

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 epanalhpshs */
#include <stdio.h>

int main()
{
    int i;

    for(i = 111; i <= 999; i++)
    {
        if((i/100 == 4) || (i%10 == 6))
            continue; /* Phgaine sthn epomenh epanalhpsh */
        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:

$!(i/100 == 4) \text{ || } (i\%10 == 6) \Leftrightarrow (i/100 != 4) \text{ \&\& } (i\%10 != 6)$

Αν θέλαμε να εμφανίσουμε τους αριθμούς που εκτός των παραπάνω περιορισμών πρέπει το άθροισμα των ψηφίων τους να είναι είτε μεγαλύτερο του 20 ή μικρότερο του 7, τότε η if γίνεται:

```
sum = i/100 + (i/10)%10 + i%10;
if((i/100 != 4) && (i%10 != 6) && ((sum > 20) || (sum < 7)))
```

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 ακολουθίας ");
    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

Να επαναλάβετε το προηγούμενο πρόγραμμα επεκτείνοντας τον πίνακα χαρακτήρων f ώστε να προσθέσουμε τον χαρακτήρα $\backslash 0$ 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 ακολουθίας %s:\n", f);

    ...

    return 0;
}
```

```
Oi 9 prwtoi oroi ths ακολουθίας Fibonacci:
1 1 2 3 5 8 13 21 34
```

Άσκηση 4

Να γραφεί ένα πρόγραμμα το οποίο να υπολογίζει το γινόμενο δύο πινάκων: Α με διαστάσεις ΝxΜ και Β με διαστάσεις ΜxΡ. Το γινόμενο των δύο πινάκων θα είναι ο πίνακας C με διαστάσεις ΝxΡ 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 ακεραίους και να τους αποθηκεύει σε έναν τετραγωνικό πίνακα (π.χ. 3x3). Στη συνέχεια, να ελέγχει αν ο πίνακας είναι άνω τριγωνικός, κάτω τριγωνικός ή και τα δύο (διαγώνιος πίνακας). Ένας τετραγωνικός πίνακας είναι άνω (κάτω) τριγωνικός όταν όλα τα στοιχεία του κάτω (πάνω) από την κύρια διαγώνιο είναι ίσα με μηδέν.

```
#include <stdio.h>
#define ROWS 3
int main(void)
{
    int i, j, up_trig, low_trig, arr[ROWS][ROWS];

    up_trig = low_trig = 1;

    for(i = 0; i < ROWS; i++)
        for(j = 0; j < ROWS; j++)
        {
            printf("Enter element [%d][%d]: ", i, j);
            scanf("%d", &arr[i][j]);
            if((i > j) && (arr[i][j] != 0))
                up_trig = 0;
            else if((i < j) && (arr[i][j] != 0))
                low_trig = 0;
        }
}
```

```
if(up_trig && low_trig)
    printf("Diagonal Matrix\n");
else if(up_trig)
    printf("Upper Triangular Matrix\n");
else if(low_trig)
    printf("Low Triangular Matrix\n");
else
    printf("No Triangular Matrix\n");

return 0;
}
```