

Κων/νου Στυλιάδη

Έτοιμες
Ασκήσεις σε
C

Φλώρινα, Οκτώβριος 1997

```
/* Πρόγραμμα c01.c - το πρώτο πρόγραμμα στη c */
#include <stdio.h>
main()
{
    int i;

    clrscr();
    printf("\nΕίναι το πρώτο μου πρόγραμμα στη c");
    i = 156;
    printf("\nΟ αριθμός μητρώου μου είναι : %d", i);
    scanf("%d", &i);    /* για να σταματήσει το κύλισμα της οθόνης */
} /* end of main */

/* Πρόγραμμα c02.c - αυτό το πρόγραμμα διαβάζει όνομα, ηλικία και μισθό */
#include <stdio.h>
main()
{
    int age;
    long misthos;
    char onoma[30];

    clrscr();
    printf("\nΠοιο είναι το όνομά σου ; ");
    scanf("%s", onoma);    /* τα ονόματα τα διαβάζουμε χωρίς το & */
    printf("\nΠοιος είναι ο μισθός σου ; ");
    scanf("%ld", &misthos);
    printf("\nΠοια είναι η ηλικία σου ; ");
    scanf("%d", &age);
    printf("\n");
    printf("\nΓεια σου.Λέγεσαι %s, είσαι %d χρονών και έχεις μισθό %ld\n",
           onoma, age, misthos);
    scanf("%d", &age);
} /* end of main */

/* Πρόγραμμα c03.c - εκμάθηση της συνάρτησης sizeof */
#include <stdio.h>
main()
{
    char i;

    clrscr();
    printf("\nμήκος ακεραίου = %d bytes", sizeof(int));
    printf("\nμήκος μεγάλου ακεραίου = %d bytes", sizeof(long));
    printf("\nμήκος πραγματικού = %d bytes", sizeof(double));
    printf("\nμήκος μεγάλου πραγματικού = %d bytes", \
           sizeof(long double));
    scanf("%c", &i);
} /* end of main */
```

/ Πρόγραμμα c04.c - ανάγνωση ενός ακεραίου και ενός χαρακτήρα */*

```
#include <stdio.h>
main()
{
    int i;
    char ch;

    clrscr();
    printf("\nΔώσε έναν ακέραιο αριθμό : ");
    scanf("%d", &i);
    printf("\nΔώσε έναν χαρακτήρα : ");
    scanf("%c", &ch);
    printf("\nΟ ακέραιος που έδωσες είναι : %d και : %c ", i, i);
    scanf("%d", &i);
} /* end of main */
```

/ Πρόγραμμα c05.c - ανάγνωση ακεραίου, μεγάλου ακεραίου, πραγματικού αριθμού και ονόματος (string) */*

```
#include <stdio.h>
main()
{
    int i;
    long li;
    char onoma[20];
    float s;

    clrscr();
    printf("\nΔώσε έναν ακέραιο αριθμό : ");
    scanf("%d", &i);
    printf("\nΔώσε έναν μεγάλο ακέραιο αριθμό : ");
    scanf("%ld", &li);
    printf("\nΔώσε ένα όνομα : ");
    scanf("%s", onoma);
    printf("\nΔώσε έναν πραγματικό αριθμό : ");
    scanf("%f", &s);

    printf("/%d\n", i);
    printf("/%2d\n", i);
    printf("/%10d\n", i);
    printf("/%-10d\n", i);
    printf("/%2s\n", onoma);
    printf("/%20s\n", onoma);
    printf("/%20.5s\n", onoma);
    printf("/%-23.5s\n", onoma);
    printf("/%+10.2f\n", s);
    printf("/%15.4f\n", s);
    scanf("%d", &i);
} /* end of main */
```

```
/* Πρόγραμμα c06.c - εμφάνιση ειδικών χαρακτήρων : \, %, ", beep */
#include <stdio.h>
main()
{
    clrscr();
    printf("\n\t\t έτσι εμφανίζεται η πλάγια κάθετος \\");
    printf("\n έτσι εμφανίζεται το επί τοις εκατό \%% ή και έτσι %%");
    printf("\n έτσι εμφανίζονται τα διπλά εισαγωγικά \"");
    printf("\n έτσι ακούγεται η ηχητική ειδοποίηση \a");
} /* end of main */
```

```
/* Πρόγραμμα c07.c - ορισμός σταθεράς - η δήλωση #define */
#include <stdio.h>
#define PI 3.1415926 /* το γνωστό μας π */
main()
{
    int radius = 10;
    long area, circum;

    clrscr();
    circum = 2 * PI * radius; /* περίμετρος κύκλου */
    area = PI * radius * radius; /* εμβαδόν κύκλου */
    printf("\nπερίμετρος = %ld και εμβαδόν = %ld", circum, area);
} /* end of main */
```

```
/* Πρόγραμμα c08.c - εύρεση του μεγίστου από τρεις ακεραίους */
#include <stdio.h>
main()
{
    int a, b, c, max;

    clrscr();
    printf("\nΔώσε τον πρώτο αριθμό : ");
    scanf("%d", &a);
    printf("\nΔώσε τον δεύτερο αριθμό : ");
    scanf("%d", &b);
    printf("\nΔώσε τον τρίτο αριθμό : ");
    scanf("%d", &c);

    max = a;
    if (b > max)
        max = b;
    if (c > max)
        max = c;
    printf("\nΟ μέγιστος είναι ο : %d", max);
    scanf("%d", &a);
} /* end of main */
```

/ Πρόγραμμα c09.c - εύρεση ακεραίου υπολοίπου - εκμάθηση της εντολής switch */*

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

```
main()
```

```
{
```

```
    int a, ypol;
```

```
    clrscr();
```

```
    printf("\nΔώσε έναν ακέραιο αριθμό : ");
```

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

```
    ypol = a % 3; /* εύρεση ακεραίου υπολοίπου με το 3 */
```

```
    switch (ypol)
```

```
    {
```

```
        case 0 :
```

```
            printf("\nΤο υπόλοιπο είναι μηδέν");
```

```
            break;
```

```
        case 1 :
```

```
            printf("\nΤο υπόλοιπο είναι ένα");
```

```
            break;
```

```
        case 2 :
```

```
            printf("\nΤο υπόλοιπο είναι δύο");
```

```
            break;
```

```
        default :
```

```
            printf("\nΑδύνατη περίπτωση");
```

```
    } /* τέλος της switch */
```

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

```
} /* end of main */
```

/ Πρόγραμμα c10.c - ελέγχει αν ένας ακέραιος αριθμός είναι μονός ή ζυγός */*

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

```
main()
```

```
{
```

```
    int a, ypol;
```

```
    clrscr();
```

```
    printf("\nΔώσε έναν ακέραιο αριθμό : ");
```

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

```
    ypol = a % 2;
```

```
    if (ypol == 1)
```

```
        printf("\nμονός αριθμός");
```

```
    else
```

```
        printf("\nζυγός αριθμός");
```

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

```
} /* end of main */
```

/ Πρόγραμμα c11.c - υπολογισμός επιδόματος παιδιών ενός μισθωτού σαν ποσοστό του βασικού του μισθού ως εξής :*

για ένα παιδί à 5%

για δύο παιδιά à 10%

για τρία παιδιά à 20%

*από το τέταρτο παιδί και πάνω à + 5% για κάθε παιδί */*

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

```
main()
```

```
{
```

```
    long misthos, epidoma;
```

```
    int child;
```

```
    clrscr();
```

```
    printf("\nΔώσε τον βασικό μισθό : ");
```

```
    scanf("%ld", &misthos);
```

```
    printf("\nΔώσε τον αριθμό των παιδιών : ");
```

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

```
    switch (child)
```

```
    {
```

```
        case 0 :
```

```
            epidoma = 0;
```

```
            break;
```

```
        case 1 :
```

```
            epidoma = misthos * 0.05;
```

```
            break;
```

```
        case 2 :
```

```
            epidoma = misthos * 0.10;
```

```
            break;
```

```
        case 3 :
```

```
            epidoma = misthos * 0.20;
```

```
            break;
```

```
        default :
```

```
            epidoma = misthos * 0.05 * (child-3) + misthos * 0.20;
```

```
    } /* end of switch */
```

```
    misthos = misthos + epidoma;
```

```
    printf("\nΤο επίδομα είναι : %ld", epidoma);
```

```
    printf("\nΟ τελικός μισθός είναι : %ld", misthos);
```

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

```
} /* end of main */
```

/ Πρόγραμμα c12.c - χαρακτηρισμός μαθητή ανάλογα με το βαθμό του */*

```
#include <stdio.h>
main()
{
    int bathmos;

    clrscr();
    printf("\nΔώσε τον βαθμό του μαθητή : ");
    scanf("%d", &bathmos);

    if (bathmos >= 0 && bathmos <= 20)
        switch (bathmos)
        {
            case 19 :
            case 20 :
                printf("\nείσαι άριστος");
                break;
            case 16 :
            case 17 :
            case 18 :
                printf("\nείσαι πολύ καλός");
                break;
            case 13 :
            case 14 :
            case 15 :
                printf("\nείσαι απλά καλός");
                break;
            case 10 :
            case 11 :
            case 12 :
                printf("\nθέλεις βελτίωση");
                break;
            default :
                printf("\nκινδυνεύεις να μείνεις στάσιμος");
        } /* end of switch */
    else
        printf("\nΈπρεπε να δώσεις έναν βαθμό από 0 - 20");

    scanf("%d", &bathmos);
} /* end of main */
```

/ Πρόγραμμα c13.c - εκμάθηση του τελεστή υπό συνθήκη ? : */*

```
#include <stdio.h>
main()
{
    int koutia;
    char cans[6];

    clrscr();
    printf("\nΔώσε τον αριθμό των κουτιών : ");
    scanf("%d", &koutia);
    strcpy(cans, (koutia==1 ? "κουτί" : "κουτιά"));
    printf("\nΥπάρχουν %d %s", koutia, cans);

    scanf("%d", &koutia);
} /* end of main */
```

/ Πρόγραμμα c14.c - εκμάθηση του βρόχου while - διαβάζουμε τους βαθμούς 10 μαθητών και αν ο βαθμός είναι μικρότερος του 0 ή μεγαλύτερος του 20, δεν θα λαμβάνεται υπόψη στο μέτρομα */*

```
#include <stdio.h>
main()
{
    int bathmos, i, sum;

    clrscr();
    i = 1;
    while (i <= 10)
    {
        printf("\nΔώσε τον βαθμό του %dου μαθητή : ", i);
        scanf("%d", &bathmos);

        if (bathmos < 0 || bathmos > 20)
            printf("\nΜη αποδεκτός βαθμός");
        else
        {
            i++;
            sum += bathmos; /* άθροισμα των βαθμών */
        } /* end of else */
    } /* end of while */

    printf("\nΜέσος όρος βαθμών = %.1f", sum/10.0);

    scanf("%d", &bathmos);
} /* end of main */
```


/ Πρόγραμμα c15.c - εύρεση ακέραιου ηηλίκου, ακέραιου υπολοίπου και δεκαδικού ηηλίκου δύο ακέραιων αριθμών - εκμάθηση του τελεστή εκμαγείο */*

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

```
main()
```

```
{
```

```
    int a, b, ypol, pil;
```

```
    float decad;
```

```
    clrscr();
```

```
    printf("\nΔώσε τον πρώτο ακέραιο αριθμό : ");
```

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

```
    printf("\nΔώσε τον δεύτερο ακέραιο αριθμό : ");
```

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

```
    pil = a / b;                /* ακέραιο ηηλίκο */
```

```
    ypol = a % b;              /* ακέραιο υπόλοιπο */
```

```
    decad = (float) a / (float) b; /* δεκαδικό ηηλίκο */
```

```
    printf("\nΤο ακέραιο ηηλίκο είναι : %d", pil);
```

```
    printf("\nΤο ακέραιο υπόλοιπο είναι : %d", ypol);
```

```
    printf("\nΤο δεκαδικό ηηλίκο είναι : %10.2f", decad);
```

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

```
} /* end of main */
```

/ Πρόγραμμα c16.c - εκμάθηση του τελεστή εκμαγείο */*

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

```
main()
```

```
{
```

```
    float a, b, sum_f;
```

```
    int sum1, sum2;
```

```
    clrscr();
```

```
    printf("\nΔώσε τον πρώτο αριθμό : ");
```

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

```
    printf("\nΔώσε τον δεύτερο αριθμό : ");
```

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

```
    sum1 = (int) a + (int) b;
```

```
    sum2 = (int) (a + b);
```

```
    sum_f = a + b;
```

```
    printf("\nΤο πρώτο αποτέλεσμα είναι : %d", sum1);
```

```
    printf("\nΤο δεύτερο αποτέλεσμα είναι : %d", sum2);
```

```
    printf("\nΤο τρίτο αποτέλεσμα είναι : %10.2f", sum_f);
```

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

```
} /* end of main */
```

/ Πρόγραμμα c17.c - εκμάθηση του βρόχου do while - εμφανίζει τα αγγλικά και τα ελληνικά κεφαλαία γράμματα σε μια σειρά */*

```
#include <stdio.h>
main()
{
    char ch = 'A';      /* αγγλικό A */

    clrscr();
    printf("\nΤα αγγλικά κεφαλαία γράμματα είναι : \n");
    do
    {
        printf("%c", ch);
        ch = ch + 1;
    } while (ch <= 'Z');
    printf("\n"); /* αλλάζει σειρά */

    ch = 'Α';      /* ελληνικό Α */
    printf("\nΤα ελληνικά κεφαλαία γράμματα είναι : \n");
    do
    {
        printf("%c", ch);
        ch = ch + 1;
    } while (ch <= 'Ω');
    printf("\n"); /* αλλάζει σειρά */
} /* end of main */
```

/ Πρόγραμμα c18.c - ανάγνωση ακεραίων με χρήση του βρόχου do while μέχρι που να συναντήσει κάποιον που να είναι μεγαλύτερος από 100 και μικρότερος από 200 */*

```
#include <stdio.h>
main()
{
    int i;

    clrscr();
    do
    {
        printf("\nΔώσε έναν ακέραιο : ");
        scanf("%d", &i);
    } while (i < 100 || i > 200);

    scanf("%d", &i);
} /* end of main */
```

```
/* Πρόγραμμα c19.c - εκμάθηση των συναρτήσεων getchar() και putchar() */
#include <stdio.h>
main()
{
    char ch;

    clrscr();
    do
    {
        printf("\nΔώσε έναν χαρακτήρα : ");
        ch = getchar();    /* διάβασμα χαρακτήρα */
        putchar(ch);      /* εκτύπωση χαρακτήρα */
    } while (ch != '#'); /* βγαίνουμε από τον βρόχο όταν δοθεί ο # */
    scanf("%c", %ch);
} /* end of main */
```

```
/* Πρόγραμμα c20.c - δήλωση πίνακα και εκμάθηση της εντολής for */
#include <stdio.h>
main()
{
    int a[10], i;
    long sum=0l;

    clrscr();
    for (i=0; i<10; i++)
    {
        printf("\nΔώσε το %do στοιχείο : ", i);
        scanf("%d", &a[i]);
        sum = sum + a[i];
    }
    printf("\nΤο άθροισμα είναι : %ld", sum);
    scanf("%d", &i);
} /* end of main */
```

```
/* Πρόγραμμα c21.c - εκτύπωση στοιχείων πίνακα με αντίστροφη σειρά */
#include <stdio.h>
main()
{
    int a[10], i;
    clrscr();
    for (i=0; i<10; i++)
    {
        printf("\nΔώσε το %do στοιχείο : ", i);
        scanf("%d", &a[i]);
    }
    for (i=9; i>=0; i--)
        printf("\nΤο %do στοιχείο είναι : %d", i, a[i]);
    scanf("%d", &i);
} /* end of main */
```

/ Πρόγραμμα c22.c - εύρεση μεγαλύτερου και μικρότερου βαθμού 10 μαθητών και ποιοι μαθητές έχουν τους βαθμούς αυτούς */*

```
#include <stdio.h>
main()
{
    int a[10], i, max, min, i_max, i_min;

    clrscr();

    /* καταχώρηση βαθμών */
    for (i=0; i<10; i++)
        do
        {
            printf("\nΔώσε τον %do βαθμό : ", i);
            scanf("%d", &a[i]);
        } while (a[i] < 0 || a[i] > 20);

    max = a[0];
    min = a[0];
    i_max = 0;
    i_min = 0;

    for (i=1; i<10; i++)
    {
        if (a[i] > max)
        {
            max = a[i];
            i_max = i;
        }
        if (a[i] < min)
        {
            min = a[i];
            i_min = i;
        }
    }
    /* end of for */

    printf("\nΟ μαθητής Νο %d έχει τον μεγαλύτερο βαθμό : %d", \
           i_max, max);
    printf("\nΟ μαθητής Νο %d έχει τον μικρότερο βαθμό : %d", \
           i_min, min);

    scanf("%d", &i);
} /* end of main */
```

/ Πρόγραμμα c23.c - εκμάθηση των εντολών break και continue */*

```
#include <stdio.h>
main()
{
    char ch;

    clrscr();
    while (1)
    {
        printf("\nΔώσε έναν χαρακτήρα : ");
        ch = getchar();
        putchar();

        if (ch == '#')
            break; /* έξοδος από τον βρόχο */

        if (ch < 'A' || ch > 'Z')
            continue; /* πηγαίνει στην κορυφή του βρόχου */
        else
            printf("έδωσες ένα γράμμα του αγγλικού αλφαβήτου");
    } /* end of while */

    scanf("%c", &ch);
} /* end of main */
```

/ Πρόγραμμα c24.c - εκμάθηση του τελεστή αύξησης ++ */*

```
#include <stdio.h>
main()
{
    int i;

    clrscr();
    i = 0;
    while (i++ == 0)
    {
        printf("\nΟ τελεστής αύξησης ++");
        printf("\ni=%d", ++i);
    }

    printf("\ni=%d", i);

    scanf("%d", &i);
} /* end of main */
```

/ Πρόγραμμα c25.c - εκμάθηση του τελεστή μείωσης - - */*

```
#include <stdio.h>
main()
{
    char ch;

    clrscr();
    ch = 'a'
    while (ch- - == 'a')
    {
        printf("\nΟ τελεστής μείωσης - -");
        printf("\nch=%c", - -ch);
    }

    printf("\nch=%c", ch);

    scanf("%c", &ch);
} /* end of main */
```

/ Πρόγραμμα c26.c - εύρεση των κορυφών μόλυνσης της ατμοσφαιρικής ρύπανσης από ένα σύνολο 20 ημερών - κορυφές μόλυνσης θεωρούνται οι μέρες εκείνες που η μόλυνσή τους είναι μεγαλύτερη από τη μόλυνση της προηγούμενης και της επόμενης μέρας - η μόλυνση έχει τιμή από 0 έως 100 */*

```
#include <stdio.h>
main()
{
    int pol[20], i;

    clrscr();
    for (i=0; i<20; i++)
    {
        do
        {
            printf("\nΔώσε τη μόλυνση της %dης μέρας : ", i);
            scanf("%d", &pol[i]);
        } while (pol[i] < 0 || pol[i] > 100);
    }

    for (i=1; i<19; i++)
        if (pol[i] > pol[i+1] && pol[i] > pol[i-1])
            printf("\nΥπάρχει κορυφή μόλυνσης την %dη μέρα", i);

    scanf("%d", &i);
} /* end of main */
```

/ Πρόγραμμα c27.c - διαβάζει τον βαθμό και την τάξη από 20 μαθητές, ελέγχει αν ο βαθμός ανήκει στο διάστημα 0-20 και αν η τάξη είναι μία από τις a, b ή c - μόλις διαβάσει τα στοιχεία ενός μαθητή, τοποθετεί τον βαθμό του σ' έναν αντίστοιχο πίνακα ανάλογα με την τάξη που ανήκει ο μαθητής */*

```
#include <stdio.h>
main()
{
    int bathmos, i;
    char taxi;
    int ia, ib, ic;          /* μετρητές για τις τρεις τάξεις */
    int a[20], b[20], c[20]; /* πίνακες βαθμών για κάθε τάξη */

    ia=ib=ic=0;           /* απόδοση τιμών σε πολλές μεταβλητές μαζί */
    clrscr();
    for (i=0; i<20; i++)
    {
        do
        {
            printf("\nΔώσε τον βαθμό του %dου μαθητή : ", i);
            scanf("%d", &bathmos);
        } while (bathmos < 0 || bathmos > 20);
        do
        {
            printf("\nΔώσε την τάξη του %dου μαθητή : ", i);
            scanf("%c", &taxi);
        } while (taxi != 'a' && taxi != 'b' && taxi != 'c');

        switch (taxi)
        {
            case 'a' :
                a[ia] = bathmos;
                ia++;
                break;
            case 'b' :
                b[ib] = bathmos;
                ib++;
                break;
            case 'c' :
                c[ic] = bathmos;
                ic++;
                break;
            default :
                printf("\n Κάτι δεν πήγε καλά");
        } /* end of switch */
    } /* end of for */

    scanf("%d", &i);
} /* end of main */
```

/ Πρόγραμμα c28.c - διαβάζει την ηλικία και το φύλο από 10 άτομα, ελέγχει αν η ηλικία είναι στο διάστημα 0-100 και αν το φύλο έχει τιμή 1 (άνδρας) ή 2 (γυναίκα) - μόλις διαβάσει τα στοιχεία ενός ατόμου, τοποθετεί την ηλικία του σ' έναν αντίστοιχο πίνακα ανάλογα με το φύλο που ανήκει και υπολογίζει τον μέσο όρο ηλικίας ανδρών και γυναικών */*

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

```
main()
```

```
{
```

```
    int age, fylo, i;
```

```
    int i_man, i_woman;           /* μετρητές για τα δύο φύλα */
```

```
    int sum_man, sum_woman;      /* αθροιστές για τα δύο φύλα */
```

```
    int man[10], woman[10];     /* πίνακες ηλικίας για κάθε φύλο */
```

```
    float mo_man, mo_woman;     /* μέσοι όροι ηλικίας */
```

```
    i_man=i_woman=sum_man=sum_woman=0;
```

```
    clrscr();
```

```
    for (i=0; i<10; i++)
```

```
    {
```

```
        do
```

```
        {
```

```
            printf("\nΔώσε το φύλο του %dου ατόμου : ", i);
```

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

```
        } while (fylo != 1 && fylo != 2);
```

```
        do
```

```
        {
```

```
            printf("\nΔώσε την ηλικία του %dου ατόμου : ", i);
```

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

```
        } while (age < 0 || age > 100);
```

```
        switch (fylo)
```

```
        {
```

```
            case 1 :
```

```
                man[i_man] = age;
```

```
                i_man++;
```

```
                sum_man += age;
```

```
                break;
```

```
            case 2 :
```

```
                woman[i_woman] = age;
```

```
                i_woman++;
```

```
                sum_woman += age;
```

```
                break;
```

```
            default :
```

```
                printf("\n Κάτι δεν πήγε καλά");
```

```
        } /* end of switch */
```

```
    } /* end of for */
```

```
    mo_man = (float) sum_man / (float) i_man;
```

```
    mo_woman = (float) sum_woman / (float) i_woman;
```

```
    printf("\n%d άνδρες με μέση ηλικία %.1f ", i_man, mo_man);
```

```
    printf("\n%d γυναίκες με μέση ηλικία %.1f", i_woman, mo_woman);
```

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

```
    } /* end of main */
```


/ Πρόγραμμα c29.c - διαβάζει το όπλο κατάταξης (1=στρατός, 2=ναυτικό, 3=αεροπορία), την ηλικία και το ύψος από 20 οπλίτες, ελέγχει αν η ηλικία είναι στο διάστημα 17-40 και αν το ύψος είναι στο διάστημα 140-220 - μόλις διαβάσει τα στοιχεία ενός οπλίτη, τοποθετεί την ηλικία του και το ύψος του σε αντίστοιχους πίνακες ανάλογα με το όπλο που ανήκει και υπολογίζει τον μέσο όρο ηλικίας και ύψους για το κάθε όπλο */*

```
#include <stdio.h>
main()
{
    int age, oplo, ypsos, i;

    int i_str, i_naut, i_aer;          /* μετρητές για τα τρία όπλα */

    int sum_str_age, sum_str_ypsos; /* αθροιστές για τα τρία όπλα */
    int sum_naut_age, sum_naut_ypsos; /* για ηλικία και ύψος */
    int sum_aer_age, sum_aer_ypsos;

    int str_age[20], str_ypsos[20]; /* πίνακες ηλικίας και ύψους */
    int naut_age[20], naut_ypsos[20];
    int aer_age[20], aer_ypsos[20];

    float mo_str_age, mo_str_ypsos; /* μέσοι όροι ηλικίας και ύψους */
    float mo_naut_age, mo_naut_ypsos;
    float mo_aer_age, mo_aer_ypsos;

    i_str=i_naut=i_aer=sum_str_age=sum_str_ypsos=0;
    sum_naut_age=sum_naut_ypsos=sum_aer_age=sum_aer_naut=0;

    clrscr();

    for (i=0; i<20; i++)
    {
        do
        {
            printf("\nΔώσε το όπλο του %dου οπλίτη : ", i);
            scanf("%d", &oplo);
        } while (oplo != 1 && oplo != 2 && oplo!=3);

        do
        {
            printf("\nΔώσε την ηλικία του %dου οπλίτη : ", i);
            scanf("%d", &age);
        } while (age < 17 || age > 40);

        do
        {
            printf("\nΔώσε το ύψος του %dου οπλίτη : ", i);
            scanf("%d", &ypsos);
        } while (ypsos < 140 || ypsos > 220);
    }
}
```

```

switch (oplo)
{
    case 1 :
        str_age[i_str] = age;
        str_ypsos[i_str] = ypsos;
        i_str++;
        sum_str_age += age;
        sum_str_ypsos += ypsos;
        break;
    case 2 :
        naut_age[i_naut] = age;
        naut_ypsos[i_naut] = ypsos;
        i_naut++;
        sum_naut_age += age;
        sum_naut_ypsos += ypsos;
        break;
    case 3 :
        aer_age[i_aer] = age;
        aer_ypsos[i_aer] = ypsos;
        i_aer++;
        sum_aer_age += age;
        sum_aer_ypsos += ypsos;
        break;
    default :
        printf("\n Κάτι δεν πήγε καλά");
} /* end of switch */

} /* end of for */

mo_str_age = (float) sum_str_age / (float) i_str;
mo_str_ypsos = (float) sum_str_ypsos / (float) i_str;
mo_naut_age = (float) sum_naut_age / (float) i_naut;
mo_naut_ypsos = (float) sum_naut_ypsos / (float) i_naut;
mo_aer_age = (float) sum_aer_age / (float) i_aer;
mo_aer_ypsos = (float) sum_aer_ypsos / (float) i_aer;

printf("\n%d στον στρατό με μέση ηλικία %.1f και μέσο ύψος %.1f",\
        i_str, mo_str_age, mo_str_ypsos);
printf("\n%d στο ναυτικό με μέση ηλικία %.1f και μέσο ύψος %.1f",\
        i_naut, mo_naut_age, mo_naut_ypsos);
printf("\n%d στην αεροπορία με μέση ηλικία %.1f και μέσο ύψος\
        %.1f", i_aer, mo_aer_age, mo_aer_ypsos);

scanf("%d", &i);

} /* end of main */

```

/ Πρόγραμμα c30.c - να ταξινομηθούν κατ' αύξουσα σειρά τα στοιχεία ενός πίνακα ακεραίων 10 θέσεων με τον αλγόριθμο Bubble-Sort - σύμφωνα με τον αλγόριθμο αυτό σαρώνουμε τα στοιχεία του πίνακα και αν ένα στοιχείο του είναι μεγαλύτερο από το επόμενο του, τότε ανταλλάσσουμε τις τιμές τους και δίνουμε την τιμή 1 σε μια μεταβλητή flag - η flag έχει την τιμή 0 κάθε φορά που μπαίνουμε στον βρόχο και αν πάρει τιμή 1 συνεχίζουμε να σαρώνουμε τον πίνακα μέχρις ότου διατηρήσει την τιμή 0 σε κάποιο πέρασμα του πίνακα - ένδειξη ότι ο πίνακας ταξινομήθηκε */*

```
#include <stdio.h>
main()
{
    int i, flag, temp, a[10];

    clrscr();

    /* καταχώρηση τιμών */
    for (i=0; i<10; i++)
    {
        printf("\nΔώσε το %do στοιχείο του πίνακα : ", i);
        scanf("%d", &a[i]);
    }

    printf("\nΟ αταξινομήτος πίνακας είναι : ");
    for (i=0; i<10; i++)
        printf("\nΤο %do στοιχείο του πίνακα είναι : %d ", i, a[i]);

    do
    {
        flag = 0;
        for (i=0; i<9; i++)
        {
            if (a[i]>a[i+1])
            {
                temp = a[i];           /* ανταλλαγή τιμών των */
                a[i] = a[i+1];       /* a[i] και a[i+1] */
                a[i+1] = temp;
                flag = 1;
            } /* end of if */
        } /* end of for */
    } while (flag==1);

    printf("\nΟ ταξινομημένος πίνακας είναι : ");
    for (i=0; i<10; i++)
        printf("\nΤο %do στοιχείο του πίνακα είναι : %d ", i, a[i]);

    scanf("%d", &i);
} /* end of main */
```

/ Πρόγραμμα c31.c - να γίνει ο αλγόριθμος της δυαδικής αναζήτησης - δηλ. να αναζητηθεί μια τιμή σ' έναν ταξινομημένο πίνακα 30 θέσεων - θα χρησιμοποιήσουμε δύο μεταβλητές, l1 και l2, και κάθε φορά θα συγκρίνουμε την τιμή που είναι στη μέση των δύο μεταβλητών με την αναζητούμενη τιμή - αν οι τιμές είναι ίσες, έχει βρεθεί η τιμή που ψάχνουμε, διαφορετικά ελαττώνουμε το l2 ή αυξάνουμε το l1 ανάλογα - αν τελικά διαπιστώσουμε ότι l1 > l2, τότε η τιμή δεν υπάρχει */*

```
#include <stdio.h>
main()
{
    int i, l1, l2, a[30];
    int value;    /* η τιμή που ψάχνουμε */
    int found;    /* σημαία που γίνεται ίση με 1 όταν βρεθεί η τιμή */
    int middle;   /* η μέση τιμή του πίνακα που συγκρίνουμε */

    clrscr();

    /* καταχώρηση τιμών */
    for (i=0; i<30; i++)
    {
        printf("\nΔώσε το %do στοιχείο του πίνακα : ", i);
        scanf("%d", &a[i]);
    }

    found = 0;    /* αρχικά η found είναι false (ψευδής) */
    l1 = 0; /* η 1η θέση στον πίνακα */
    l2 = 29;     /* η τελευταία θέση στον πίνακα */
    while (!found && l1 <= l2)
    {
        middle = (l1 + l2) / 2;    /* η μέση τιμή των l1 και l2 */
        if (a[middle] == value)    /* έχει βρεθεί η τιμή */
            found = 1;
        if (a[middle] > value)    /* ψάχνουμε στο 1ο μισό του πίνακα */
            l2 = middle - 1;
        if (a[middle] < value)    /* ψάχνουμε στο 2ο μισό του πίνακα */
            l1 = middle + 1;
    } /* end of while */

    if (found)
        printf("\nΗ τιμή βρέθηκε στη θέση : %d", middle);
    else
        printf("\nΗ τιμή δεν υπάρχει στον πίνακα");

    scanf("%d", &i);
} /* end of main */
```

/ Πρόγραμμα c32.c - να καταχωρηθούν αμέραιες τιμές σ' έναν πίνακα 10 θέσεων και να βρεθεί η μέγιστη και η ελάχιστη τιμή του πίνακα καθώς και η μεταξύ τους διαφορά */*

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

```
main()
```

```
{
```

```
    int i, a[10], max, min;
```

```
    clrscr();
```

```
    /* καταχώρηση τιμών */
```

```
    for (i=0; i<10; i++)
```

```
    {
```

```
        printf("\nΔώσε το %do στοιχείο του πίνακα : ", i);
```

```
        scanf("%d", &a[i]);
```

```
    } /* end of for */
```

```
    max = a[0];
```

```
    min = a[0];
```

```
    for (i=1; i<10; i++)
```

```
    {
```

```
        if (a[i] > max)
```

```
            max = a[i];
```

```
        if (a[i] < min)
```

```
            min = a[i];
```

```
    } /* end of for */
```

```
    printf("\nΗ μέγιστη τιμή του πίνακα είναι : %d", max);
```

```
    printf("\nΗ ελάχιστη τιμή του πίνακα είναι : %d", min);
```

```
    printf("\nΗ διαφορά μέγιστης και ελάχιστης τιμής είναι : %d", \
        max-min);
```

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

```
} /* end of main */
```

/ Πρόγραμμα c33.c - εκμάθηση των δομών (struct) - να δημιουργηθεί μια δομή υπαλλήλου με τα εξής στοιχεία : επώνυμο, όνομα, ηλικία και μισθός και να καταχωρηθούν τιμές για 5 υπαλλήλους - να βρεθεί και να εκτυπωθεί το επώνυμο και το όνομα του υπαλλήλου που έχει τη μεγαλύτερη ηλικία καθώς και το επώνυμο και το όνομα του υπαλλήλου που έχει το μεγαλύτερο μισθό */*

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

```
struct ypalilos {
    char eponymo[15];
    char onoma[10];
    int age;
    long misthos;
}; /* end of struct */
```

/ η δομή δηλώνεται πριν από τη main() και αποτελεί έναν νέο τύπο δεδομένων - μια δομή περιέχει πεδία, τα οποία μπορεί να είναι γνωστοί τύποι δεδομένων της C ή και άλλες δομές */*

```
main()
```

```
{
```

```
    struct ypalilos ypal;
```

```
    /* η μεταβλητή ypal είναι του τύπου δεδομένων ypalilos,
```

```
    δηλ. είναι δομή (struct) */
```

```
    int i, max_age, max_misthos;
```

```
    char max_age_eponymo[15];
```

```
    /* το επώνυμο του υπαλλήλου που έχει τη μεγαλύτερη ηλικία */
```

```
    char max_age_onoma[10];
```

```
    /* το όνομα του υπαλλήλου που έχει τη μεγαλύτερη ηλικία */
```

```
    char max_misthos_eponymo[15];
```

```
    /* το επώνυμο του υπαλλήλου που έχει το μεγαλύτερο μισθό */
```

```
    char max_misthos_onoma[10];
```

```
    /* το όνομα του υπαλλήλου που έχει το μεγαλύτερο μισθό */
```

```
    clrscr();
```

```
    max_age=0;
```

```
    max_misthos=0;
```

```
    for (i=0; i<5; i++)
```

```
    {
```

```
        printf("\nΔώσε τα στοιχεία του %dου υπαλλήλου :", i);
```

```
        printf("\nΕπώνυμο : ");
```

```
        scanf("%s", ypal.eponymo);
```

```
        printf("\nΌνομα : ");
```

```
        scanf("%s", ypal.onoma);
```

```
        printf("\nΗλικία : ");
```

```
        scanf("%d", &ypal.age);
```

```
        printf("\nΜισθός : ");
```

```
        scanf("%ld", &ypal.misthos);
```

```
        printf("\n"); /* αφήνει μία σειρά κενή */
```

```
        /* για να αποκτήσουμε πρόσβαση στα πεδία μιας δομής, γρά-
```

φουμε το όνομα της δομής, μετά τελεία (.) και μετά το όνομα του πεδίου, π.χ.

*ypal.eponymo - τα πεδία μιας δομής αντιμετωπίζονται όπως όλοι οι τύποι δεδομένων της C */*

```
        if (ypal.age > max_age)
        {
            max_age = ypal.age;
            strcpy(max_age_eponymo, ypal.eponymo);
            strcpy(max_age_onoma, ypal.onoma);
        }
    /* η συνάρτηση strcpy() αντιγράφει τα περιεχόμενα του δεύτερου ορίσματός
    της, που είναι μια συμβολοσειρά χαρακτήρων, στο πρώτο όρισμά της - χρησι-
    μοποιείται για να καταχωρούμε τιμές σε συμβολοσειρές χαρακτήρων */

        if (ypal.misthos > max_misthos)
        {
            max_misthos = ypal.misthos;
            strcpy(max_misthos_eponymo, ypal.eponymo);
            strcpy(max_misthos_onoma, ypal.onoma);
        }

    } /* end of for */

    printf("\nΟ %s %s έχει τη μεγαλύτερη ηλικία : %d", \
           max_age_eponymo, max_age_onoma, max_age);
    printf("\nΟ %s %s έχει το μεγαλύτερο βαθμό : %d", \
           max_misthos_eponymo, max_misthos_onoma, max_misthos);

    scanf("%d", &i);
} /* end of main */
```

/ Πρόγραμμα c34.c - εκμάθηση των πινάκων δομών (struct) - να δημιουργηθεί μια δομή πελάτη με τα εξής στοιχεία : κωδικός, επώνυμο, όνομα, διεύθυνση, πόλη και υπόλοιπο και να καταχωρηθούν τιμές για 10 υπαλλήλους σ' έναν πίνακα δομών - να εκτυπωθούν το επώνυμο και το όνομα των πελατών που έχουν υπόλοιπο μεγαλύτερο από 100000 καθώς και το συνολικό υπόλοιπο όλων των πελατών */*

```
#include <stdio.h>
struct pelates {
    int code;
    char eponymo[15];
    char onoma[10];
    char address[20];
    char poli[10];
    long ypoloipo;
}; /* end of struct */
main()
{
    int i;
    long sum=0l;
    struct pelates pel[10];

    clrscr();
    for (i=0; i<10; i++)
    {
        printf("\nΔώστε τα στοιχεία του %dου πελάτη : ", i);
        printf("\nΚωδικός : ");
        scanf("%d", &pel[i].code);
        printf("\nΕπώνυμο : ");
        scanf("%s", pel[i].eponymo);
        printf("\nΌνομα : ");
        scanf("%s", pel[i].onoma);
        printf("\nΔιεύθυνση : ");
        scanf("%s", pel[i].address);
        printf("\nΠόλη : ");
        scanf("%s", pel[i].poli);
        printf("\nΥπόλοιπο : ");
        scanf("%ld", &pel[i].ypoloipo);
        sum += pel[i].ypoloipo;
        if (pel[i].ypoloipo > 100000)
            printf("\nΟ %s %s έχει υπόλοιπο %ld",\
                pel[i].eponymo, pel[i].onoma, pel[i].ypoloipo);
    } /* end of for */
    printf("\nΤο συνολικό υπόλοιπο των πελατών είναι : %ld", sum);
    scanf("%d", &i);
} /* end of main */
```


/ Πρόγραμμα c35.c - εκμάθηση των πινάκων δομών (struct) - να δημιουργηθεί μια δομή μισθωτού με τα εξής στοιχεία : ονοματεπώνυμο, ημερομίσθιο, ημέρες εργασίας, μικτά, κρατήσεις, φόρος και καθαρά και να καταχωρηθούν τιμές για 10 μισθωτούς σ' έναν πίνακα δομών - οι κρατήσεις είναι το 20% των μικτών αποδοχών και ο φόρος είναι το 5% στα μικτά-κρατήσεις - να γίνουν οι υπολογισμοί μισθοδοσίας και να υπολογιστεί και να εκτυπωθεί το σύνολο των καθαρών αποδοχών όλων των μισθωτών */*

```
#include <stdio.h>
struct misthotos {
    char eponymia[25];
    int imerom;
    int meres;
    long mikta;
    long kratisis;
    long foros;
    long kathara;
}; /* end of struct */

main()
{
    int i;
    long sum=0l;
    struct misthotos mist[10];

    clrscr();
    for (i=0; i<10; i++)
    {
        printf("\nΔώστε τα στοιχεία του %dου μισθωτού : ", i);
        printf("\nΟνοματεπώνυμο : ");
        scanf("%s", mist[i].eponymia);
        printf("\nΗμερομίσθιο : ");
        scanf("%d", &mist[i].imerom);
        printf("\nΜέρες εργασίας : ");
        scanf("%d", &mist[i].meres);
        mist[i].mikta = (long) mist[i].imerom * (long) mist[i].meres;
        mist[i].kratisis = (long) mist[i].mikta * 0.2;
        mist[i].foros = (long) (mist[i].mikta - mist[i].kratisis) * 0.05;
        mist[i].kathara = mist[i].mikta - mist[i].kratisis - mist[i].foros;
        sum += mist[i].kathara;
    } /* end of for */
    printf("\nΤο σύνολο των καθαρών αποδοχών είναι : %ld", sum);
    scanf("%d", &i);
} /* end of main */
```

/ Πρόγραμμα c36.c - να διαβαστούν 10 τιμές σ' έναν πίνακα ακεραίων και να βρεθεί η μεγαλύτερη και η δεύτερη μεγαλύτερη τιμή */*

```
#include <stdio.h>
main()
{
    int i, max, max2, a[10];

    clrscr();
    for (i=0; i<10; i++)
    {
        printf("\nΔώστε τον %do αριθμό του πίνακα : ", i);
        scanf("%d", &a[i]);
    } /* end of for */

    max=a[0];
    /* εδώ βρίσκουμε τη μεγαλύτερη τιμή του πίνακα */
    for (i=1; i<10; i++)
        if (a[i]>max)
            max=a[i];

    max2=a[0];
    /* εδώ βρίσκουμε τη 2ο μεγαλύτερη τιμή του πίνακα */
    for (i=1; i<10; i++)
        if (a[i]>max2 && a[i]!=max)
            max2=a[i];

    printf("\nΗ μεγαλύτερη τιμή του πίνακα είναι : %d", max);
    printf("\nΗ δεύτερη μεγαλύτερη τιμή του πίνακα είναι : %d", max2);
    scanf("%d", &i);
} /* end of main */
```

/ Πρόγραμμα c37.c - να διαβαστεί ένας αριθμός ημερών και να βρεθεί από πόσες εβδομάδες και από πόσες μέρες αποτελείται */*

```
#include <stdio.h>
main()
{
    int a, weeks, days;

    clrscr();
    printf("\nΔώστε τον αριθμό των ημερών : ");
    scanf("%d", &a);
    weeks = a / 7;      /* ακέραιο πηλίκο */
    days = a % 7;      /* ακέραιο υπόλοιπο */
    printf("\nΟι %d ημέρες αποτελούνται από %d εβδομάδες και %d\
    ημέρες", a, weeks, days);
    scanf("%d", &i);
} /* end of main */
```

/ Πρόγραμμα c38.c - μια εταιρεία μεταφορών χρησιμοποιεί ένα δικό της αεροπλάνο για τις μεταφορές που κάνει στο εξωτερικό - το αεροπλάνο μπορεί κάθε φορά να μεταφέρει ένα μέγιστο συνολικό φορτίο - τα εμπορεύματα είναι συσκευασμένα σε πακέτα και κάθε πακέτο έχει έναν αύξοντα αριθμό και ένα βάρος - το πρόγραμμα θα διαβάξει τα βάρη των πακέτων και θα υπολογίζει και θα τυπώνει πόσες αεροπορικές πτήσεις θα χρειαστούν για να μεταφερθούν όλα τα πακέτα και πόσο συνολικό φορτίο θα μεταφέρει η κάθε πτήση */*

```
#include <stdio.h>
main()
{
    int i, max_fortio, fortio;
    int paketa;          /* το πλήθος των πακέτων */
    int synolo=0;        /* το άθροισμα των φορτίων */
    int flight=0;        /* ο μετρητής των πτήσεων */

    clrscr();
    printf("\nΔώσε το πλήθος των πακέτων : ");
    scanf("%d", &paketa);
    printf("\nΔώσε το μέγιστο φορτίο : ");
    scanf("%d", &max_fortio);
    for (i=0; i<paketa; i++)
    {
        printf("\nΔώσε το %do φορτίο : ", i);
        scanf("%d", &fortio);
        synolo += fortio;
        if (synolo > max_fortio)
        {
            flight++;
            synolo -= fortio;
            printf("\nΘα φύγει η πτήση %d με %d φορτίο : ",\
                flight, synolo);
            synolo = fortio;
        }
        if (synolo == max_fortio)
        {
            flight++;
            printf("\nΘα φύγει η πτήση %d με %d φορτίο : ",\
                flight, synolo);
            synolo = 0;
        }
    } /* end of for */
    if (synolo > 0)
    {
        flight++;
        printf("\nΘα φύγει η πτήση %d με %d φορτίο : ", flight, synolo);
    }
} /* end of main */
/* για να φύγει μια πτήση, θα πρέπει το άθροισμα των φορτίων να γίνει ίσο με ή να ξεπεράσει το μέγιστο φορτίο */
```

```

/* Πρόγραμμα c39.c - να καταχωρούνται τιμές σε τρεις ακέραιες μεταβλητές
και μετά να καλείται μια συνάρτηση που θα υπολογίζει το άθροισμά τους -
εκμάθηση των συναρτήσεων */
#include <stdio.h>
main()
{
    int a, b, c;
    int athroisma;
    int sum();
    /* εδώ δηλώνουμε τις συναρτήσεις που θα χρησιμοποιήσει το πρόγραμ-
μα - μια συνάρτηση επιστρέφει μία και μόνο μία τιμή στο όνομά της και όταν
την δηλώνουμε, γράφουμε πριν από το όνομά της τον τύπο δεδομένων της, τον
τύπο δεδομένων δηλ. της τιμής που επιστρέφει */

    clrscr();
    printf("\nΔώσε τον πρώτο αριθμό : ");
    scanf("%d", &a);
    printf("\nΔώσε τον δεύτερο αριθμό : ");
    scanf("%d", &b);
    printf("\nΔώσε τον τρίτο αριθμό : ");
    scanf("%d", &c);
    athroisma = sum(a, b, c); /* καλείται η συνάρτηση */
    printf("\nΤο άθροισμα είναι ίσο με : ", athroisma);
    scanf("%d", &a);
} /* end of main */

/* εδώ γράφουμε τις εντολές της συνάρτησης */
int sum(a1, b1, c1)
int a1, b1, c1;
/* τα a1, b1 και c1 είναι τα ορίσματα της συνάρτησης, αλλά είναι και τοπικές
μεταβλητές και εδώ δηλώνεται ο τύπος δεδομένων τους */
{
    int sum1; /* εδώ δηλώνονται οι τοπικές μεταβλητές της συνάρτησης */

    sum1=a1+b1+c1;
    return sum1;
    /* με την εντολή return επιστρέφουμε στο κυρίως πρόγραμμα */
} /* end of sum() */

```

/ Πρόγραμμα c40.c - να καταχωρούνται τιμές σε τρεις ακεραίες μεταβλητές και μετά να καλείται μια συνάρτηση που θα υπολογίζει τη μεγαλύτερη τιμή - εκμάθηση των συναρτήσεων */*

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

```
main()
```

```
{
```

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

```
    int max;
```

```
    int maximum();
```

/ εδώ δηλώνουμε τις συναρτήσεις που θα χρησιμοποιήσει το πρόγραμμα - μια συνάρτηση επιστρέφει μία και μόνο μία τιμή στο όνομά της και όταν την δηλώνουμε, γράφουμε πριν από το όνομά της τον τύπο δεδομένων της, τον τύπο δεδομένων δηλ. της τιμής που επιστρέφει */*

```
    clrscr();
```

```
    printf("\nΔώσε τον πρώτο αριθμό : ");
```

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

```
    printf("\nΔώσε τον δεύτερο αριθμό : ");
```

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

```
    printf("\nΔώσε τον τρίτο αριθμό : ");
```

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

```
    max = maximum(a, b, c); /* καλείται η συνάρτηση */
```

```
    printf("\nΗ μεγαλύτερη τιμή είναι : ", max);
```

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

```
} /* end of main */
```

/ εδώ γράφουμε τις εντολές της συνάρτησης */*

```
int maximum(a1, b1, c1)
```

```
int a1, b1, c1;
```

/ τα a1, b1 και c1 είναι τα ορίσματα της συνάρτησης, αλλά είναι και τοπικές μεταβλητές και εδώ δηλώνεται ο τύπος δεδομένων τους */*

```
{
```

```
    int max1; /* εδώ δηλώνονται οι τοπικές μεταβλητές της συνάρτησης */
```

```
    max1=a1;
```

```
    if (b1>max1)
```

```
        max=b1;
```

```
    if (c1>max1)
```

```
        max=c1;
```

```
    return max1;
```

/ με την εντολή return επιστρέφουμε στο κυρίως πρόγραμμα */*

```
} /* end of sum() */
```

/ Πρόγραμμα c41.c - να καταχωρούνται τιμές σε μια ακέραια μεταβλητή και σε μια μεταβλητή χαρακτήρα και μετά να καλείται μια συνάρτηση τύπου void που θα εκτυπώνει τον χαρακτήρα τόσες φορές όσο είναι η τιμή της ακέραιας μεταβλητής - εκμάθηση των συναρτήσεων τύπου void */*

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

```
main()
```

```
{
```

```
    int a;
```

```
    char ch;
```

```
    void ektyp();
```

/ μια συνάρτηση τύπου void δεν επιστρέφει καμία τιμή ή επηρεάζει δύο ή περισσότερες τιμές στο κυρίως πρόγραμμα και όταν τη δηλώνουμε, γράφουμε πριν από το όνομά της τη λέξη void */*

```
    clrscr();
```

```
    printf("\nΔώσε τον αριθμό : ");
```

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

```
    printf("\nΔώσε τον χαρακτήρα : ");
```

```
    scanf("%c", &ch);
```

ektyp(a, ch); / μια συνάρτηση τύπου void καλείται με απλή αναγραφή του ονόματός της και των ορισμάτων της */*

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

```
} /* end of main */
```

```
void ektyp(a1, ch1)
```

```
int a1;
```

```
char ch1;
```

```
{
```

```
    int i;
```

```
    for (i=1; i<=a1; i++)
```

```
        printf(ch);
```

/ μια συνάρτηση τύπου void δεν έχει εντολή return */*

```
} /* end of ektyp() */
```

/ Πρόγραμμα c42.c - να εκτυπωθούν 20 γραμμές, όπου η 1^η θα έχει ένα *, η 2^η δύο *, η 3^η τρία * κοκ και η τελευταία 20 * - για να γίνονται οι εκτυπώσεις θα καλείται μια συνάρτηση τύπου void που θα εκτυπώνει το * τόσες φορές όσο είναι η τιμή της ακεραίας μεταβλητής που της περνιέται σαν όρισμα - εκμάθηση των συναρτήσεων τύπου void */*

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

```
main()
```

```
{
```

```
    int i;
```

```
    void ektyp();
```

/ μια συνάρτηση τύπου void δεν επιστρέφει καμία τιμή ή επηρεάζει δύο ή περισσότερες τιμές στο κυρίως πρόγραμμα και όταν τη δηλώνουμε, γράφουμε πριν από το όνομά της τη λέξη void */*

```
    clrscr();
```

```
    for (i=1; i<=20; i++)
```

```
    {
```

```
        ektyp(i);    /* κλήση της συνάρτησης */
```

```
        printf("\n"); /* αλλαγή γραμμής */
```

```
    }
```

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

```
} /* end of main */
```

```
void ektyp(i1)
```

```
int i1;
```

/ το όρισμα i1 γίνεται ίσο με τη μεταβλητή i του κυρίως προγράμματος */*

```
{
```

```
    int i;
```

/ αυτή η τοπική μεταβλητή με όνομα i δεν έχει καμία απολύτως σχέση με τη μεταβλητή i του κυρίως προγράμματος - μέσα στις συναρτήσεις επικρατούν οι τοπικές μεταβλητές, οι οποίες δεν αναγνωρίζονται εκτός της συνάρτησης */*

```
    for (i=1; i<=i1; i++)
```

```
        printf('*');
```

```
    /* μια συνάρτηση τύπου void δεν έχει εντολή return */
```

```
} /* end of ektyp() */
```

/ Πρόγραμμα c43.c - να γίνει η εκτύπωση ενός χριστουγεννιάτικου δένδρου που θα αποτελείται από αστεράκια με τη βοήθεια μιας συνάρτησης τύπου void - το δένδρο θα αποτελείται από 10 γραμμές και θα έχει την παρακάτω μορφή - η 1^η γραμμή θα έχει 1 *, η 2^η γραμμή θα έχει 3 *, η 3^η γραμμή 5 * κοκ και η τελευταία γραμμή θα έχει 19 * */*

```

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

```

```

#include <stdio.h>
main()
{
    int i;
    void tree();

    clrscr();
    a=9; /* ο αριθμός των κενών χαρακτήρων πριν από τα αστεράκια */
    b=1; /* ο αριθμός των * σε κάθε γραμμή */
    for (i=1; i<=10; i++)
    {
        tree(a, b); /* κλήση της συνάρτησης */
        a--; /* τα κενά μειώνονται κατά 1 */
        b=b+2; /* τα αστεράκια αυξάνονται κατά 2 */
        printf("\n"); /* αλλαγή γραμμής */
    }
    scanf("%d", &a);
} /* end of main */

```

```

void tree(a1, b1)
int a1, b1;
{
    int i;

    for (i=0; i<a1; i++)
        printf(" "); /* αφήνει κενά πριν τυπώσει τα αστεράκια */
    for (i=0; i<b1; i++)
        printf("*"); /* τυπώνει τα αστεράκια */
} /* end of tree() */

```


/ Πρόγραμμα c44.c - να καταχωρηθούν τιμές σε δύο αμέριαιες μεταβλητές, να οριστούν δύο δείκτες σ' αυτές τις μεταβλητές και να τυπωθούν οι τιμές των μεταβλητών, οι διευθύνσεις τους, οι τιμές των δεικτών, οι τιμές των διευθύνσεων των δεικτών και οι τιμές που δείχνουν οι δείκτες */*

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

```
main()
```

```
{
```

```
    int a, b;
```

```
    int *pa, *pb; /* έτσι ορίζεται ένας δείκτης σ' έναν τύπο δεδομένων */
```

```
    clrscr();
```

```
    printf("\nΔώσε τιμή για την πρώτη μεταβλητή : ");
```

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

```
    printf("\nΔώσε τιμή για τη δεύτερη μεταβλητή : ");
```

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

```
    pa = &a; /* το &a είναι η διεύθυνση της μεταβλητής a */
```

```
    pb = &b;
```

```
    printf("\nΟι τιμές των a και b είναι : %d και %d", a, b);
```

```
    printf("\nΟι διευθύνσεις των a και b είναι : %p και %p", &a, &b);
```

```
    printf("\nΟι τιμές των δεικτών των a και b είναι : %p και %p", pa, pb);
```

```
    printf("\nΟι διευθύνσεις των δεικτών είναι : %p και %p", &pa, &pb);
```

```
    printf("\nΟι τιμές που δείχνουν οι δείκτες είναι : %d και %d", *pa, *pb);
```

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

```
} /* end of main */
```

/ Πρόγραμμα c45.c - να καταχωρηθούν τιμές σε μια μεταβλητή τύπου integer, σε μια τύπου long, σε μια τύπου float, σε μια τύπου double και σε μια τύπου char και να οριστούν δείκτες γι' αυτές τις μεταβλητές και να τυπωθούν οι τιμές των δεικτών - μετά θα αυξηθούν οι τιμές των δεικτών κατά ένα και θα τυπωθούν ξανά - τι παρατηρείτε ; */*

```
#include <stdio.h>
main()
{
    int i, *pi;
    long li, *pli;
    float fi, *pfi;
    double di, *pdi;
    char ch, *pch;

    clrscr();
    printf("\nΔώσε τιμή για τον ακέραιο : ");
    scanf("%d", &i);
    printf("\nΔώσε τιμή για τον μεγάλο ακέραιο : ");
    scanf("%ld", &li);
    printf("\nΔώσε τιμή για τον πραγματικό : ");
    scanf("%f", &fi);
    printf("\nΔώσε τιμή για τον μεγάλο πραγματικό : ");
    scanf("%lf", &di);
    printf("\nΔώσε τιμή για τον χαρακτήρα : ");
    scanf("%c", &ch);
    pi = &i;
    pli = &li;
    pfi = &fi;
    pdi = &di;
    pch = &ch;
    printf("\nΟι τιμές των δεικτών είναι : %p %p %p %p %p", \
        pi, pli, pfi, pdi, pch);
    pi++;
    pli++;
    pfi++;
    pdi++;
    pch++;
    printf("\nΟι νέες τιμές των δεικτών είναι : %p %p %p %p %p", \
        pi, pli, pfi, pdi, pch);
    scanf("%d", &a);
} /* end of main */
```

/ παρατηρούμε ότι αν και αυξήσαμε τις τιμές των δεικτών κατά ένα, μόνο ο pch αυξήθηκε κατά ένα, ο pi αυξήθηκε κατά 2, ο pdi κατά 8 και οι pli και pfi αυξήθηκαν κατά 4 - αυτό γίνεται γιατί ένας δείκτης αυξάνεται σύμφωνα με τα bytes που καταλαμβάνει ο τύπος δεδομένων στον οποίο δείχνει */*

```
/* Πρόγραμμα c46.c - να καταχωρηθούν τιμές σε δύο ακέραιες μεταβλητές και
μετά να κληθεί μια συνάρτηση τύπου void που θα ανταλλάσσει τις τιμές τους -
εκμάθηση της κλήσης με αναφορά στη C */
#include <stdio.h>
main()
{
    int a, b;
    void change();

    clrscr();
    printf("\nΔώσε τιμή για τον πρώτο ακέραιο : ");
    scanf("%d", &a);
    printf("\nΔώσε τιμή για τον δεύτερο ακέραιο : ");
    scanf("%d", &b);
    change(&a, &b);
    /* περνάμε στη συνάρτηση τις διευθύνσεις των ορισμάτων */
    printf("\nΟι νέες τιμές των a και b είναι : %d και %d", a, b);
    scanf("%d", &a);
} /* end of main */

void change()
int *a1, *b1;      /* τα a1 και b1 είναι δείκτες σε ακέραιους */
{
    int temp;

    temp = *a1; /* τα *a1 και *b1 είναι οι τιμές που δείχνουν οι δείκτες */
    *a1 = *b1; /* στην ουσία δηλ. είναι οι τιμές των a και b */
    *b1 = temp;
    /* έτσι αλλάζουν οι τιμές των μεταβλητών a και b του κυρίως προγράμ-
ματος από τα αντίστοιχα ορίσματά τους στη συνάρτηση change(), a1 και b1 */
} /* end of change() */
```

/ Πρόγραμμα c47.c - να καταχωρηθούν τιμές σε δύο ακεραίες μεταβλητές και μετά να κληθεί μια συνάρτηση τύπου void που θα επιστρέφει στον πρώτο ακεραίο το άθροισμα των δύο ακεραίων και στον δεύτερο ακεραίο τη διαφορά των δύο ακεραίων - εκμάθηση της κλήσης με αναφορά στη C */*

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

```
main()
```

```
{
```

```
    int a, b;
```

```
    void abplus();
```

```
    clrscr();
```

```
    printf("\nΔώσε τιμή για τον πρώτο ακεραίο : ");
```

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

```
    printf("\nΔώσε τιμή για τον δεύτερο ακεραίο : ");
```

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

```
    abplus(&a, &b);
```

```
    printf("\nΟι νέες τιμές των a και b είναι : %d και %d", a, b);
```

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

```
} /* end of main */
```

```
void abplus()
```

```
int *a1, *b1;
```

```
{
```

```
    int sum, diff;
```

```
    sum = *a1 + *b1;
```

```
    diff = *a1 - *b1;
```

```
    *a1 = sum;
```

```
    *b1 = diff;
```

```
} /* end of abplus() */
```

/ Πρόγραμμα c48.c - να δημιουργηθεί ένας πίνακας ακεραίων δύο διαστάσεων και να καταχωρηθούν σ' αυτόν οι 6 προσπάθειες 8 αθλητών στο άλμα εις μήκος, δηλ. σύνολο 48 τιμές και να βρεθεί το καλύτερο άλμα, ποιος αθλητής το έκανε και σε ποια προσπάθεια - εκμάθηση και επεξεργασία των πινάκων δύο διαστάσεων */*

```
#include <stdio.h>
main()
{
    int a[6][8]; /* ορισμός πίνακα 6 γραμμών και 8 στηλών */
    int max, athlete, trial, i, j;

    clrscr();
    max=0; /* η καλύτερη προσπάθεια */
    athlete=0; /* ο αριθμός του καλύτερου αθλητή */
    trial=0; /* η προσπάθεια του καλύτερου αθλητή */
    for (i=0; i<6; i++)
        /* σ' αυτό το for διαβάζουμε τους 6 αθλητές */
        for (j=0; j<8; j++)
            /* σ' αυτό το for διαβάζουμε τις 8 προσπάθειες ενός αθλητή */
            {
                printf("\nΔώσε την %dη προσπάθεια του %dου αθλητή : \
                    ", j, i);
                scanf("%d", &a[i][j]);
                if (a[i][j] > max)
                {
                    max = a[i][j];
                    athlete = i; /* ο καλύτερος αθλητής */
                    trial = j; /* η προσπάθεια του καλύτερου αθλητή */
                } /* end of if */
            } /* end of for-j */

    printf("\nΗ καλύτερη προσπάθεια ήταν %d και την έκανε ο %δος \
        αθλητής στην %dη προσπάθειά του", max, athlete, trial);
    scanf("%d", &i);
} /* end of main */
```

/ Πρόγραμμα c49.c - να δημιουργηθεί ένας πίνακας ακεραίων 10 θέσεων, να καταχωρηθούν σ' αυτόν τιμές με τη χρήση δεικτών και να βρεθεί η μεγαλύτερη τιμή και σε ποια θέση του πίνακα είναι - εκμάθηση της επεξεργασίας πίνακα με τη χρήση δεικτών */*

```
#include <stdio.h>
main()
{
    int a[10]; /* Το όνομα ενός πίνακα είναι η διεύθυνση του 1ου στοιχείου
του πίνακα, δηλ. ισχύει a == &a[0] και γενικά a+i == &a[i] */
    int max, i_max, i;

    clrscr();
    for (i=0; i<10; i++)
    {
        printf("\nΔώσε την %dη τιμή του πίνακα : ", i);
        scanf("%d", a+i);
    } /* end of for */

    max = *a;
    i_max = 0;
    for (i=1; i<10; i++)
    {
        if (*(a+i) > max) /* Το *(a+i) = a[i] */
        {
            max = *(a+i);
            i_max = i;
        }
    } /* end of for */

    printf("\nΗ μεγαλύτερη τιμή είναι : %d και είναι στη θέση : %d", \
max, i_max);
    scanf("%d", &i);
} /* end of main */
```

/ Πρόγραμμα c50.c - να δημιουργηθεί ένας πίνακας ακεραίων 10 θέσεων, να καταχωρηθούν σ' αυτόν τιμές με τη χρήση δεικτών και να κληθεί μια συνάρτηση τύπου void στην οποία θα περαστεί σαν όρισμα ο πίνακας και η συνάρτηση θα βρούκει τη μεγαλύτερη τιμή και σε ποια θέση του πίνακα είναι - εκμάθηση της επεξεργασίας πίνακα με τη χρήση δεικτών και του περάσματος ενός πίνακα σαν όρισμα σε μια συνάρτηση */*

```
#include <stdio.h>
main()
{
    int a[10]; /* Το όνομα ενός πίνακα είναι η διεύθυνση του 1ου στοιχείου
του πίνακα, δηλ. ισχύει a == &a[0] και γενικά a+i == &a[i] */
    int i;
    void maximum();

    clrscr();
    for (i=0; i<10; i++)
    {
        printf("\nΔώσε την %dη τιμή του πίνακα : ", i);
        scanf("%d", a+i);
    } /* end of for */

    maximum(a);
    /* Στη συνάρτηση περνάμε το όνομα του πίνακα, δηλ. τη διεύθυνση του
1ου στοιχείου του πίνακα που είναι στην ουσία ένας δείκτης σ' ακέραιο */

    scanf("%d", &i);
} /* end of main */
```

```
void maximum(a1)
int *a1;
/* Το a1 είναι δείκτης σ' ακέραιο και με τη βοήθειά του έχουμε απευθείας
πρόσβαση στα στοιχεία του πίνακα a[i] */
{
    int max, i_max, i; /* Τα max και i_max είναι τοπικές μεταβλητές */

    max = *a1;
    i_max = 0;
    for (i=1; i<10; i++)
    {
        if (*(a1+i) > max) /* Το *(a1+i) = a1[i] = a[i] */
        {
            max = *(a1+i);
            i_max = i;
        }
    } /* end of for */
    printf("\nΗ μεγαλύτερη τιμή είναι : %d και είναι στη θέση : %d", \
max, i_max);
} /* end of maximum() */
```

/ Πρόγραμμα c51.c - να οριστούν τρεις πίνακες ακεραίων a, b, και c, 10 θέσεων, να δημιουργηθεί μια συνάρτηση τύπου void με την οποία θα δοθούν τιμές στους πίνακες a και b και μια άλλη συνάρτηση τύπου void η οποία θα προσθέτει τις τιμές των δύο πρώτων πινάκων και θα τις καταχωρεί στον τρίτο πίνακα */*

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

```
main()
```

```
{
```

```
    int a[10], b[10], c[10], i;
```

```
    void store_pinakas();
```

```
    void add_pinakas();
```

```
    clrscr();
```

```
    store_pinakas(a);
```

```
    store_pinakas(b);
```

```
    add_pinakas(a, b, c);
```

```
    printf("\nΟι αριθμοί του 1ου πίνακα είναι : ");
```

```
    for (i=0; i<10; i++)
```

```
        printf("%d ", *(a+i));
```

```
    printf("\nΟι αριθμοί του 2ου πίνακα είναι : ");
```

```
    for (i=0; i<10; i++)
```

```
        printf("%d ", *(b+i));
```

```
    printf("\nΟι αριθμοί του 3ου πίνακα είναι : ");
```

```
    for (i=0; i<10; i++)
```

```
        printf("%d ", *(c+i));
```

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

```
} /* end of main */
```

```
void store_pinakas(a1);
```

```
int *a1;
```

```
{
```

```
    int i;
```

```
    for (i=0; i<10; i++)
```

```
    {
```

```
        printf("\nΔώσε την %dη τιμή του πίνακα : ", i);
```

```
        scanf("%d", a1+i);
```

```
    } /* end of for */
```

```
} /* end of store_pinakas() */
```

```
void add_pinakas(a1, b1, c1);
```

```
int *a1, *b1, *c1;
```

```
{
```

```
    int i;
```

```
    for (i=0; i<10; i++)
```

```
        *(c1+i) = *(a1+i) + *(b1+i);
```

```
} /* end of add_pinakas() */
```


/ Πρόγραμμα c52.c - να καταχωρηθούν ονόματα σε τρεις συμβολοσειρές χαρακτήρα και να χρησιμοποιηθούν οι συναρτήσεις αλφαριθμητικών της C : strlen(), strcat(), strcpy() και strcmp() - εκμάθηση των συναρτήσεων αλφαριθμητικών */*

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

```
main()
```

```
{
```

```
    char name1[20], name2[20], name3[40];
    int mikos;
```

```
    clrscr();
    printf("\nΔώσε το πρώτο όνομα : ");
    scanf("%s", name1);
    printf("\nΔώσε το δεύτερο όνομα : ");
    scanf("%s", name2);
    printf("\nΔώσε το τρίτο όνομα : ");
    scanf("%s", name3);
```

```
/* με τη συνάρτηση strlen() παίρνουμε το μήκος ενός string */
```

```
    mikos = strlen(name1);
    printf("\nΤο πρώτο όνομα έχει %d χαρακτήρες : ", mikos);
    mikos = strlen(name2);
    printf("\nΤο δεύτερο όνομα έχει %d χαρακτήρες : ", mikos);
    mikos = strlen(name3);
    printf("\nΤο τρίτο όνομα έχει %d χαρακτήρες : ", mikos);
```

```
/* με τη συνάρτηση strcat() ενώνουμε δύο strings και το αποτέλεσμα το καταχωρούμε στο πρώτο string */
```

```
    strcat(name3, name1);
    printf("\nΤο τρίτο όνομα είναι τώρα ίσο με το τρίτο και το πρώτο \
        όνομα μαζί %s : ", name3);
```

```
/* με τη συνάρτηση strcpy() καταχωρούμε το δεύτερο string στο πρώτο */
```

```
    strcpy(name3, name2);
    printf("\nΤο τρίτο όνομα είναι τώρα ίσο με το δεύτερο %s και %s : ", \
        name3, name2);
```

```
/* με τη συνάρτηση strcmp() συγκρίνουμε δύο strings και αν είναι ίδια, η συνάρτηση επιστρέφει την τιμή 0 */
```

```
    if (strcmp(name1, name2) == 0)
        printf("\nΤα δύο πρώτα ονόματα είναι ίδια");
    else
        printf("\nΤα δύο πρώτα ονόματα δεν είναι ίδια");
```

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

```
} /* end of main */
```

/ Πρόγραμμα c53.c - να οριστούν τρεις πίνακες ακεραίων a, b, και c, 10, 10 και 20 θέσεων αντίστοιχα - να δημιουργηθεί μια συνάρτηση τύπου void με την οποία θα δοθούν τιμές στους πίνακες a και b, μια άλλη συνάρτηση τύπου void η οποία θα ταξινομεί κατ' αύξουσα σειρά τις τιμές των πινάκων a και b και μια τρίτη συνάρτηση τύπου void η οποία θα συγχωνεύει τις τιμές του πίνακα a και του πίνακα b και θα τις καταχωρεί ταξινομημένες στον τρίτο πίνακα c - εκμάθηση της ταξινόμησης με συγχώνευση (sorting by merging) */*

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

```
main()
```

```
{
```

```
    int a[10], b[10], c[20], i;
```

```
    void store_pinakas();    /* συνάρτηση καταχώρησης */
```

```
    void sort_pinakas();    /* συνάρτηση ταξινόμησης */
```

```
    void merge_pinakas();   /* συνάρτηση συγχώνευσης */
```

```
    clrscr();
```

```
    store_pinakas(a);
```

```
    store_pinakas(b);
```

```
    sort_pinakas(a);
```

```
    sort_pinakas(b);
```

```
    merge_pinakas(a, b, c);
```

```
    printf("\nΟι αριθμοί του 1ου πίνακα είναι : ");
```

```
    for (i=0; i<10; i++)
```

```
        printf("%d  ", *(a+i));
```

```
    printf("\nΟι αριθμοί του 2ου πίνακα είναι : ");
```

```
    for (i=0; i<10; i++)
```

```
        printf("%d  ", *(b+i));
```

```
    printf("\nΟι αριθμοί του 3ου πίνακα είναι : ");
```

```
    for (i=0; i<20; i++)
```

```
        printf("%d  ", *(c+i));
```

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

```
} /* end of main */
```

```
void store_pinakas(a1);
```

```
int *a1;
```

```
{
```

```
    int i;
```

```
    for (i=0; i<10; i++)
```

```
    {
```

```
        printf("\nΔώσε την %dη τιμή του πίνακα : ", i);
```

```
        scanf("%d", a1+i);
```

```
    } /* end of for */
```

```
} /* end of store_pinakas() */
```

```

void sort_pinakas(a1);
int *a1;
{
    int i, flag, temp;

    do
    {
        flag = 0;
        for (i=0; i<9; i++)
        {
            if (*(a1+i) > *(a1+i+1))
            {
                temp = *(a1+i);
                *(a1+i) = *(a1+i+1);
                *(a1+i+1)=temp;
                flag = 1;
            } /* end of if */
        } /* end of for */
    }
    while (flag==1);
} /* end of sort_pinakas() */
/* για την ταξινόμηση του πίνακα χρησιμοποιούμε τη μέθοδο Bubble-Sort */

```

```

void merge_pinakas(a1, b1, c1);
int *a1, *b1, *c1;
{
    int i, ia=0, ib=0, ic=0;
    /* τα ia, ib και ic δείχνουν στη θέση του αντίστοιχου πίνακα όπου θα
    γίνει η επόμενη καταχώρηση */

    /* ο παρακάτω βρόχος θα εκτελείται όσο δεν έχουμε φθάσει στο τέλος
    κάποιου πίνακα */
    while (ia<10 && ib<10)
    {
        if (*(a1+ia) < *(b1+ib))
        {
            *(c1+ic) = *(a1+ia);
            ia++;
            ic++;
        }
        else
        {
            *(c1+ic) = *(b1+ib);
            ib++;
            ic++;
        }
    }
} /* end of while */

```

/ μόλις βγούμε από τον βρόχο while, θα υπάρχουν ακόμα τιμές για καταχώρηση στον πίνακα c, από τον πίνακα a ή από τον πίνακα b */*

```

if (ia<10)
{
    /* σ' αυτή την περίπτωση ib == 10 */
    for (i=ia; i<10; i++)
    {
        *(c1+ic) = *(a1+i);
        ic++;
    }
} /* end of if-ia */

```

```

if (ib<10)
{
    /* σ' αυτή την περίπτωση ia == 10 */
    for (i=ib; i<10; i++)
    {
        *(c1+ic) = *(b1+i);
        ic++;
    }
} /* end of if-ib */

```

/ end of merge_pinakas() */*

/ για να δουλέψει ο αλγόριθμος της ταξινόμησης με συγχώνευση (sorting by merging), πρέπει να είναι ταξινομημένοι και οι δύο πίνακες (a και b) που θα συγχωνεύσουμε - ορίζουμε τρεις δείκτες, ia, ib και ic, για τους τρεις πίνακες, με αρχικές τιμές 0 για όλους και συγκρίνουμε την τιμή του πίνακα a που υπάρχει στη θέση ia με την τιμή του πίνακα b που υπάρχει στη θέση ib - όποια τιμή είναι μικρότερη, καταχωρείται στον πίνακα c και στη θέση ic - μόλις γίνει μια καταχώρηση στον πίνακα c, αυξάνονται κατά ένα το ic και το ia ή το ib, ανάλογα - μόλις τελειώσουν τα στοιχεία του ενός πίνακα, τότε το ia ή το ib θα γίνει ίσο με 10 - σ' αυτή την περίπτωση βγαίνουμε από τον βρόχο και παίρνουμε τα υπόλοιπα στοιχεία του άλλου πίνακα και τα τοποθετούμε στον πίνακα c */*

/ Πρόγραμμα c54.c - να οριστούν μια εξωτερική μεταβλητή i και μια τοπική a και μια στατική b σε μια συνάρτηση test() τύπου void - να δοθούν αρχικές τιμές 0 στις a και b - να κληθεί η συνάρτηση test() τρεις φορές και κάθε φορά να αυξάνουν οι μεταβλητές a και b κατά ένα - να τυπώνονται οι τιμές των a και b και να δικαιολογηθούν τα αποτελέσματα */*

```
#include <stdio.h>
int i;
main()
{
    extern int i; /* δηλώνουμε ότι η i είναι εξωτερική μεταβλητή */
    void test();

    clrscr();
    for(i=0; i<3; i++)
        test();
    scanf("%d", &i);
} /* end of main */

void test()
{
    auto int i;          /* τοπική μεταβλητή */
    int a=0;             /* τοπική μεταβλητή */
    static int b=0;     /* στατική τοπική μεταβλητή */

    printf("\na=%d b=%d", a++, b++);
} /* end of test() */
```

/ το a θα τυπώνεται συνέχεια σαν 0, ενώ το b θα τυπώνεται με τις τιμές 0, 1 και 2 και αυτό γιατί είναι στατική μεταβλητή και κάθε φορά που καλείται η συνάρτηση test(), η b κρατάει την τιμή της από την προηγούμενη κλήση της συνάρτησης - με τη δήλωση extern δηλώνουμε (υπενθυμίζουμε) ότι μια μεταβλητή είναι εξωτερική, δηλ. καθολική, και η δήλωσή της είναι πριν από τη main() - μια εξωτερική μεταβλητή αναγνωρίζεται από όλες τις συναρτήσεις του προγράμματος, εκτός από εκείνες που έχουν μια τοπική μεταβλητή με το ίδιο όνομα, όπως συμβαίνει με τη μεταβλητή i παραπάνω - με τη δήλωση auto, που είναι προαιρετική, δηλώνουμε ότι μια μεταβλητή είναι τοπική, δηλ. αναγνωρίζεται μόνο μέσα στη συνάρτηση όπου ανήκει και υπερισχύει έτσι των εξωτερικών μεταβλητών που έχουν το ίδιο όνομα */*

/ Πρόγραμμα c55.c - να διαβαστεί μια ακέραια μεταβλητή a και να τυπωθούν οι τιμές της στην 8δική και στη 16δική μορφή */*

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

```
main()
```

```
{
```

```
    int a;
```

```
    clrscr();
```

```
    printf("\nΔώσε τιμή για τον ακέραιο");
```

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

```
    printf("\nH 8δική τιμή του ακεραίου είναι : %o", a);
```

```
    printf("\nH 16δική τιμή του ακεραίου είναι : %x", a);
```

```
    printf("\nH 16δική τιμή του ακεραίου είναι : %X", a);
```

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

```
} /* end of main */
```

/ ο προσδιοριστής %o εμφανίζει την 8δική μορφή, ο προσδιοριστής %x εμφανίζει 16δική μορφή με τα ψηφία-γράμματα (a έως f) μικρά, ενώ ο προσδιοριστής %X εμφανίζει 16δική μορφή με τα ψηφία-γράμματα (A έως F) κεφαλαία */*

/ Πρόγραμμα c56.c - να διαβαστούν δύο ακέραιες μεταβλητές a και b, να τυπωθούν οι αντίστροφες τιμές τους στην 8δική μορφή και να υπολογιστούν σε μια τρίτη μεταβλητή c η εφαρμογή του δυαδικού τελεστή AND (&) και του δυαδικού τελεστή OR (|) στις μεταβλητές a και b */*

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

```
main()
```

```
{
```

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

```
    clrscr();
```

```
    printf("\nΔώσε τιμή για τον ακέραιο a");
```

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

```
    printf("\nΔώσε τιμή για τον ακέραιο b");
```

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

```
    printf("\nH αντίστροφη 8δική τιμή του a είναι : %o", a);
```

```
    printf("\nH αντίστροφη 8δική τιμή του b είναι : %o", b);
```

```
    c = a & b;
```

```
    printf("\nH 8δική τιμή του a AND b είναι : %o", c);
```

```
    c = a | b;
```

```
    printf("\nH 8δική τιμή του a OR b είναι : %o", c);
```

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

```
} /* end of main */
```

/ χρησιμοποιήσαμε 8δική μορφή για τους αριθμούς γιατί δεν υπάρχει προσδιοριστής για να εμφανίσουμε τη 2δική μορφή ενός αριθμού - ο τελεστής αντιστρέφει τα δυαδικά ψηφία, δηλ. μετατρέπει το 1 σε 0 και το 0 σε 1 - ο τελεστής & επιδρά στα δυαδικά ψηφία των a και b και δίνει αποτέλεσμα 1 όταν και τα δύο ψηφία είναι 1, ενώ ο τελεστής | επιδρά στα δυαδικά ψηφία των a και b και δίνει αποτέλεσμα 0 όταν και τα δύο ψηφία είναι 0 */*

/ Πρόγραμμα c57.c - να διαβαστούν 8 2δικά ψηφία (0 ή 1) και να υπολογιστεί και να εκτυπωθεί ο αντίστοιχος δεκαδικός αριθμός */*

```
#include <stdio.h>
main()
{
    int a[i], i, sum;

    clrscr();
    for(i=0; i<8; i++)
        do
        {
            printf("\nΔώσε το %do ψηφίο : ", i);
            scanf("%d", &a[i]);
        } while (a[i]!=0 && a[i]!=1);
    sum=a[0]*1+a[1]*2+a[2]*4+a[3]*8+a[4]*16+a[5]*32+a[6]*64+\
        a[7]*128;
    printf("\nΟ αντίστοιχος δεκαδικός είναι : %d", sum);
    scanf("%d", &i);
} /* end of main */
```

/ για παράδειγμα, αν ο 2δικός αριθμός είναι 10011011, τότε ο αντίστοιχος δεκαδικός αριθμός είναι :*

$$1X1 + 1X2 + 0X4 + 1X8 + 1X16 + 0X32 + 0X64 + 1X128 = 155 */$$

/ Πρόγραμμα c58.c - να διαβαστούν 4 8δικά ψηφία (0 έως 7) και να υπολογιστεί και να εκτυπωθεί ο αντίστοιχος δεκαδικός αριθμός */*

```
#include <stdio.h>
main()
{
    int a[i], i, sum;

    clrscr();
    for(i=0; i<4; i++)
        do
        {
            printf("\nΔώσε το %do ψηφίο : ", i);
            scanf("%d", &a[i]);
        } while (a[i]<0 || a[i]>7);
    sum=a[0]*1+a[1]*8+a[2]*64+a[3]*512;
    printf("\nΟ αντίστοιχος δεκαδικός είναι : %d", sum);
    scanf("%d", &i);
} /* end of main */
```

/ για παράδειγμα, αν ο 8δικός αριθμός είναι 4207, τότε ο αντίστοιχος δεκαδικός αριθμός είναι :*

$$7X1 + 0X8 + 2X64 + 4X512 = 2183 */$$

/ Πρόγραμμα c59.c - να διαβαστούν οι χαρακτήρες από ένα αρχείο κειμένου και να εμφανιστούν στην οθόνη */*

```
#include <stdio.h>
main()
{
    FILE *fp;    /* δηλώνουμε έναν δείκτη σε αρχείο */
    char ch;
    int i;
    clrscr();
    fp=fopen("C:\IEK.BAT", "r");    /* άνοιγμα αρχείου για διάβασμα */
    while(!feof(fp))    /* έλεγχος τέλους αρχείου */
    {
        ch=fgetc(fp);    /* διάβασμα χαρακτήρα από αρχείο */
        printf("%c", ch);
    } /* end of while */
    fclose(fp);    /* κλείσιμο αρχείου */
    scanf("%d", &i);
} /* end of main */
```

/ Πρόγραμμα c60.c - να καταχωρηθούν διάφορες λέξεις σ' ένα αρχείο, να κλείσει το αρχείο και μετά να ανοίξει πάλι για να διαβαστούν τα στοιχεία του και να εμφανιστούν στην οθόνη - η καταχώρηση λέξεων θα τελειώσει όταν δώσουμε * */*

```
#include <stdio.h>
main()
{
    FILE *fp;
    char word[20];    /* η λέξη που θα χρησιμοποιεί το αρχείο */
    int i;
    clrscr();
    fp=fopen("C:\IEK.DAT", "w");    /* άνοιγμα αρχείου για γράψιμο */
    do
    {
        printf("Δώσε μια λέξη : ");
        scanf("%s", word);
        fputs(word, fp);    /* γράψιμο λέξης σε αρχείο */
        fputs("\n", fp);    /* γράψιμο του χαρακτήρα <enter> */
    } while (strcmp(word, ""));
    fclose(fp);    /* κλείσιμο αρχείου */

    pfb=fopen("C:\IEK.DAT", "r");    /* άνοιγμα αρχείου για διάβασμα */
    while(!feof(fp))    /* έλεγχος τέλους αρχείου */
    {
        fgets(word, 20, fp);    /* διάβασμα λέξης από αρχείο */
        printf("%s", word);
    } /* end of while */
    fclose(fp);    /* κλείσιμο αρχείου */
    scanf("%d", &i);
} /* end of main */
```


/ Πρόγραμμα c61.c - να καταχωρηθούν το επώνυμο, το όνομα, η διεύθυνση και το υπόλοιπο για 5 πελάτες στο αρχείο pelates.dat και μετά να εμφανιστούν αυτά τα στοιχεία στην οθόνη */*

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    FILE *fp;
    char eponymo[15], onoma[10], address[15], ypolipo[10];
    int i;
    clrscr();

    pfb=fopen("C:\PELATES.DAT", "w");
    for (i=0; i<5; i++)
    {
        printf("\nΔώσε το επώνυμο : ");
        scanf("%s", eponymo);
        fputs(eponymo, fp);
        printf("\nΔώσε το όνομα : ");
        scanf("%s", onoma);
        fputs(onoma, fp);
        printf("\nΔώσε τη διεύθυνση : ");
        scanf("%s", address);
        fputs(address, fp);
        printf("\nΔώσε το υπόλοιπο : ");
        scanf("%s", ypolipo);
        fputs(ypolipo, fp);
    } /* end of for */
    fclose(pfb);
    getch();

    pfb=fopen("C:\PELATES.DAT", "r");
    for (i=0; i<5; i++)
    {
        fgets(eponymo, 15, fp);
        printf("%s", eponymo);
        fgets(onoma, 10, fp);
        printf("%s", onoma);
        fgets(address, 15, pfb);
        printf("%s", address);
        fgets(ypolipo, 10, fp);
        printf("%d", atoi(ypolipo));
        printf("\n");
    } /* end of for */
    fclose(pfb);
    scanf("%d",&i);
} /* end of main */
/* τις αριθμητικές μεταβλητές τις καταχωρούμε στο αρχείο σαν strings και εμφανίζουμε τις τιμές τους με τη συνάρτηση atoi() */
```

/ Πρόγραμμα c62.c - να δημιουργηθεί μια δομή αποθήκης με τα εξής πεδία : κωδικός, περιγραφή, τιμή, ποσότητα και stock ασφαλείας - να καταχωρηθούν τιμές για 10 είδη με τη χρήση δείκτη σε δομή - να υπολογιστεί η συνολική αξία της αποθήκης και να εμφανιστούν στην οθόνη τα είδη από τα οποία πρέπει να γίνει παραγγελία - εκμάθηση της χρήσης δείκτη σε δομή */*

```
#include <stdio.h>
struct apothiki {
    int kodikos;
    char perigrifi[20];
    int timi;
    int posotita;
    int stock;
}; /* end struct */

main()
{
    int i;
    long sum=0l; /* συνολική αξία της αποθήκης */
    struct apothiki apoth[10]; /* δήλωση πίνακα από δομές */
    struct apothiki *p_apoth; /* δήλωση δείκτη σε δομή */

    clrscr();
    for(i=0; i<10; i++)
    {
        p_apoth = &apoth[i]; /* πού δείχνει ο δείκτης */
        printf("\n\nΣτοιχεία %dου είδους : ", i);
        printf("\nΔώσε κωδικό : ");
        scanf("%d", &p_apoth->kodikos);
        printf("\nΔώσε περιγραφή : ");
        scanf("%s", p_apoth->perigrifi);
        printf("\nΔώσε τιμή : ");
        scanf("%d", &p_apoth->timi);
        printf("\nΔώσε ποσότητα : ");
        scanf("%d", &p_apoth->posotita);
        printf("\nΔώσε stock ασφαλείας : ");
        scanf("%d", &p_apoth->stock);
        if ((p_apoth->posotita) < (p_apoth->stock))
            printf("\nΠαραγγελία από το : %s", p_apoth->perigrifi);
        sum += (long)(p_apoth->timi)*(long)(p_apoth->posotita);
    } /* end of for */
    printf("\nΗ συνολική αξία της αποθήκης είναι : %ld", sum);
    scanf("%d", &i);
} /* end of main */
```

/ για να έχουμε πρόσβαση στα πεδία μιας δομής με τη χρήση δείκτη, μπορούμε να χρησιμοποιήσουμε τη μορφή :*

*p_apoth->kodikos ή τη μορφή : (*p_apoth).kodikos,
ενώ για να μπορούμε να διαβάσουμε με την εντολή scanf(), χρησιμοποιούμε :
scanf("%d", &p_apoth->kodikos); ή scanf("%d", &(*p_apoth).kodikos); */*

/ Πρόγραμμα c63.c - τρία υποκαταστήματα διακινούν 6 εφημερίδες το καθένα - να καταχωρηθούν οι ποσότητες των εφημερίδων που διακινούν τα υποκαταστήματα σε μία ημέρα και να υπολογιστεί :*

α) η συνολική ποσότητα των εφημερίδων που διακινούν όλα τα υποκαταστήματα,

β) η ποσότητα των εφημερίδων που διακινεί το κάθε υποκατάστημα,

γ) η ποσότητα που διακινείται για κάθε εφημερίδα και από τα τρία υποκαταστήματα συνολικά και

*δ) η μέγιστη ποσότητα εφημερίδων, ποιο υποκατάστημα τη διακινεί και για ποια εφημερίδα */*

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

```
main()
```

```
{
```

```
    int a[i][j];    /* ορισμός πίνακα δύο διαστάσεων */
```

```
    int i, j, max, i_max, j_max;
```

```
    int sum=0, sum_ypok, sum_efim;
```

```
    clrscr();
```

/ στο παρακάτω nested-for διαβάζουμε τα στοιχεία του πίνακα - το i συμβολίζει τα υποκαταστήματα και το j τις εφημερίδες */*

```
    for(i=0; i<3; i++)
```

```
        for(j=0; j<6; j++)
```

```
        {
```

```
            printf("\nΤιμή για %do υποκ. και %dh εφημερίδα : ", i, j);
```

```
            scanf("%d", &a[i][j]);
```

```
            sum += a[i][j];    /* συνολική διακίνηση */
```

```
        } /* end of for */
```

/ εδώ υπολογίζουμε το σύνολο των εφημερίδων που διακινεί το κάθε υποκατάστημα - το sum_ypok υπολογίζει το άθροισμα αυτό για κάθε υποκατάστημα ξεχωριστά */*

```
    for(i=0; i<3; i++)
```

```
    {
```

```
        sum_ypok=0;
```

```
        for(j=0; j<6; j++)
```

```
            sum_ypok += a[i][j];
```

```
        printf("\nΤο %do υποκ. διακίνησε %d εφημ.: ", i, sum_ypok);
```

```
    } /* end of for-i */
```

/ εδώ υπολογίζουμε το σύνολο των εφημερίδων που διακινείται για κάθε εφημερίδα ξεχωριστά και για όλα τα υποκαταστήματα - το sum_efim υπολογίζει το άθροισμα αυτό για κάθε εφημερίδα ξεχωριστά */*

```
    for(j=0; j<6; j++)
```

```
    {
```

```
        sum_efim=0;
```

```
        for(i=0; i<3; i++)
```

```
            sum_efim += a[i][j];
```

```
        printf("\nΗ %dh εφημ. διακίνησε %d αντίτυπα : ", j, sum_efim);
```

```
    } /* end of for-j */
```

```
/* εδώ υπολογίζουμε τη μέγιστη ποσότητα των αντιτύπων που διακινή-
θηκε (max), για ποια εφημερίδα (j_max) έγινε η διακίνηση και σε ποιο υπο-
κατάστημα (i_max) */
max = a[0][0];
i_max = 0;
j_max = 0;
for(i=0; i<3; i++)
    for(j=0; j<6; j++)
    {
        if (a[i][j] > max)
        {
            max = a[i][j];
            i_max = i;
            j_max = j;
        } /* end of if */
    } /* end of for */
printf("\nΟι περισσότερες πωλήσεις είναι : %d", max);
printf("\nΤις έκανε το %do υποκατάστημα", i_max);
printf("\nΓια την %dh εφημερίδα", j_max);

scanf("%d", &i);
} /* end of main */
```

/ Πρόγραμμα c64.c - να γραφεί συνάρτηση με όνομα htoi() που θα μετατρέψει ένα αλφαριθμητικό από 16δικούς χαρακτήρες στον αντίστοιχο ακέραιο */*

```

long htoi(s)
char *s;          /* s είναι το string που θα διαβάσουμε */
{
int length, timh, i=1;
long sum = 0L;
    length = strlen(s); /* πρέπει να ξέρουμε το μήκος του s */
    while (*s != '\0') /* τελειώνουμε όταν βρούμε τον χαρακτήρα '\0' */
    {
        switch (*s) /* εδώ υπολογίζεται η αξία του κάθε ψηφίου */
        {
            case '0' : timh = 0; break;
            case '1' : timh = 1; break;
            case '2' : timh = 2; break;
            case '3' : timh = 3; break;
            case '4' : timh = 4; break;
            case '5' : timh = 5; break;
            case '6' : timh = 6; break;
            case '7' : timh = 7; break;
            case '8' : timh = 8; break;
            case '9' : timh = 9; break;
            case 'a' :
            case 'A' : timh = 10; break;
            case 'b' :
            case 'B' : timh = 11; break;
            case 'c' :
            case 'C' : timh = 12; break;
            case 'd' :
            case 'D' : timh = 13; break;
            case 'e' :
            case 'E' : timh = 14; break;
            case 'f' :
            case 'F' : timh = 15; break;
        } /* end of switch */
        sum = sum + power(16, length-i)*timh;
        i++;
        *s++; /* πάει στον επόμενο χαρακτήρα */
    } /* end of while */
    return sum;
    /* το sum είναι η δεκαδική τιμή του 16δικού string και η τιμή επιστροφής της συνάρτησης */
} /* end of htoi() */

```

/ για να μετατρέψουμε έναν 16δικό αριθμό σε ακέραιο κάνουμε τα εξής :
ξεκινάμε από τα δεξιά και πολλαπλασιάζουμε την τιμή κάθε ψηφίου του α-
ριθμού με μια δύναμη του 16. Οι δυνάμεις του 16 ξεκινούν από 0 από το πρώ-
το δεξιά ψηφίο και φθάνουν μέχρι την τιμή του πλήθους των 16δικών ψηφίων
μείον 1 στο πρώτο αριστερά ψηφίο. Οι τιμές των 16δικών ψηφίων είναι από 0
- 9 για τα ψηφία 0 - 9 και μετά είναι :*

*a, A : 10 b, B : 11 c, C : 12 d, D : 13 e, E : 14 f, F : 15
Π.χ.: 6F0d = 6 X 16³ + 15 X 16² + 0 X 16¹ + 13 X 16⁰ */*

/ η συνάρτηση htoi() είναι τύπου long γιατί επιστρέφει στο όνομά της την α-
ντίστοιχη ακέραια τιμή ενός string από 16δικά ψηφία. Χρησιμοποιούμε τη συ-
νάρτηση power(16, length-i), όπου υψώνουμε το 16 στη δύναμη length-i, όπου
το i ξεκινά από την τιμή 1 και αυξάνει κάθε φορά κατά 1 έως ότου γίνει ίσο με
0. Το αποτέλεσμα από την ύψωση σε δύναμη το πολλαπλασιάζουμε με τη με-
ταβλητή timh, που είναι η αξία του κάθε ψηφίου, και μετά τα προσθέτουμε
στη μεταβλητή sum, η οποία και θα χρησιμοποιηθεί στην εντολή return */*

/ Πρόγραμμα c65.c - να γραφεί συνάρτηση με όνομα reverse() που θα αντιμε-
ταθέτει τους χαρακτήρες ενός αλφαριθμητικού */*

```
void reverse(s)
```

```
char *s;
```

```
{
```

```
    int length;
```

```
    char *t;
```

```
    length = strlen(s);
```

```
    for (i=0; i<=length-1; i++)    /* καταχωρούμε έναν-έναν τους
```

```
        *(t+i) = *(s+length-1-i); /* χαρακτήρες του s στο t */
```

```
    while(*s++ = *t++);    /* εδώ καταχωρούμε το string t στο s */
```

```
} /* end of reverse() */
```

/ η συνάρτηση αυτή δεν επιστρέφει κάποια τιμή, αλλά επηρεάζει το string s -
βάζουμε στη μεταβλητή length το μήκος του s και χρησιμοποιούμε το βοηθητι-
κό string t όπου καταχωρούμε τους χαρακτήρες του s από το τέλος προς την
αρχή - μετά καταχωρούμε στο string s τα περιεχόμενα του string t */*

/ Πρόγραμμα c66.c - να γραφεί συνάρτηση με όνομα strlen() που θα επιστρέ-
φει το μήκος ενός αλφαριθμητικού */*

```
int strlen(s)
```

```
char *s;
```

```
{
```

```
    int length = 0;
```

```
    /* τελειώνουμε όταν φθάσουμε στον χαρακτήρα '\0' */
```

```
    while (*s++ != '\0')
```

```
        length++;
```

```
    return length;
```

```
} /* end of strlen() */
```

/ η συνάρτηση είναι τύπου int και χρησιμοποιεί την τοπική μεταβλητή length
για να υπολογίσει το μήκος του string s */*

/ Πρόγραμμα c67.c - να γραφεί συνάρτηση με όνομα strcmp() που θα δέχεται δύο αλφαριθμητικά σαν είσοδο και θα επιστρέφει 0 αν τα δύο αλφαριθμητικά είναι ίσα, -1 αν το πρώτο αλφαριθμητικό είναι μικρότερο από το δεύτερο και 1 αν το πρώτο αλφαριθμητικό είναι μεγαλύτερο από το δεύτερο */*

```
int strcmp(s, t)
char *s, *t;
{
    int timh, i=0;

    while (*s++ == *t++) /* όσο οι χαρακτήρες των s και t είναι ίσοι */
        i++;           /* αυξάνουμε το i κατά ένα */
    if (i==strlen(s) && i==strlen(t)) /* τα δύο strings είναι ίσα */
        timh = 0;
    else
    {
        /* *(s+i) και *(t+i) είναι οι χαρακτήρες των δύο strings
           που είναι οι επόμενοι από τους τελευταίους ίσους
           χαρακτήρες των δύο strings */
        if (*(s+i) < *(t+i)) /* το s είναι μικρότερο από το t */
            timh = -1;
        if (*(s+i) > *(t+i)) /* το s είναι μεγαλύτερο από το t */
            timh = 1;
    } /* end of if-else */
    return timh; /* η μεταβλητή timh επιστρέφεται από τη συνάρτηση */
} /* end of strcmp() */
```

/ εδώ χρησιμοποιούμε την εξής τεχνική : μια μεταβλητή i βρίσκει πόσοι χαρακτήρες είναι ίσοι από τα δύο strings ξεκινώντας φυσικά από τον πρώτο χαρακτήρα από αριστερά - μόλις βρούμε έναν χαρακτήρα που δεν είναι ίσος, τότε βγαίνουμε από τον βρόχο while και ελέγχουμε με μια εντολή if αν το i είναι ίσο με τα μήκη και των δύο strings - αν συμβαίνει αυτό, τότε τα δύο strings είναι ίσα και θέτουμε timh = 0 - αλλιώς, ελέγχουμε τους επόμενους χαρακτήρες και από τα δύο strings και τους συγκρίνουμε και θέτουμε timh = 1 ή timh = -1 ανάλογα */*

Δείτε τα επόμενα παραδείγματα :

```
*s = "ΑΘΗΝΑ";           *s = "ΒΑΣΙΛΗΣ";   *s = "ΦΛΩΡΙΝΑ";
*t = "ΑΘΗΝΑΙΚΟΣ";     *t = "ΑΓΓΕΛΟΣ";   *t = "ΦΛΩΡΙΝΑ";
i = 5 και timh = -1    i = 0 και timh = 1   i = 7 και timh = 0
```

/ Πρόγραμμα c68.c - να γραφεί συνάρτηση που θα δέχεται σαν είσοδο το όνομα ενός μήνα (π.χ. Jan, Feb, κλπ) και το έτος και θα επιστρέφει τον αριθμό των ημερών του μήνα, αφού λάβει υπόψη του τα δίσεκτα έτη - δίσεκτα είναι τα έτη που διαιρούνται ακριβώς με τον αριθμό 4, εκτός απ' αυτά που διαιρούνται ακριβώς με το 100 και απ' αυτά εξαιρούνται τα έτη που διαιρούνται ακριβώς με το 400 - δηλ. το 1996 είναι δίσεκτο, το 1900 όχι και το 2000 ναι */*

```
int hmeres_mhna(mhnas, etos)
char *mhnas;
int etos;
{
    int hmeres, ypol4, ypol100, ypol400, disekto;

    ypol4 = etos % 4;          /* υπόλοιπο της διαίρεσης με το 4 */
    ypol100 = etos % 100;     /* υπόλοιπο της διαίρεσης με το 100 */
    ypol400 = etos % 400;    /* υπόλοιπο της διαίρεσης με το 400 */

    if (ypol4 != 0)
        disekto = 0;         /* όχι δίσεκτο */
    if (ypol4 == 0 && ypol100 != 0)
        disekto = 1;        /* δίσεκτο */
    if (ypol100 == 0 && ypol400 != 0)
        disekto = 0;        /* όχι δίσεκτο */
    if (ypol400 == 0)
        disekto = 1;        /* δίσεκτο */

    if (strcmp(mhnas, 'Jan') == 0)
        hmeres = 31;
    if (strcmp(mhnas, 'Feb') == 0)
    {
        if (disekto)
            hmeres = 29;
        else
            hmeres = 28;
    } /* end of if */
    if (strcmp(mhnas, 'Mar') == 0)
        hmeres = 31;
    if (strcmp(mhnas, 'Apr') == 0)
        hmeres = 30;
    if (strcmp(mhnas, 'May') == 0)
        hmeres = 31;
    if (strcmp(mhnas, 'Jun') == 0)
        hmeres = 30;
    if (strcmp(mhnas, 'Jul') == 0)
        hmeres = 31;
    if (strcmp(mhnas, 'Aug') == 0)
        hmeres = 31;
    if (strcmp(mhnas, 'Sep') == 0)
        hmeres = 30;
```



```
    if (strcmp(mhnas, 'Oct') == 0)
        hmeres = 31;
    if (strcmp(mhnas, 'Nov') == 0)
        hmeres = 30;
    if (strcmp(mhnas, 'Dec') == 0)
        hmeres = 31;

    return hmeres;    /* επιστρέφουμε τις μέρες που έχει ο μήνας */
} /* end of hmeres_mhna() */
```

/ Πρόγραμμα c69.c - ο υπολογισμός του φόρου των μισθωτών γίνεται με βάση τον παρακάτω πίνακα :*

Κλιμάκιο Εισοδήματος	Φορολογικός Συντελεστής	Φόρος Κλιμακίου	Σύνολο Εισοδήματος	Σύνολο Φόρου
1.355.000	0	0	1.355.000	0
1.282.500	5	64.125	2.637.500	64.125
1.582.500	15	237.375	4.220.000	301.500
3.165.000	30	949.500	7.385.000	1.251.000
8.440.000	40	3.376.000	15.825.000	4.627.000
Υπερβάλλον	45			

τα πρώτα 1.355.000 είναι αφορολόγητα, τα επόμενα 1.282.500 έχουν φόρο 5%, δηλ. 64.125 κοκ - για παράδειγμα αν το φορολογητέο εισόδημα είναι 2.800.000, τότε αφαιρούνται τα πρώτα 1.355.000, τα επόμενα 1.282.500 έχουν φόρο 5%, δηλ. 64.125 και τα υπόλοιπα 2.800.000 - 1.355.000 - 1.282.500 = 162.500 έχουν φόρο 15%, δηλ. 24.375 - ο συνολικός φόρος είναι έτσι : 64.125 + 24.375 = 88.500 δραχ. - με βάση τα παραπάνω να γίνει πρόγραμμα που να υπολογίζει τον φόρο ενός μισθωτού */

```
main()
{
    long foros, eisodhma;

    printf("\nΔώσε το εισόδημα : ");
    scanf("%ld", &eisodhma);
    if (eisodhma < 1355000)
        foros = 0;
    if (eisodhma >= 1355000 && eisodhma < 2637500)
        foros = int((eisodhma-1355000)*0.05+0.5);
    if (eisodhma >= 2637500 && eisodhma < 4220000)
        foros = int((eisodhma-2637500)*0.15+64125+0.5);
    if (eisodhma >= 4220000 && eisodhma < 7385000)
        foros = int((eisodhma-4220000)*0.30+301500+0.5);
    if (eisodhma >= 7385000 && eisodhma < 15825000)
        foros = int((eisodhma-7385000)*0.40+1251000+0.5);
    if (eisodhma >= 15825000)
        foros = int((eisodhma-15825000)*0.45+4627000+0.5);

    printf("\nΟ φόρος είναι : %ld", foros);
    scanf("%ld", &eisodhma);
} /* end of main */
```