

Άσκηση 1

Να γραφεί ένα πρόγραμμα το οποίο να εμφανίζει όλους τους αριθμούς από το 111 μέχρι το 999, εκτός από αυτούς που αρχίζουν από 4 ή τελειώνουν σε 6.

```

/* los tropos: me continue gia termatismo mias epanalhpshts */

#include <stdio.h>

int main()
{
    int i;

    for(i = 111; i < 999; i++)
    {
        if((i/100 == 4) || (i%10 == 6))
            continue; /* Paradeigma pros apofygh!! */
        printf("%d\n", i);
    }
    return 0;
}

```

Άσκηση 1 (συνέχεια)

Να ξαναγραφεί το προηγούμενο πρόγραμμα χωρίς την continue.

```

/* 2os tropos: xwris continue*/

#include <stdio.h>

int main()
{
    int i;

    for(i = 111; i < 999; i++)
        if((i/100 != 4) && (i%10 != 6))
            printf("%d\n", i);

    return 0;
}

```

Σημείωση. Η συνθήκη που ελέγχουμε τώρα είναι η άρνηση της προηγούμενης, δηλαδή:

```
!((i/100 == 4) || (i%10 == 6)) ⇔ ((i/100 != 4) && (i%10 != 6))
```

Άσκηση 2

Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει συνεχώς ημερήσιες θερμοκρασίες και να εμφανίζει το ποσοστό των ημερών που η θερμοκρασία ήταν ίση, μικρότερη ή μεγαλύτερη από 40°C. Αν ο χρήστης εισάγει τιμή ίση με 100, η εισαγωγή των θερμοκρασιών να τερματίζει.

```

/* los tropos: me break gia termatismo tou atermonos loop */

#include <stdio.h>
int main()
{
    int days, eq, less, more;
    float temp;

    days = eq = less = more = 0;
    while(1)
    {
        printf("Enter temperature (to STOP, type 100): ");
        scanf("%f", &temp);
        if(temp == 100)
            break;          /* Paradeigma pros apofygh!! */

        days++;
        if(temp < 40)
            less++;
        else if(temp > 40)
            more++;
        else
            eq++;
    }
    if(days)
        printf("L: %.2f M: %.2f Eq: %.2f\n", 100.0*less/days,
              100.0*more/days, 100.0*eq/days);
    return 0;
}

```

```

/* 2os tropos: xwris break */

#include <stdio.h>
int main()
{
    int days, eq, less, more;
    float temp;

    days = eq = less = more = 0;
    printf("Enter temperature (to STOP, type 100): ");
    scanf("%f", &temp);
    while(temp != 100)
    {
        days++;
        if(temp < 40)
            less++;
        else if(temp > 40)
            more++;
        else
            eq++;
        printf("Enter temperature (to STOP, type 100): ");
        scanf("%f", &temp);
    }
    if(days)
        printf("L: %.2f M: %.2f Eq: %.2f\n", 100.0*less/days,
            100.0*more/days, 100.0*eq/days);
    return 0;
}

```

### Άσκηση 3

Να γραφεί ένα πρόγραμμα το οποίο να δημιουργεί τους 9 πρώτους όρους της ακολουθίας Fibonacci  $x[n]$ , όπου:  $x[n] = x[n-1] + x[n-2]$  με αρχικές συνθήκες  $x[0] = x[1] = 1$ .

```

#include <stdio.h>

int main()
{
    int i, fib[9];
    char f[9]; /* Προς υποδοχή της λέξης Fibonacci */

    fib[0] = fib[1] = 1;
    for(i=2; i<9; i++)
        fib[i] = fib[i-1] + fib[i-2];

    f[0]='F'; f[1]='i'; f[2]='b'; f[3]='o'; f[4]='n';
    f[5]='a'; f[6]='c'; f[7]=f[6]; f[8]=f[1];

    printf("Oi 9 prwtoi oroi ths akolouthias ");
    for(i=0; i<9; i++)
        printf("%c", f[i]);
    printf("\n");

    for(i=0; i<9; i++)
        printf("%d ", fib[i]);
    printf("\n");

    return 0;
}

```

Άσκηση 3 (συνέχεια)

Να επαναλάβετε το προηγούμενο πρόγραμμα επεκτείνοντας τον πίνακα χαρακτήρων  $f$  ώστε να προσθέσουμε τον χαρακτήρα '\0' με ASCII κωδικό 0 στο τέλος. Ο χαρακτήρας αυτός έχει μνημονικό όνομα NULL στον πίνακα ASCII και σηματοδοτεί το τέλος μιας συμβολοσειράς σε συναρτήσεις που χειρίζονται συμβολοσειρές. Μία τέτοια συνάρτηση είναι και η printf() όταν χρησιμοποιείται το προσδιοριστικό μετατροπής %s.

```
int main()
{
    int i, fib[9];
    char f[10]; /* Προς υποδοχή της λέξης Fibonacci */

    ...

    f[0]='F'; f[1]='i'; f[2]='b'; f[3]='o'; f[4]='\n';
    f[5]='a'; f[6]='c'; f[7]=f[6]; f[8]=f[1]; f[9]='\0';

    printf("Oi 9 prwtoi oroi ths akolouthias %s:\n", f);

    ...

    return 0;
}
```

```
Oi 9 prwtoi oroi ths akolouthias Fibonacci:
0 1 1 2 3 5 8 13 21
```

Άσκηση 4

Να γραφεί ένα πρόγραμμα το οποίο να υπολογίζει το γινόμενο δύο πινάκων:  $A$  με διαστάσεις  $N \times M$  και  $B$  με διαστάσεις  $M \times P$ . Το γινόμενο των δύο πινάκων θα είναι ο πίνακας  $C$  με διαστάσεις  $N \times P$  και στοιχεία που υπολογίζονται από τη σχέση:

$$C(i,j) = \sum_{k=1}^M A(i,k) * B(k,j)$$

```
#include <stdio.h>
int main()
{
    int i,j,k;
    int A[][3]={{1,0,2},{2,1,1}}; /* Emesos prosdiorismos grammwn */
    int B[3][4]={{0,1,1,2},{2,0,3,1},{1,3,0,2}};
    int C[2][4]={0}; /* Arxikopoihsh olou tou pinaka sto 0 */

    for(i=0; i<2; i++)
        for(j=0; j<4; j++)
            for(k=0; k<3; k++)
                C[i][j] += A[i][k] * B[k][j];

    printf("C = AB :\n");
    for(i=0; i<2; i++)
    { for(j=0; j<4; j++)
        printf("%d ", C[i][j]);
        printf("\n"); }

    return 0;
}
```

Άσκηση 5

Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει τρεις ακεραίους (π.χ. a, b και c) τον έναν μετά τον άλλο, όχι όλους μαζί. Το πρόγραμμα να αναγκάζει τον χρήστη να εισάγει τους αριθμούς σε φθίνουσα διάταξη (π.χ. a > b > c).

```
#include <stdio.h>

int main(void)
{
    int a, b, c;

    printf("Enter first number: ");
    scanf("%d", &a);
    do
    {
        printf("\nEnter second number: ");
        scanf("%d", &b);
    } while(a <= b);
    do
    {
        printf("\nEnter third number: ");
        scanf("%d", &c);
    } while(b <= c);

    printf("\nNumbers typed: %d %d %d\n", a, b, c);

    return 0;
}
```

Άσκηση 6

Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει δύο ακεραίους (π.χ. M, N) και να εμφανίζει το M×N πλέγμα. Οι διαστάσεις κάθε κελιού στο πλέγμα πρέπει να είναι 3×2. Για παράδειγμα, δείτε ένα 3×5 πλέγμα:

```
+---+---+---+---+
| | | | | |
+---+---+---+---+
| | | | | |
+---+---+---+---+
| | | | | |
+---+---+---+---+
```

Οι τρεις οριζόντιοι χαρακτήρες κάθε κελιού να είναι: + - - και οι δύο κάθετοι: + |.

```

#include <stdio.h>
int main(void)
{
    int i, j, m, n, height, width;

    printf("Enter the number of vertical cells: ");
    scanf("%d", &m);

    printf("Enter the number of horizontal cells: ");
    scanf("%d", &n);

    height = m*2;
    width = n*3;

    for(i = 0; i <= height; i++)
    {
        for(j = 0; j <= width; j++)
        {
            if((i%2) == 0)      /* if i is even */
            {
                if(j%3 == 0)    /* if j = 0,3,6,9,... */
                    printf("+");
                else
                    printf("-");
            }
        }
    }

```

```

        else      /* ... i is odd */
        {
            if(j%3 == 0)
                printf("|");
            else
                printf(" ");
        }
    }
    printf("\n");
}

return 0;
}

```

Άσκηση 7

Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει έναν ακέραιο (π.χ. N) και να εμφανίζει το αποτέλεσμα της παράστασης:

$$\frac{1}{1 \times 3} + \frac{1}{3 \times 5} + \frac{1}{5 \times 7} + \dots + \frac{1}{(N-2) \times N}$$

Το πρόγραμμα να υποχρεώνει τον χρήστη να εισάγει έναν περιττό αριθμό μεγαλύτερο ή ίσο του 3.

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

    do
    {
        printf("Enter odd number (>=3): ");
        scanf("%d", &N);
    } while(N < 3 || ((N%2) == 0));

    for(i = 1, sum = 0.0; i <= N-2; i += 2)
        sum += 1.0/(i*(i+2));

    printf("Sum: %.10lf\n", sum);

    return 0;
}
```

Άσκηση 8

Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει τους βαθμούς N φοιτητών (το N να καθορίζεται με define) και να τους αποθηκεύει σε έναν πίνακα. Στη συνέχεια, το πρόγραμμα να υπολογίζει τον μέσο όρο τους m, την τυπική απόκλιση s και το πλήθος των φοιτητών που πήραν βαθμό A, B, C και F λαμβάνοντας υπόψη ότι:

- F είναι βαθμός στο διάστημα [0 , m-s)
- C είναι βαθμός στο διάστημα [m-s , m)
- B είναι βαθμός στο διάστημα [m , m+s)
- A είναι βαθμός στο διάστημα [m+s , 10]

Υπενθυμίζεται ότι αν  $x_i$  είναι ο βαθμός του i-στού φοιτητή τότε:

$$m = (1/N) \sum_{i=1}^N x_i$$

$$s^2 = (1/N) \sum_{i=1}^N [x_i - m]^2$$

```
#include <stdio.h>
#include <math.h>
#define N 10

int main(void)
{
    int i, A, B, C, F;
    float m, s, sum, var, grade[N];

    for(i = 0, sum = 0.0F; i < N; i++)
    {
        printf("\nPlease enter grade[%d]: ", i);
        scanf("%f", &grade[i]);
        sum += grade[i];
    }

    m = sum/N;          /* mean of student grades */

    for(i=0, var=0.0F; i < N; i++)
        var += (grade[i] - m)*(grade[i] - m);
}
```

```

var = var/N;      /* variance of student grades */
s = sqrt(var);   /* standard deviation of student grades */

for(i=0, A=B=C=F=0; i<N; i++)
    if (grade[i] >= m+s) A++;
    else if (grade[i] >= m) B++;
    else if (grade[i] >= m-s) C++;
    else F++;

printf("\n\n");
printf("Mean: %.1f - Standard Deviation: %.1f\n\n",m,s);
printf("A: %d (%.f%%)\n",A,100.0*A/N);
printf("B: %d (%.f%%)\n",B,100.0*B/N);
printf("C: %d (%.f%%)\n",C,100.0*C/N);
printf("F: %d (%.f%%)\n",F,100.0*F/N);

system("pause");
return 0;
}

```

### Άσκηση 8 (συνέχεια)

Να δημιουργήσετε ένα αρχείο στο ίδιο directory με το εκτελέσιμο πρόγραμμά σας (.exe), με όνομα vathmoi.dat, στο οποίο να αποθηκεύσετε τους βαθμούς 40 φοιτητών. Στη συνέχεια, να εκτελέσετε το πρόγραμμα της Άσκησης 5 με χρήση ανακατεύθυνσης εισόδου / εξόδου. Την πρώτη φορά να διαβάζει τα δεδομένα από το αρχείο και να εμφανίζει τα αποτελέσματα στην οθόνη ενώ τη δεύτερη φορά να στέλνει (αποθηκεύει) τα αποτελέσματα στο αρχείο results.txt. Χρειάζονται τα printf που καθοδηγούν τον χρήστη όταν διαβάζετε από αρχείο;

Για ανακατεύθυνση της εισόδου έτσι ώστε η εισαγωγή των δεδομένων στο πρόγραμμα να μη γίνει από το πληκτρολόγιο αλλά από αρχείο (π.χ. το αρχείο "vathmoi.dat"), πρώτα δημιουργήστε το αρχείο "vathmoi.dat" με έναν text editor, βάλτε 40 βαθμούς που να χωρίζονται με «λευκούς» χαρακτήρες (space ' ', tab '\t', newline '\n') και σώστε το στο directory που βρίσκεται και το εκτελέσιμο αρχείο σας (.exe). Στη συνέχεια, εκτελέστε το πρόγραμμα από τη γραμμή εντολών (command line) όπως στο παράδειγμα:

```
C:\>grades.exe < vathmoi.dat
```

Αν θέλετε να ανακατευθύνετε και την έξοδο (δηλαδή τα printf) ώστε να μην εμφανίζεται στην οθόνη αλλά να αποθηκεύεται σε αρχείο (π.χ. στο "results.txt") εκτελέστε το ως εξής:

```
C:\>grades.exe < vathmoi.dat > results.txt (δημιουργεί νέο αρχείο)
```

ή

```
C:\>grades.exe < vathmoi.dat >> results.txt (προσαρτά στο τέλος υπάρχοντος αρχείου ή
δημιουργεί νέο αρχείο αν δεν υπάρχει)
```