

Πρακτικό 4

0. Βγάλτε στυλό και χαρτί για να σχηματίσετε τον σκελετό και την λογική του προγράμματός σας πριν το περάσετε στην οθόνη.

1. Γράψτε προγράμματα script, που να τυπώνουν τους παρακάτω αριθμούς, σχηματίζοντας διαφορετικές εκδοχές χρησιμοποιώντας:
- Την εντολή disp για κάθε αριθμό χωριστά
 - Την εντολή disp αφότου πληκτρολογήσετε όλα τα στοιχεία του διανύσματος
 - Την εντολή disp αφότου σχηματίσετε το διάνυσμα με αυτοματοποιημένο τρόπο
 - Την εντολή disp αφότου σχηματίσετε το διάνυσμα με επαναληπτική διαδικασία

0

50

100

150

200

```
clear
disp(0)
disp(50)
disp(100)
disp(150)
disp(200)
```

```
-----
clear
x=[0 50 100 150 200]';
disp(x)
```

```
-----
clear
x=[0; 50; 100; 150; 200];
disp(x)
```

```
-----
clear
x=[0:50:200]';
disp(x)
```

```
-----
clear
athroisma=0;
x=athroisma;
for i=1:4
    athroisma=athroisma+50;
    x(i+1)=athroisma;
end
disp(x')
```

2. Γράψτε πρόγραμμα script, που να ρωτάει τον χρήστη εάν δεν πρόσεχε στο προηγούμενο μάθημα και, εάν ναι, να εμφανίζει στην οθόνη "I will pay more attention during lectures." 100 φορές. Χρησιμοποιήστε στην συνάρτηση strcmp και δείτε από την βοήθεια του Matlab την διαφορά με την strcmpr.

```

clear;clc;
x=input('Were you talking during class? Answer yes or no');
tf = strcmpi(x,'Yes')
if tf==1
    for i=1:100
        disp('I will pay more attention during lectures')
    end
else
    disp('Keep up the good work, ALWAYS have questions!')
end

```

3. Μετατρέψτε το script της 2^{ης} άσκησης σε πρόγραμμα function,
Είσοδος: το κείμενο 'YES' ή 'NO' (απάντηση στο ερώτημα: *Were you talking during class?*), ανάλογα με την απάντηση.
Έξοδος: το κατάλληλο μήνυμα

```

function answer = ex2function(attention)
    tf = strcmpi(attention,'Yes')
    if tf==1
        answer='I will pay more attention during lectures';
        for i=1:100
            disp('I will pay more attention during lectures')
        end
    else
        answer='Keep up the good work, ALWAYS have questions!';
    end
end

```

Script που καλεί την ex2function

```

clear; clc;
x = input('Were you talking during class? Answer yes or no');
y = ex2function(x);

```

4. Να γράψετε ένα πρόγραμμα script, το οποίο θα διαβάζει από τον χρήστη 10 αριθμούς και θα καλεί ένα function, το οποίο θα έχει ως είσοδο αυτούς τους 10 αριθμούς και θα επιστρέφει σε ξεχωριστές μεταβλητές τις αρνητικές από τις θετικές τιμές σε αύξουσα σειρά.

```

function [xpositive xnegative]=posnegSorter(x)
    xpositive=[];
    xnegative=[];
    for i=1:length(x)
        if x(i)>=0
            xpositive=[xpositive x(i)];
        else
            xnegative=[xnegative x(i)];
        end
    end
end

```

```
xpositive=sort(xpositive);
xnegative=sort(xnegative);
end
```

Script που καλεί την posnegSorter

```
clear; clc;
x=input('Please provide an array of 10 numbers');
[thetika arnhtika]=posnegSorter(x)
```

5. Να υπολογιστεί το παραγοντικό ενός θετικού ακέραιου αριθμού. Το παραγοντικό ενός θετικού ακέραιου αριθμού $N \neq 0$ συμβολίζεται $N!$ και ορίζεται ως το γινόμενο των αριθμών 1, 2, 3, ..., (N-1), N. Αν $N=0$ τότε $N! = 1$.

Παράδειγμα: $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$

Ελέγξτε τα αποτελέσματα του κώδικά σας σύμφωνα με τον παρακάτω πίνακα (Πηγή: <http://en.wikipedia.org/wiki/Factorial>)

N	N!
0	1
1	1
2	2
3	6
4	24
5	120
6	720
7	5 040
8	40 320
9	362 880
10	3 628 800
15	1 307 674 368 000
20	2 432 902 008 176 640 000

Λύση 1 (Με χρήση επανάληψης for)

```
clear;clc;

n=input('Give the number for which you to calculate the factorial');
paragontiko=1;
for i=1:n
    paragontiko=paragontiko*i;
end
fprintf('The value of the factorial is: %d\n',paragontiko)
```

Λύση 2

```
clear;clc;
n=input('Give the number for which you to calculate the factorial');
prod(1:n)
```

6. Μετατρέψτε τον παραπάνω κώδικα σε function,
Είσοδος: ο αριθμός για τον οποίο θέλουμε να υπολογίσουμε το παραγοντικό
Έξοδος: η τιμή του παραγοντικού του αριθμού αυτού

Λύση 1

```
function [y]=paragontiko(n)
    y=1;
    for i=1:n
        y=y*i;
    end
end
```

Λύση 2

```
function [y]=paragontiko(n)
    y=prod(1:n)
end
```

Script που καλεί την paragontiko

```
clear;clc;
n=input('Give the number for which you to calculate the factorial');
x=paragontiko(n);
disp(x)
```

Για το σπίτι:

1. Ως βοηθός του Σερλοκ Χομς προσπαθείτε να διαλευκάνετε ένα φόνο μιας κοπέλας. Σας ζητείται να καταγράψετε για 10 πολίτες στο περιβάλλον του θύματος εάν είχαν τα τρία απαραίτητα στοιχεία: Μέσο, κίνητρο και ευκαιρία.
Αποθηκεύστε σε ξεχωριστές arrays τα στοιχεία που μαζέψατε από τις συνεντεύξεις σας και ξεχωρίστε ποιοί συγκεντρώνουν και τα τρία στοιχεία για να διερευνηθούν περαιτέρω.

Έπειτα τυπώστε ένα μήνυμα με τον αύξοντα αριθμό των πολιτών.

https://en.wikipedia.org/wiki/Means,_motive,_and_opportunity

```
clear;clc;

means= [1 0 0 1 0 1 1 1 0 0];
motive=[0 0 1 1 0 0 1 1 1 1];
oppty= [1 0 1 1 0 1 0 1 0 1];

citizens=means.*motive.*oppty;
suspects=[];
for i=1:length(means)
    if citizens(i)==1
        suspects=[suspects i];
    end
end

disp('The suspect citizens are:')
fprintf('%d\n',suspects)
```

2. Η συνάρτηση sort του MATLAB χρησιμοποιεί έναν αλγόριθμο με το όνομα Quick Sort για να ταξινομήσει N αριθμούς.

Δίνονται N αριθμοί. Να φτιάξετε script στο MATLAB που να ταξινομεί σε αύξουσα τάξη μεγέθους του αριθμούς αυτούς σύμφωνα με την παρακάτω διαδικασία:

Έστω ότι θέλετε να ταξινομήσετε $N = 5$ αριθμούς οι οποίοι είναι αποθηκευμένοι σε μια μονοδιάστατη array A.

Η διαδικασία που πρέπει να ακολουθήσετε είναι:

Βήμα 1:

Συγκρίνουμε το A(1) με το A(2) και εναλλάσσουμε τις τιμές τους μόνο αν $A(1) > A(2)$
Συγκρίνουμε το A(1) με το A(3) και εναλλάσσουμε τις τιμές τους μόνο αν $A(1) > A(3)$
Συγκρίνουμε το A(1) με το A(4) και εναλλάσσουμε τις τιμές τους μόνο αν $A(1) > A(4)$
Συγκρίνουμε το A(1) με το A(5) και εναλλάσσουμε τις τιμές τους μόνο αν $A(1) > A(5)$

Βήμα 2:

Συγκρίνουμε το A(2) με το A(3) και εναλλάσσουμε τις τιμές τους μόνο αν $A(2) > A(3)$
Συγκρίνουμε το A(2) με το A(4) και εναλλάσσουμε τις τιμές τους μόνο αν $A(2) > A(4)$
Συγκρίνουμε το A(2) με το A(5) και εναλλάσσουμε τις τιμές τους μόνο αν $A(2) > A(5)$

Βήμα 3:

Συγκρίνουμε το A(3) με το A(4) και εναλλάσσουμε τις τιμές τους μόνο αν $A(3) > A(4)$
Συγκρίνουμε το A(3) με το A(5) και εναλλάσσουμε τις τιμές τους μόνο αν $A(3) > A(5)$

Βήμα 4:

Συγκρίνουμε το A(4) με το A(5) και εναλλάσσουμε τις τιμές τους μόνο αν $A(4) > A(5)$

```
clear; clc;

a=[-5 3 0 -6 1];
disp('The input vector is:')
disp(a)

n=length(a);
for i=1:n-1
    for j=i+1:n
        if a(i)>a(j)
            k=a(i);
            a(i)=a(j);
            a(j)=k;
        end
    end
end
disp('The sorted vector is:')
disp(a)
```

3. Υπολογίστε την τετραγωνική ρίζα ενός θετικού πραγματικού αριθμού x σύμφωνα με την παρακάτω διαδικασία.

Βήμα 1: Επιλογή των τιμών των: x , N , $error$

Βήμα 2: Επιλογή μιας αρχικής τιμής του $y = x/2$

Βήμα 3: For $i = 1, 2, \dots, N$

 Βήμα 4: $y_{old} = y$

 Βήμα 5: Υπολόγισε το $y = (x / y + y) / 2$

 Βήμα 6: Αν $|y - y_{old}| < error$ πήγαινε στο Βήμα 8

Βήμα 7: EndFor

Βήμα 8: Τύπωσε το y

Βήμα 9: End

Η παραπάνω διαδικασία ονομάζεται **αλγόριθμος** για την εύρεση της τετραγωνικής ρίζας ενός πραγματικού αριθμού.

Ο παραπάνω αλγόριθμος τερματίζεται σε δύο περιπτώσεις:

αν τα βήματα 4, 5, 6 γίνουν N φορές (βλ. Βήμα 3)

ή

αν δύο διαδοχικές προσεγγίσεις διαφέρουν (κατ' απόλυτη τιμή – βλ. Βήμα 6) λιγότερο από την επιθυμητή ακρίβεια $error$ (accuracy) - <http://mathworld.wolfram.com/Accuracy.html>

β) Συγκρίνετε τα αποτελέσματά σας με την εσωτερική συνάρτηση (intrinsic function) του MATLAB `sqrt`

γ) Υπολογίστε την τετραγωνική ρίζα του 2 χρησιμοποιώντας $N=10$ και $error=1.E-6$.

δ) Μετατρέψτε τον παραπάνω κώδικα σε function,

Είσοδος: Οι επαναλήψεις N , το σφάλμα $error$, ο αριθμός x

Εξοδος: η τετραγωνική ρίζα του x

```
clear;clc;
disp('This program computes the square root')
disp('of a positive real number')
x=input('Give the positive real number: ');
N=input('Give the number of iterations: ');
error=input('Give the accuracy: ');
y=x/2;
for i=1:N
    y_old=y;
    y=(x/y+y)/2;
    if (abs(y-y_old)<error);break;end
end

fprintf('The square root is: %.15f\n',y)
y_intrinsic=sqrt(x);
fprintf('Use of matlab function sqrt: %.15f\n',y_intrinsic)
difference = abs(y - y_intrinsic);
fprintf('Difference: %.15f\n',difference)
```

Συνάρτηση

```
function [y] = mysqrt(x,n,error)
% [y] = mysqrt(x,N,error)
% This function computes the square root
% of a positive real number
% INPUT ARGUMENTS
% x: positive real number
% N: number of iterations
% error: the accuracy
% OUTPUT ARGUMENTS
% y: The approximated square root of x

    y=x/2;
    for i=1:n
        y_old=y;
        y=(x/y+y)/2;
        if (abs(y-y_old)<error);break;end
    end

end
```

Script που καλεί την mysqrt

```
clear;clc;
disp('This program calls the function mysqrt')
disp('to compute the square root')
disp('of a positive real number')
x=input('Give the positive real number: ');
N=input('Give the number of iterations: ');
error=input('Give the accuracy: ');

y=mysqrt(x,N,error);

fprintf('The square root is: %.15f\n',y)
y_intrinsic=sqrt(x);
fprintf('Use of matlab function sqrt: %.15f\n',y_intrinsic)
difference = abs(y - y_intrinsic);
fprintf('Difference: %.15f\n',difference)
```