριθμούς της σειράς Fibonacci. Το n είναι ακέραιος και θα διαβάζεται από το πληκτρολόγιο. Η σειρά Fibonacci είναι αυτή στην οποία κάθε αριθμός ισούται με το άθροισμα των δύο προηγούμενων, με τις πρώτες δύο τιμές της σειράς να είναι το 0 και το 1.

Επεξηγήσεις - υπενθυμίσεις:

- «**Αναδρομικές**» λέγονται οι συναρτήσεις, οι οποίες καλούν τον εαυτό τους.
- Κάθε αναδρομική συνάρτηση **περιλαμβάνει μια if** για τον καθορισμό της συνθήκης τερματισμού.

Γ'. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

1. (Η άσκηση είναι παραλλαγή της άσκησης B2 του εργαστηρίου 6) Να γραφεί ένα πρόγραμμα, στη `main()` του οποίου θα διαβάζεται ένας χαρακτήρας από το πληκτρολόγιο, έστω ο `ch`. Στη συνέχεια θα καλείται μια συνάρτηση, η οποία να διαβάζει συνεχώς χαρακτήρες από το πληκτρολόγιο, μέχρι να δοθεί ο χαρακτήρας `ch` (με την `getche()` ή την `getch()`). Η συνάρτηση μετρά τους χαρακτήρες που διαβάζει αυξάνοντας ένα μετρητή, επιστρέφει το πλήθος τους στη `main()`, η οποία και το γράφει στην οθόνη.
2. (Η άσκηση είναι παραλλαγή της άσκησης Γ1 του εργαστηρίου 5) Να γραφεί μια συνάρτηση (την οποία να ενσωματώσετε σε ένα πρόγραμμα), στην οποία θα διαβάσετε 30 float αριθμούς από το πληκτρολόγιο. Η συνάρτηση να υπολογίζει την τυπική απόκλιση (standard deviation) αυτών των αριθμών. Η τυπική απόκλιση είναι η τετραγωνική ρίζα της εξής διαφοράς: μέση τιμή των τετραγώνων των αριθμών μείον το τετράγωνο της μέσης τιμής. Το αποτέλεσμα να επιστρέφεται στην `main()` από όπου θα γράφεται στην οθόνη.

ΕΡΓΑΣΤΗΡΙΟ 8

Α'. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 8 καλύπτονται τα παρακάτω θέματα:

- Δείκτες. Δηλώσεις δεικτών. Περιεχόμενα δείκτη.
- «Πέρασμα» διευθύνσεων σε συνάρτηση.

Β'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

Δείκτες. Δηλώσεις δεικτών.

1. Εκτελέστε το παρακάτω πρόγραμμα και παρατηρήστε τι εμφανίζει στην οθόνη.

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

main( )
{
    int a, b, *p;
    p = &a;
    *p = 5;
    p = &b;
    *p = 6;
    printf("a=%d, b=%d, p=%p\n", a, b, p );
    system("pause");
}
```

2. Να γράψετε ένα πρόγραμμα, στο οποίο να δηλώσετε ένα πίνακα 10 θέσεων ακεραίων. Να γεμίσετε τον πίνακα από το πληκτρολόγιο και στη συνέχεια να εμφανίσετε στην οθόνη τη διεύθυνση της κάθε θέσης του πίνακα και τα περιεχόμενά της.

Επεξηγήσεις - υπενθυμίσεις:

- Η τιμή ενός δείκτη είναι η διεύθυνση μνήμης στην οποία είναι τοποθετημένος. Η τιμή αυτή μπορεί να εμφανιστεί στην οθόνη με τον προσδιοριστή %p.

«Πέρασμα» διευθύνσεων σε συνάρτηση.

3. Να γράψετε ένα πρόγραμμα, στη `main()` του οποίου να δηλώσετε και να δώσετε τιμές σε δυο ακέραιες μεταβλητές. Να καλέσετε μετά μια συνάρτηση, η οποία με μία μόνο κλήση θα εναλλάσσει τις τιμές των μεταβλητών της `main()`.
4. Να γραφεί μία συνάρτηση που λέγεται `praxeis()`. Η συνάρτηση να καλείται με ορίσματα δυο ακεραίους, τα `x` και `y` ή τις διευθύνσεις τους (ό,τι νομίζετε σωστό). Η συνάρτηση να υπολογίζει το άθροισμα, τη διαφορά και το γινόμενο των `x` και `y`. Τελειώνοντας, η συνάρτηση να επιστρέψει στη `main()` το γινόμενο που υπολόγισε. Πριν επιστρέψει ον έλεγχο, η συνάρτηση να αλλάζει την τιμή της `x` στη `main()` και να την κάνει ίση με το άθροισμα που υπολόγισε. Να αλλάζει επίσης την τιμή της `y` στη `main()` και να την κάνει ίση με τη διαφορά που υπολόγισε.
Στη `main()` να καλείται η συνάρτηση και να γράφονται οι τιμές των `x` και `y` πριν και μετά την κλήση, καθώς και η τιμή επιστροφής της συνάρτησης.
5. Στη `main()` ενός προγράμματος έχετε διαβάσει ένα ακέραιο, τον `akc`, ο οποίος αντιπροσωπεύει μια θερμοκρασία Κελσίου. Ζητείται να γράψετε μια συνάρτηση, η οποία θα λέγεται `thermo()` και θα κάνει τα εξής:
- Θα υπολογίζει την θερμοκρασία Φαρενάιτ, στην οποία αντιστοιχούν οι `akc` βαθμοί Κελσίου. Αν έχετε για παράδειγμα `c` βαθμούς Κελσίου, τότε οι αντίστοιχοι Φαρενάιτ (έστω `f`), δίνονται από τον τύπο $f = 9c/5 + 32$. Οι βαθμοί Φαρενάιτ που θα υπολογιστούν να προκύπτουν όχι από ακέραιες πράξεις, αλλά ως `float` τιμή.
 - Θα υπολογίζει την θερμοκρασία Κέλβιν, στην οποία αντιστοιχούν οι `akc` βαθμοί Κελσίου. Η θερμοκρασία Κέλβιν είναι κατά 273 μεγαλύτερη από την αντίστοιχη Κελσίου.
 - Η συνάρτηση θα έχει τιμή επιστροφής την θερμοκρασία Φαρενάιτ που υπολόγισε.
 - Η συνάρτηση επίσης θα αλλάζει την τιμή του `akc` στη `main()` και θα την κάνει ίση με την θερμοκρασία Κέλβιν που υπολόγισε.
- Η συνάρτηση `thermo()` να οριστεί και να δηλωθεί.
Στη `main()` να καλείτε την `thermo()` και να εμφανίζεται στην οθόνη η τιμή που επιστρέφεται, καθώς και η αλλαγμένη τιμή του `akc`.

Γ'. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

1. Να γράψετε ένα πρόγραμμα, στη `main()` του οποίου να διαβάσετε ένα ακέραιο, τον `kwh`, ο οποίος αντιπροσωπεύει την κατανάλωση σε κιλοβατώρες ενός σπιτιού. Στη συνέχεια να καλείτε μια συνάρτηση, την `kostos`, η οποία να υπολογίζει πόσα χρήματα κοστίζει το ρεύμα που καταναλώθηκε. Η συνάρτηση θα κάνει τα εξής:
- Εάν η κατανάλωση είναι μικρότερη από 100 κιλοβατώρες, κάθε μία κοστίζει 0,10 ΕΥΡΩ.
 - Εάν η κατανάλωση είναι μεταξύ 100 και 200 κιλοβατωρών, οι πρώτες 100 κοστίζουν 0,10 ΕΥΡΩ κάθε μία και οι υπόλοιπες 0,15 ΕΥΡΩ κάθε μία.
 - Εάν η κατανάλωση είναι πάνω από 200 κιλοβατώρες, οι πρώτες 100 κοστίζουν 0,10 ΕΥΡΩ κάθε μία, οι επόμενες 100 κοστίζουν 0,15 ΕΥΡΩ κάθε μία και οι υπόλοιπες 0,20 ΕΥΡΩ κάθε μία.

- Αν το κόστος που προκύπτει είναι μεγαλύτερο από 100 ΕΥΡΩ, να επιβάλεται και πρόστιμο 10% στο πάνω από τα 100 ΕΥΡΩ ποσόν.
- Η συνάρτηση να έχει τιμή επιστροφής το ποσόν που υπολογίστηκε.
- Η συνάρτηση επίσης να ενημερώνει μέσω των παραμέτρων της την `main()` για το ποσόν του προστίμου.

Η `main()` να γράφει στην οθόνη το ποσόν που πρέπει να πληρωθεί, καθώς και το πρόστιμο.

ΟΧΙ ΓΙΑ ΠΩΛΗΣΗ. ΔΩΡΕΑΝ ΒΟΗΘΗΜΑ ΣΕ ΦΟΙΤΗΤΕΣ ΕΠΙΠ

ΕΡΓΑΣΤΗΡΙΟ 9

Α'. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 9 καλύπτονται τα παρακάτω θέματα:

- Εξωτερικές μεταβλητές.
- Στατικές μεταβλητές.
- Πίνακες και δείκτες.
- Πίνακες ως ορίσματα συνάρτησης.

Β'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ/ΕΠΕΞΗΓΗΣΕΙΣ

Εξωτερικές μεταβλητές.

1. Εκτελέστε το παρακάτω πρόγραμμα και παρατηρήστε τι εμφανίζει στην οθόνη.

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

int a;
float x;

void test ();

main()
{
    printf("a=%d, x=%f \n", a,x);
    test();
    printf("a=%d, x=%f \n", a,x);
    system("pause");
}

void test ()
{ a=5; x=3.14; }
```

2. Εκτελέστε το παρακάτω πρόγραμμα και παρατηρήστε τι εμφανίζει στην οθόνη.

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

int a = 3;

void test ( );
void test2 (int);
void test3 ( );

main( )
{
    test( );
    printf("a=%d \n", a);
    test2(a);
    printf("a=%d \n", a);
    test3( );
    printf("a=%d \n", ++a);
    system("pause");
}

void test ( )
{
    int a;
    a = 4;
}

void test2 (int a)
{
    a = 5;
}

void test3 ( )
{
    printf("a=%d \n", a);
    a = 5;
}
```

Επεξηγήσεις - υπενθυμίσεις:

- Οι εξωτερικές μεταβλητές είναι γνωστές σε όλες τις συναρτήσεις, οι οποίες δεν έχουν τοπικές μεταβλητές με το ίδιο όνομα.
- Οι τοπικές μεταβλητές υπερισχύουν των εξωτερικών.

Στατικές μεταβλητές.

3. Εκτελέστε το παρακάτω πρόγραμμα και παρατηρήστε τι εμφανίζει στην οθόνη.

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

void test ( );
void test2 ( );

main( )
{
    test( );
    test( );
    test( );
    test2( );
    test2( );
    test2( );
    system("pause");
}

void test ( )
{
    static int a=1;
    a++;
    printf("a=%d \n", a);
}

void test2 ( )
{
    static int a;
    a=1;
    a++;
    printf("a=%d \n", a);
}
```

Επεξηγήσεις - υπενθυμίσεις:

Οι στατικές μεταβλητές είναι γνωστές μόνο στη συνάρτηση στην οποία έχουν δηλωθεί. Μετά το τέλος της συνάρτησης δεν καταστρέφονται, αλλά η συνάρτηση τις «ξαναβρίσκει» στην επόμενη κλήση της.

Πίνακες και δείκτες.

4. Εκτελέστε το παρακάτω πρόγραμμα και παρατηρήστε τι εμφανίζει στην οθόνη.

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

main( )
{
    int pin[4];          // Πίνακας
    int *p;              // Δείκτης

    p = pin;             // ή p=&pin[0];
    p[1] = 5;
    *(pin+2) = 6;
    printf("%d %d \n", *(p+1), pin[2] );
    system("pause");
}
```

5. Χρησιμοποιώντας συμβολισμό δεικτών να δώστε τιμές από το πληκτρολόγιο σε ένα πίνακα ακεραίων N θέσεων.

Επεξηγήσεις - υπενθυμίσεις:

- Θυμηθείτε ότι ισχύουν τα εξής:

```
pin = &pin[0]
pin+i = &pin[i]
*pin = pin[0]
*(pin+i) = pin[i]
```

Πίνακες ως ορίσματα συνάρτησης

6. Στη main() ενός προγράμματος έχετε δηλώσει δύο πίνακες ακεραίων N θέσεων, τους pin και mat (το N είναι μια σταθερά). Οι πίνακες να γεμίσουν από το πληκτρολόγιο. Να γραφεί μια συνάρτηση, η οποία θα κάνει τα εξής:
- Θα αντιγράψει τα περιεχόμενα του πίνακα mat στον πίνακα pin.
 - Αφού κάνει την αντιγραφή, η συνάρτηση θα ελέγχει τα στοιχεία του πίνακα pin μέχρι να βρει το πρώτο αρνητικό στοιχείο και θα επιστρέφει στη main() ένα δείκτη σε αυτό το πρώτο αρνητικό στοιχείο.
 - Εάν δεν υπάρχει κανένα αρνητικό στοιχείο στον πίνακα pin, η συνάρτηση θα επιστρέφει στη main() ένα δείκτη στο πρώτο στοιχείο του πίνακα pin.

Η main() να εμφανίζει στην οθόνη τα περιεχόμενα της θέσης που δείχνει ο δείκτης τον οποίο επέστρεψε η συνάρτηση.

7. Στη main() ενός προγράμματος:

- Διαβάζονται δυο συμβολοσειρές από το πληκτρολόγιο, οι symb και syn.
- Καλείται στη συνέχεια μια συνάρτηση, η strsrch(), η οποία χρειάζεται πληροφορίες για τις δυο συμβολοσειρές, δέχεται δε μεταξύ των άλλων παραμέτρων της και ένα ακέραιο, τον k. Η συνάρτηση κάνει τα εξής:
 - Τοποθετεί μέσα στη συμβολοσειρά symb, από την θέση k και μετά, όλα τα περιεχόμενα της συμβολοσειράς syn, χωρίς τον χαρακτήρα \0.
 - Τελειώνοντας, η συνάρτηση επιστρέφει ένα δείκτη στη συμβολοσειρά symb, k θέσεις από την αρχή της.

Να διαβαστούν οι συμβολοσειρές, να κληθεί από την main() η συνάρτηση και να εμφανίσει η main() στην οθόνη τα περιεχόμενα του δείκτη τον οποίο επιστρέφει η συνάρτηση.

Επεξηγήσεις - υπενθυμίσεις:

- Για να «περάσω» σε μια συνάρτηση ως παράμετρο ένα πίνακα, περνάω ένα δείκτη στην αρχή του πίνακα και (αν χρειάζεται) το μέγεθος του πίνακα.
- Στον ορισμό μια συνάρτησης (της paradeigm) οι παρακάτω συμβολισμοί είναι ισοδύναμοι:

void paradeigm (int *pin)

void paradeigm (int pin[])

Γ'. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

8. Στη main() ενός προγράμματος έχουν δηλωθεί δυο πίνακες ακεραίων N θέσεων, οι pin και mat. Από τη main() καλείται μια συνάρτηση, η comp(), η οποία κάνει τα εξής:

- Βρίσκει την πρώτη θέση στην οποία διαφέρουν οι pin και mat. Έστω ότι αυτή είναι d θέσεις από την αρχή των πινάκων.
- Τα στοιχεία του mat, τα οποία δεν είναι ίδια με τα αντίστοιχα στοιχεία του pin τα κάνει μηδέν.
- Εάν οι πίνακες έχουν τα ίδια στοιχεία, η συνάρτηση επιστρέφει ένα δείκτη στην αρχή του mat.
- Εάν οι πίνακες δεν έχουν τα ίδια στοιχεία, η συνάρτηση επιστρέφει ένα δείκτη d θέσεις πιο κάτω από την αρχή του pin.

Να δηλωθεί κι να οριστεί η συνάρτηση comp().

Η main() να γράφει στην οθόνη τα περιεχόμενα του δείκτη, τον οποίο επέστρεψε η comp().

Δεν επιτρέπεται η χρήση εξωτερικών μεταβλητών.

ΕΡΓΑΣΤΗΡΙΟ 10

Α'. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 10 καλύπτονται τα παρακάτω θέματα:

- Συμβολοσειρές.
- Συναρτήσεις gets, puts, scanf (με %s),
- Συναρτήσεις strlen, strcpy, strcat, strcmp, strchr, strstr.

Β'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

Συμβολοσειρές και συναρτήσεις χειρισμού συμβολοσειρών.

1. Εκτελέστε το παρακάτω πρόγραμμα και παρατηρήστε τι εμφανίζει στην οθόνη.

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h> // Συμβολοσειρές

main( )
{
    char s[80], t[80];
    int i;

    printf ("give s -> ");
    gets(s);
    printf ("give t -> ");
    gets(t);
    i = strcmp(s,t);
    printf ("strcmp = %d \n", i );
    if (i<0)
        printf (" s<t \n");
    if (i==0)
        printf (" s=t \n");
    if (i>0)
        printf (" s>t \n");
    system("pause");
}
```

2. Εκτελέστε το παρακάτω πρόγραμμα και παρατηρήστε τι εμφανίζει στην οθόνη.

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

main( )
{
    char s[80], t[80];
    char *p;

    printf ("give t -> ");
    gets(t);
    p = strcpy(s,t);          // s = t; p = s;
    printf ("s = %s \n", s );
    printf ("%c\n", *p);
    system("pause");
}
```

3. Εκτελέστε το παρακάτω πρόγραμμα και παρατηρήστε τι εμφανίζει στην οθόνη.

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

main( )
{
    char s[80], t[80];
    char *p;

    printf ("give s -> ");
    gets(s);
    printf ("give t -> ");
    gets(t);
    p = strcat(s,t);          // s = s + t; p = s;
    printf("s = %s \n", s );
    printf ("%c\n", *p);
    system("pause");
}
```

4. Εκτελέστε το παρακάτω πρόγραμμα και παρατηρήστε τι εμφανίζει στην οθόνη.

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

main( )
{
    char s[80];
    char *p;

    strcpy( s, "ABCABEB" );           // s = "ABCABEB"
    p = strchr( s, 'D' );              // Search for 'D' gives NULL
    if (p==NULL)
        printf("Den yparxei D \n");
    else
        printf("Yparxei D, thesh = %d \n", p-s );
    p = strchr( s, 'B' );              // Search for character 'B' in s
    if (p==NULL)
        printf("Den yparxei B \n");
    else
        printf("Yparxei B, thesh = %d \n", p-s );
    p = strchr( s, 'C' );              // Search for character 'C' in s
    if (p==NULL)
        printf("Den yparxei C\n");
    else
        printf("Yparxei C, thesh = %d \n", p-s );
    p = strchr( s, 'E' );              // Search for character 'E' in s
    if (p==NULL)
        printf("Den yparxei E\n");
    else
        printf("Yparxei E, thesh = %d \n", p-s );
    system("pause");
}
```

5. Εκτελέστε το παρακάτω πρόγραμμα και παρατηρήστε τι εμφανίζει στην οθόνη.

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

main( )
{
    char s[80];
    char *p;

    strcpy( s, "ABCBE" );           // s = "ABCBE"
    p = strstr( s, "CD" );           // Search for "CD" in s
    if (p==NULL)
        printf ("Den yparxei CD\n");
    else
        printf ("Yparxei CD, thesh = %d \n", p-s);
    p = strstr( s, "CB" );           // Search for string "CB" in s
    if (p==NULL)
        printf ("Den yparxei CB\n");
    else
        printf ("Yparxei CB, thesh = %d \n", p-s);
    system("pause");
}
```

6. Στη main() ενός προγράμματος:

- Διαβάζονται δυο συμβολοσειρές από το πληκτρολόγιο, οι symb και syn.
- Καλείται στη συνέχεια μια συνάρτηση, η strstrch(), η οποία χρειάζεται πληροφορίες για τις δυο συμβολοσειρές, δέχεται δε μεταξύ των άλλων παραμέτρων της και ένα ακέραιο, τον k. Η συνάρτηση κάνει τα εξής:
 - Τοποθετεί μέσα στη συμβολοσειρά symb, από την θέση k και μετά, όλα τα περιεχόμενα της συμβολοσειράς syn, χωρίς τον χαρακτήρα \0.
 - Τελειώνοντας, η συνάρτηση επιστρέφει ένα δείκτη στη συμβολοσειρά symb, k θέσεις από την αρχή της.

Να διαβαστούν οι συμβολοσειρές, να κληθεί από την main() η συνάρτηση και να εμφανίσει η main() στην οθόνη τα περιεχόμενα του δείκτη τον οποίο επιστρέφει η συνάρτηση.

7. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση του πιο κάτω προγράμματος και γιατί:

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

char words[10];

main( )
{
    char wra[10] = "TI WRA";
    char *ptr = "ENTEKA";
    char *dkt;
    int k;

    puts(&wra[1]);
    puts(ptr+1);
    dkt = strcpy(words+1, ptr);
    printf("%s\n", dkt+3);
    printf("%c\n", *dkt+3);
    words[0] = 'A';
    puts(words);
    for (k=1; k<3; k++)
        ptr++;
    dkt = strcat(wra, ptr+2);
    puts(dkt);
    k = strcmp(words+1, ptr-2);
    printf("%d\n", k);
    putchar(*strcat(words+4,ptr+2));
}
```

Επεξηγήσεις - υπενθυμίσεις:

- Στην άσκηση 1 γίνεται χρήση της strcmp, η οποία συγκρίνει αλφαβητικά δυο συμβολοσειρές. Η τιμή επιστροφής της είναι θετικός αριθμός ή αρνητικός ή μηδέν, ανάλογα με το ποια συμβολοσειρά προηγείται αλφαβητικά της άλλης.
- Στην άσκηση 2 γίνεται χρήση της strcpy, η οποία αντιγράφει μια συμβολοσειρά σε μια άλλη και επιστρέφει δείκτη σε χαρακτήρα ίσο με το πρώτο της όρισμα.

- Στην άσκηση 3 γίνεται χρήση της `strcat`, η οποία επικολλά μια συμβολοσειρά στο τέλος μιας άλλης και επιστρέφει ένα δείκτη σε χαρακτήρα ίσο με το πρώτο της όρισμα.
- Στην άσκηση 4 γίνεται χρήση της `strchr`, η οποία αναζητά την πρώτη εμφάνιση ενός χαρακτήρα σε μια συμβολοσειρά και επιστρέφει ένα δείκτη σε χαρακτήρα στην θέση του χαρακτήρα αυτού, αλλιώς επιστρέφει `NULL`.
- Στην άσκηση 5 γίνεται χρήση της `strstr`, η οποία αναζητά την πρώτη εμφάνιση μιας συμβολοσειράς μέσα σε μια άλλη συμβολοσειρά. Επιστρέφει ένα δείκτη σε χαρακτήρα στην θέση της συμβολοσειράς που επεσήμανε, αλλιώς επιστρέφει `NULL`.

Γ'. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

8. Να γραφεί μια συνάρτηση, η οποία θα προσομοιώνει την `strlen`. Δηλαδή θα κάνει ό,τι και η `strlen`, η οποία υποτίθεται ότι δεν είναι γνωστή. Να γράψετε ένα πρόγραμμα, το οποίο θα χρησιμοποιεί τη συνάρτηση που γράψατε.
9. Να γραφεί μια συνάρτηση, η οποία θα προσομοιώνει την `strcpy`. Δηλαδή θα κάνει ό,τι και η `strcpy`, η οποία υποτίθεται ότι δεν είναι γνωστή. Να γράψετε ένα πρόγραμμα, το οποίο θα χρησιμοποιεί τη συνάρτηση που γράψατε.
10. Να γραφεί μια συνάρτηση, η οποία θα προσομοιώνει την `strcat`. Δηλαδή θα κάνει ό,τι και η `strcat`, η οποία υποτίθεται ότι δεν είναι γνωστή. Να γράψετε ένα πρόγραμμα, το οποίο θα χρησιμοποιεί τη συνάρτηση που γράψατε.
11. Να γραφεί μια συνάρτηση, η οποία θα προσομοιώνει την `strcmp`. Δηλαδή θα κάνει ό,τι και η `strcmp`, η οποία υποτίθεται ότι δεν είναι γνωστή. Να γράψετε ένα πρόγραμμα, το οποίο θα χρησιμοποιεί τη συνάρτηση που γράψατε.
12. Να γραφεί μια συνάρτηση, η οποία θα προσομοιώνει την `strchr`. Δηλαδή θα κάνει ό,τι και η `strchr`, η οποία υποτίθεται ότι δεν είναι γνωστή. Να γράψετε ένα πρόγραμμα, το οποίο θα χρησιμοποιεί τη συνάρτηση που γράψατε.
13. Να γραφεί μια συνάρτηση, η οποία θα προσομοιώνει την `strstr`. Δηλαδή θα κάνει ό,τι και η `strstr`, η οποία υποτίθεται ότι δεν είναι γνωστή. Να γράψετε ένα πρόγραμμα, το οποίο θα χρησιμοποιεί τη συνάρτηση που γράψατε.

ΕΡΓΑΣΤΗΡΙΟ 11

Α'. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 11 καλύπτονται τα παρακάτω θέματα:

- Συναρτήσεις `exp`, `abs`, `fabs`, `pow`, `sqrt`, `sin`, `cos`, `tan`.
- Συναρτήσεις `atoi`, `atof`, `atol`.
- Δομές (structures): Περιγραφή, πεδία δομής, δηλώσεις και δεδομένα στις δομές.
- Φωλιασμένες δομές.

Β'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

Συναρτήσεις `exp`, `abs`, `fabs`, `pow`, `sqrt`, `sin`, `cos`, `tan`

1. Να γραφεί ένα πρόγραμμα, το οποίο να υπολογίζει και να εμφανίζει στην οθόνη τις δέκα πρώτες δυνάμεις του 10.
2. Να γραφεί ένα πρόγραμμα, το οποίο να υπολογίζει και να εμφανίζει στην οθόνη τα ημίτονα, τα συνημίτονα των τιμών -1 έως 1 σε βήματα του ενός δεκάτου..
3. Να γραφεί ένα πρόγραμμα, το οποίο να διαβάζει N φορές από το πληκτρολόγιο ανά ένα ακέραιο και ένα `float` κάθε φορά και να εμφανίζει στην οθόνη τις απόλυτες τιμές των αριθμών που διαβάστηκαν.

Επεξηγήσεις - υπενθυμίσεις:

- Στην άσκηση 1 χρειάζεται η συνάρτηση υπολογισμού δύναμης `pow`.
- Στην άσκηση 2 χρειάζονται οι συναρτήσεις `sin`, `cos` και `tan`. Το όρισμα των συναρτήσεων πρέπει να δίνεται σε ακτίνια.
- Στην άσκηση 3 χρειάζονται οι συναρτήσεις `abs` και `fabs`.

Συναρτήσεις `atoi`, `atof`, `atol`

4. Να γραφεί ένα πρόγραμμα, το οποίο να διαβάζει δύο ακέραιους, δύο `float` και δύο `long` από το πληκτρολόγιο με την χρήση των `gets( )`, `atoi( )`, `atof( )` και `atol( )` και να εμφανίζει στην οθόνη τα αθροίσματα των αριθμών κάθε είδους.

Επεξηγήσεις - υπενθυμίσεις:

- Οι συναρτήσεις `atoi( )`, `atof( )` και `atol( )` βρίσκονται στο `stdlib.h`.

Δομές: Περιγραφή, πεδία δομής. Δηλώσεις και δεδομένα στις δομές.

5. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση του πιο κάτω προγράμματος και γιατί.

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

struct complex { float re, im; };    // Περιγραφή νέου τύπου

main( )
{
    struct complex z1, z2;          // Δήλωσεις μεταβλητών
    printf("sizeof = %d \n", sizeof (struct complex));
    printf("give z1.re -> ");
    scanf( "%f", &z1.re );          // Διάβασμα
    z1.im = 55;
    z2 = z1;                        // Επιτρεπτή εκχώρηση τιμής
    printf("z2 = %f + i * %f \n", z2.re, z2.im );
    system("pause");
}
```

6. Σε συνάχεια της παραπάνω άσκησης να γράψετε ένα πρόγραμμα, στο οποίο να διαβάζετε από το πληκτρολόγιο τιμές για δύο μιγαδικούς αριθμούς. Θα διαβάζετε επίσης ένα χαρακτήρα. Αν ο χαρακτήρας είναι το + θα κάνετε πρόσθεση των μιγαδικών αριθμών, αν είναι - θα κάνετε αφαίρεση, αν είναι * θα κάνετε πολλαπλασιασμό και αν είναι / θα κάνετε διαίρεση. Για οποιονδήποτε άλλο χαρακτήρα θα γράφεται στην οθόνη «Λάθος», θα ζητείται καινούργιος χαρακτήρας και θα επαναλαμβάνεται όλη η διαδικασία. Το αποτέλεσμα της πράξης να γράφεται στην οθόνη.

7. Σε ένα πρόγραμμα έχετε τον εξής τύπο δομών:

```
struct stype
{
    int j;
    char ch[30];
    float fp;
};
```

Να δηλώσετε στη main() δυο μεταβλητές δομής του πιο πάνω τύπου, τις str1 και str2 και να τους δώσετε τιμές από το πληκτρολόγιο. Στα πεδία ch να καταχωρήσετε συμβολοσειρές.

Στη συνέχεια να εναλλάξετε τα περιεχόμενα των str1 και str2 και να γράψετε τα πεδία τους στην οθόνη.

Επεξηγήσεις - υπενθυμίσεις:

- Ο τελεστής sizeof δίνει το μέγεθος σε byte ενός είδους δεδομένων ή μιας μεταβλητής.

- Έστω δύο μιγαδικοί αριθμοί, οι $z_1=a+jb$ και $z_2=c+jd$. Τότε:
 - $w_1 = z_1+z_2 = (a+c) + j (b+d)$
 - $w_2 = z_1-z_2 = (a-c) + j (b-d)$
 - $w_3 = z_1 \cdot z_2 = (ac-bd) + j (bc+ad)$
 - $w_4 = z_1/z_2 = (ac+bd)/(c^2+d^2) + j (bc-ad)/(c^2+d^2)$

Φωλιασμένες δομές.

8. Κάθε εικονοστοιχείο (pixel) στην οθόνη χαρακτηρίζεται από την θέση του (δομή του είδους pixel) και το χρώμα του (δομή του είδους color). Το χρώμα καθορίζεται από τρεις αριθμούς. Εκτελέστε το παρακάτω πρόγραμμα, στο οποίο δίνουμε τιμές στα διάφορα πεδία. Τι θα εμφανιστεί στην οθόνη μετά την εκτέλεση του προγράμματος και γιατί.

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

struct color { int r,g,b; };           // red, green, blue

struct pixel
{
    int x,y;                          // Συντεταγμένες pixel
    struct color c;                   // Δομή color μέσα σε pixel
};

main( )
{
    struct pixel p;
    struct color blue;

    blue.r = blue.g = 0; blue.b = 255;    // create blue
    p.x = 12; p.y = 50;
    p.c = blue;                          // Το χρώμα γίνεται blue
    p.c.r = 50;                          // Διόρθωση του red του p, αλλά ΟΧΙ του blue
    printf("x=%d, y=%d, r=%d, g=%d, b=%d \n",
           p.x, p.y, p.c.r, p.c.g, p.c.b );
    system("pause");
}
```

9. Να γράψετε ένα πρόγραμμα, στο οποίο να περιγράψετε δύο είδη δομών, τις one και two. Οι δομές του είδους one έχουν δύο πεδία: το ak (ακέραιος) και το pin (πίνακας χαρακτήρων 30 θέσεων). Οι δομές του είδους two έχουν τέσσερα πεδία: το data (ακέραιος), το mat (πίνακας ακεραίων 20 θέσεων), το item (δομή του είδους one) και το melos (δομή του είδους one). Να δηλώσετε δυο μεταβλητές, την person (δομή του είδους one) και την memb (δομή του είδους two). Να γράψετε τις εντολές (κάθε μια είναι ανεξάρτητη από τις υπόλοιπες), με τις οποίες

θα γίνονται τα παρακάτω, μετά δε από κάθε εντολή να γράφονται στην οθόνη τα περιεχόμενα των δομών για επιβεβαίωση:

- Διαβάζουμε τιμές από το πληκτρολόγιο για τα πεδία των δομών `person` και `memb`. Στους πίνακες χαρακτήρων να τοποθετήσετε συμβολοσειρές. Το διάβασμα του πεδίου `ak` της `person` να γίνει με την `gets( )` και την `atoi( )` και του πεδίου `data` της `memb` με την `scanf( )`.
- Αντιγράφουμε τη συμβολοσειρά που υπάρχει στον πίνακα `pin` της `person` στον αντίστοιχο πίνακα του πεδίου `item` της `memb`.
- Γράφουμε στην οθόνη το μήκος της συμβολοσειράς που υπάρχει στον πίνακα χαρακτήρων του πεδίου `meios` της δομής `memb`.
- Στο τέλος της συμβολοσειράς που υπάρχει στον πίνακα `pin` της `person` κάνουμε επικόλληση της συμβολοσειράς που υπάρχει στον αντίστοιχο πίνακα χαρακτήρων του πεδίου `meios`, της δομής της `memb`, εφ' όσον υπάρχει αρκετός χώρος.
- Αντιγράφουμε το πεδίο `item` της δομής `memb`, στο πεδίο `meios` της ίδιας δομής.

Επεξηγήσεις - υπενθυμίσεις:

- Η προσπέλαση των πεδίων στις φωλιασμένες δομές γίνεται με την πολλαπλή χρήση της τελείας (`.`).

ΕΡΓΑΣΤΗΡΙΟ 12

Α'. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 12 καλύπτονται τα παρακάτω θέματα:

- Πίνακες δομών.
- Δομές ως παράμετροι και ως τιμή επιστροφής συναρτήσεων.
- Δείκτες σε δομές.

Β'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

Πίνακες δομών.

1. Πληκτρολογήστε και εκτελέστε το πιο κάτω πρόγραμμα και παρατηρήστε τι κάνει:

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

#define N 100

struct student { int am; char name[30]; };

main( )
{
    struct student pin[N];          // Πίνακας δομών
    int i;

    for (i=0; i<N; i++)              // Διάβασμα
    {
        printf ("Dwste AM toy spoudasth %d -> ", i );
        scanf ("%d", &pin[i].am );  // Προσέξτε το &
        getchar ( );                 // Τι χρειάζεται εδώ η getchar;
        printf ("Dwste onoma -> ");
        gets (pin[i].name);
    }
    for (i=0; i<N; i++)              // Εκτύπωση
    {
        printf ("Spoudasths %2d, AM=%4d, Onoma=%s \n",
                                                         i, pin[i].am, pin[i].name );
    }
    system("pause");
}
```

Δομές ως παράμετροι και ως τιμή επιστροφής συναρτήσεων.

2. Σε ένα πρόγραμμα έχετε τον εξής τύπο δομών:

```
struct stype
{
    int j;
    char ch[30];
    float fp;
}
```

Να δηλώσετε στη main() δυο μεταβλητές δομές του πιο πάνω τύπου, τις str1 και str2 και να τους δώσετε τιμές από το πληκτρολόγιο. Στα πεδία ch να καταχωρήσετε συμβολοσειρές.

Στη συνέχεια να καλέσετε μια συνάρτηση, η οποία θα λέγεται struct_swap(). Η συνάρτηση θα καλείται μια μόνο φορά στη main() και θα εναλλάσσει τα περιεχόμενα των str1 και str2. Η συνάρτηση να δηλωθεί, να οριστεί.

Να μη χρησιμοποιείτε εξωτερικές μεταβλητές.

3. Σε ένα πρόγραμμα έχετε τον εξής τύπο δομών:

```
struct funds
{
    char name[20];
    float poson;
}
```

Να δηλώσετε στη main() ένα πίνακα δομών του πιο πάνω τύπου, τον persons και να του δώσετε τιμές από το πληκτρολόγιο. Στο πεδίο name να καταχωρήσετε συμβολοσειρές. Στη συνέχεια:

- Να διαβάσετε ένα χαρακτήρα στη main(), τον ch.
- Να καλέσετε μια συνάρτηση, την athroisma(), η οποία θα καλείται μια μόνο φορά στη main() και θα κάνει τα εξής:
 - Θα ελέγχει ποιες από τις συμβολοσειρές των στοιχείων του πίνακα δομών έχουν πρώτο γράμμα τον χαρακτήρα ch.
 - Θα δημιουργεί το άθροισμα των πεδίων poson αυτών των στοιχείων του πίνακα.
 - Θα επιστρέφει το άθροισμα στη main().

Η main() να εμφανίζει στην οθόνη την τιμή που επέστρεψε η athroisma().

Να μη χρησιμοποιείτε εξωτερικές μεταβλητές.

Δείκτες σε δομές.

4. Επαναλαμβάνεται μέρος της άσκησης 9 του εργαστηρίου 11, με καινούργια ζητούμενα:

Να γράψετε ένα πρόγραμμα, στο οποίο να περιγράψετε δύο είδη δομών, τις one και two. Οι δομές του είδους one έχουν δύο πεδία: το ak (ακέραιος) και το pin (πίνακας χαρακτήρων 30 θέσεων). Οι δομές του είδους two έχουν τέσσερα πεδία: το data (ακέραιος), το mat (πίνακας ακεραίων 20 θέσεων), το item (δομή του είδους one) και το melos (δομή του είδους one). Να δηλώσετε δυο μεταβλητές, την person (δομή του είδους one) και την memb (δομή του είδους two). Να δηλώσετε επίσης ένα δείκτη σε δομές του είδους one τον ptr και ένα δείκτη σε δομές του είδους two, τον dkt. Να γράψετε τις εντολές (κάθε μια είναι ανεξάρτητη από τις υπόλοιπες), με τις οποίες θα γίνονται τα παρακάτω, μετά δε από κάθε εντολή να γράφονται στην οθόνη τα περιεχόμενα των δομών για επιβεβαίωση:

Όπου δηλώνεται πίνακας χαρακτήρων να υποθέσετε ότι αυτός περιέχει συμβολοσειρά.

Να κάνετε τον δείκτη ptr να δείχνει στην δομή person και τον dkt να δείχνει στην δομή memb και να γράψετε τις εντολές (κάθε μια είναι ανεξάρτητη από τις υπόλοιπες), με τις οποίες:

- Διαβάζουμε με την gets μια συμβολοσειρά για το πεδίο pin της δομής στην οποία δείχνει ο ptr.
- Αντιγράφουμε τη συμβολοσειρά που υπάρχει στον πίνακα pin τ της δομής στην οποία δείχνει ο ptr στον αντίστοιχο πίνακα του πεδίου item της δομής στην οποία δείχνει ο dkt.
- Δίνουμε τιμή από το πληκτρολόγιο στο πεδίο data της δομής στην οποία δείχνει ο dkt.
- Δίνουμε τιμή στην δεύτερη θέση του πίνακα mat της δομής, στην οποία δείχνει ο dkt.
- Γράφουμε στην οθόνη τα περιεχόμενα των πεδίων της δομής στην οποία δείχνει ο ptr.
- Αντιγράφουμε την δομή person, το item πεδίο της δομής στην οποία δείχνει ο dkt.
- Αντιγράφουμε το πεδίο item της δομής στην οποία δείχνει ο dkt, στο πεδίο melos της ίδιας δομής.
- Γράφουμε στην οθόνη το μήκος της συμβολοσειράς που υπάρχει στον πίνακα χαρακτήρων του πεδίου melos της δομής στην οποία δείχνει ο dkt.
- Στο τέλος της συμβολοσειράς που υπάρχει στον πίνακα pin της person κάνουμε επικόλληση της συμβολοσειράς που υπάρχει στον αντίστοιχο πίνακα χαρακτήρων του πεδίου melos, της δομής στην οποία δείχνει ο dkt.