

## Προγραμματισμός Ι – Εργαστήριο 03-12-2012

### 1<sup>η</sup> Εργαστηριακή Άσκηση

Να δημιουργήσετε πρόγραμμα που θα τυπώνει το ημερολόγιο ενός μηνός. Ο χρήστης θα ορίζει των αριθμό των ημερών του μήνα και την ημέρα της εβδομάδας όπου αρχίζει ο μήνας.

Enter number of days in month: 31  
Enter starting day of the week (1 = Sun, 7 = Sat): 3

```
      1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```

Ενδεικτική λύση:

```
#include <stdio.h>

int main(void)
{
    int i, n, start_day;

    printf("Enter number of days in month: ");
    scanf("%d", &n);
    printf("Enter starting day of the week (1=Sun, 7=Sat): ");
    scanf("%d", &start_day);

    /* print "blank dates" */
    for (i = 1; i < start_day; i++)
        printf("  ");

    /* print the calendar */
    for (i = 1; i <= n; i++)
    {
        printf("%3d", i);
        if ((start_day + i - 1) % 7 == 0)
            printf("\n");
    }

    printf("\n");
    return 0;
}
```

## 2<sup>η</sup> Εργαστηριακή Άσκηση\_a

Το παρακάτω πρόγραμμα διαβάζει έναν ακέραιο και εμφανίζει το άθροισμα του τετραγώνου και του κύβου του αριθμού, με τη χρήση των συναρτήσεων *square* και *cube*.

Συμπληρώστε τα κενά.

```
#include <stdio.h>

int square(int a);
int cube(int a);

int main()
{
    int i, j, k;

    printf("Enter number: ");
    scanf("%d", &i);

    j = square(i);
    k = cube(i);
    printf("sum = %d\n", j+k);
    return 0;
}

int square(int a)
{
    return a*a;
}

int cube(int a)
{
    return a*a*a;
}
```

## 2<sup>η</sup> Εργαστηριακή Άσκηση\_b

Το παρακάτω πρόγραμμα διαβάζει δυο θετικούς ακέραιους αριθμούς και υπολογίζει το ΜΚΔ (σύμφωνα με τον αλγόριθμο του Ευκλείδη), με τη χρήση της συνάρτησης *gcd*.

Συμπληρώστε τα κενά.

```
#include <stdio.h>

int gcd(int a, int b);

int main()
{
    int num1, num2;

    do
    {
        printf("Enter the first number: ");
        scanf("%d", &num1);

        printf("Enter the second number (equal or less than the first one): ");
        scanf("%d", &num2);
    } while((num2 > num1) || (num1 <= 0) || (num2 <= 0));
}
```

```

    printf("GCD of %d and %d is %d\n", num1, num2, gcd(num1,num2));
    return 0;
}

int gcd(int a, int b)
{
    int r;

    while(1) /* αόριστες επαναλήψεις */
    {
        r = a%b;
        if(r == 0)
            return b;
        else /* Σύμφωνα με τον αλγόριθμο, διαιρούμε τον b με τον r, οπότε
αλλάζουμε τις τιμές των a και b αντίστοιχα. */
        {
            a = b;
            b = r;
        }
    }
}

```

### 3<sup>η</sup> Εργαστηριακή Άσκηση

Να δημιουργήσετε πρόγραμμα που θα τυπώνει τα στοιχεία ενός δισδιάστατου πίνακα *array* ακεραίων  $4 \times 4$ . Δώστε αρχικές τιμές κατευθείαν από τον ορισμό και στη συνέχεια υπολογίστε και τυπώστε το άθροισμα κάθε γραμμής σε νέο πίνακα *line* και το άθροισμα κάθε στήλης σε νέο πίνακα *row*, με τη χρήση συναρτήσεων.

*Ενδεικτική λύση χωρίς συναρτήσεις:*

```

#include <stdio.h>
#define SIZE 4 /* μέγιστο πλήθος στοιχείων για πίνακα */

int main(void)
{
    /* αρχικοποίηση πίνακα */
    int a[SIZE][SIZE] = { {6, 7, 9, 1},
                          {4, 2, 3, 7},
                          {2, 9, 9, 6},
                          {1, 2, 3, 4} };

    int i, j;
    int line[4] = { 0 }; /* ισοδύναμο με {0,0,0,0} */
    int row[4] = { 0 };

    /* εκτύπωση πίνακα */
    printf("array\n");
    for(i = 0; i < SIZE; i++)
    {
        for(j = 0; j < SIZE; j++)
            printf("%d", a[i][j]);
        printf("\n");
    }

    /* πίνακας line */
    printf("sum of each line\n");
    for(i = 0; i < SIZE; i++)
    {
        for(j = 0; j < SIZE; j++)
            line[i] += a[i][j];
    }
}

```

```

        printf("%d\n", line[i]);
    }

    /* πίνακας row */
    printf("sum of each row\n");
    for(j = 0; j < SIZE; j++)
    {
        for(i = 0; i < SIZE; i++)
            row[j] += a[i][j];
        printf("%d ", row[j]);
    }

    return 0;
}

```

*Ενδεικτική λύση με συναρτήσεις:*

```

#include <stdio.h>
#define SIZE 4 /* μέγιστο πλήθος στοιχείων για πίνακα */

/* πρωτότυπα συναρτήσεων */
void printArray(int a[][SIZE], int size); /* two dimension array to a function */
void arrayLine(int line[], int size, int a[][SIZE]);
void arrayRow(int row[], int size, int a[][SIZE]);

int main(void)
{
    /* αρχικοποίηση πινάκων */
    int a[SIZE][SIZE] = { {6, 7, 9, 1},
                          {4, 2, 3, 7},
                          {2, 9, 9, 6},
                          {1, 2, 3, 4} };

    int line[SIZE] = { 0 }; /* μηδενισμός πίνακα line */
    int row[SIZE] = { 0 }; /* μηδενισμός πίνακα row */

    /* call of functions */
    printArray(a, SIZE);
    arrayLine(line, SIZE, a);
    arrayRow(row, SIZE, a);

    return 0;
} /* end main */

/* printArray function */
void printArray(int a[][SIZE], int size)
{
    int i, j;
    printf("array\n");
    for(i = 0; i < SIZE; i++)
    {
        for(j = 0; j < SIZE; j++)
            printf("%d", a[i][j]);
        printf("\n");
    }
} /* end of function printArray */

/* arrayLine function */
void arrayLine(int line[], int size, int a[][SIZE])
{
    int i, j;

```

```

    printf("sum of each line\n");
    for(i = 0; i < SIZE; i ++)
    {
        for(j = 0; j < SIZE; j ++)
            line[i] += a[i][j];
        printf("%d\n", line[i]);
    }
} /* end of function arrayLine */

/* arrayRow function */
void arrayRow(int row[], int size, int a[][SIZE])
{
    int i, j;
    printf("sum of each row\n");
    for(j = 0; j < SIZE; j ++)
    {
        for(i = 0; i < SIZE; i ++)
            row[j] += a[i][j];
        printf("%d ", row[j]);
    }
} /* end of function arrayRow */

```

#### 4<sup>η</sup> Εργαστηριακή Άσκηση

Να δημιουργήσετε πρόγραμμα για τον υπολογισμό του παραγοντικού ενός δοσμένου φυσικού αριθμού  $n$ . Να χρησιμοποιηθεί συνάρτηση και με αναδρομή.

*Ενδεικτικές λύσεις:*

```

/* factorial using for loop */
#include <stdio.h>

int main()
{
    int i, num, fact = 1;

    printf("Enter a number to calculate it's factorial: ");
    scanf("%d", &num);

    for (i = 1; i <= num; i++)
        fact *= i;

    printf("n! = %d\n", fact);

    return 0;
}

```

```

/* factorial using function */
#include <stdio.h>

long factorial(int); // τύπου long για μεγάλους αριθμούς

int main()
{
    int number;
    long fact = 1;

    printf("Enter a number to calculate it's factorial: ");
}

```

```

scanf("%d", &number);

printf("%d! = %ld\n", number, factorial(number));

return 0;
}

long factorial(int n)
{
    int i;
    long result = 1;

    for (i = 1; i <= n; i++)
        result *= i;

    return result;
}

/* factorial using recursion */
#include<stdio.h>

long factorial(int);

int main()
{
    int num;
    long f;

    printf("Enter a number to find factorial: ");
    scanf("%d", &num);

    if (num < 0)
        printf("Negative numbers are not allowed.\n");
    else
    {
        f = factorial(num);
        printf("%d! = %ld\n", num, f);
    }

    return 0;
}

long factorial(int n)
{
    if (n == 0)
        return 1;
    else
        return(n * factorial(n-1));
}

```