Εργαστήριο MATLAB

- Δημιουργία / Διαχείριση μητρών και διανυσμάτων:

  a. zeros(m,n), ones(m,n), eye(m,n), eye(n), rand(m,n), randn(m,n).

  - Πράξεις μεταξύ πινάκων:
    - addition (+)
    - subtraction (-)
    - transposition (‘)
    - multiplication (*)
    - point wise multiplication (.*)
    - point wise division (./)
    - power exposition (^)
    - point wise power exposition (.^)
  - Χρήσιμες συναρτήσεις:
    - sum(), diag(), inv(), reshape(), length(), size(), numel(), det, triu(), tril().
  - Δεικτοδότηση μητρών:
    - A(row_index,column_index)
  - Γραμμική Προσπέλαση:
    - A = [1 2 3;4 5 6;6 7 9];
    - I = [1:1:9];
    - B = A(I);
    - B = [1 4 7 2 5 8 3 6 9]
  - Προσπέλαση Πολλαπλών Στοιχείων:
    - A([1:row_index],column_index) (προσπέλαση στοιχείων στήλης).
    - A(row_index,[1:column_index]) (προσπέλαση στοιχείων γραμμής).
    - A(row_index,[1:end]) ⇔ A(row_index,:) (προσπέλαση γραμμής).
    - A([1:end],column_index) ⇔ A(:,column_index) (προσπέλαση στήλης).

- **Ενδεικτικές Ασκήσεις**

1. Δημιουργήστε το διάνυσμα x με στοιχεία ...
  - ***a.*** 2, 4, 6, 8, ...
  - ***b.*** 10, 8, 6, 4, 2, 0, -2, -4
  - ***c.*** 1, 1/2, 1/3, 1/4, 1/5, ...
  - ***d.*** 0, 1/2, 2/3, 3/4, 4/5, ...

2. Δημιουργήστε ένα διάνυσμα με όλους τους μονούς αριθμούς μεταξύ 31 και 75.

3. Αν x = [2 5 1 6].
  - ***b.*** Προσθέστε 16 στο κάθε στοιχείο.
  - ***c.*** Υπολογίστε την τετραγωνική ρίζα του κάθε στοιχείου
  - ***d.*** Υπολογίστε το τετράγωνο κάθε στοιχείου

4. Αν x = [3 2 6 8]' και y = [4 1 3 5]' (x και y θα πρέπει να είναι διανύσματα στήλης).
  - ***a.*** Προσθέστε τα στοιχεία του x στο y
  - ***b.*** Υψώστε κάθε στοιχείο του x στην δύναμη που προσδιορίζεται από το αντίστοιχο στοιχείο του y.
  - ***c.*** Διαιρέστε κάθε στοιχείο του y με το αντίστοιχο στοιχείο του x.
  - ***d.*** Πολλαπλασιάστε κάθε στοιχείο του x με το αντίστοιχο στοιχείο του y, αποκαλώντας το αποτέλεσμα "z".
  - ***e.*** Προσθέστε τα στοιχεία του z και αναθέστε τα αποτελέσματα σε μια μεταβλητή"w".
  - ***f.*** Υπολογίστε το x'*y – w.

5. Δημιουργήστε ένα διάνυσμα x με στοιχεία,

$$x_n = (-1)^{n+1}/(2n-1)$$

Προσθέστε τα στοιχεία αυτού του διανύσματος σε ένα νέο διάνυσμα (n=100).

6. Να πραγματοποιηθεί η αντιστροφή της σειράς των στοιχείων ενός δοσμένου διανύσματος **x**.

```
x = x(length(x):-1:1)
```

7. Να κατασκευάστε ένα πίνακα $\overline{\overline{A}}$, $\overline{\overline{B}} \in M_{15\times 15}$ όπου $\overline{\overline{A}} = [a_{ij}]$ όπου $a_{ij} = \cos[(i+j)*\pi + \frac{\pi}{2}]$ και $\overline{\overline{B}} = [b_{ij}]$ όπου $b_{ij} = \sin[(i+j)*\pi + \frac{\pi}{2}]$.

Να υλοποιηθεί ένα script file που να υπολογίζει:

  - ***a.*** Τον πίνακα $\overline{\overline{C}} = A^{-1} * B$
  - ***b.*** Η μέση τιμή των στοιχείων του $\overline{\overline{A}}$, $\overline{\overline{B}}$.
  - ***c.*** Η διακύμανση των στοιχείων του $\overline{\overline{C}}$.
  - ***d.*** Η μέση τιμή ανά στήλη του $\overline{\overline{C}}$

```
% ΑΣΚΗΣΗ 1 - ΕΝΔΕΙΚΤΙΚΗ ΛΥΣΗ
k = [1:1:15];
lk = length(k);
I = k'*ones(1,lk);
A = cos((I+I')*pi + pi/2);
B= sin((I+I')*pi + pi/2);
%ΕΡΩΤΗΜΑ Α -  μέση τιμή των στοιχείων του
C = inv(A)*B;
%ΕΡΩΤΗΜΑ Β - Η  μέση τιμή των στοιχείων του Α, Β
mA = sum(sum(A))/numel(A);
%or
%mA=mean(mean(A));
mB = sum(sum(B))/numel(B);
%or
%mB=mean(mean(B));
%ΕΡΩΤΗΜΑ Γ - Η διακύμανση των στοιχείων του C
[Cm,Cn] = size(C);
m=mean(mean(C));
Cv = C-ones(Cm,Cn)*m;
Cv = Cv.^2;
S = sum(sum(Cv))/numel(Cv);
% % ΑΥΤΟ ΕΙΝΑΙ ΛΑΘΟΣΣΣΣΣΣΣΣ s=var(var(X));
%ΕΡΩΤΗΜΑ Δ - Η μέση τιμή ανά στήλη
mc = mean(C);
```

8. Να δημιουργήσετε μια μήτρα $A \in M_{100x100}$ τα στοιχεία της οποίας είναι ομοιόμορφα κατανεμημένα στο διάστημα $[1:1:10]$.
   a. Να υπολογισθεί η μέση τιμή των άρτιων γραμμών της μήτρας.
   b. Να υπολογισθεί η μέση τιμή των περιττών στηλών της μήτρας.
   c. Να υπολογισθούν οι συχνότητες εμφάνισης καθενός εκ των στοιχείων του διαστήματος $[1:1:10]$ και να αποθηκευθούν σε ένα νέο διάνυσμα F.
   d. Να παραστήσετε γραφικά το ιστόγραμμα συχνοτήτων που αντιστοιχεί στο διάνυσμα F.
   e. Να υπολογιστούν οι συχνότητες εμφάνισης των άρτιων και περιττών στοιχείων της μήτρας A.

```
clc
clear all
% Set the elements of the random matrix A that are uniformly
distributed in
% the [1..10] interval.
A = ceil(10*rand(100,100));
EvenRows = A([2:2:100],:);
EvenRowsMean = sum(sum(EvenRows))/numel(EvenRows);
% or equivalently
EvenRowsMean = mean(mean(EvenRows));
OddColumns = A(:,[1:2:99]);
OddColumnsMean = mean(mean(OddColumns));
% or equivalently
OddColumnsMean = sum(sum(OddColumns))/numel(OddColumns);
% Transform matrix A into a row vector.
V = reshape(A,1,numel(A));
% Compute the histogram of frequencies.
```

```
F = hist(V,[1:1:10]);
figure('Name','Frequency Histogram')
bar([1:1:10],F,'b')
xlabel('Integer Values')
ylabel('Absolute Frequencies')
grid on
% Compute the mean frequencies of even and odd elements in the
[1..10] elements.
Feven_mean = mean(F([2:2:10]))
Fodd_mean = mean(F([1:2:9]))
```

9. Να υπολογισθεί η μέση τιμή του αθροίσματος $\sum_{m=1}^{100}\sum_{n=1}^{100} m^2 + n^2$
Έστω ότι η τιμή του n είναι 3, τότε για τον υπολογισμό του αθροίσματος θα έχουμε ότι:

```
clc
clear all
N = 100;
% Version 1 - Non Vectorized Code.
tic
S = 0;
for m = 1:1:N
    for n = 1:1:N
        S = S + m^2 + n^2;
    end;
end;
toc
S
% Version 2 - Vectorized Code.
tic
S = 0;
I = [1:1:N]' * ones(1,N);
W = I.^2 + (I').^2;
S = sum(sum(W));
toc
S
```

10. Θεωρούμε ένα γραμμικό χρονικά αναλλοίωτο σύστημα διακριτού χρόνου η κρουστική απόκριση του οποίου, $h[n]$, έχει διάρκεια N, ίση με την περίοδο του σήματος εισόδου, $x[n]$. Να υλοποιήσετε μια συνάρτηση MatLab myconv.m η οποία δεχόμενη ως είσοδο τα διανύσματα **x** και **h**, μήκους N, θα επιστρέφει την έξοδο του συστήματος $y[n]$. Με χρήση της συνάρτησης που θα υλοποιήσετε να υπολογίσετε το σήμα εξόδου που αντιστοιχεί στο σήμα εισόδου $x[n] = \left\{\cos\left(\pi + \frac{\pi}{64} \times n\right), 0 \le n \le N-1\right\}$ με κρουστική απόκριση $h[n] = \{e^{0.01 \times n}, 0 \le n \le N-1\}$ για N = 128.

Γνωρίζουμε από τη θεωρία σημάτων και συστημάτων ότι η απόκριση ενός Γ.Χ.Α συστήματος θα δίνεται από την σχέση $y[n] = \sum_{k=0}^{n} x[k]h[n-k]$. Με βάση την παραπάνω σχέση θα έχουμε ότι:

$$y[0] = x[0] * h[0]$$
$$y[1] = x[0] * h[1] + x[1] * h[0]$$
$$\vdots$$
$$y[N-2] = x[0] * h[N-2] + \cdots + x[N-2] * h[0]$$
$$y[N-1] = x[0] * h[N-1] + \cdots + x[N-1] * h[N-1]$$

```
% Define the number of samples for both the signal and the impulse
% response.
N = 128;
% Define the interval corresponding to the input signal and the impulse
% response.
n = [0:1:N-1];
% Define the signal and the impulse response.
x = sin(pi+(pi/64)*n);
h = exp(0.01*n);
% Plot the signal.
figure('Name','Input Signal')
stem(n,x,'-r','LineWidth',1.5);
xlabel('Time Interval');
ylabel('Signal Value')
grid on
% Plot the impulse response.
figure('Name','Impulse Response')
stem(n,h,'-r','LineWidth',1.8);
xlabel('Time Interval');
ylabel('Signal Value')
grid on
% Compute the output signal.
y = myconv(x,h)
% Plot input signal, impulse response and ouput signal in the same
window.
figure('Name','Output Signal')
subplot(3,1,1)
stem(n,x,'-r','LineWidth',1.5);
xlabel('Time Interval');
ylabel('Signal Value')
grid on
subplot(3,1,2)
stem(n,h,'-b','LineWidth',1.8);
xlabel('Time Interval');
ylabel('Signal Value')
grid on
subplot(3,1,3)
stem(n,y,'-g','LineWidth',1.8);
xlabel('Time Interval');
```

```
ylabel('Signal Value')
grid on
```

11. Να γραφεί ρουτίνα MatLab η οποία θα κατασκευάζει τον τετραγωνικό πίνακα $\mathbf{A} \in \mathbf{M}_{n\times n}$ με στοιχεία που δίνονται από την παρακάτω σχέση A(r,c) = (r-1)*10*n + 10*c.
Στη συνέχεια η ρουτίνα θα πρέπει να αντιμεταθέτει μεταξύ τους τα στοιχεία του άνω τριγωνικού και κάτω τριγωνικού υποπίνακα εξαιρώντας τα στοιχεία της κύριας διαγωνίου.

Η ρουτίνα που περιγράφεται παρακάτω πραγματοποιεί την αντιμετάθεση των στοιχείων του πίνακα με άξονα συμμετρίας την κύρια διαγώνιο; [όχι, π.χ. για n = 4.]

Π.χ. για n = 2 θα πρέπει να πραγματοποιείται η εξής αντιστροφή:

$$\begin{bmatrix} 10 & 20 \\ 30 & 40 \end{bmatrix} \rightarrow \begin{bmatrix} 10 & 30 \\ 20 & 40 \end{bmatrix}$$

Π.χ. για n = 3 θα πρέπει να πραγματοποιείται η εξής αντιστροφή:

$$\begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{bmatrix} \rightarrow \begin{bmatrix} 10 & 40 & 70 \\ 20 & 50 & 80 \\ 30 & 60 & 90 \end{bmatrix}$$

```
% Clear screen.
clc
% Clear all variables in the working space.
clear all
% Set the dimensionality of the square matrix.
n = 3;
% Compute the number of elements in the corresponding square matrix.
N = n * n;
% Set the elements of matrix M, where M is a n x n matrix whose
elements
% are given by the following equation:
% M(r,c) = (r-1)*10*n + 10*c
% Matrix M is initialy defined as a row vector.
M1 = [10:10:10*N];
% Reshape the row vector M in a corresponding n x n square matrix.
M1 = reshape(M1,n,n)'
```

```
M2 = M1;
% Get the main diagonal of matrix M.
Diag = diag(M2);
% Get the positions of the main diagonal elements in the original
matrix
% M. Keep in mind that the intersect routine requires a row vector
version
% of the matrix M, thus the reshape operation is used in order to
% internally trasform matrix M into a row vector.
[Diagonal,DiagonalIndices] = intersect(reshape(M2,1,N),Diag);
% Replace the main diagonal elements with zeros;
M2(DiagonalIndices) = 0;
% Get the upper and lower triangle matrix corresponding to the original
% matrix M.
UpperTriangle = triu(M2);
LowerTriangle = tril(M2);
% Get the non-zero elements positions of the upper and lower triangle
matrices.
UpperTriangleNonZeroIndices = find(UpperTriangle~=0);
LowerTriangleNonZeroIndices = find(LowerTriangle~=0);
% Get the non-zero elements of the upper and lower triangle matrices.
NonZeroUpperTriangle = UpperTriangle(UpperTriangleNonZeroIndices)
NonZeroLowerTriangle = LowerTriangle(LowerTriangleNonZeroIndices)
M2(UpperTriangleNonZeroIndices) = NonZeroLowerTriangle;
M2(LowerTriangleNonZeroIndices) = NonZeroUpperTriangle;
M2(DiagonalIndices) = Diagonal;
M2
```

12. Να γραφεί συνάρτηση MatLab η οποία θα πραγματοποιεί το k fold cross validation διαμερισμό ενός δοσμένου συνόλου δεικτών.

```
function [TrainIndices,TestIndices] = kfoldIndices(N,K)
Indices = [1:1:N];
M = N / K;
if (mod(N,K)~=0)
    error('The number of elements within vector Indices must be fully
devided by K');
else
    TrainIndices = cell(1,K);
    TestIndices = cell(1,K);
    for k = 1:1:K
        test_indices = [(k-1)*M+1:1:k*M]
        train_indices = setdiff(Indices,test_indices);
        TrainIndices{k} = train_indices;
        TestIndices{k} = test_indices;
    end;
end
```