

Μάθημα: Γλώσσες Προγραμματισμού
Λύσεις των θεμάτων της εξέτασης της περιόδου Ιουνίου

Λύση 1ου θέματος

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

double mt(double x[], int n)
{
    int i;
    double sum;

    sum = 0.0;

    for (i=0; i < n; i++)
        sum += x[i];

    return (sum / n);
}

double SD(double x[], int n)
{
    int i;

    double sum, term, mesitimi, result;

    if (n < 2)
        return (-1.0);

    mesitimi = mt(x,n);

    sum = 0.0;
    for (i=0; i < n; i++) {
        term = mesitimi - x[i];
        sum += term * term;
    }

    result = sum/(n*(n-1));
    result = sqrt(result);
    return result;
}

/* Για τον έλεγχο της παραπάνω συνάρτησης. ΔΕΝ απαιτείται από την
εκφώνηση */

main()
{
    double y[] = {1.0, 2.0, 3.0};

    printf("Typiki apoklisi :%g\n", SD(y,3));
}
```

Λύση 2ου θέματος

```
#include <stdio.h>

#include "genlib.h"
#include "simpio.h"

/* Endeiktikes times gia ta N kai M */

#define N 3
#define M 2

main()
{
    double a[N][M], b[M], c[N];

    double sum;
    int i, j, k;

    /* Είσοδος των στοιχείων του πίνακα a */
    for (i = 0; i < N; i++)
        for (j = 0; j < M; j++) {
            printf("a[%d][%d] = ? ", i, j);
            a[i][j] = GetReal();
            /* Εναλλακτικά
            scanf("&f", &a[i][j]);
            */
        }

    /* Είσοδος των στοιχείων του πίνακα b */
    for (j = 0; j < M; j++) {
        printf("b[%d] = ? ", j);
        b[j] = GetReal();
        /* Εναλλακτικά
        scanf("&f", &b[j]);
        */
    }

    /* Υπολογισμός των στοιχείων του πίνακα c */
    for (i = 0; i < N; i++) {
        sum = 0.0;
        for (j = 0; j < M; j++)
            sum += a[i][j]* b[j];

        c[i] = sum;
    }

    /*
    * Εκτύπωση των στοιχείων των πινάκων (χωρίς τη χρήση
    * συναρτήσεων)
    */

    for (i = 0; i < N; i++) {
```

```

        for (j = 0; j < M; j++)
            printf("%g ", a[i][j]);
        printf("\n");
    }
    printf("\n");

    for (j = 0; j < M; j++)
        printf("%g ", b[j]);

    printf("\n\n");

    for (j = 0; j < N; j++)
        printf("%g ", c[j]);

    printf("\n");
}

```

Λύση 3ου θέματος

α)

```

double LeastPositive(double a[], int n)
{
    int i;
    double least;

    least = a[0];

    for (i=1; i < n; i++)
        if (a[i] < least && a[i] > 0)
            least = a[i];

    return least;
}

```

β)

```

#include "genlib.h"

bool sortIsOK(int array[], int n)
{
    int i;

    /* Μη επιτρεπτό μέγεθος πίνακα */
    if (n < 0)
        return FALSE;

    for (i = 1; i < n; i++)
        if (array[i-1] < array[i])
            return FALSE;

    return TRUE;
}

```

/* Για τον έλεγχο της παραπάνω συνάρτησης. ΔΕΝ απαιτείται από την

```

    * εκφώνηση
    */
main()
{
    int x[] = {4,3,3,2,2,1,0};

    printf("%d\n", sortIsOK(x,7));
}

```

Λύση 4ου θέματος

```

main()

{

    int x, y, z; /*Δήλωση μεταβλητών */

    int p, q; /* Δήλωση δεικτών */

    x = 1; y = 2; /* Ανάθεση τιμών στις μεταβλητές x, y */

    p = &x; q = &y; /* Ο δείκτης p δείχνει στη διεύθυνση της
μεταβλητής x και ο q στη διεύθυνση της y */

    z = x; /* Ανάθεση στη μεταβλητή z της τιμής της μεταβλητής
x */

    *p = *q; /* Το περιεχόμενο της διεύθυνσης στην οποία
δείχνει ο δείκτης x (δηλαδή το περιεχόμενο της μεταβλητής x)
γίνεται ίσο με το περιεχόμενο της διεύθυνσης που δείχνει ο δείκτης
q (δηλαδή το περιεχόμενο της μεταβλητής y) */

    *q = z; /* Το περιεχόμενο της διεύθυνσης στην οποία δείχνει
ο δείκτης q γίνεται ίσο με το περιεχόμενο της μεταβλητής z */

    /* Αποτέλεσμα της εκτέλεσης του παραπάνω προγράμματος είναι η
αντιμετάθεση των τιμών των μεταβλητών x και y */

}

```

Λύση 5ου θέματος

```

#include <stdio.h>

int CharFrequency(char s[], char c)
{
    int i, count;

    count = 0;

```

```
        for (i = 0; s[i] != '\0'; i++)
            if ( s[i] == c)
                count++;

    return count;
}

/* Για τον έλεγχο της παραπάνω συνάρτησης. ΔΕΝ απαιτείται από την
 * εκφώνηση
 */
main()
{
    printf("%d\n", CharFrequency("Another green world", 'e'));
}
```