ΕΘΝΙΚΟ ΚΑΙ
ΚΑΠΟΔΙΣΤΡΙΑΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

Α.Μ: *201100079*	**Επώνυμο:** *Κορομηλάς*	**Όνομα:** *Χρήστος*
Ημ/νία(Ημέρα,Ώρα) εκτέλεσης της άσκησης: *30/03/2022 (Τετάρτη, 15:30-18:00)*		
Ημ/νία παράδοσης της άσκησης: *13/04/2022*		

MATLAB

Εισαγωγικές ασκήσεις στη Matlab με σκοπό την εκμάθηση και εξοικείωση με το περιβάλλον της.

➢ ***Άσκηση 1η :***

Ακολουθεί το πρόγραμμα $\boldsymbol{deyterobathmia.m}$ στη λειτουργική του μορφή:

```
%We solve the Equation ax^2+bx+c=0

clear;
clc;
close all;

%Parametroi Programmatos

a = 1;
b = 1;
c = 1;

D = sqrt(b^2-4*a*c);

p1 = (-b+D)/(2*a);
p2 = (-b-D)/(2*a);

p1r = real(p1);
p1i = imag(p1);

p2r = real(p2);
p2i = imag(p2);

fprintf('p1=%.4f %.4fi\np2=%.4f %.4fi\n',p1r,p1i,p2r,p2i);
```

Το πρόγραμμα πραγματοποιεί την επίλυση μίας δευτεροβάθμιας εξίσωσης, όπως αναγράφεται στο εργαστηριακό οδηγό και όταν γίνει run επιστρέφει:

p1=-0.5000 0.8660i

p2=-0.5000 -0.8660i

➢ ***<u>Άσκηση 2η :</u>***

Ακολουθεί το πρόγραμμα $\boldsymbol{deyterobathmia2.m}$ στην λειτουργική του μορφή:

```
%We solve the Equation a1x^2 + a2x^2 + a3x = 0

clear;
clc;
close all;

for i=1:1:3
    fprintf('a%d=',i);
    a(i)=input('');
    fprintf('\n');
end

res = roots([a(1) a(2) a(3)])
```

Το πρόγραμμα αρχικά εκκινεί ένα $\boldsymbol{for\ loop}$ κατά το οποίο ζητάει και δέχεται τους τρείς συντελεστές του πολυωνύμου δεύτερου βαθμού. Έπειτα αφού κλείσει η επανάληψη επιστρέφει στην μεταβλητή $\boldsymbol{res}$ το αποτέλεσμα της συνάρτησης $\boldsymbol{roots}$ η οποία δέχεται τους πολυωνυμικούς συντελεστές και να επιστρέφει τις λύσεις του πολυωνύμου.

➢ ***Άσκηση 3η :***

Ακολουθεί το πρόγραμμα: $\boldsymbol{askisi3.m}$ στην λειτουργική του μορφή:

```
%Loop for_end

clear;
clc;
close all;

a0 = 1;

an = a0;

for (i=1:1:10)
    %fprintf('an=%.2f\n',an); gia na doume endiamesa ta apotelesmata
    an = - 0.8*an + 2;
end

fprintf('an=%.2f\n',an);
```

Απάντηση ερωτήσεων:

Ερώτημα 1: Το πρόγραμμα στην έξοδό του τυπώνει:
an=1.10

Ερώτημα 2: Εάν τοποθετήσουμε την εντολή $\boldsymbol{fprintf}$ μέσα στην επανάληψη $\boldsymbol{for}$ τότε θα μπορούσαμε εύκολα να κάνουμε το πρόγραμμά μας να τυπώνει κάθε αλλαγή που γίνεται στη μεταβλητή $\boldsymbol{an}$

Ερώτημα 3: Το πρόγραμμα αρχικά ορίζει ως αρχική τιμή $\boldsymbol{a0}$ ίση με το $\mathbf{1}$ και αρχικοποιεί τη μεταβλητή $\boldsymbol{an}$ θέτοντας την ίση με την $\boldsymbol{a0}$. Στη συνέχεια αρχίζει ένα $\boldsymbol{for\ loop}$ που θα κάνει $\mathbf{10}$ επαναλήψεις στο οποίο η τιμή an αντικαθίσταται από την σχέση $\boldsymbol{-0.8 * an + 2}$ όπου an η τιμή της μεταβλητής στη προηγούμενη επανάληψη. Τέλος αφού τελειώσουν όλες οι επαναλήψεις τυπώνεται στην οθόνη το τελικό αποτέλεσμα της $\boldsymbol{an}$ με ακρίβεια δύο δεκαδικών ψηφίων.

Ερώτημα 4: Η γραμμή 5 και 6 αντίστοιχα:

5η `an = a0;`

6η `for (i=1:1:10)`

Η 5η γραμμή θέτει την αρχική τιμή που θα έχει το an καθώς θα ξεκινάει η επανάληψη , ενώ η 6η γραμμή περιέχει τις τρεις απαραίτητες συνθήκες που χρειάζεται για να πραγματοποιηθεί η επανάληψη, οι οποίες είναι αντίστοιχα αρχική τιμή, βήμα αύξησης, τελική τιμή.

➢ ***Άσκηση 4η :***

Ακολουθεί το πρόγραμμα: $\boldsymbol{askisi4.m}$ στην λειτουργική του μορφή:

```
%Loop while_end

clear;
clc;
close all;

i = 0;
a0 = 1;
an = a0;

while (an<1.110 || an>1.118) % || is 'or' operator
    i = i+1;
    an = - 0.8*an+2;
end

fprintf('an=%.2f\t i=%.0f\n',an,i);
```

Απάντηση ερωτήσεων:

Ερώτημα 1: Το πρόγραμμα καθώς το τρέξουμε τυπώνει:

an=1.12 i=13

Ερώτημα 2: Αρχικά το πρόγραμμα αρχικοποιεί τρεις μεταβλητές την $\boldsymbol{i=0}$, $\boldsymbol{a0=1}$ και την $\boldsymbol{an=a0}$. Στη συνέχεια το πρόγραμμα εκκινεί μία $\boldsymbol{while}$ επανάληψη κατά την οποία πρέπει να ισχύει είτε η συνθήκη $\boldsymbol{an<1.110}$ είτε η συνθήκη $\boldsymbol{an>1.118}$, για να συνεχίσει η επανάληψη. Η μεταβλητή $\boldsymbol{i}$ λειτουργεί ως μετρητής των πόσων επαναλήψεων έχουν πραγματοποιηθεί ενώ σε κάθε επανάληψη η μεταβλητή $\boldsymbol{an}$ παίρνει την τιμή $\boldsymbol{-0.8*an+2}$ όπου η $\boldsymbol{an}$ μέσα σε αυτή τη σχέση έχει τη τιμή που είχε στη περασμένη επανάληψη. Τέλος , αφού τερματίσει το $\boldsymbol{while}$ τυπώνεται η τελική τιμή του $\boldsymbol{an}$ με ακρίβεια δύο δεκαδικών ψηφίων και ο αριθμός των επαναλήψεων που έχουν πραγματοποιηθεί (δηλαδή η τιμή που έχει η μεταβλητή $\boldsymbol{i}$).

Ερώτημα 3: Η γραμμή 5 και 6 αντίστοιχα:

5η `i = 0;`

6η `a0 = 1;`

Η 5η γραμμή θέτει την μεταβλητή $\boldsymbol{i=0}$ η οποία λειτουργεί ως μετρητής θέτοντας την ίση με το $\boldsymbol{0}$ πρακτικά μηδενίζει τον μετρητή, έτσι ώστε να μπορεί να μετρήσει σωστά τον αριθμό επαναλήψεων που θα ακολουθήσουν. Ακόμη, η γραμμή 6η αρχικοποεί τη μεταβλητή $\boldsymbol{a0=1}$ η οποία θα αποτελέσει την αρχική τιμή της μεταβλητής $\boldsymbol{an}$ πριν ξεκινήσει η επανάληψη.

Ερώτημα 4: Σύμφωνα με τον εργαστηριακό μας οδηγό η γραμμή που θα έπρεπε να αλλάξει είναι η γραμμή 9:

από

```
while(an<1.110 || an>1.118)
```

σε

```
while(an<=1.11 || an>=1.115)
```

➢ *Άσκηση 5η :*

Ακολουθεί το πρόγραμμα: $\boldsymbol{askisi5.m}$ στην λειτουργική του μορφή:

```
%Ploting example

clear;
clc;
close all;

a0 = 1;
an = a0;
epsilon = 1e-3;

res = 1;
n = 1;

while (res>epsilon)
    an_in = an;
    res1(n) = an;
    x(n) = n;
    an = -0.8*an + 2;
    n = n+1;
    res = abs(abs(an_in)-abs(an));
end

fprintf('an=%.2f\nn=%d\n',an,n);

plot(x,res1);
drawnow;
```

Απάντηση ερωτήσεων:

Ερώτημα 1: Το πρόγραμμα πραγματοποιεί την ακολουθία:

$$\boldsymbol{a_{n+1} = -0.8 * a_n + 2}$$

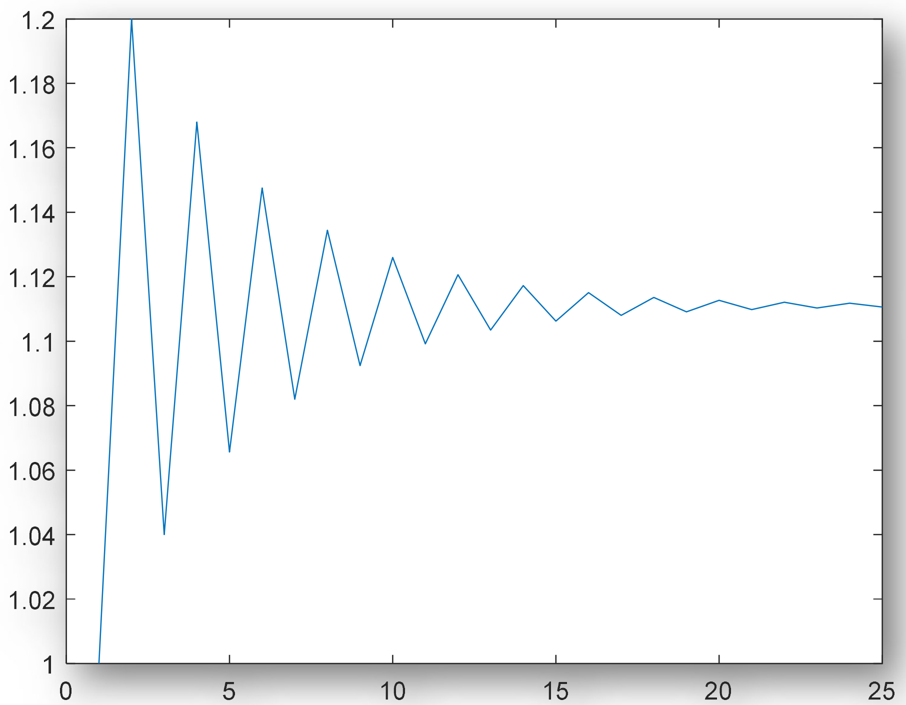
έως ότου το απόλυτο σφάλμα του $\boldsymbol{a_{n+1} - a_n}$ (δηλαδή δυο διαδοχικών τιμών) να είναι μικρότερο από το $\boldsymbol{10^{-3}}$. Τέλος αφού πραγματοποιηθούν όλοι οι υπολογισμοί το πρόγραμμα τυπώνει τη τιμή του τελευταίου $\boldsymbol{an}$ που υπολογίστηκε, τον αριθμό των επαναλήψεων που πραγματοποιήθηκαν και το διάγραμμα που βρίσκεται στο επόμενο ερώτημα.

Ερώτημα 2: Καθώς τρέχουμε το πρόγραμμα λαμβάνουμε την ακόλουθη έξοδο:

an=1.11

n=26

Όπου $\boldsymbol{a_n}$ η τιμή που υπολογίστηκε στη τελευταία επανάληψη και n ο αριθμός των επαναλήψεων που πραγματοποιήθηκαν. Ακόμη τυπώνεται το ακόλουθο διάγραμμα όπου στον άξονα $\boldsymbol{x}$ αναγράφεται ο αριθμός επανάληψης, ενώ στον άξονα $\boldsymbol{y}$ αναγράφεται η υπολογιζόμενη τιμή an στο αντίστοιχο σημείο $\boldsymbol{n}$.



***Ερώτημα 3**:* Το πρόγραμμα καθώς ξεκινάει αρχικοποιεί τη μεταβλητή $\boldsymbol{a_n}$ με τη τιμή $\mathbf{1}$, ορίζει το σφάλμα με το οποίο θα γίνεται ο έλεγχος τερματισμού δηλαδή το $\boldsymbol{epsilon} = \mathbf{1}\boldsymbol{e} - \mathbf{3}$ και αρχικοποιεί το απόλυτο σφάλμα δύο διαδοχικών τιμών $\boldsymbol{res}$ με τη τιμή $\mathbf{1}$ καθώς και το βήμα επανάληψης $\boldsymbol{n}$. Έπειτα ξεκινάει μία $\boldsymbol{while}$ επανάληψη κατά την οποία θέτουμε $\boldsymbol{an_{in}} = \boldsymbol{a_n}$ δηλαδή ως$\boldsymbol{an_{in}}$ τη προηγούμενη τιμή $\boldsymbol{a_n}$, αποθηκεύουμε τη τιμή του $\boldsymbol{a_n}$ στο array $\boldsymbol{res}\mathbf{1}(\boldsymbol{n})$ (με το επόμενο βήμα της ακολουθίας που αναφέραμε στο ερώτημα 1, ανεβάζουμε κατά $\mathbf{1}$ το μετρητή επανάληψης και υπολογίζουμε το απόλυτο σφάλμα των δύο διαδοχικών τιμών. Εάν το απόλυτο σφάλμα είναι μικρότερο ή ίσο με τη τιμή $\mathbf{10^{-3}}$ τότε η επανάληψη τερματίζει, διαφορετικά η επανάληψη συνεχίζεται. Μετά το τερματισμό της, τυπώνονται οι έξοδοι που αναφέρθηκαν στα προηγούμενα ερωτήματα.

Η συνάρτηση $\boldsymbol{plot}(\boldsymbol{x}, \boldsymbol{y})$ πραγματοποιεί γραφικές αναπαραστάσεις παίρνοντας δεδομένα για τον $\boldsymbol{x}$ και τον $\boldsymbol{y}$ άξονα αντίστοιχα, ενώ η συνάρτηση $\boldsymbol{drawnow}$ ανανεώνει τα δεδομένα που πρόκειται να εμφανιστούν μέσα στα παράθυρα γραφικών παραστάσεων.

επαναλήψεων που θα ακολουθήσουν. Ακόμη, η γραμμή 6η αρχικοποιεί τη μεταβλητή $\boldsymbol{a}\mathbf{0} = \mathbf{1}$ η οποία θα αποτελέσει την αρχική τιμή της μεταβλητής $\boldsymbol{an}$ πριν ξεκινήσει η επανάληψη.

➢ *Άσκηση 6η :*

Ακολουθεί το πρόγραμμα: $\boldsymbol{grammiko1.m}$ στην λειτουργική του μορφή:

```
%We solve the linear 3X3 system:
%a11*x + a12*y + a13*z = a14
%a21*x + a22*y + a23*z = a24
%a31*x + a32*y + a33*z = a34
clear;
clc;
close all;
for (i=1:1:3)
    for (j=1:1:4)
        fprintf('a%d%d=',i,j);
        a(i,j)=input('');
        fprintf('\n');
    end
end
for (i=1:1:3)
    for (j=1:1:3)
        Dxyz1(i,j) = a(i,j);
    end
end
for (i=1:1:3)
    for (j=1:1:4)
        if (j==4)
            k=1;
        else
            k=j;
        end
        Dx1(i,k) = a(i,j);
    end
end
for (i=1:1:3)
    for (j=1:1:4)
        if (j==4)
            k=2;
        else
            k=j;
        end
        Dy1(i,k) = a(i,j);
    end
end
for (i=1:1:3)
    for (j=1:1:4)
        if (j==4)
            k=3;
        else
            k=j;
        end
        Dz1(i,k) = a(i,j);
    end
end
Dxyz = det(Dxyz1);
Dx = det(Dx1); Dy = det(Dy1); Dz = det(Dz1);
 x = Dx/Dxyz;   y = Dy/Dxyz;   z = Dz/Dxyz;
fprintf('x=%.4f\ny=%.4f\nz=%.4f\n',x,y,z);
```

Απάντηση ερωτήσεων:

***Ερώτημα 1**:* Καθώς τρέξουμε το πρόγραμμα μας το πρώτο πράγμα που θα εκτυπωθεί θα η ένδειξη: $\boldsymbol{a11 =}$

Η οποία μας επιτρέπει να δώσουμε τη τιμή $\boldsymbol{\alpha_{11}}$ του πίνακα $\boldsymbol{A}$ δεδομένου ότι επιλύουμε σύστημα μορφής: $\boldsymbol{Ax = b}$.

***Ερώτημα 2**:* Ουσιαστικά το πρόγραμμα εκτελεί επίλυση συστήματος εξισώσεων $\boldsymbol{3X3}$ με τη μέθοδο $\boldsymbol{Cramer}$.

***Ερώτημα 3**:* Αφού περάσουμε όλα τα δεδομένα μέσα στο πρόγραμμα και το αφήσουμε να κάνει τους τελικούς υπολογισμούς του η έξοδος που θα πάρουμε θα έχει τη μορφή:

x=##.####

y=##.####

z=##.####

(##.#### λύση για την εκάστοτε μεταβλητή)

***Ερώτημα 4**:* Το πρόγραμμα μας χρησιμοποιεί 5 βασικές επαναλήψεις για να καταφέρει να τρέξει. Καθώς ξεκινάει το πρόγραμμα η πρώτη επανάληψη υπάρχει για να εισάγει ο χρήστης όλες τις τιμές του πίνακα $\boldsymbol{A}$ και του πίνακα $\boldsymbol{b}$ καθώς χωρίς αυτές δεν μπορεί να υπάρξει κανένας υπολογισμός. Έπειτα στη δεύτερη επανάληψη, αξιοποιούνται τα δεδομένα που έδωσε ο χρήστης για τη κατασκευή του πίνακα $\boldsymbol{A}$ ο οποίος αναγράφεται ως $\boldsymbol{Dxyz1}$. Στις επόμενες 3 επαναλήψεις δημιουργούμε τους πίνακες $\boldsymbol{Dx1}$, $\boldsymbol{Dy1}$, $\boldsymbol{Dz1}$ οι οποίοι είναι ο πίνακας $\boldsymbol{Dxyz1}$ στον οποίο έχουμε αντικαταστήσει τη πρώτη δεύτερη και τρίτη στήλη αντίστοιχα με το πίνακα /διάνυσμα $\boldsymbol{b}$ (ο οποίος δεν έχει σχηματιστεί αλλά υπάρχει μέσα στα δεδομένα του χρήστη που έχουν δοθεί στο $\boldsymbol{a}$). Αφού τερματίσουν όλες οι επαναλήψεις υπολογίζονται οι ορίζουσες των πινάκων που σχηματίστηκαν στις $\boldsymbol{for}$ επαναλήψεις και επιστρέφονται στις αντίστοιχες μεταβλητές τους όπως βλέπουμε ακολούθως:

```
Dxyz = det(Dxyz1);
Dx = det(Dx1);
Dy = det(Dy1);
Dz = det(Dz1);
```

Συνεπώς, τώρα μένει μόνο να γίνουν οι υπολογισμοί της μεθόδου του $\boldsymbol{Cramer}$ όπως:

```
x = Dx/Dxyz;
y = Dy/Dxyz;
z = Dz/Dxyz;
```

Τελειώνοντας το πρόγραμμα τυπώνει τις μεταβλητές $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ που περιέχουν την επίλυση του συστήματος, όπως αναγράφεται στην απάντηση του ερωτήματος 3 .

Για να υπάρξουν επαναλήψεις όπου σχηματίζουμε πίνακες χρησιμοποιούμε διπλή επανάληψη for, έτσι ώστε να μπορούμε να γράψουμε σε δύο διαστάσεις.

- ***Άσκηση 7η :***

Ακολουθεί το πρόγραμμα: $\boldsymbol{grammiko2.m}$ στην λειτουργική του μορφή:

```
%We solve a linear NXN system:

clear;
clc;
close all;

fprintf('please insert the number N: ');
n = input('');

for (i=1:1:n)
    for (j=1:1:n+1)
        fprintf('a%d%d=',i,j);
        a(i,j)=input('');
        fprintf('\n');
    end
end
for (i=1:1:n)
    for (j=1:1:n)
        Dxyz1(i,j) = a(i,j);
    end
end

Dxyz = det(Dxyz1);

for (m=1:1:n)
    for (i=1:1:n)
        for (j=1:1:n+1)
            if (j==n+1)
                k=m;
            else
                k=j;
            end
            Dx1(i,k) = a(i,j);
        end
    end
    Dx=det(Dx1);
    x=Dx/Dxyz;
    fprintf('The answer for x_%d=%.4f\n',m,x);
end
```

➢ ***Άσκηση 8η :***

Ακολουθεί το πρόγραμμα: $\boldsymbol{grammiko3.m}$ στην λειτουργική του μορφή:

```
%%% We solve the linear 3X3 system with matrix:
%%% a11*x + a12*y + a13*z = a14
%%% a21*x + a22*y + a23*z = a24
%%% a31*x + a32*y + a33*z = a34

clear;
clc;
close all;

li = 3;
co = li+1;
[b] = fun1(li,co);

for(i=1:1:li)
    for(j=1:1:co-1)
        Dxyz1(i,j)=b(i,j);
    end
end
for (i=1:1:li)
    for(j=1:1:co)
         if (j==4)
            k=1;
        else
            k=j;
        end
        Dx1(i,k) = b(i,j);
    end
end
for (i=1:1:li)
    for(j=1:1:co)
         if (j==4)
            k=2;
        else
            k=j;
        end
        Dy1(i,k) = b(i,j);
    end
end
for (i=1:1:li)
    for(j=1:1:co)
         if (j==4)
            k=3;
        else
            k=j;
        end
        Dz1(i,k) = b(i,j);
    end
end
fun2(Dxyz1,Dx1,Dy1,Dz1);
```

Ακολουθούν οι 2 συναρτήσεις:$\boldsymbol{fun1.m}$ και $\boldsymbol{fun2.m}$ αντίστοιχα:

```
function [b]=fun1(lin,col);

for (i=1:1:lin)
    for (j=1:1:col)
        fprintf('elmnt%d%d=',i,j);
        b(i,j)=input('');
        fprintf('\n');
    end
end
```

```
function [a]=fun2(Dxyz2,Dx2,Dy2,Dz2);

Dxyz = det(Dxyz2);
Dx = det(Dx2);
Dy = det(Dy2);
Dz = det(Dz2);

x = Dx/Dxyz;
y = Dy/Dxyz;
z = Dz/Dxyz;

fprintf('x=%.4f\ny=%.4f\nz=%.4f\n',x,y,z);
```

Απάντηση ερωτήσεων:

***Ερώτημα 1**:* Μετά τη την εκτέλεση της εντολής εμφανίζεται στην οθόνη μας το ακόλουθο:

```
>> help grammiko1.m

 We solve the linear 3X3 system:

 a11*x + a12*y + a13*z = a14

 a21*x + a22*y + a23*z = a24

 a31*x + a32*y + a33*z = a34


>> help grammiko3.m

 % We solve the linear 3X3 system with matrix:

 % a11*x + a12*y + a13*z = a14

 % a21*x + a22*y + a23*z = a24

 % a31*x + a32*y + a33*z = a34
```

Στη MATLAB τα σχόλια εμφανίζονται με % ή %% παραπάνω % απλά θα εμφανίζονται στο command center της MATLAB χωρίς όμως να επηρεάζουν το κώδικα όταν γίνεται run.

***Ερώτημα 2**: Το πρόγραμμα μας πραγματοποιεί ακριβώς την ίδια διαδικασία που πραγματοποιεί το πρόγραμμα* $\boldsymbol{grammiko1.m}$ *με τη διαφορά πως έχουμε τοποθετήσει δύο λειτουργίες του μέσα σε εξωτερικές συναρτήσεις. Στην συνάρτηση* $\boldsymbol{fun1.m}$ *βρίσκεται το κομμάτι του κώδικα που ζητάει και καταχωρεί τα δεδομένα από το χρήστη σε έναν ενιαίο πίνακα, ενώ στη συνάρτηση* $\boldsymbol{fun2.m}$ *βρίσκεται το κομμάτι του κώδικα που υπολογίζει τις ορίζουσες , πραγματοποιεί τη τελευταία πράξη υπολογισμού στον αλγόριθμο του* $\boldsymbol{Cramer}$ *και τυπώνει τις λύσεις* $\boldsymbol{x, y, z}$ *.*

***Ερώτημα 3**:* Η έξοδος που θα πάρουμε από το πρόγραμμα αφού καταχωρηθούν τα δεδομένα θα είναι της μορφής:

x=##.####

y=##.####

z=##.####

(##.#### λύση για την εκάστοτε μεταβλητή όπως ακριβώς και στο $\boldsymbol{grammiko1.m}$)

***Ερώτημα 4**:* Όπως ακριβώς είχαμε απαντήσει και στην Έκτη Άσκηση, αυτής της εργαστηριακής μας άσκησης , το πρόγραμμα αυτό επιλύει ένα γραμμικό σύστημα εξισώσεων $\boldsymbol{3X3}$ με τη μέθοδο $\boldsymbol{Cramer}$. Λειτουργεί και υπολογίζει όλες τις μεταβλητές με ίδιο τρόπο με αυτού του προγράμματος $\boldsymbol{grammiko1.m}$ όμως κατά την εκκίνηση του, το πρόγραμμα $\boldsymbol{grammiko3.m}$ καλεί μία συνάρτηση, τη fun1.m να τρέξει το κομμάτι του αλγόριθμου που δέχεται δεδομένα από το χρήστη και στο τελείωμα του, το πρόγραμμα καλεί τη συνάρτηση fun2.m για τους τελικούς υπολογισμούς και την εκτύπωση αποτελεσμάτων που είδαμε στα ερωτήματα 2 και 3 αντίστοιχα.

➢ ***Άσκηση 9η :***

Ακολουθεί το πρόγραμμα: $\boldsymbol{pinakas1.m}$ στην λειτουργική του μορφή:

```
%Computation of a weird matrix:
% F_ij = 2*(a_ij)^5 + 3*(b_ij)^3 + 5*(a_ij*b_ij) +1
% A,B: 3X4

clear;
clc;
close all;

li = 3;
co = li+1;

fprintf('\nGive me the elements of the first (A) matrix \n') ;
[A] = fun1(li,co);
fprintf('\nGive me the elements of the second (B) matrix \n') ;
[B] = fun1(li,co);

fprintf('\n Printing Answers:\n');

for(i=1:1:li)
    for(j=1:1:co)
        F(i,j)=2.*A(i,j).^5+3.*B(i,j).^3+5*A(i,j).*B(i,j)+1;
        fprintf('F%d%d=%.4f',i,j,F(i,j));
        fprintf('\n');
    end
end
```

Έγινε χρήση της συνάρτησης $\boldsymbol{fun1.m}$ για τη καταχώρηση των πινάκων από τη προηγούμενη άσκηση.

```
function [b]=fun1(lin,col);

for (i=1:1:lin)
    for (j=1:1:col)
        fprintf('elmnt%d%d=',i,j);
        b(i,j)=input('');
        fprintf('\n');
    end
end
```