

Άσκηση 1

Να γραφεί πρόγραμμα που να δημιουργεί τυχαίο αριθμό στο διάστημα [0, 100] και να καλεί τον χρήστη να μαντέψει τον αριθμό με 5 το πολύ προσπάθειες. Στην περίπτωση εύρεσης του αριθμού, το πρόγραμμα θα εμφανίζει το πλήθος των προσπαθειών ενώ στην περίπτωση αποτυχίας θα εμφανίζει κατάλληλο μήνυμα.

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

#define YES 1
#define NO 0
#define MAX_GUESSES 5
int main(void)
{
    int num, nr_guesses, guess = -1, found, cont;
    srand(time(NULL)); /* Initializes random number generator */
    do
    {
        num = rand()%101; /* rand() returns int in [0, RAND_MAX] */
        nr_guesses = 0;
        found = NO;
        while (found == NO && nr_guesses < MAX_GUESSES) /* ή !found */
        {
            printf("Guess a number between 0 and 100.\n");
            scanf("%d", &guess);
            nr_guesses++;
            if (num == guess)
                { printf("BINGO!\n"); found = YES; }
            else
                if (num < guess)
                    printf("High guess!\n");
                else
                    printf("Low guess!\n");
        }
        if (found)
            printf("Congratulations! Found in %d guesses.\n", nr_guesses);
        else
            printf("Sorry, next time!\n");

        printf("Continue with another number? (1 for YES, 0 for NO): \n");
        scanf("%d", &cont);
    }
    while (cont == YES); /* ή while (cont) */
    return 0;
}
```

Άσκηση 2

Να γραφεί πρόγραμμα που να υπολογίζει τις ρίζες της β'θμιας εξίσωσης $ax^2 + bx + \gamma = 0$, όπου $a \neq 0$.

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

int main(void)
{
    float a, b, c, x1, x2, D;
    /* float tol = 1.0e-6; */

    printf("Enter coefficients a, b and c: ");
    scanf("%f%f%f",&a, &b, &c);

    D = b*b - 4*a*c;    /* Diakrinousa */

    if (D>0)    /* if (D>tol) */
    {
        x1 = (-b + sqrt(D))/(2*a);
        x2 = (-b - sqrt(D))/2/a;
        printf("Two real roots:  x1 = %f, x2 = %f\n",x1,x2);
    }
}
```

```
else if (D == 0)    /* else if (D>-tol && D<tol) */
    printf("Double real root:  x1 = x2 = %f\n",-b/2/a);
else    /* Complex roots */
{
    printf("Complex root x1 = %f + j %f\n",
        -b/2/a,sqrt(-D)/2/a);
    printf("Complex root x2 = %f - j %f\n",
        -b/2/a,sqrt(-D)/2/a);
}

return 0;
}
```

Άσκηση 3

Τι θα εμφανίσει το παρακάτω πρόγραμμα;

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

    for(i=0, j=6; (i < j) || (i = j-2); i++, --j)
        printf("i = %d, j = %d\n",i,j);

    printf("i = %d, j = %d\n",i,j);
    return 0;
}
i = 0, j = 6
i = 1, j = 5
i = 2, j = 4
i = 1, j = 3
i = 0, j = 2
```

Άσκηση 4

Να γραφεί ένα πρόγραμμα το οποίο να διαβάζει δύο αριθμούς και έναν τελεστή και να προσομοιώνει μια απλή αριθμομηχανή που θα εκτελεί την πράξη και θα εμφανίζει το αποτέλεσμα. Να χρησιμοποιηθεί η switch και να ερωτάται ο χρήστης για την ακρίβεια σε δεκαδικά ψηφία.

```
#include <stdio.h>
#include <stdlib.h> /* edw vrisketai h exit() */

int main(void)
{
    int n;
    char op;
    float x,y,res=0.0; /* res = x <op> y */

    printf("MIA APLH ARITHMOMHXHNH");
    printf("-----\n");

    printf("Dose ton prwto arithmo: ");
    scanf("%f",&x);

    /* Adeiase to input buffer apo to newline */
    while (getchar() != '\n');

    printf("\Dose enan telesth apo {+ - * /}: ");
    scanf("%c",&op);
```

```

printf("\nDose ton deytero arithmo: ");
scanf("%f", &y);

switch (op)
{
    case '+': res = x + y; break;
    case '-': res = x - y; break;
    case '*': res = x * y; break;
    case '/': if (y!=0.0F) res = x / y; break;
    default: printf("\n\nInvalid operator!"); exit(0);
}

if (op == '/' && y == 0.0F && x != 0.0F)
    printf("\n\nAdynath diairesh\n");
else if (op == '/' && y == 0.0F && x == 0.0F)
    printf("\n\nAprosdioristh morfh 0/0\n");
else
{
    printf("\n\nTi akriveia se dekadika psifia theleis?\n\n");
    scanf("%d", &n);

    printf("\n%.*f %c %.*f = %.*f\n",n,x,op,n,y,n,res);
}

system("pause"); /* ή getch(); */
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("Dwse enan akeraio: ");
    scanf("%d", &a);
    do
    {
        printf("\nDwse enan akeraio < %d : ",a);
        scanf("%d", &b);
    } while(a <= b);
    do
    {
        printf("\n... kai enan akeraio < %d : ",b);
        scanf("%d", &c);
    } while(b <= c);

    printf("\nOi treis arithmoi einai: %d %d %d\n", a, b, c);

    return 0;
}

```

Άσκηση 6

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 εισαγάγει 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: %.10f\n", sum);

    return 0;
}
```

Άσκηση 7

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 = 0.0; 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=0.0; 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

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 7 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 διαβάσετε 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)
```