ΤΕΙ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΚΑΙ ΑΝΑΠΤΥΞΗΣ ΑΛΓΟΡΙΘΜΩΝ

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΡΓΑΣΤΗΡΙΟ

(Γ' ΕΚΔΟΣΗ)

ΗΡΑΚΛΕΙΟ 2017

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.

Στη διάρκεια του εξαμήνου διεξάγονται τρία τουλάχιστον διαγωνίσματα (τεστ) Τ1, Τ2 και Τ3, διάρκειας περίπου 45 λεπτών (ή όσων κρίνει ο διδάσκων), ο δε τελικός βαθμός του εργαστηρίου προκύπτει από τους βαθμούς σε αυτά τα τέστ, με ενδεικτικό ποσοστό συμμετοχής καθενός στον τελικό βαθμό 30%, 30% και 40% αντίστοιχα. Ο τελικός βαθμός στο εργαστήριο θα διαμορφώνεται δηλαδή σύμφωνα με τον εξής τύπο (ή περίπου αυτόν, ανάλογα με τα τελικά ποσοστά των τριών τέστ):

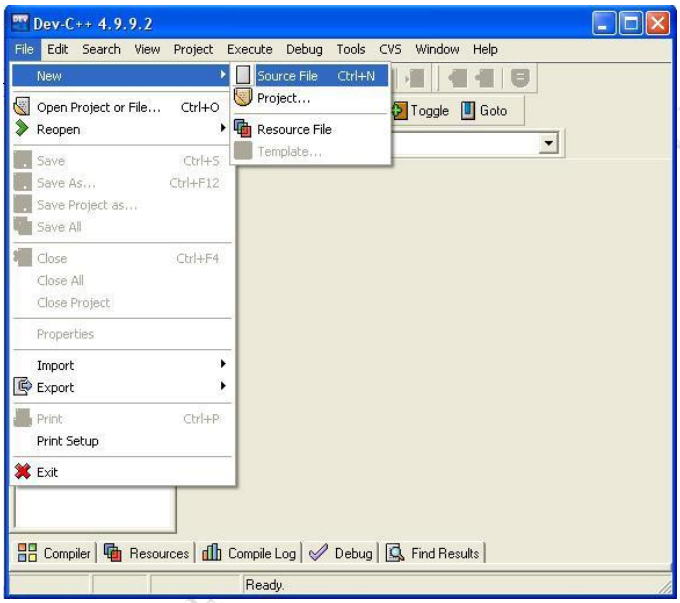
Τελικός Βαθμός = 0.3 * Τ1 + 0.3 * Τ2 + 0.4 * Τ3.

ΤΟ ΠΕΡΙΒΑΛΛΟΝ ΕΡΓΑΣΙΑΣ
Dev C++ IDE

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

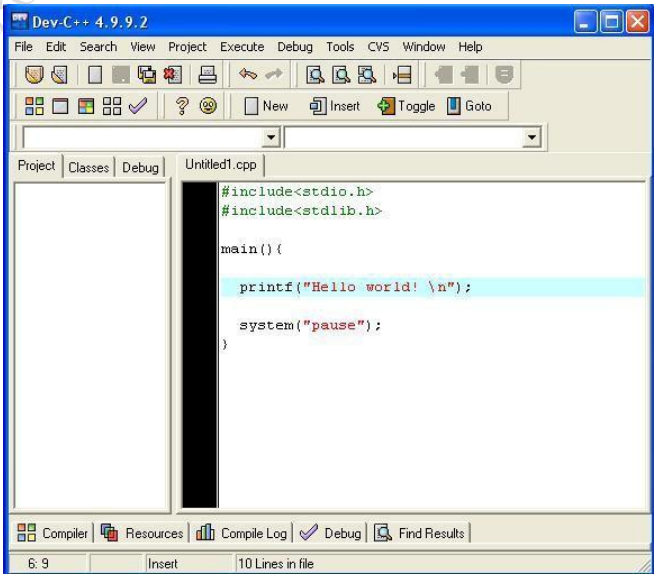
1. Δημιουργία του νέου προγράμματος

File/New/SourceFile, ή Ctrl-N



Εικ. 1

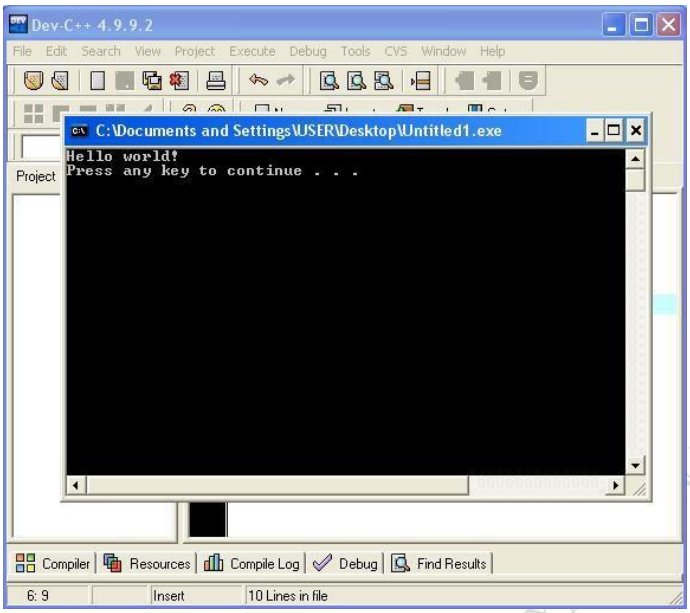
2. Σύνταξη του προγράμματος στον κειμενογράφο του Dev C++



Εικ. 2

3. Εκτέλεση του προγράμματος

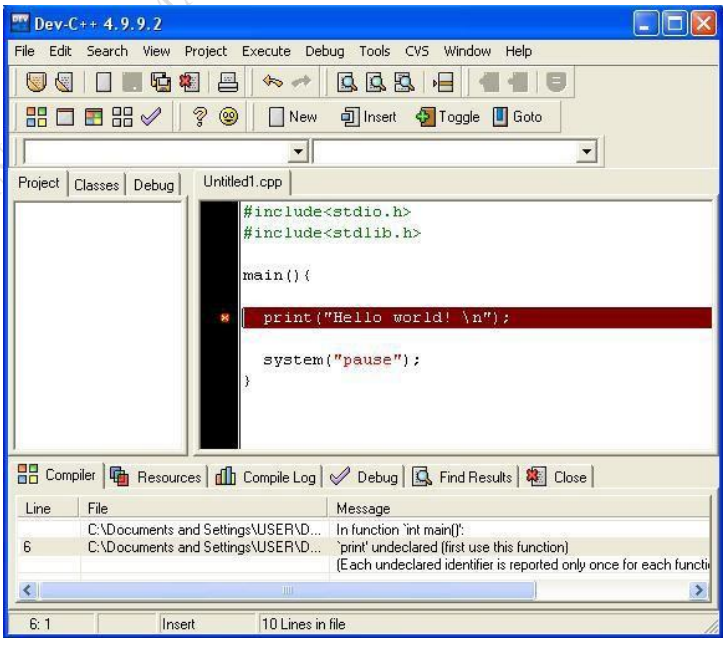
Πλήκτρο **F9 ή F11**(ανάλογα με την έκδοση): Compile and Run program.



Εικ. 3

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



Εικ. 4

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.

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)

ΕΡΓΑΣΤΗΡΙΟ 1

Α΄. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 1 καλύπτονται τα παρακάτω θέματα:

- **Μορφή προγράμματος** της C.
- **Μεταβλητές**.
- Μια πρώτη ματιά στην **printf( )**.
- Ειδικοί χαρακτήρες: **\n \b \r \t \a \0 \\ \' \" %%**
- Τύποι **int** και **long**. Προσδιοριστές **%d**, **%x**, **%o** και χρήση της printf( ) με αυτούς.

Β΄. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

Επεξηγήσεις – υπενθυμίσεις (άσκηση 1) :

#include <stdio.h> : Χρήση της βιβλιοθήκης stdio.h

main( ) : Κύρια συνάρτηση, αρχή του προγράμματος

{ } : Οι εντολές της main βρίσκονται ανάμεσα στα άγκιστρα

printf ("Hello world! \n"); : Εμφάνιση στην οθόνη

system ("pause"); : Περιμένει το πάτημα ενός πλήκτρου για την συνέχεια. Έτσι, η οθόνη αποτελεσμάτων παραμένει «ανοιχτή» μέχρι να πατήσετε ένα πλήκτρο. Πρέπει να υπάρχει το #include <stdlib.h> στην αρχή του προγράμματος.

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 (ασκήσεις 2, 3, 4) :

Τα παρακάτω ζευγάρια χαρακτήρων έχουν ειδική χρήση στην C και συγκεκριμένα, όταν τα συναντήσουμε μέσα στην printf, κάνουν τα εξής:

\n	αλλαγή γραμμής (new line)
\b	χαρακτήρας οπισθοχώρησης (αριστερό βελάκι)
\r	επιστροφή στην αρχή της γραμμής
\t	χαρακτήρας tab (8 κενά)
\a	beep, κουδούνι, ηχείο του Η/Υ
\0	κάνει «αόρατο» το υπόλοιπο τμήμα της γραμμής
****	κάθετος γραμμή
\'	μονά εισαγωγικά (απόστροφος)
\"	διπλά εισαγωγικά
%%	ποσοστό

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 ("\nTEI KRHTHS\n");
    printf ("SXOLH TEXNOLOGIKWN\n EFARMOGWN\n");
    system ("pause");
}
```

3. Τι θα εμφανίσει στην οθόνη το πιο κάτω πρόγραμμα; Προσπαθήστε 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");
    }
```

4. Να γράψετε ένα πρόγραμμα, το οποίο θα εμφανίζει στην οθόνη το πιο κάτω σχήμα:

```
                    """""
                      """""""""
%%%%%%%%%%%%%%%%%%% """""""""
%%%%%%%%%%%%%%%%%%% """""""""""""""
                        """""""""
                      """""""""
                    """""
```

(Οι άκρες του βέλους είναι διαδοχικά διπλά εισαγωγικά)

Επεξηγήσεις – υπενθυμίσεις (άσκηση 5) :

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

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

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

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

   num = 1;
   printf ("O ARITHMOS  ISOYTAI  ME %d\n", num);
   art = num + 1;
   printf ("AN  PROSTHESOYME  1  PAIRNOYME %d\n", art);
   printf ("ATHROISMA = %d. DIAFORA = %d\n", num+art, num-
   art); system("pause");
}
```

Επεξηγήσεις – υπενθυμίσεις (άσκηση 6) :

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

```
#include<stdio.h>                    /*Αλλα αριθμητικά συστήματα*/
#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 (άσκηση 7) :

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

```
int i;
float x;
long z;
charch;
```

- **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. 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 τρείς 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 (άσκηση 8) :

- Η τήρηση των ορίων των τιμών των μεταβλητών είναι ευθύνη του προγραμματιστή.
- Στα 32-bit συστήματα τύπος **int** ταυτίζεται με **long** (bytes). Τα όρια των ακεραίων των 2 byte είναι από -32768 έως +32767, ενώ για τους ακεραίους των 4 byte τα όρια αυτά είναι από -2147483648 έως +2147483647

8. 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");
}
```

Γ΄. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

9. Στο παρακάτω πρόγραμμα να διορθώσετε τα συντακτικά και τα λογικά λάθη, ώστε να δίνει σωστά αποτελέσματα:

```
include studio.h
main [ ]
{
    integer k
    k:=60;
    print {SE ENA ETOS YPARXOYN k EBDOMADES};
}
```

10. Να γράψετε ένα πρόγραμμα που να εκτυπώνει στην οθόνη το όνομα και την διεύθυνσή σας.

11. Να γράψετε ένα πρόγραμμα που να εκτυπώνει στην οθόνη το όνομα και το επίθετό σας σε μια γραμμή με μια printf( ), σε δυό ξεχωριστές γραμμές με μια printf( ) και σε μία γραμμή με δύο ξεχωριστές printf( ).

12. Να γράψετε ένα πρόγραμμα, στο οποίο να δηλώσετε δύο μεταβλητές. Η μια θα έχει τιμή όση η ηλικία σας σε έτη. Το πρόγραμμα θα υπολογίζει την ηλικία σας σε μήνες και θα την εμφανίζει στην οθόνη.

13. 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.

14. Στα παλαιά χρόνια (όταν η μνήμη του Η/Υ ήταν πολύτιμη), κάποιος βρήκε την εξής "αριθμητική" λύση για την αντιμετάθεση τιμών (στην οποία αναφέρεται και η πιο πάνω άσκηση): **a=a+b; b=a-b; a=a-b;** Είναι σωστή; Να το διαπιστώσετε δίνοντας δύο τιμές στα a και b, μετά τις πιο πάνω εντολές και ξαναεμφανίζοντας τις τιμές των a και b.

ΕΡΓΑΣΤΗΡΙΟ 2

Α΄. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 2 καλύπτονται τα παρακάτω θέματα:

- Τύπος **char**. Προσδιοριστής **%c**. Χρήση της printf( ) με αυτόν.
- Τύποι **float** και **double**. Προσδιοριστές **%f** και **%lf**. Χρήση της printf( ).
- Η συνάρτηση **scanf( )**.
- **Σταθερές**.
- **Πίνακες** (εισαγωγικά).

Β΄. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

char:

Επεξηγήσεις – υπενθυμίσεις (άσκηση 1) :

Οι μεταβλητές τύπου **χαρακτήρα (char)** είναι ένας (ακριβώς) χαρακτήρας, π.χ. 'A', 'B', 'C', '2', '3', '#', '╝' κλπ. Αποθηκεύονται σε ένα (1) byte στη μνήμη.

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

```
#include <stdio.h>

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

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

Επεξηγήσεις – υπενθυμίσεις (άσκηση 2) :

Οι μεταβλητές τύπου χαρακτήρα μπορούν να εμφανιστούν στην οθόνη **με μορφή χαρακτήρα** χρησιμοποιώντας τον προδιοριστή %c ή **με την μορφή ακεραίου** χρησιμοποιώντας τον προδιοριστή %d

2. Τι θα εμφανίσει στην οθόνη το πιο κάτω πρόγραμμα και γιατί;

```
#include <stdio.h>

main( )
{
    char c, d;

    c = 'A';
    d = '$';
    printf("\n");
    printf("%c\n", c);
    printf("%c\n", d);
    printf("%d\n", c);
    printf("%d\n", d);
    printf("%c\n", c+d);
    printf("%d\n", c+d);
}
```

float:

Επεξηγήσεις – υπενθυμίσεις (άσκηση 3) :

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

Στο παρακάτω πρόγραμμα εμφανίζεται η πρόσθετη μορφοποίηση %k.mf, πχ. %12.4f. Ο αριθμός k παριστάνει θέσεις και ο m το κλασματικό μέρος (τα δεκαδικά). Δηλαδή, ο αριθμός τυπώνεται σε k θέσεις (με στοίχιση προς τα δεξιά) **από τις οποίες** οι m για τα δεκαδικά.

3. Τι θα εμφανίσει στην οθόνη το πιο κάτω πρόγραμμα και γιατί;

```
#include <stdio.h>

main( )
{
    float 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);
}
```

scanf( ):

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 4, 5, 6) :

- Η συνάρτηση **scanf** διαβάζει από το πληκτρολόγιο.
- Η scanf( ) **μετά το κόμμα** χρειάζεται **δείκτη**.
- Όταν αναφερόμαστε σε **απλή μεταβλητή, ο δείκτης αποτελείται από το & και το όνομα της μεταβλητής**. Π.χ. &ak, αν το ak έχει δηλωθεί ως int ή char ή float κλπ.

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

```
#include <stdio.h>

main( )
{
    int i;

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

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

```
#include<stdio.h>

main( )
{
    float salary;
    printf ("TI  MISTHO  THELETE? ");
    printf ("SYMPLHRWSTE (EYRW)_________\b\b\b\b\b\b\b\b\b");
                                        /* Βάλτε 9 κάτω παύλες */
    scanf ("%f", &salary);
    printf ("\n%5.0f EYRW TO MHNA EINAI %7.0f EYRW TON XRONO\n",
        salary, 12*salary );
}
```

6. 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. Για F βαθμούς Φαρενάιτ, οι αντίστοιχοι βαθμοί Κελσίου C δίνονται από τον τύπο: C=(F-32)*5/9.
- Γράφει στην οθόνη: Η θερμοκρασία είναι -------- βαθμοί Κελσίου. (αντί για τις παύλες να γράψετε τους βαθμούς Κελσίου).

Σταθερές:

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 7, 8) :

- Οι **σταθερές** σε ένα πρόγραμμα ορίζονται με **#define**.
- **Οι τιμές τους δεν αλλάζουν** σε όλη τη διάρκεια εκτέλεσης του προγράμματος.

7. Παρακολουθείστε στο παρακάτω πρόγραμμα την χρήση σταθερών:

```
#include <stdio.h>

#define PI  3.14159
#define DOL  1.17

main( )
{
    printf ("pi = %f \n",PI);
    printf ("euro = %f dollaria.\n", DOL);
}
```

8. Να γραφεί ένα πρόγραμμα που θα διαβάζει από το πληκτρολόγιο 4 float αριθμούς (τις τιμές δηλαδή 4 float μεταβλητών), οι οποίοι αντιπροσωπεύουν το κόστος μιας τηλεόρασης, ενός dvd, 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:

Επεξηγήσεις – υπενθυμίσεις (άσκηση 9) :

- Οι **πίνακες** είναι ομάδες μεταβλητών του **ίδιου τύπου**, οι οποίες έχουν αποθηκευτεί σε **διαδοχικές θέσεις μνήμης**.
  - int pin[100]; /* 100 ακέραιοι */
  - float math[200]; /* 200 float */
  - char name[80]; /* 80 χαρακτήρες */
- Δήλωση μεταβλητής πίνακα: **τύπος όνομα[θέσεις];** (βλ. ανωτέρω)
- Οι αναφορές στα στοιχεία ενός πίνακα ξεκινάνε από το μηδέν
  - pin[0] : πρώτο στοιχείο, δηλαδή ο πρώτος ακέραιος του pin (ακέραια μεταβλητή)
  - pin[99] : τελευταίο στοιχείο, δηλαδή ο τελευταίος ακέραιος του pin (ακέραια μεταβλητή)

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

```
#include <stdio.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]);
}
```

Γ΄. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

10. Να γραφεί ένα πρόγραμμα, το οποίο θα διαβάζει από το πληκτρολόγιο δύο float, οι οποίοι αντιπροσωπεύουν τις δύο κάθετες πλευρές ενός ορθογωνίου τριγώνου. Το πρόγραμμα να υπολογίζει και να γράφει στην οθόνη σε χώρο 8 διαστημάτων, με 2 κλασματικά ψηφία το εμβαδόν του ορθογωνίου τριγώνου.

11. Να γραφεί ένα πρόγραμμα, το οποίο θα διαβάζει από το πληκτρολόγιο ένα ακέραιο, ο οποίος αντιπροσωπεύει μια γωνία σε μοίρες. Το πρόγραμμα να υπολογίζει και να γράφει στην οθόνη σε χώρο 10 διαστημάτων, με 6 κλασματικά ψηφία το μέγεθος της γωνίας σε ακτίνια (rad). Υπενθύμιση: οι 180 μοίρες αντιστοιχούν σε π ακτίνια, δηλαδή (προσεγγιστικά) σε 3.14 ακτίνια.

12. 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 τον μηνιαίο 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 fp, 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.

13. Να γραφεί ένα πρόγραμμα, στο οποίο να δηλώσετε ένα πίνακα float 5 θέσεων, τον pin. Να διαβάσετε τιμές από το πληκτρολόγιο, τις οποίες να καταχωρήσετε στις 5 θέσεις του πίνακα. Στη συνέχεια να γράψετε τις τιμές του πίνακα στην οθόνη ως εξής: σε μια γραμμή θα γράφεται η τιμή του πρώτου στοιχείου του πίνακα με δύο δεκαδικά, θα αφήνονται 5 κενά και δίπλα θα γράφεται η τιμή πάλι του πρώτου στοιχείου του πίνακα με τρία δεκαδικά. Στην από κάτω γραμμή θα γράφεται η τιμή του δεύτερου στοιχείου του πίνακα με δύο δεκαδικά, θα αφήνονται 5 κενά και δίπλα θα γράφεται η τιμή πάλι του δεύτερου στοιχείου του πίνακα με τρία δεκαδικά. Το ίδιο να γίνει και για τα υπόλοιπα τρία στοιχεία του πίνακα.

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:

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 1, 2, 3) :

- Η διαίρεση ακεραίου με ακέραιο δίνει **ακέραιο αποτέλεσμα.**
- Για **πλήρη διαίρεση** πρέπει ο αριθμητής ή ο παρονομαστής (ή και οι δύο) να είναι float.
- Το **modulo (%)** είναι πράξη μεταξύ ακεραίων τύπων και δίνει **το υπόλοιπο της ακέραιας διαίρεσης**.

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

```
#include <stdio.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);
}
```

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 ίσούται 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:

Επεξηγήσεις – υπενθυμίσεις (άσκηση 3) :

- Προτεραιότητα τελεστών (από μεγαλύτερη προς μικρότερη):

  ( ) παρενθέσεις

  \+ - (ως πρόσημα), ++ --

  * / %

  \+ -

  = *= /= %= += -=

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

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

```
#include <stdio.h>

main( )
{
    int   i=1, j=1, k1=10, k2=20, k3=30, k4=40, k5=50, k, h;
    float a=7.0, b=6.0, c=5.0, d=4.0, e, x, y, z ;

    printf ("ARXIKES TIMES i, j, i=%3d, j=%3d\n", i, j);
    k = i++;
    h = ++j;
    printf ("META TIS AYKSHSEIS, i=%3d, j=%3d\n"
            "                    i=%3d, j=%3d\n", i, j, k, h);
```

```
        printf ("ARXIKES TIMES, k1=%3d, k2=%3d, k3=%3d, "
                "k4=%3d, k5=%3d\n", k1, k2, k3, k4, k5);
        k1 += 2;
        k2 -= i;
        k3 *= (8/4);
        k4 /= 2;
        k5 %= 2;
        printf ("META TIS PRAKSEIS, k1=%3d, k2=%3d, k3=%3d, "
                "k4=%3d, k5=%3d\n", k1, k2, k3, k4, k5);
        printf ("ARXIKES TIMES, a=%3.0f, b=%3.0f, c=%3.2f, d=%3.2f\n",
                                        a, b, c, d);
        e = 2.0;
        x = a + b – c / d * e;
        y = a + (b – c) / d * e;
        z = ((a + b) – c / d) * e;
        printf ("META TIS  PRAKSEIS, e=%10.1f\nx=%10.2f\ny=%10.3f"
                                                "\nz=%10.4fn", e, x, y, z);
}
```

Επεξηγήσεις – υπενθυμίσεις (άσκηση 4) :

Εμφάνιση ακεραίου σε χώρο 8 διαστημάτων (στοίχιση δεξιά): με τον προσδιοριστή **%8d**. Για στοίχιση αριστερά σε χώρο 8 διαστημάτων: προσδιοριστής **%-8d**.

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

```
#include <stdio.h>

main( )
{
    int top, score;

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

Μετατροπές τύπων:

Επεξηγήσεις – υπενθυμίσεις (άσκηση 5) :

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

```
float x;
int n; n
= 5;
x = n;          /* int σε float */
```

είτε «εξαναγκασμένες» (**casting**):

```
float x;
int n;
x = 1.5;
n = (int) x;    /* int casting σε float, το n ισούται με 1 */
```

- Συμπληρωματικά, υπολογίστε τις τιμές των 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 = 2/4*10.0 + 2%4 + 10;
b = (int)5.99 + 6.2*2;      b = 3.0/4.0;
b = 3/4;                    b = 3.0/4;
b = 3./4;                   b = 3/4.0;
```

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

```
/* Μετατροπές τύπων */
#include<stdio.h>

main( )
{
    float fl;
    int ak;


    printf("DWSTE ENA float: ");
    scanf("%f", &fl);
    ak = (int) fl;
    printf("%5d\n", ak);
}
```

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

Συμβολοσειρές (γενικά):

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 6, 7, 8, 9, 10) :

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

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

```
#include <stdio.h>

main( )
{
    char name[80];

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

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

```
#include <stdio.h>

#define GOOD "EXEIS WRAIO ONOMA" /* Σταθερή συμβολοσειρά */

main( )
{
    char onoma[80];

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

Μήκος συμβολοσειράς:

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 8, 9, 10) :

- Η συνάρτηση strlen (s) **επιστρέφει το μήκος** της συμβολοσειράς s.
- Προσοχή στην ύπαρξη του **<string.h>**.

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

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

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

    printf ("DWSTE TO ONOMA SAS");
    scanf ("%s", name);
    m = strlen (name);
    printf ("TO ONOMA SAS EXEI %d GRAMMATA \n", m);
}
```

Γ΄. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

Επεξηγήσεις – υπενθυμίσεις (άσκηση 9) :

Με την εντολή **printf ("%*d", a, b);** γράφεται στην οθόνη ο ακέραιος b σε χώρο a διαστημάτων.

9. 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

Επεξηγήσεις – υπενθυμίσεις (άσκηση 10) :

- Μια συμβολοσειρά t **αντιγράφεται** σε μια άλλη s μόνο με τη χρήση της **strcpy (s,t);** (αναλυτικότερα γι' αυτήν στα επόμενα μαθήματα).
- Ελέγχουμε εάν το μήκος της t είναι μεγαλύτερο από το μήκος της s, χρησιμοποιώντας την εντολή if της C. (αναλυτικότερα γι' αυτήν στα επόμενα μαθήματα).

10. 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 αντιγράφει την μικρότερη στην μεγαλύτερη και να εμφανίζει και πάλι τις συμβολοσειρές στην οθόνη.

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( ).

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 1, 2, 3) :

- **ch = getchar( ); διαβάζει** χαρακτήρα ch **με πλήκτρο <enter>**. Έχει τιμή επιστροφής τον χαρακτήρα που διάβασε. Είναι περίπου ισοδύναμη με scanf ("%c", &ch);
- **putchar(ch); γράφει** στην οθόνη τον χαρακτήρα ch. Είναι ισοδύναμη με printf ("%c", ch);
- Βιβλιοθήκη <conio.h>. Πρέπει να έχει γίνει include.
- **ch = getche( );** διαβάζει χαρακτήρα ch **χωρίς πλήκτρο <enter>**. Έχει τιμή επιστροφής τον χαρακτήρα που διάβασε.
- **ch = getch( );** διαβάζει τον χαρακτήρα ch **χωρίς πλήκτρο <enter>** και **ΔΕΝ τον εμφανίζει στην οθόνη**. Έχει τιμή επιστροφής τον χαρακτήρα που διάβασε.

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

```
#include <stdio.h>

main( )
{
    char ch;

    printf ("DWSTE XARAKTHRA KAI META<enter> -> ");
    ch = getchar( );
    printf ("DWSATE %c \n", ch);
    putchar ('B');
    putchar ('Y');
    putchar ('E');
    putchar('\n');
}
```

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

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

main( )
{
    char ch;

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

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 τρείς χαρακτήρες. Το πρόγραμμα θα διαβάζει ένα-ένα χαρακτήρα, κάθε ένα από τους οποίους θα τον αποθηκεύει σε μια μεταβλητή τύπου char. Κάθε χαρακτήρας που διαβάζεται να μην εμφανίζεται στην οθόνη, αλλά αντί γι' αυτόν να εμφανίζεται ένα αστεράκι. Αφού διαβαστούν οι τρείς χαρακτήρες να γραφεί στην οθόνη: *"YOUR PIN IS"* και δίπλα να γραφεί ο κωδικός που δόθηκε

Εντολή if και παραλλαγές της.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 4) :

- Τελεστές : Μεγαλύτερο (**>**), μικρότερο (**<**), μεγαλύτερο ή ίσο (**>=**), μικρότερο ή ίσο (**<=**), ίσο (**==**), διάφορο (**!=**).
- Παράδειγμα σύνταξης της if : **if (a > 0) k++;**
- Παράδειγμα σύνταξης της if…else :

```
if (a > 0) k++;
   else b+=5;
```

- Με την **if…else** μπορούμε να επιλέξουμε **μια από δύο** περιπτώσεις.
- Για **k περιπτώσεις** χρειαζόμαστε k μεμονωμένες if ή k-1 if…else. Προσοχή! **Όχι** μεμονωμένες if και else στο τέλος.
- **Σύνθετη εντολή** : ομάδα απλών εντολών που εκτελούνται όλες αν ισχύει μια συνθήκη ή δεν εκτελείται καμμιά αν δεν ισχύει η συνθήκη. Οι σύνθετες εντολές σηματοδοτούνται από σε άγκιστρα. Μετά το **}** δεν υπάρχει **;**

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

- Το πρόγραμμα αρχικά θα γράφει στην οθόνη: **DWSTE VATHMOYS PROODOY KAI EKSETASTIKHS** και θα περιμένει να διαβάσει δυο 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 ERGASTHRIOY?** και περιμένει να διαβάσει ένα χαρακτήρα, τον ch.
- Αν το ch είναι ίσο με N (εννοούμε δηλαδή ΝΑΙ), τότε το πρόγραμμα κάνει τα εξής:
  - Γράφει στην οθόνη **DWSTE VATHMO ERGASTHRIOY** και 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 Ν, τότε το πρόγραμμα γράφει στην οθόνη: **VATHMOS THEWRIAS** και δίπλα την τιμή του bm, σε χώρο 5 διαστημάτων με 1 δεκαδικό.

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 5, 6, 7, 8, 9):

- Τελεστές : Λογικό και (**&&**), λογικό ή (**||**), λογικό όχι (**!**).
- Εάν θέλουμε να ισχύει μια συνθήκη όταν ισχύουν **ταυτόχρονα δύο (ή περισσότερες) επιμέρους συνθήκες**, αυτές τις συνδέουμε με το **λογικό και (&&)**.
- Εάν θέλουμε να ισχύει μια συνθήκη όταν ισχύει **μία τουλάχιστον από δύο (ή περισσότερες) επιμέρους συνθήκες**, αυτές τις συνδέουμε με το **λογικό ή (||)**.
- Για να διακρίνουμε μεταξύ **k περιπτώσεων** χρειαζόμαστε **k μεμονωμένες if** ή **k-1 if…else**. Προσοχή! **Όχι** μεμονωμένες if και μία else στο τέλος.

5. 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 **LATHOS**.
- Εάν ο the είναι μεταξύ 40 και 50, τότε θα γράφεται στην οθόνη **POLLH ZESTH**.
- Εάν ο the είναι μεταξύ 10 και 40, τότε θα γράφεται στην οθόνη **KALOS KAIROS.**
- Εάν ο the είναι κάτω από 10 (αλλά προφανώς πάνω από -30), τότε θα γράφεται στην οθόνη **KRYO**.

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

- Θα διαβάζει από το πληκτρολόγιο τρείς 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.

Γ'. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

7. 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.

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

- Αρχικά θα γράφεται στην οθόνη: **H EKSISWSH EINAI ax2+bx+c=0.**
- Μετά θα γράφεται: **DWSTE TA a, b KAI c** και θα περιμένει να διαβάσει τρείς 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: **DEN YPARXEI EKSISWSH**
- Εάν ισχύει ότι το a είναι μηδέν και το b είναι διάφορο του μηδενός, τότε θα γράφεται στην οθόνη: **MIA RIZA** 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: **OI RIZES EINAI** 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 ($D=b^2-4*a*c$) είναι θετική, τότε θα υπολογίσει τις ρίζες, θα γράψει στην οθόνη:

  DYO PRAGMATIKES RIZES

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

  MIGADIKES RIZES

  και από κάτω τις τιμές τους. (Για τις τιμές αυτές θα υπολογιστεί χωριστά το πραγματικό μέρος (το οποίο ισούται με –b/(2*a)) και χωριστά το φανταστικό μέρος (το οποίο ισούται με √|D|/(2*a)) . Μετά, αυτά θα γραφτούν δίπλα-δίπλα, έχοντας ανάμεσά τους το +j ή το –j).

9. Να γραφεί ένα πρόγραμμα που θα λύνει και θα υπολογίζει τον φόρο που θα πληρώσει ένας μισθωτός όπως παρακάτω. Το πρόβλημα είναι ένα κλασικό παράδειγμα της λεγόμενης κλιμακωτής χρέωσης:

- Αρχικά θα διαβάζεται από το πληκτρολόγιο ένας float, ο mis, ο οποίος αντιπροσωπεύει το μισθό ενός υπαλλήλου.
- Μετά διαβάζεται ένας float, ο en, ο οποίος αντιπροσωπεύει το ενοίκιο που πληρώνει ο υπάλληλος.
- Στη συνέχεια διαβάζεται ένας float, ο ex, ο οποίος αντιπροσωπεύει τα έξοδα διαβίωσης του υπαλλήλου.
- Το φορολογητέο ποσόν (fp) υπολογίζεται εάν από τους 14 μισθούς ενός έτους αφαιρέσουμε 12 ενοίκια και τα έξοδα διαβίωσης. Το fp δεν θα επιτρέπεται να είναι αρνητικό. Εάν είναι, τότε τίθεται ίσο με μηδέν.
- Ο φόρος υπολογίζεται από το fp ως εξής:
  - Εάν το fp είναι μέχρι 10000, τότε ο φόρος θα είναι ίσος με το 10% του fp.
  - Εάν το fp 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 fp 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**.
- Επαναληπτική εντολή **for**.

Β'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

Εντολή switch.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 1):

- Η switch είναι ισοδύναμη με **πολλές εντολές if**, από τις οποίες **θα εκτελεστεί μόνο μία**. Σύνταξη:

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

- Τα c, c1, c2 κλπ είναι **ακέραιοι** ή **χαρακτήρες**.
- Αν το c είναι διάφορο από τα c1, c2 κλπ, εκτελούνται οι εντολές μετά το default.
- Η **break** προκαλεί **άμεσο τερματισμό** της switch.

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

```
#include <stdio.h>
#include <conio.h>          // Γιατην getche( )
main( )
{
    char ch;
    printf ("Menu :\n");
    printf ("A – Play \n");
    printf ("B – Work \n");
    printf ("C – EXIT \n");
    printf ("\n Your choice -> ");
```

```
        ch = getche( );
        printf ("\n\n");
        switch (ch)
        {
            case 'A' : printf ("Now playing ... \n"); break;
            case 'B' : printf ("Now working ... \n"); break;
            case 'C' : printf ("Bye ... \n"); break;
            default : printf ("Bad choice\n"); break;
        }
    }
```

Επαναληπτική εντολή for

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 2, 3):

- Η **for** έχει την εξής μορφή:

  for (Αρχική; Συνθήκη; Επόμενη) Ομάδα Εντολών
- Η **"Αρχική"** πρόταση εκτελείται **μόνο την πρώτη φορά** που φτάνουμε στην for.
- Η **"Συνθήκη"** ελέγχεται και εάν ισχύει, εκτελείται η **"Ομάδα Εντολών"**, αλλιώς η επανάληψη τερματίζεται.
- Μετά την εκτέλεση της **"Ομάδας Εντολών"** εκτελείται η **"Επόμενη"** πρόταση.
- Ελέγχεται πάλι η **"Συνθήκη"** και συνεχίζεται ο πιο πάνω κύκλος, μέχρι η **"Συνθήκη"** να μην ισχύει πλέον.

  Π.χ. **for (i=1; i<=3; i++) printf ("%3d",i);** // γράφει **υυ1υυ2υυ3**
- Η **"Ομάδα Εντολών"** μπορεί να αποτελείται από μια **απλή εντολή** (όπως παραπάνω) ή μια **σύνθετη εντολή**, οπότε βρίσκεται μέσα σε άγκιστρα.

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.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 4):

Οι χαρακτήρες του κώδικα ASCII έχουν αύξοντες αριθμούς από 0 έως 255.

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. Ο χαρακτήρας z βρίσκεται μετά τον χαρακτήρα # στον κώδικα ASCII.

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 6, 7, 9):

Οι **μετρητές** και οι **αθροιστές** σε ένα πρόγραμμα είναι μεταβλητές, στις οποίες πρέπει να αποδίδουμε **αρχική τιμή μηδέν**.

6. Να γραφεί ένα πρόγραμμα, το οποίο θα διαβάζει 10 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 + …….

1.0 - 1.0/2.0 + 1.0/3.0 - 1.0/4.0 + …….

Γράψτε ένα πρόγραμμα, το οποίο να υπολογίζει το συνολικό άθροισμα κάθε μιας από τις δύο σειρές μέχρι κάποιο αριθμό όρων (π.χ. μέχρι τον 100ο όρο ή τον 200ο κλπ). Το πρόγραμμα ξεκινώντας, να ζητάει και να διαβάζει από το πληκτρολόγιο το πλήθος των όρων που θα αθροιστούν.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 8):

- Το **μήκος μιας συμβολοσειράς** δίνεται από τη συνάρτηση **strlen( )**
- Αν δηλώσουμε τον πίνακα χαρακτήρων char nom[30]; τότε τα nom[0], nom[1], nom[2] κλπ… είναι χαρακτήρες.

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.

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 9, 12):

Έστω int pin[30]; Για να διαβάσουμε το 1ο στοιχείο του πίνακα: scanf (**"**%d**"**, &pin[0]); Για το 2ο στοιχείο: scanf (**"**%d**"**, &pin[1]); κλπ για όλα τα στοιχεία του πίνακα.

9. Γράψτε ένα πρόγραμμα, το οποίο να γεμίζει από το πληκτρολόγιο ένα πίνακα ακεραίων 10 θέσεων, ο οποίος λέγεται mat. Αφού γεμίσει ο πίνακας, το πρόγραμμα να αθροίζει μεταξύ τους τα στοιχεία του πίνακα και επίσης να μετρά πόσα από τα στοιχεία αυτά είναι θετικοί ακέραιοι. Να εμφανίζονται τα αποτελέσματα στην οθόνη.

Γ’. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

Επεξηγήσεις – υπενθυμίσεις (άσκηση 10):

Η **τετραγωνική ρίζα** δίδεται από τη συνάρτηση **sqrt ( )**, η οποία έχει float τιμή επιστροφής. Πρέπει επίσης να έχετε κάνει include το αρχείο επικεφαλίδας **math.h**

10. Να γραφεί ένα πρόγραμμα, το οποίο θα διαβάζει 10 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 να ελέγξετε το πρόγραμμά σας, οι αριθμοί: 1.1 2 3.4 6 4.3 2.9 1 2.1 9 6.5 δίνουν τυπική απόκλιση 2.482761).

Επεξηγήσεις – υπενθυμίσεις (άσκηση 11):

- Ο **αλγόριθμος εύρεσης του μεγίστου** σε ένα πίνακα είναι ο εξής:
  - Θέτουμε ως μέγιστο (meg) την πρώτη τιμή του πίνακα.
  - Συγκρίνουμε την δεύτερη τιμή του πίνακα με το meg. Αν η δεύτερη τιμή είναι μεγαλύτερη από το meg, κάνουμε το meg ίσο με την δεύτερη τιμή.
  - Συνεχίζουμε το ίδιο για όλα τα στοιχεία του πίνακα.
  - Όταν τελειώσουμε, το meg περιέχει την μέγιστη τιμή.
- Ο **αλγόριθμος εύρεσης του ελαχίστου** είναι αντίστοιχος, αλλά ξεκινούμε θέτοντας ως ελάχιστη την πρώτη τιμή του πίνακα και συγκρίνουμε κάθε στοιχείο με την ελάχιστη τιμή

11. Γράψτε ένα πρόγραμμα, το οποίο θα διαβάζει ακέραιους από το πληκτρολόγιο και θα γεμίζει ένα πίνακα ακεραίων 10 θέσεων. Στη συνέχεια θα βρίσκει και θα γράφει στην οθόνη τον μέγιστο και τον ελάχιστο ακέραιο του πίνακα.

12. Γράψτε ένα πρόγραμμα, το οποίο θα διαβάζει ακέραιους από το πληκτρολόγιο και θα γεμίζει ένα πίνακα ακεραίων 10 θέσεων, τον pin. Στη συνέχεια θα μεταφέρει τους ακέραιους αυτούς σε ένα άλλο πίνακα, τον mat, αλλά από το τέλος προς την αρχή. Δηλαδή ο 1ος ακέραιος του pin θα γίνεται τελευταίος στον mat, ο 2ος του pin θα γίνεται προτελευταίος στον mat κ.ο.κ. Αφού ολοκληρωθεί η μεταφορά, οι πίνακες να γράφονται στην οθόνη.

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 δύο φορές, μια φορά με την χρήση της while και μια φορά με την χρήση της do-while. Τα ίδια προγράμματα μπορούν φυσικά να πραγματοποιηθούν και με την χρήση της εντολής for. Ομοίως, τα προγράμματα που εκπονήθηκαν στο προηγούμενο εργαστήριο με την χρήση της for, μπορούν να προαγματοποιηθούν και με την χρήση της while ή της do...while. Συνιστάται να το κάνετε.

Επεξηγήσεις – υπενθυμίσεις (Για όλες τις ασκήσεις):

- 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 **"Ομάδα Εντολών"** μπορεί να αποτελείται από μια **απλή εντολή** ή μια **σύνθετη εντολή**, οπότε βρίσκεται μέσα σε άγκιστρα.
- 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**.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 1) :

- **ch = getche( );** διαβάζει τον χαρακτήρα ch **χωρίς πλήκτρο <enter>**. Έχει τιμή επιστροφής τον χαρακτήρα που διάβασε.
- **ch = getch( );** διαβάζει τον χαρακτήρα ch **χωρίς πλήκτρο <enter>** και **ΔΕΝ τον εμφανίζει στην οθόνη**. Έχει τιμή επιστροφής τον χαρακτήρα που διάβασε.
- Βιβλιοθήκη <conio.h>. Πρέπει να έχει γίνει include.
- Οι **μετρητές** και οι **αθροιστές** σε ένα πρόγραμμα είναι μεταβλητές, στις οποίες πρέπει να αποδίδουμε **αρχική τιμή μηδέν**.

1. Να γραφεί ένα πρόγραμμα, το οποίο να διαβάζει συνεχώς χαρακτήρες από το πληκτρολόγιο (με την getche( ) ή την getch( )), μέχρι να δοθεί ο χαρακτήρας Α. Το πρόγραμμα μετρά τους χαρακτήρες που διαβάζει αυξάνοντας ένα μετρητή. Αφού ολοκληρωθεί το διάβασμα, να γράφετε στην οθόνη την τιμή του μετρητή.

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 2, 3) :

- **Προσδιοριστής %s** για διάβασμα από το πληκτρολόγιο και εμφάνιση στην οθόνη συμβολοσειρών.
- Οι συμβολοσειρές αποθηκεύονται σε **πίνακες τύπου char**.
- **Σταθερές συμβολοσειρές**: Γράφονται μέσα σε **διπλά εισαγωγικά**.

2. Να γραφεί ένα πρόγραμμα, στο οποίο να διαβάζετε 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.

3. Να γραφεί πρόγραμμα, το οποίο θα δέχεται μια σταθερά, την ΜΗΝ με τιμή ERGASIA SE EKSELIKSH. Θα εμφανίζει στην οθόνη την τιμή της ΜΗΝ και μετά την φράση:

SYNEXEIA? (N/O)

Το πρόγραμμα θα περιμένει να διαβάσει ένα χαρακτήρα από το πληκτρολόγιο, τον ch και θα συνεχίζει το ίδιο, όσο ο χαρακτήρας που διαβάστηκε δεν είναι το Ο (με το Ο δηλαδή σταματάει). Όσο συνεχίζεται η εκτέλεση του προγράμματος (όσο δηλαδή δεν έχει πατηθεί το Ο), το πρόγραμμα θα κάνει τα εξής:

- Εάν ο χαρακτήρας ch δεν είναι το Ν, τότε γράφεται στην οθόνη:

  LATHOS. DWSTE ALLO XARAKTHRA

- Εάν ο χαρακτήρας που πατήθηκε είναι Ν, γράφεται στην οθόνη η τιμή του ΜΗΝ, από κάτω η φράση:

  SYNEXEIA? (N/O)

  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 Ν.

Προτάσεις break και continue.

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 4, 5, 6) :

- 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).

4. Εκτελέστε το παρακάτω πρόγραμμα:

```
#include <stdio.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; */
}
```

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 5, 6) :

- **ch = getche( );** διαβάζει τον χαρακτήρα ch **χωρίς πλήκτρο <enter>**. Έχει τιμή επιστροφής τον χαρακτήρα που διάβασε.
- **ch = getch( );** διαβάζει τον χαρακτήρα ch **χωρίς πλήκτρο <enter>** και **ΔΕΝ τον εμφανίζει στην οθόνη**. Έχει τιμή επιστροφής τον χαρακτήρα που διάβασε.
- Βιβλιοθήκη <conio.h>. Πρέπει να έχει γίνει include.
- Οι **μετρητές** και οι **αθροιστές** σε ένα πρόγραμμα είναι μεταβλητές, στις οποίες πρέπει να αποδίδουμε **αρχική τιμή μηδέν**.

5. Γράψτε ένα πρόγραμμα, το οποίο θα κάνει τα εξής:

Θα διαβάζει 15 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 15 επαναλήψεων μόνο 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 5, αλλά δεν θα μετράει τους χαρακτήρες B και b.

Γ'. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

7. Να γραφεί ένα πρόγραμμα, το οποίο να κάνει τα εξής:

Θα δημιουργεί ένα τυχαίο θετικό ακέραιο αριθμό μεταξύ 0 και 100. Για να το κάνετε αυτό χρειάζεστε ένα "σπόρο" για να ξεκινήσετε την παραγωγή τυχαίων αριθμών. Αυτό τον "σπόρο" τον παίρνετε από το ρολόι του υπολογιστή δίνοντας σαν πρώτη εντολή του προγράμματός σας το εξής:

srand (time (NULL));

Στη συνέχεια θα καλέσετε την συνάρτηση rand( ), η οποία δεν χρειάζεται ορίσματα, δημιουργεί ένα τυχαίο θετικό ακέραιο και έχει τιμή επιστροφής αυτόν τον ακέραιο που δημιούργησε. Σκεφθείτε με ποιον τρόπο θα περιορίσετε τον ακέραιο αυτόν, ώστε να μείνει στο διάστημα 0 έως 100.

Οι συναρτήσεις rand( ) και srand( )χρειάζονται τα αρχεία επικεφαλλίδας (header) time.h και stdlib.h.

Αφού παραχθεί ο τυχαίος ακέραιος (π.χ. ο num), να διαβάζονται συνεχώς 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.

8. 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.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 9) :

Για να μετατρέψουμε ένα αριθμό του δεκαδικού συστήματος σε δυαδικό, κάνουμε **συνεχείς** (ακέραιες) **διαιρέσεις με το 2, όσο το πηλίκο που προκύπτει δεν είναι μηδέν**. Σε κάθε διαίρεση **κρατούμε το υπόλοιπο**. Ο δυαδικός αριθμός αποτελείται από **τα υπόλοιπα των διαιρέσεων, αν τα πάρουμε όμως από το τέλος προς την αρχή**.

9. Γράψτε ένα πρόγραμμα, το οποίο, να διαβάζει ένα θετικό ακέραιο και στη συνέχεια να τον παρουσιάζει στην οθόνη στο δυαδικό αριθμητικό σύστημα.

ΕΡΓΑΣΤΗΡΙΟ 7

Α'. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 7 καλύπτονται οι **εμφωλευμένες** επαναληπτικές εντολές (**for μέσα σε for, while μέσα σε while** κλπ).

Β'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

Εμφωλευμένες επαναληπτικές εντολές (for μέσα σε for κλπ).

Συνιστάται να γράψετε τα παρακάτω προγράμματα χρησιμοποιώντας κάθε δυνατό συνδυασμό εμφωλευμένων επαναληπτικών εντολών (π.χ. for μέσα σε for, for μέσα σε while, while μέσα σε for, while μέσα σε while, do-while μέσα σε for κλπ). Για τήν ώρα του εργαστηρίου είναι αρκετό να χρησιμοποιήσετε δύο από τους συνδυασμούς (όποιους θέλετε).

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 1, 2, 3) :

Σε κάθε κύκλο της εξωτερικής επανάληψης εκτελούνται όλοι οι κύκλοι τής εσωτερικής επανάληψης. Στο παρακάτω, για παράδειγμα:

```
for (j=1;j<=5;j++) {
    for (k=1;k<4;k++)
        printf ("%5d", j+k); }
```

η printf θα εκτελεστεί 15 φορές.

1. Γράψτε ένα πρόγραμμα, το οποίο να εμφανίζει στην οθόνη το εξής (με τη χρήση εμφωλευμένων επαναληπτικών εντολών και ΟΧΙ προφανώς με πέντε printf, κάθε μια από τις οποίες να γράφει 123456789):

```
123456789
123456789
123456789
123456789
123456789
```

Επεξηγήσεις – υπενθυμίσεις (άσκηση 2) :

Παρατηρείστε στην άσκηση 2, ότι οι γραμμές παραμένουν πέντε, όπως και στην άσκηση 1. Δείτε στην 1 ποιος καθόριζε το πλήθος των γραμμών και τον αριθμό στον οποίο τελειώνει κάθε γραμμή (δεν θα τα αλλάξετε αυτά). Παρατηρείστε στην άσκηση 2 ότι την 1η φορά οι αριθμοί ξεκινούν από το 1, την 2η φορά οι αριθμοί ξεκινούν από το 2 κλπ, άρα στο πρόγραμμα 1 θα τροποποιήσετε κατάλληλα τον αριθμό από τον οποίο αρχίζει κάθε γραμμή.

2. Να τροποποιηθεί το πρόγραμμα της άσκησης 1 για να εμφανίσει:

```
123456789
23456789
3456789
456789
56789
```

Επεξηγήσεις – υπενθυμίσεις (άσκηση 3) :

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". Άρα **με την printf ("\xDB"); μπορείτε να γκριζάρετε στην οθόνη χώρο ενός διαστήματος**.

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

Επεξηγήσεις – υπενθυμίσεις (άσκηση 4) :

- Το **μέγεθος ενός πίνακα** δίνεται πάντα ως **ακέραιος αριθμός** ή ορίζεται ως **ακέραια σταθερά**.
- Οι **μετρητές** και οι **αθροιστές** σε ένα πρόγραμμα είναι μεταβλητές, στις οποίες πρέπει να αποδίδουμε **αρχική τιμή μηδέν**.
- Η **break διακόπτει οριστικά** την εκτέλεση μιας επαναληπτικής εντολής, μέσα στην οποία βρίσκεται (for, while, do-while). Αν η **break** βρίσκεται μέσα **σε εμφωλευμένη επανάληψη**, **διακόπτεται μόνο η εσωτερική επανάληψη**.

4. Γράψτε ένα πρόγραμμα, στο οποίο να κάνετε τα εξής:

- Να δηλώσετε ένα πίνακα ακεραίων Ν θέσεων, τον pin, τον οποίο να γεμίσετε από το πληκτρολόγιο.
- Αφού γεμίσει ο πίνακας να αθροίζετε μεταξύ τους τα στοιχεία του. Εάν το άθροισμα (έστω sum) ξεπεράσει το 100, να κάνετε τα εξής:
  - Θα ελέγχετε ένα-ένα τα υπόλοιπα στοιχεία του πίνακα (μετά δηλαδή από αυτό που έκανε το sum να ξεπεράσει το 100). Κάθε θετικό από αυτά θα το προσθέτει σε ένα άθροισμα, το sump.
  - Κάθε αρνητικό από τα παραπάνω στοιχεία θα το προσθέτει σε ένα άθροισμα, το sumn.
  - Εάν συναντήσετε στοιχείο ίσο με μηδέν, τότε τερματίζονται οι επαναλήψεις σας, δεν εξετάζετε δηλαδή άλλο στοιχείο του πίνακα και το πρόγραμμα τερματίζεται. Πριν τον τερματισμό, να γράφεται στην οθόνη η τιμή του sum.
- Εάν ο τερματισμός των επαναλήψεων δεν προέκυψε επειδή συναντήσατε στοιχείο ίσο με μηδέν, όπως αναφέρεται παραπάνω, στην οθόνη να γράφονται οι τιμές των sump και sumn κια μετά το πρόγραμμα τερματίζεται.

Γ'. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

Επεξηγήσεις – υπενθυμίσεις (άσκηση 5) :

Παρόμοιο πρόβλημα με την άσκηση 2. Εδώ οι γραμμές παραμένουν πέντε, ξεκινούν πάντα από το 1, αλλά 1η φορά οι αριθμοί τελειώνουν στο 1, την 2η φορά τελειώνουν στο 2 κλπ.

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

```
1
12
123
1234
12345
```

6. Παραλλαγή της πιο πάνω άσκησης 3: Να ζητάει το πρόγραμμα τη στήλη από όπου θα αρχίσει το "γκριζάρισμα", καθώς και το μέγεθος της γκρίζας περιοχής (πλήθος γραμμών και στηλών).

7. Γράψτε ένα πρόγραμμα, το οποίο να εμφανίζει στην οθόνη τον πίνακα προπαίδειας των αριθμών 1 έως 10 ως εξής:

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

8. Γράψτε προγράμματα, τα οποία να εμφανίζουν τρίγωνα μορφής:

α)

```
*****
****
***
**
*
```

β)

```
*
**
***
****
*****
```

ΕΡΓΑΣΤΗΡΙΟ 8

Α'. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 8 καλύπτονται τα παρακάτω θέματα:

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

Β'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

Συναρτήσεις με ορίσματα, τιμή επιστροφής συνάρτησης.

Επεξηγήσεις – υπενθυμίσεις (για όλες τις ασκήσεις) :

- Η **δήλωση** μιας συνάρτησης βρίσκεται πριν την main( ) και περιλαμβάνει τον **τύπο** της συνάρτησης, το **όνομά** της, το **είδος των παραμέτρων** και **ελληνικό ερωτηματικό**.
- Η **κλήση** μιας συνάρτησης, αποτελείται από το **όνομά** της και τις **τιμές των ορισμάτων**.
- Στην επικεφαλλίδα του **ορισμού** της συνάρτησης δηλώνονται **τόσες παράμετροι**, **όσα και τα ορίσματα** και η αντιστοίχιση ορισμάτων σε παραμέτρους γίνεται με τη σειρά, ένα προς ένα, πριν αρχίσει να εκτελείται η συνάρτηση.
- Οι **παράμετροι** της συνάρτησης **δηλώνονται ΜΟΝΟ μέσα στις παρενθέσεις** της επικεφαλλίδας και όχι μέσα στη συνάρτηση.
- Μεταβλητές που έχουν δηλωθεί **μέσα σε κάποια συνάρτηση** είναι γνωστές **ΜΟΝΟ στη συνάρτηση, στην οποία έχουν δηλωθεί**. Το ίδιο ισχύει **και για τις παραμέτρους** της συνάρτησης.
- Μια συνάρτηση, η οποία **τερματίζεται με return** γίνεται **ίση με την τιμή που ακολουθεί το return** (**τιμή επιστροφής** της συνάρτησης) και μπορεί να αποδοθεί σε μια μεταβλητή κατά την κλήση της. Αν π.χ. η συνάρτηση max(x, y) έχει τιμή επιστροφής ακέραιο, ισχύει: ak=max(x,y); όπου το ak είναι ακέραια μεταβλητή.

1. Να παρατηρήσετε το παρακάτω πρόγραμμα και να δείτε τι κάνει πριν το εκτελέσετε (υπάρχουν σχετικά σχόλια).

```
#include <stdio.h>

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

main( )
{
    int m;

    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 */
}


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 ΜΕΤΑ 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;
}
```

Επεξηγήσεις – υπενθυμίσεις (άσκηση 4) :

Για τον υπολογισμό της τετραγωνικής ρίζας, η C διαθέτει την συνάρτηση sqrt( ) στο math.h.

2. Να γραφεί μία συνάρτηση, η οποία λέγεται computeraki( ).

Στην main( ) θα διαβάζονται δύο ακέραιοι, οι a και b και ένας χαρακτήρας, ο ch. Στη συνέχεια θα καλείται η συνάρτηση computeraki( ), η οποία 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.

Από την main( ) θα γράφεται στην οθόνη το αποτέλεσμα της πράξης, το οποίο επέστρεψε η συνάρτηση.

Από τη main( ) να αποκλείεται χαρακτήρας, ο οποίος δεν είναι ένας από τους παραπάνω, καθώς και η διαίρεση με μηδέν.

3. Να γραφεί μία συνάρτηση, η οποία θα 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 power( ) θα υπολογίζει τον float υψωμένο στην ακέραια δύναμη.

Να ΜΗ χρησιμοποιήσετε την έτοιμη συνάρτηση της C για ύψωση σε δύναμη, η οποία λέγεται pow( ) και ευρίσκεται στο math.h

Επεξηγήσεις – υπενθυμίσεις (άσκηση 4) :

Για τον υπολογισμό της τετραγωνικής ρίζας, η C διαθέτει την συνάρτηση sqrt( ) στο math.h.

4. Να γράψετε ένα πρόγραμμα, το οποίο θα ζητά και θα διαβάζει από το πληκτρολόγιο τρείς ακεραίους, τους a, b και c, οι οποίοι υποτίθεται ότι αντιπροσωπεύουν τις τρείς 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. Είναι αρκετό να διαπιστώσουμε ότι ισχύει αυτό για ένα μόνο συνδυασμό των τριών πλευρών.

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 τρείς ακέραιοι δεν αποτελούν τρίγωνο, τότε η συνάρτηση embadon() να επιστρέφει τιμή 0.

Το πρόγραμμα να συνεχίζεται, όσο και οι τρείς ακέραιοι που διαβάζονται δεν ξεπερνούν το 100. Να καλείτε δηλαδή συνεχώς την συνάρτηση, όσο ισχύει το παραπάνω. Κάθε φορά (σε κάθε κλήση) 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 τρείς ακέραιοι που διαβάστηκαν).

Αναδρομικές συναρτήσεις.

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 5, 6, 14) :

- **«Αναδρομικές»** λέγονται οι συναρτήσεις, οι οποίες καλούν τον εαυτό τους.
- Κάθε αναδρομική συνάρτηση **περιλαμβάνει μια if** (ή πιο γενικά μια εντολή ελέγχου) για τον καθορισμό της συνθήκης τερματισμού.

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

```
#include<stdio.h>

void  test (int);

main( )
{
    test (5);                          /* «Πυροδοτεί» 6 κλήσεις της test( ) */
}

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

6. Προσπαθείστε να γράψετε ένα πρόγραμμα, στο οποίο θα διαβάζετε 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.

Γ’. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

7. Να γραφεί ένα πρόγραμμα, στη main( ) του οποίου θα διαβάζονται δύο ακέραιοι από το πληκτρολόγιο, έστω οι row και cols και δύο χαρακτήρες, οι ch και xar. Στη συνέχεια θα καλείται μια συνάρτηση, η plot( ), η οποία να γράφει στην οθόνη μια γραμμή με cols φορές τον χαρακτήρα ch, στη συνέχεια μια γραμμή με cols φορές τον χαρακτήρα xar, ξανά μια γραμμή με cols φορές τον χαρακτήρα ch κ.ο.κ. Αυτό θα επαναλαμβάνεται για rows γραμμές.

8. Στη main( ) ενός προγράμματος να διαβάζετε τρείς ακεραίους, οι οποίοι αντιστοιχούν σε κάποια χρονική διάρκεια σε ώρες, λεπτά και δευτερόλεπτα. Να καλείτε μια συνάρτηση, την seconds, η οποία θα υπολογίζει την χρονική αυτή διάρκεια σε δευτερόλεπτα και θα επιστρέφει τον αριθμό των δευτερολέπτων στην main( ), από όπου θα εμφανίζεται στην οθόνη.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 9) :

- Η άσκηση είναι παραλλαγή της άσκησης 3 του εργαστηρίου 5.
- Οι **μετρητές** και οι **αθροιστές** σε ένα πρόγραμμα είναι μεταβλητές, στις οποίες πρέπει να αποδίδουμε **αρχική τιμή μηδέν**.

9. Να γραφεί ένα πρόγραμμα, στη main( ) του οποίου θα διαβάζεται ένας ακέραιος από το πληκτρολόγιο, έστω ο ak. Στη συνέχεια θα καλείται μια συνάρτηση, η οποία θα αθροίζει τους ακέραιους αριθμούς από το 1 έως το ak. Η συνάρτηση θα επιστρέφει στη main( ) την τιμή που υπολόγισε, και η main( ) θα εμφανίζει το αποτέλεσμα στην οθόνη.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 10) :

- Η άσκηση είναι παραλλαγή της άσκησης 6 του εργαστηρίου 5.
- Οι **μετρητές** και οι **αθροιστές** σε ένα πρόγραμμα είναι μεταβλητές, στις οποίες πρέπει να αποδίδουμε **αρχική τιμή μηδέν**.

10. Να γραφεί ένα πρόγραμμα, στη main( ) του οποίου θα διαβάζεται ένας ακέραιος από το πληκτρολόγιο, έστω ο ak. Στη συνέχεια θα καλείται μια συνάρτηση, η οποία θα διαβάζει ak float αριθμούς από το πληκτρολόγιο. Η συνάρτηση θα μετρά πόσοι από τους αριθμούς αυτούς είναι θετικοί, θα τους αθροίζει (τους θετικούς) και θα επιστρέφει στη main( ) το άθροισμα που υπολόγισε. Η main( ) θα γράφει το άθροισμα στην οθόνη.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 11) :

- Η άσκηση είναι παραλλαγή της άσκησης 7 του εργαστηρίου 5.
- Οι **μετρητές** και οι **αθροιστές** σε ένα πρόγραμμα είναι μεταβλητές, στις οποίες πρέπει να αποδίδουμε **αρχική τιμή μηδέν**.
- Στην δεύτερη σειρά οι περιττοί όροι προστίθενται, ενώ οι άρτιοι αφαιρούνται.

11. Θεωρείστε τις παρακάτω σειρές:

1.0+ 1.0/2.0 + 1.0/3.0 + 1.0/4.0 + …….

1.0 - 1.0/2.0 + 1.0/3.0 - 1.0/4.0 + …….

Να γραφεί ένα πρόγραμμα, στη main( ) του οποίου θα διαβάζεται ένας ακέραιος από το πληκτρολόγιο, έστω ο ak. Στη συνέχεια θα καλούνται δυο συναρτήσεις, η one( ) και η two( ).

Η συνάρτηση one( ) να υπολογίζει το άθροισμα της πρώτης σειράς μέχρι τον ak όρο, να επιστρέφει το άθροισμα αυτό στη main ( ), η οποία να γράφει το άθροισμα στην οθόνη.

Η συνάρτηση two( ) να υπολογίζει το άθροισμα της δεύτερης σειράς μέχρι τον ak όρο, να επιστρέφει το άθροισμα αυτό στη main ( ), η οποία να γράφει το άθροισμα στην οθόνη.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 12) :

- Η άσκηση είναι παραλλαγή της άσκησης 10 του εργαστηρίου 5.
- Οι **αθροιστές** σε ένα πρόγραμμα είναι μεταβλητές, στις οποίες πρέπει να αποδίδουμε **αρχική τιμή μηδέν**.
- Η τυπική απόκλιση ενός πλήθους αριθμών είναι η τετραγωνική ρίζα της εξής διαφοράς: μέση τιμή των τετραγώνων των αριθμών μείον το τετράγωνο της μέσης τιμής. (Για να ελέγξετε το πρόγραμμά σας, οι αριθμοί: 1.1 2 3.4 6 4.3 2.9 1 2.1 9 6.5 δίνουν τυπική απόκλιση 2.482761).

12. Να γραφεί ένα πρόγραμμα, στη main( ) του οποίου θα διαβάζεται ένας ακέραιος από το πληκτρολόγιο, έστω ο ak. Στη συνέχεια θα καλείται μια συνάρτηση, στην οποία θα διαβάζονται ak float αριθμοί από το πληκτρολόγιο. Η συνάρτηση θα υπολογίζει την τυπική απόκλιση (standard deviation) αυτών των αριθμών και θα επιστρέφει το αποτέλεσμα στην main( ). Από την main( ) το αποτέλεσμα να γράφεται στην οθόνη.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 13) :

- Η άσκηση είναι παραλλαγή της άσκησης 1 του εργαστηρίου 6.
- Οι **μετρητές** και οι **αθροιστές** σε ένα πρόγραμμα είναι μεταβλητές, στις οποίες πρέπει να αποδίδουμε **αρχική τιμή μηδέν**.
- Να διαβάσετε χαρακτήρες με την χρήση της συνάρτησης getche( ).

13. 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 χαρακτήρας ίσος με τον ch. Η συνάρτηση μετρά τους χαρακτήρες που διαβάζει αυξάνοντας ένα μετρητή, επιστρέφει το πλήθος τους στη main( ), η οποία και το γράφει στην οθόνη.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 14) :

Η σειρά Fibonacci είναι μια ακολουθία ακεραίων, στην οποία κάθε αριθμός ισούται με το άθροισμα των δύο προηγούμενων. Οι πρώτες δύο τιμές της σειράς είναι το 0 και το 1. Άρα η ακολουθία είναι:

0 1 1 2 3 5 8 13 21 34... κλπ

14. (**Αναδρομική συνάρτηση**). Να γράψετε ένα πρόγραμμα, το οποίο υπολογίζει τους n πρώτους αριθμούς της σειράς Fibonacci. Το n είναι ακέραιος και θα διαβάζεται από το πληκτρολόγιο.

ΕΡΓΑΣΤΗΡΙΟ 9

Α'. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 9 καλύπτονται τα παρακάτω θέματα:

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

Β'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

Δείκτες. Δηλώσεις δεικτών.

Επεξηγήσεις – υπενθυμίσεις (όλες οι ασκήσεις) :

- Η δήλωση μιας μεταβλητής (π.χ: int ak;) έχει ως επακόλουθο τη δημιουργία του δείκτη (σε ακέραιο) &ak.
- Η δήλωση: int *ptr; δημιουργεί απ'ευθείας τον δείκτη σε ακέραιο ptr.
- Το *** μέσα στο πρόγραμμα** επιτρέπεται να υπάρχει **ΜΟΝΟ μπροστά από όνομα δείκτη** και σημαίνει **«τα περιεχόμενα του δείκτη»**.
- Η **τιμή ενός δείκτη** ισούται με τη διεύθυνση μνήμης του byte στο οποίο είναι τοποθετημένος ο δείκτης και εμφανίζεται στην οθόνη με την χρήση του **προσδιοριστή %p**.

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

```
#include<stdio.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 );
}
```

Επεξηγήσεις – υπενθυμίσεις (άσκηση 2) :

Στην θέση k ενός πίνακα ακεραίων pin βρίσκεται ο **ακέραιος pin[k]** και σε αυτόν **δείχνει ο δείκτης &pin[k]**, άρα το διάβασμα αυτού του ακεραίου γίνεται με: scanf (“%d”, &pin[k]);

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.

«Πέρασμα» διευθύνσεων σε συνάρτηση.

Προσοχή!! Όλες οι συναρτήσεις που ζητούνται στην συνέχεια υποτίθεται ότι **καλούνται μια μόνο φορά** από την main( ), **δεν επιτρέπεται δε η χρήση εξωτερικών μεταβλητών**.

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 3, 4, 5, 6, 7, 8, 9, 10) :

- Κάθε συνάρτηση πριν αρχίσει να δουλεύει **δημιουργεί** τις παραμέτρους της, οι οποίες είναι **αντίγραφα των ορισμάτων** με τα οποία έγινε η κλήση της.
- Εάν θέλουμε μια συνάρτηση **να ενημερώνει** την main( ) για κάποιες τιμές **εκτός από αυτήν την οποία επιστρέφει** (με το return), η κλήση της συνάρτησης γίνεται με **ορίσματα δείκτες**.

3. Να γράψετε ένα πρόγραμμα, στη main( ) του οποίου να δηλώσετε και να δώσετε τιμές σε δυό ακέραιες μεταβλητές. Να καλέσετε μετά μια συνάστηση, 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 τον 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.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 6) :

- Η άσκηση είναι παραλλαγή της άσκησης 6 του εργαστηρίου 5 και της άσκησης 10 του εργαστηρίου 8.
- Οι **μετρητές** και οι **αθροιστές** σε ένα πρόγραμμα είναι μεταβλητές, στις οποίες πρέπει να αποδίδουμε **αρχική τιμή μηδέν**.

6. Να γραφεί ένα πρόγραμμα, στη main( ) του οποίου θα διαβάζεται ένας ακέραιος από το πληκτρολόγιο, έστω ο ak. Στη συνέχεια θα καλείται μια συνάρτηση, η οποία θα διαβάζει ak float αριθμούς από το πληκτρολόγιο. Η συνάρτηση:

- Θα μετρά πόσοι από τους αριθμούς αυτούς είναι θετικοί (έστω cp το πλήθος) και θα τους αθροίζει (τους θετικούς, έστω sump το άθροισμα)
- Θα μετρά πόσοι από τους αριθμούς αυτούς είναι αρνητικοί (έστω cn το πλήθος) και θα τους αθροίζει (τους αρνητικούς, έστω sumn το άθροισμα)

Η main( ) να γράφει στην οθόνη τα cp, sump, cn και sumn.

Γ'. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

7. 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 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.

8. Να γράψετε ένα πρόγραμμα, στη main( ) του οποίου να διαβάσετε τρείς float μεταβλητές, τις bs, yps και emb. Η bs αντιπροσωπεύει την βάση ενός τριγώνου, η yps αντιπροσωπεύει το ύψος του τριγώνου και η emb το εμβαδόν κάποιας έκτασης. Να γράψετε και να καλέσετε στη συνέχεια μια συνάρτηση, την area, η οποία θα κάνει τα εξής:

- Θα δέχεται ως παραμέτρους τους τρεις float. Το εάν χρειάζεται να «περάσετε» ως παραμέτρους μεταβλητές ή δείκτες ή κάποιες μεταβλητές και κάποιους δείκτες, θα το κρίνετε εσείς.

- Η συνάρτηση υπολογίζει το εμβαδόν του τριγώνου που έχει βάση όση το bs και ύψος όσο το yps. (Εμβαδόν τριγώνου = Βάση * ύψος / 2).
- Όσο το εμβαδόν που υπολογίζεται είναι μεγαλύτερο από την τιμή emb, αλλά και η τιμή του bs παραμένει θετική, μειώνεται η τιμή του bs κατά 0.1 και υπολογίζεται ξανά το εμβαδόν.
- Όταν τελειώσουν οι επαναλήψεις αυτές, ελέγχουμε εάν το εμβαδόν που υπολογίστηκε έχει τελικά γίνει μικρότερο από την τιμή του emb. Αν ναι, η συνάρτηση θα επιστρέψει στη main( ) τιμή ίση με 1. Αν όχι, η συνάρτηση θα επιστρέψει τιμή 0.
- Εκτός από την τιμή επιστροφής, η συνάρτηση ενημερώνει τη main( ) μέσω των παραμέτρων της για τα εξής:
    - Αλλάζει την τιμή που αντιστοιχεί στο bs στη main( ) και την κάνει ίση με την τιμή του bs, όπως διαμορφώθηκε μετά τις συνεχείς μειώσεις κατά 0.1 που έγιναν.
    - Αλλάζει την τιμή που αντιστοιχεί στην emb στη main( ) και την κάνει ίση με την τιμή του εμβαδού που υπολογίστηκε.

Στη main( ), μετά την κλήση της συνάρτησης, να γράψετε στην οθόνη την τιμή επιστροφής της, καθώς και τις τιμές των bs, yps και emb.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 9) :

- Η άσκηση είναι παραλλαγή της άσκησης 7 του εργαστηρίου 6.
- Η **δημιουργία ενός τυχαίου θετικού ακεραίου** γίνεται με την χρήση της συνάρτησης **rand( )**. Πριν την κλήση της πρέπει να έχει προηγηθεί το:

  srand (time (NULL));
- Οι συναρτήσεις rand( ) και srand( ) χρειάζονται τα αρχεία επικεφαλλίδας (header) **time.h** και **stdlib.h**.

9. Να γραφεί ένα πρόγραμμα, το οποίο θα δημιουργεί ένα τυχαίο ακέραιο αριθμό (π.χ. τον num) μεταξύ -100 και 100.

Αφού παραχθεί ο τυχαίος ακέραιος, να καλείται μια συνάρτηση, η guess( ), η οποία θα διαβάζει συνεχώς ακέραιους από το πληκτρολόγιο μέχρι να βρεθεί ο num. Όσο ο αριθμός που διαβάζεται (π.χ. ο gs) δεν είναι ίσος με num, η συνάρτηση να κάνει τα εξής:

- Εάν ο gs είναι μεγαλύτερος από τον num, γράφει στην οθόνη DWSE MIKROTERO και αυξάνεται ένας μετρητής, ο meg.
- Εάν ο gs είναι μικρότερος από τον num, γράφει στην οθόνη DWSE MEGALYTERO και αυξάνεται ένας μετρητής, ο mik.

Αφού βρεθεί ο τυχαίος ακέραιος num, η συνάρτηση να τερματίζεται και η main( ) να γραφεί στην οθόνη τις τιμές των mik και meg, καθώς και τον num.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 10) :

- Η άσκηση είναι παραλλαγή της άσκησης 10 του εργαστηρίου 5 και της άσκησης 12 του εργαστηρίου 8.
- Οι **αθροιστές** σε ένα πρόγραμμα είναι μεταβλητές, στις οποίες πρέπει να αποδίδουμε **αρχική τιμή μηδέν**.
- Η τυπική απόκλιση ενός πλήθους αριθμών είναι η τετραγωνική ρίζα της εξής διαφοράς: μέση τιμή των τετραγώνων των αριθμών μείον το τετράγωνο της μέσης τιμής. (Για να ελέγξετε το πρόγραμμά σας, οι αριθμοί: 1.1 2 3.4 6 4.3 2.9 1 2.1 9 6.5 δίνουν τυπική απόκλιση 2.482761).

10. Να γραφεί ένα πρόγραμμα, στη main( ) του οποίου θα διαβάζεται ένας float από το πληκτρολόγιο, έστω ο fp.

Στη συνέχεια θα καλείται μια συνάρτηση, η stddev( ), στην οποία θα διαβάζονται συνεχώς float αριθμοί από το πληκτρολόγιο, όσο οι αριθμοί αυτοί είναι μικρότεροι από τον fp. Η συνάρτηση να υπολογίζει την τυπική απόκλιση (standard deviation) των float αριθμών που διαβάστηκαν σε αυτήν και θα επιστρέφει το αποτέλεσμα στην main( ).

Η main( ) να γράφει στην οθόνη την τυπική απόκλιση, καθώς και το πλήθος των αριθμών που διαβάστηκαν.

ΕΡΓΑΣΤΗΡΙΟ 10

Α'. ΠΕΡΙΕΧΟΜΕΝΑ ΜΑΘΗΜΑΤΟΣ

Στο εργαστήριο 10 καλύπτονται τα παρακάτω θέματα:

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

Β'. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

Εξωτερικές μεταβλητές.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 1) :

- 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.
- Εξωτερικές μεταβλητές στις οποίες **δεν έχουμε αποδώσει αρχική τιμή**, παίρνουν τιμή **μηδέν**.
- Τιμή **μηδέν για ένα χαρακτήρα** είναι ο μηδενικός χαρακτήρας του κώδικα ASCII, δηλαδή **ο χαρακτήρας \x0.**

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

```
#include<stdio.h>
int a = 3;
float x;

void test ( );
void test2 (int);
void test3 ( );
main( )
{
        printf ("a=%d, x=%f \n", a, x);
        test ( );
        printf ("a=%d \n", a);
        test2 (a);
        printf ("a=%d \n", a);
        test3 ( );
        printf ("a=%d, x=%f \n", ++a, x); }
```

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

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

void test3 ( ) {
        x=3.14;
        printf ("a=%d, x=%f \n", ++a, x);
        a = 5; }
```

Στατικές μεταβλητές.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 2) :

- Οι **στατικές μεταβλητές** είναι γνωστές **μόνο στη συνάρτηση στην οποία έχουν δηλωθεί**.
- Μετά το τέλος της συνάρτησης οι στατικές μεταβλητές **δεν καταστρέφονται,** αλλά **η συνάρτηση τις «ξαναβρίσκει» στην επόμενη κλήση της**.
- Η συνάρτηση **«περνάει» από την δήλωση των στατικών μεταβλητών μόνο την πρώτη φορά** που καλείται.
- Στατικές μεταβλητές στις οποίες **δεν έχουμε αποδώσει αρχική τιμή**, παίρνουν τιμή **μηδέν**.
- Τιμή **μηδέν για ένα χαρακτήρα** είναι ο μηδενικός χαρακτήρας του κώδικα ASCII, δηλαδή **ο χαρακτήρας \x0.**

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

```
#include<stdio.h>
void test ( );
void test2 ( );

main( )
{
    test( );
    test( );
    test( );
    test2( );
    test2( );
    test2( );
}
```

```
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); }
```

Πίνακες και δείκτες.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 3) :

- Θυμηθείτε ότι ισχύουν τα εξής:

  pin == &pin[0]

  pin+k == &pin[k]

  ***pin == pin[0]**

  ***(pin+k) == pin[k]**

- Η **τιμή ενός δείκτη** ισούται με τη διεύθυνση μνήμης του byte στο οποίο είναι τοποθετημένος ο δείκτης και εμφανίζεται στην οθόνη με την χρήση του **προσδιοριστή %p**.

3. Χρησιμοποιώντας συμβολισμό δεικτών να δώσετε τιμές από το πληκτρολόγιο σε ένα πίνακα ακεραίων Ν θέσεων. Στη συνέχεια να εμφανίσετε στην οθόνη τις διευθύνσεις και τις τιμές των στοιχείων του πίνακα

Πίνακες ως ορίσματα συνάρτησης

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 4, 5, 6) :

- 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 μιας συνάρτησης (της parad) **οι παρακάτω συμβολισμοί είναι ισοδύναμοι**:

  void parad (int *pin)

  void parad (int pin[ ])

  Σε κάθε περίπτωση, **το pin είναι δείκτης σε ακέραιο.**

Επεξηγήσεις – υπενθυμίσεις (συνέχεια) :

- Μια συνάρτηση, εκτός από int, float, char κλπ, μπορεί να έχει **τιμή επιστροφής δείκτη.** Π.χ. η παρακάτω συνάρτηση test( ) δέχεται ως όρισμα δείκτη σε ακέραιο και επιστρέφει δείκτη σε float:

  float *test (int *ptr)

- Το **NULL είναι τιμή δείκτη.**

4. Στη main( ) ενός προγράμματος έχετε δηλώσει δύο πίνακες ακεραίων Ν θέσεων, τους pin και mat (το Ν είναι μια σταθερά). Οι πίνακες να γεμίσουν από το πληκτρολόγιο. Να γραφεί μια συνάρτηση, η οποία θα κάνει τα εξής:

- Θα αντιγράφει 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. Η συνάρτηση 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 ίσο με NULL.

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

Δεν επιτρέπεται η χρήση εξωτερικών μεταβλητών.

5. Στη main( ) ενός προγράμματος:

- Δηλώνονται δύο πίνακες χαρακτήρων, οι symb και syn.
- Διαβάζονται δυο συμβολοσειρές από το πληκτρολόγιο και τοποθετούνται στους πίνακες symb και syn.
- Καλείται στη συνέχεια μια συνάρτηση, η strsrch( ), η οποία δέχεται μεταξύ των άλλων παραμέτρων της και ένα ακέραιο, τον k. Η συνάρτηση κάνει τα εξής:
  - Τοποθετεί μέσα στον πίνακα όπου έχει αποθηκευτεί η συμβολοσειρά symb και από την θέση k και μετά του πίνακα, όλα τα περιεχόμενα της συμβολοσειράς syn, χωρίς τον χαρακτήρα \x0. Αυτό να γίνει μόνο εάν το πλήθος των στοιχείων που θα αντιγραφούν χωράει από την θέση k του πίνακα symb και κάτω.
  - Εάν γίνει η αντιγραφή, η συνάρτηση να επιστρέφει ένα δείκτη στη συμβολοσειρά symb, k θέσεις από την αρχή της, αλλιώς να επιστρέφει ένα δείκτη ίσο με NULL.

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, εφ' όσον ο δείκτης αυτός δεν είναι NULL. Εάν είναι NULL να γράφεται στην οθόνη η λέξη NULL.

Δεν επιτρέπεται η χρήση εξωτερικών μεταβλητών.

Γ'. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

6. Στη main( ) ενός προγράμματος έχουν δηλωθεί δυο πίνακες ακεραίων Ν 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 (η σύγκριση γίνεται στοιχείο προς στοιχείο, δηλαδή το $1^{ο}$ του pin με το $1^{ο}$ του mat, το $2^{ο}$ με το $2^{ο}$ κλπ).
- Εάν τα στοιχεία των δυο πινάκων είναι όλα ίδια (ένα προς ένα, όπως και παραπάνω), 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 την τιμή και τα περιεχόμενα του δείκτη, τον οποίο επέστρεψε η comp( ).

Δεν επιτρέπεται η χρήση εξωτερικών μεταβλητών.

7. Να γραφεί ένα πρόγραμμα, στο οποίο να διαβάζετε μια συμβολοσειρά (π.χ. την wrd) από το πληκτρολόγιο. Στη συνέχεια να καλείται μια συνάρτηση, η reverse( ), η οποία θα αντιστρέφει την συμβολοσειρά wrd (δηλαδή ο πρώτος χαρακτήρας της θα γίνεται τελευταίος, ο δεύτερος προτελευταίος κλπ).

Η συνάρτηση να έχει τιμή επιστροφής ένα δείκτη στην θέση όπου βρίσκεται ο χαρακτήρας \x0 της αρχικής συμβολοσειράς.

Η main( ) να γράφει στην οθόνη την συμβολοσειρά μετά την αντιστροφή της, καθώς και την τιμή του δείκτη, τον οποίο επέστρεψε η συνάρτηση.

ΕΡΓΑΣΤΗΡΙΟ 11

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.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 1) :

- Οι συμβολοσειρές αποθηκεύονται σε **πίνακες τύπου char**.
- Η **gets διαβάζει συμβολοσειρά** από το πληκτρολόγιο. Δέχεται **όρισμα δείκτη σε χαρακτήρα** και τοποθετεί την συμβολοσειρά από εκεί που δείχνει ο δείκτης και κάτω. **Στο τέλος τοποθετεί τον χαρακτήρα '\x0'**. **Διαβάζει και κενά** (κάτι που δεν κάνει η scanf).
- Η **strcmp συγκρίνει αλφαβητικά δυο συμβολοσειρές**. Δέχεται ο**ρίσματα δύο δείκτες σε χαρακτήρα.** Επιστρέφει **μηδέν** εάν **οι συμβολοσειρές είναι ίδιες**, **αρνητικό** αριθμό εάν **η πρώτη προηγείται αλφαβητικά της δεύτερης** και **θετικό** αριθμό εάν **η δεύτερη προηγείται αλφαβητικά της πρώτης**.

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

```
#include <stdio.h>
#include <conio.h>
#include <string.h>          /* Για 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"); }
```

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 2) :

- Οι συμβολοσειρές αποθηκεύονται σε **πίνακες τύπου char**.
- Η **gets διαβάζει συμβολοσειρά** από το πληκτρολόγιο. Δέχεται **όρισμα δείκτη σε χαρακτήρα** και τοποθετεί την συμβολοσειρά από εκεί που δείχνει ο δείκτης και κάτω. **Στο τέλος τοποθετεί τον χαρακτήρα '\x0'**. **Διαβάζει και κενά** (κάτι που δεν κάνει η scanf).
- Η **strcpy** δέχεται ορίσματα **δύο δείκτες σε χαρακτήρα. Αντιγράφει** την συμβολοσειρά που ξεκινάει **από το δεύτερο όρισμα** εκεί που δείχνει **το πρώτο όρισμα**. **Επιστρέφει δείκτη ίσο με το πρώτο της όρισμα**.

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

```
#include <stdio.h>
#include <conio.h>
#include <string.h>              /* Για συμβολοσειρές */

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

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

Επεξηγήσεις – υπενθυμίσεις (άσκηση 3) :

- Οι συμβολοσειρές αποθηκεύονται σε **πίνακες τύπου char**.
- Η **gets διαβάζει συμβολοσειρά** από το πληκτρολόγιο. Δέχεται **όρισμα δείκτη σε χαρακτήρα** και τοποθετεί την συμβολοσειρά από εκεί που δείχνει ο δείκτης και κάτω. **Στο τέλος τοποθετεί τον χαρακτήρα '\x0'**. **Διαβάζει και κενά** (κάτι που δεν κάνει η scanf).
- Η **strcat** δέχεται ορίσματα **δύο δείκτες σε χαρακτήρα. Επικολλά** την συμβολοσειρά που ξεκινάει **από το δεύτερο όρισμα στο τέλος** της συμβολοσειράς που ξεκινάει **από το πρώτο όρισμα** (αφαιρεί το \x0 από την πρώτη συμβολοσειρά και μεταφέρει από εκεί και κάτω την δεύτερη όπως είναι). **Επιστρέφει δείκτη ίσο με το πρώτο της όρισμα**.

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

```
#include <stdio.h>
#include <conio.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    και   p <--- s; */
    printf ("s = %s \n", s );
    printf ("%c\n", *p); }
```

Επεξηγήσεις – υπενθυμίσεις (άσκηση 4) :

- Οι συμβολοσειρές αποθηκεύονται σε **πίνακες τύπου char**.
- Η **strchr** δέχεται ορίσματα **ένα δείκτη σε χαρακτήρα** και **ένα χαρακτήρα. Αναζητά** στην συμβολοσειρά που ξεκινάει **από το πρώτο όρισμα** (τον δείκτη), **το δεύτερο όρισμα** (τον χαρακτήρα). **Επιστρέφει ένα δείκτη** στην θέση όπου βρέθηκε για **πρώτη φορά** το δεύτερο όρισμα**. Αν δεν υπάρχει** το δεύτερο όρισμα στην συμβολοσειρά, επιστρέφει **NULL**.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 4 συνέχεια) :

- Η **strcpy** δέχεται ορίσματα **δύο δείκτες σε χαρακτήρα. Αντιγράφει** την συμβολοσειρά που ξεκινάει **από το δεύτερο όρισμα** εκεί που δείχνει **το πρώτο όρισμα**. **Επιστρέφει δείκτη ίσο με το πρώτο της όρισμα**.

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

```
#include <stdio.h>
#include <conio.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 ); }
```

Επεξηγήσεις – υπενθυμίσεις (άσκηση 5) :

- Οι συμβολοσειρές αποθηκεύονται σε **πίνακες τύπου char**.
- Η **strstr** δέχεται ορίσματα **δύο δείκτες σε χαρακτήρα. Αναζητά** στην συμβολοσειρά που ξεκινάει από το **πρώτο όρισμα** την συμβολοσειρά που ξεκινάει από **το δεύτερο όρισμα**. **Επιστρέφει ένα δείκτη** στην θέση όπου βρέθηκε για **πρώτη φορά** το δεύτερο όρισμα**. Αν δεν υπάρχει** το δεύτερο όρισμα στην συμβολοσειρά, επιστρέφει **NULL**.
- Η **strcpy** δέχεται ορίσματα **δύο δείκτες σε χαρακτήρα. Αντιγράφει** την συμβολοσειρά που ξεκινάει **από το δεύτερο όρισμα** εκεί που δείχνει **το πρώτο όρισμα**. **Επιστρέφει δείκτη ίσο με το πρώτο της όρισμα**.

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

```
#include <stdio.h>
#include <conio.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 );
}
```

Επεξηγήσεις – υπενθυμίσεις (άσκηση 6) :

- Οι συμβολοσειρές αποθηκεύονται σε **πίνακες τύπου char**.
- Η **gets διαβάζει συμβολοσειρά** από το πληκτρολόγιο. Δέχεται **όρισμα δείκτη σε χαρακτήρα** και τοποθετεί την συμβολοσειρά από εκεί που δείχνει ο δείκτης και κάτω. **Στο τέλος τοποθετεί τον χαρακτήρα '\x0'**. **Διαβάζει και κενά** (κάτι που δεν κάνει η scanf).
- Τα **περιεχόμενα ενός πίνακα** τα διαβάζουμε, τα γράφουμε, τα αντιγράφουμε κλπ με την **χρήση επαναληπτικής εντολής.**

6. 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 (εφ' όσον χωράνε), χωρίς τον χαρακτήρα \x0.
  - 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 (άσκηση 7) :

- Οι συμβολοσειρές αποθηκεύονται σε **πίνακες τύπου char**.
- Η **puts** δέχεται **όρισμα δείκτη σε χαρακτήρα. Γράφει στην οθόνη** την συμβολοσειρά από εκεί που δείχνει ο δείκτης και κάτω και αλλάζει γραμμή.
- Η **strcpy** δέχεται ορίσματα **δύο δείκτες σε χαρακτήρα. Αντιγράφει** την συμβολοσειρά που ξεκινάει **από το δεύτερο όρισμα** εκεί που δείχνει **το πρώτο όρισμα**. **Επιστρέφει δείκτη ίσο με το πρώτο της όρισμα**.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 7 συνέχεια) :

- Η **strcat** δέχεται ορίσματα **δύο δείκτες σε χαρακτήρα. Επικολλά** την συμβολοσειρά που ξεκινάει **από το δεύτερο όρισμα στο τέλος** της συμβολοσειράς που ξεκινάει **από το πρώτο όρισμα** (αφαιρεί το \x0 από την πρώτη συμβολοσειρά και μεταφέρει από εκεί και κάτω την δεύτερη όπως είναι). **Επιστρέφει δείκτη ίσο με το πρώτο της όρισμα**.
- Η **strcmp συγκρίνει αλφαβητικά δυο συμβολοσειρές**. Δέχεται ο**ρίσματα δύο δείκτες σε χαρακτήρα.** Επιστρέφει **μηδέν** εάν **οι συμβολοσειρές είναι ίδιες**, **αρνητικό** αριθμό εάν **η πρώτη προηγείται αλφαβητικά της δεύτερης** και **θετικό** αριθμό εάν **η δεύτερη προηγείται αλφαβητικά της πρώτης**.

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

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

char words[10];

main( ) {
    char wra[10] = "TI WRA";
    char *dkt, *ptr = "ENTEKA";
    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)); }
```

Γ'. ΕΝΔΕΙΚΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

Επεξηγήσεις – υπενθυμίσεις (άσκηση 8) :

- Οι συμβολοσειρές αποθηκεύονται σε **πίνακες τύπου char**.
- Η **strlen** δέχεται όρισμα ένα **δείκτη σε χαρακτήρα. Μετρά** το **πλήθος των χαρακτήρων** της συμβολοσειράς που ξεκινάει από το όρισμά της μέχρι το \x0 **(δεν μετράει το \x0)** και **επιστρέφει** το **πλήθος των χαρακτήρων** που μέτρησε**.**

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.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 9) :

- Οι συμβολοσειρές αποθηκεύονται σε **πίνακες τύπου char**.
- Η **strcpy** δέχεται ορίσματα **δύο δείκτες σε χαρακτήρα. Αντιγράφει** την συμβολοσειρά που ξεκινάει **από το δεύτερο όρισμα** εκεί που δείχνει **το πρώτο όρισμα**. **Επιστρέφει δείκτη ίσο με το πρώτο της όρισμα**.

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.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 10) :

- Οι συμβολοσειρές αποθηκεύονται σε **πίνακες τύπου char**.
- Η **strcat** δέχεται ορίσματα **δύο δείκτες σε χαρακτήρα. Επικολλά** την συμβολοσειρά που ξεκινάει **από το δεύτερο όρισμα στο τέλος** της συμβολοσειράς που ξεκινάει **από το πρώτο όρισμα** (αφαιρεί το \x0 από την πρώτη συμβολοσειρά και μεταφέρει από εκεί και κάτω την δεύτερη όπως είναι). **Επιστρέφει δείκτη ίσο με το πρώτο της όρισμα**.

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.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 11) :

- Οι συμβολοσειρές αποθηκεύονται σε **πίνακες τύπου char**.
- Η **strcmp συγκρίνει αλφαβητικά δυο συμβολοσειρές**. Δέχεται ο**ρίσματα δύο δείκτες σε χαρακτήρα.** Επιστρέφει **μηδέν** εάν **οι συμβολοσειρές είναι ίδιες**, **αρνητικό** αριθμό εάν **η πρώτη προηγείται αλφαβητικά της δεύτερης** και **θετικό** αριθμό εάν **η δεύτερη προηγείται αλφαβητικά της πρώτης**.

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.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 12) :

- Οι συμβολοσειρές αποθηκεύονται σε **πίνακες τύπου char**.
- Η **strchr** δέχεται ορίσματα **ένα δείκτη σε χαρακτήρα** και **ένα χαρακτήρα. Αναζητά** στην συμβολοσειρά που ξεκινάει **από το πρώτο όρισμα** (τον δείκτη), **το δεύτερο όρισμα** (τον χαρακτήρα). **Επιστρέφει ένα δείκτη** στην θέση όπου βρέθηκε για **πρώτη φορά** το δεύτερο όρισμα**. Αν δεν υπάρχει** το δεύτερο όρισμα στην συμβολοσειρά, επιστρέφει **NULL**.

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.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 13) :

- Οι συμβολοσειρές αποθηκεύονται σε **πίνακες τύπου char**.
- Η **strstr** δέχεται ορίσματα **δύο δείκτες σε χαρακτήρα. Αναζητά** στην συμβολοσειρά που ξεκινάει από το **πρώτο όρισμα** την συμβολοσειρά που ξεκινάει από **το δεύτερο όρισμα**. **Επιστρέφει ένα δείκτη** στην θέση όπου βρέθηκε για **πρώτη φορά** το δεύτερο όρισμα**. Αν δεν υπάρχει** το δεύτερο όρισμα στην συμβολοσειρά, επιστρέφει **NULL**.

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:

- Μαθηματικές συναρτήσεις (π.χ. abs, fabs, pow, sqrt, sin, cos).
- Συναρτήσεις atoi, atof, atol.
- Δομές (structures): Περιγραφή, πεδία δομής, δηλώσεις και δεδομένα στις δομές.
- Φωλιασμένες δομές.

Β΄. ΑΣΚΗΣΕΙΣ ΓΙΑ ΕΚΤΕΛΕΣΗ – ΕΠΙΔΕΙΞΗ / ΕΠΕΞΗΓΗΣΕΙΣ

Συναρτήσεις abs, fabs, pow, sqrt, sin, cos

Επεξηγήσεις – υπενθυμίσεις (άσκηση 1) :

Η συνάρτηση **pow** δέχεται ως **ορίσματα δύο double** και **επιστρέφει double** ίσο με το **πρώτο όρισμα υψωμένο στο δεύτερο όρισμα.**

1. Να γραφεί ένα πρόγραμμα, το οποίο να υπολογίζει και να εμφανίζει στην οθόνη τις δέκα πρώτες δυνάμεις του 10.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 2) :

- Η συνάρτηση **sin** δέχεται ως **όρισμα ένα double** και **επιστρέφει double** ίσο με το **ημίτονο** του ορίσματός της. **Η γωνία** (το όρισμα) πρέπει να έχει δοθεί **σε ακτίνια** (rad).
- Η συνάρτηση **cos** δέχεται ως **όρισμα ένα double** και **επιστρέφει double** ίσο με το **συνημίτονο** του ορίσματός της. **Η γωνία** (το όρισμα) πρέπει να έχει δοθεί **σε ακτίνια** (rad).

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.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 3) :

- Η συνάρτηση **abs** δέχεται ως **όρισμα ένα ακέραιο** και **επιστρέφει ακέραιο** ίσο με την **απόλυτη τιμή** του ορίσματός της.
- Η συνάρτηση **fabs** δέχεται ως **όρισμα ένα double** και **επιστρέφει double** ίσο με την **απόλυτη τιμή** του ορίσματός της.

3. Να γραφεί ένα πρόγραμμα, το οποίο να διαβάζει Ν φορές από το πληκτρολόγιο ανά ένα ακέραιο και ένα float κάθε φορά και να εμφανίζει στην οθόνη τις απόλυτες τιμές των αριθμών που διαβάστηκαν.

Συναρτήσεις atoi, atof, atol.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 4) :

- Η συνάρτηση **atoi** δέχεται ως όρισμα ένα **δείκτη σε χαρακτήρα** και **επιστρέφει ακέραιο. Μετατρέπει την συμβολοσειρά** που βρίσκεται εκεί που δείχνει το όρισμά της σε **ακέραιο**.
- Η συνάρτηση **atof** δέχεται ως όρισμα ένα **δείκτη σε χαρακτήρα** και **επιστρέφει float. Μετατρέπει την συμβολοσειρά** που βρίσκεται εκεί που δείχνει το όρισμά της σε **float**.
- Η συνάρτηση **atol** δέχεται ως όρισμα ένα **δείκτη σε χαρακτήρα** και **επιστρέφει long. Μετατρέπει την συμβολοσειρά** που βρίσκεται εκεί που δείχνει το όρισμά της σε **long**.

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.

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

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 5, 6, 7, 8, 9) :

- Η **περιγραφή της δομής** προϋπάρχει της δήλωσης για να «διδάξει» στον compiler πώς είναι το νέο είδος μεταβλητών που δημιουργούμε.
- Η **περιγραφή της δομής** βρίσκεται **μέσα σε άγκιστρα**, **μετά τα οποία** υπάρχει το **;**

Επεξηγήσεις – υπενθυμίσεις (συνέχεια, ασκήσεις 5, 6, 7, 8, 9) :

- Στο κάθε πεδίο μιας δομής αναφερόμαστε ως εξής:

  Όνομα δομής . Όνομα πεδίου

- Η εμφάνιση **μόνο του είδους** της δομής (της λέξης δηλαδή που ακολουθεί την λέξη struct, όπως η λέξη complex στο παράδειγμα της άσκησης 1) αποτελεί **συντακτικό λάθος**.
- Η εμφάνιση **μόνο του ονόματος κάποιου πεδίου της δομής** (των λέξεων δηλαδή που εμφανίζονται στην περιγραφή μιας δομής, όπως οι λέξεις re και im στο παράδειγμα της άσκησης 1) αποτελεί επίσης **συντακτικό λάθος.**

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

```
#include <stdio.h>

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

main( )
{
    struct complex z1, z2;             /* Δήλωσεις μεταβλητών *
    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 );
}
```

Επεξηγήσεις – υπενθυμίσεις (συνέχεια, άσκηση 6) :

Έστω δύο μιγαδικοί αριθμοί, οι z1=a+jb και z2=c+jd. Τότε:

- w1 = z1+z2 = (a+c) + j (b+d)
- w2 = z1-z2 = (a-c) + j (b-d)
- w3 = z1*z2 = (ac-bd) + j (bc+ad)
- w4 = z1/z2 = (ac+bd)/(c^2+d^2) + j (bc-ad)/(c^2+d^2)

6. Σε συνέχεια της άσκησης 5, 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 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 εναλλάξετε τα περιεχόμενα των str1 και str2 και να γράψετε τα πεδία τους στην οθόνη.

Φωλιασμένες δομές.

Επεξηγήσεις – υπενθυμίσεις (συνέχεια, ασκήσεις 8, 9) :

- **Φωλιασμένες** λέγονται οι δομές των οποίων **κάποιο ή κάποια πεδία** είναι **δομή**, η οποία **έχει ήδη περιγραφεί** προηγουμένως.
- Η **προσπέλαση των πεδίων στις φωλιασμένες δομές** γίνεται με την **πολλαπλή χρήση της τελείας (.)**.

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 τρείς 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>

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 );
}
```

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.
- 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 melos 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 melos, της δομής της memb, εφ’ όσον υπάρχει αρκετός χώρος.
- Αντιγράφουμε το πεδίο item της δομής memb, στο πεδίο melos της ίδιας δομής.

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.

Επεξηγήσεις – υπενθυμίσεις (ασκήσεις 1, 3) :

- Σε ένα πίνακα δομών, **κάθε στοιχείο του πίνακα είναι μια δομή**. Στην άσκηση 1, το pin είναι ένας πίνακας Ν δομών του είδους student. Άρα, το pin[0], το pin[1] κλπ είναι δομές του είδους student.
- **Ο χαρακτήρας** για παράδειγμα **της 3ης θέσης του πίνακα name του τέταρτου στοιχείου του πίνακα pin** λέγεται **pin[3] . name[2]**
- Το **fflush(stdin) αδειάζει τον buffer εισόδου** κι έτσι η gets δεν επηρρεάζεται από το <enter> της scanf που έχει προηγηθεί.

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

```
#include<stdio.h>

#define N 10

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

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

    for (k=0; k<N; k++) {                                  /* Διάβασμα */
        printf ("Dwste AM toy spoudasth %d -> ", k+1);
        scanf ("%d", &pin[k].am );                  /* Προσέξτε το &*/
        fflush (stdin);
        printf ("Dwste onoma -> ");
        gets (pin[k].name); }
```

```
for (k=0; k<N; k++)                                   /* Εκτύπωση */
    printf ("Spoudasths %3d\n", k+1);
    printf ("AM=%4d\n", pin[k].am);
    printf ("Arxiko gramma onomatos=%3c\n", pin[k].name[0]);
    printf ("Onoma=%s \n", pin[k].name ); } }
```

Δομές ως παράμετροι και ως τιμή επιστροφής συναρτήσεων.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 2) :

- Μια **συνάρτηση** μπορεί να **δέχεται ως ορίσματα δομές**, όπως και οποιονδήποτε άλλο τύπο δεδομένων. Π.χ.: Η επικεφαλλίδα μιας συνάρτησης που είναι void και δέχεται ως παραμέτρους δύο δομές, τις s1 και s2 του τύπου stype της άσκησης 2, θα είναι:

  void example (struct stype s1, struct stype s2)

- Η **κλήση** της συνάρτησης θα είναι:

  example (s1, s2);

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.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 3) :

- Μια **συνάρτηση** μπορεί να **δέχεται ως όρισμα πίνακα δομών**, όπως και οποιονδήποτε άλλο πίνακα. Π.χ.: Η επικεφαλλίδα μιας συνάρτησης που είναι void και δέχεται ως παράμετρο ένα πίνακα δομών του τύπου funds της άσκησης 3, τον pin, θα είναι:

  void paradeigma (struct funds *pin) ή

  void paradeigma (struct funds pin[ ])

- Η **κλήση** της συνάρτησης θα είναι:

  paradeigma (pin);

3. Σε ένα πρόγραμμα έχετε τον εξής τύπο δομών:

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

Να δηλώσετε στη main( ) ένα πίνακα Ν 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.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 4) :

- Μια **συνάρτηση** μπορεί να έχει **τιμή επιστροφής δομή**, όπως και οποιοδήποτε άλλο είδος δεδομένων. Π.χ.: Η επικεφαλλίδα μιας συνάρτησης που είναι void και δέχεται ως μια δομή του τύπου funds της άσκησης 4, την str, θα είναι:

  struct funds reading (struct funds str)

- Η **κλήση** της συνάρτησης θα είναι:

  x = reading (str);

  όπου το x είναι δομή της main( ).

4. Σε ένα πρόγραμμα έχετε τον εξής τύπο δομών:

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

Να δηλώσετε στη main( ) ένα πίνακα Ν δομών του πιο πάνω τύπου, τον persons. Για κάθε στοιχείο του πίνακα (άρα Ν φορές), να καλείτε μια συνάρτηση, έστω την reading( ), η οποία θα διαβάζει μια δομή του είδους funds, θα την επιστρέφει στην main( ), από όπου και θα καταχωρείται στον πίνακα.

Αφού γεμίσει ο πίνακας persons, να καλείται μια άλλη συνάρτηση, η display( ), η οποία θα γράφει τα περιεχόμενά του persons στην οθόνη.

Να μη χρησιμοποιείτε εξωτερικές μεταβλητές.

Δείκτες σε δομές.

Επεξηγήσεις – υπενθυμίσεις (άσκηση 5) :

- Στα **πεδία μιας δομής** στην οποία **δείχνει ένας δείκτης** αναφερόμαστε ως εξής:

  Όνομα δείκτη -> Όνομα πεδίου

- Σε μια παράσταση όπου υπάρχουν τελεστές **.** και **-> όλη η παράσταση αριστερά από την . πρέπει να είναι δομή**, ενώ **όλη η παράσταση αριστερά από το -> πρέπει να είναι δείκτης σε δομή**. Σε κάθε άλλη περίπτωση υπάρχει συντακτικό λάθος

5. 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 της δομής στην οποία δείχνει ο ptr στον αντίστοιχο πίνακα του πεδίου item της 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 στην οποία δείχνει ο ptr.
- Αντιγράφουμε την δομή person, το item πεδό 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.