



Γλώσσες Προγραμματισμού

Πίνακες και πράξεις με πίνακες/Αρχεία Μ

Χαρά Πρασσά, chprassa@gmail.com

Ένα μεγάλο μέρος αυτών των σημειώσεων έχει βασιστεί:

- Hanselman, D. & Littlefield, B. (2007), Μάθετε το Matlab 7, εκδόσεις Κλειδάριθμος.
- Γιαμουρίδης, Δ. (2004), Financial Modeling with Matlab, Πανεπιστημιακές Σημειώσεις.
- Γεωργίου, Γ. & Ξενοφώντος, Χ. (2007), Εισαγωγή στη Matlab, Πανεπιστημιακές Σημειώσεις.
- Ζαπράνης, Α. & Λιβάνης, Ε. (2009), Χρηματοοικονομικές εφαρμογές με το Matlab, εκδόσεις Κλειδάριθμος.

ο Η εντολή **dot(a,b)** υπολογίζει το εσωτερικό γινόμενο δύο διανυσμάτων a και b ίσου μήκους (και τα δύο διανύσματα γραμμές ή στήλες ή το ένα διάνυσμα γραμμή και το άλλο στήλη). Ένας άλλος τρόπος να υπολογίσουμε το εσωτερικό γινόμενο δύο διανυσμάτων είναι με πολλαπλασιασμό τους.

```
>> a=[1 2 3];
```

```
>> b=[3 4 5]';
```

```
>> dot(a,b)
```

```
ans =
```

```
26
```

```
>> dot(b,a)
```

```
ans =
```

```
26
```

```
>> a*b
```

```
ans =
```

```
26
```

ο Η εντολή **cross(u,v)** υπολογίζει το εξωτερικό γινόμενο $u \times v$ δύο διανυσμάτων u και v ίσου μήκους (και τα δύο διανύσματα γραμμές ή στήλες ή το ένα διάνυσμα γραμμή και το άλλο στήλη). Το εξωτερικό γινόμενο μας δίνει ένα διάνυσμα γραμμή εκτός αν και τα δύο διανύσματα είναι στήλες.

```
>> cross(a,b)
```

```
ans =
```

```
-2  4 -2
```

```
>> cross(b,a)
```

```
ans =
```

```
2 -4  2
```

Ο πιο εύκολος τρόπος για να δημιουργήσουμε σύνθετους πίνακες είναι από συνένωση μικρότερων υποπινάκων. Για την κατασκευή ενός σύνθετου πίνακα χειριζόμαστε τους υποπίνακες όπως και τα απλά στοιχεία ενός πίνακα.

- η $[A \ B]$ μας δίνει τον σύνθετο πίνακα $[A \ B]$

- η $[A; B; C]$ μας δίνει τον σύνθετο πίνακα $\begin{bmatrix} A \\ B \\ C \end{bmatrix}$

- η $[A \ B; C \ D]$ μας δίνει τον σύνθετο πίνακα $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$

□ Έστω οι πίνακες $A = \begin{bmatrix} -1 & 2 \\ 3 & -4 \end{bmatrix}$, $B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $C = \begin{bmatrix} 0 & 2 \\ 3 & -3 \end{bmatrix}$ και $D = \begin{bmatrix} -1 & 2 \\ 5 & -4 \end{bmatrix}$

Να κατασκευάσετε τους σύνθετους πίνακες: $E = [A \ B]$, $F = \begin{bmatrix} A \\ B \\ C \end{bmatrix}$ και $G = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$.

```
>> A=[-1 2;3 -4];
>> B=abs(A);
>> C=eye(2)+A;
>> D=A;
>> D(2,1)=5;
>> E=[A B]
```

E =

```
-1  2  1  2
 3 -4  3  4
```

```
>> F=[A;B;C]
```

F =

```
-1  2
 3 -4
 1  2
 3  4
 0  2
 3 -3
```

```
>> G=[A B;C D]
```

G =

```
-1  2  1  2
 3 -4  3  4
 0  2 -1  2
 3 -3  5 -4
```

Παράδειγμα 1:

Να κατασκευάσετε τον πίνακα $A = \begin{bmatrix} 2 & 0 & 0 & 5 & 5 \\ 0 & 2 & 0 & 5 & 5 \end{bmatrix}$ χρησιμοποιώντας τους ειδικούς πίνακες, eye και zeros.

```
>> A=[2*eye(2,3) zeros(2)+5]
```

```
A =
```

```
2 0 0 5 5
0 2 0 5 5
```

Παράδειγμα 2:

Να κατασκευάσετε τον πίνακα $B = \begin{bmatrix} 2 & 0 & 3 & 0 & 0 \\ 0 & 2 & 0 & 3 & 0 \\ 0 & 0 & 4 & 4 & 4 \\ 0 & 0 & 4 & 4 & 4 \end{bmatrix}$ χρησιμοποιώντας τους ειδικούς πίνακες, eye, ones και zeros.

```
>> B=[2*eye(2) 3*eye(2,3); zeros(2) 4*ones(2,3)]
```

```
B =
```

```
2 0 3 0 0
0 2 0 3 0
0 0 4 4 4
0 0 4 4 4
```

Παράδειγμα 3:

Να κατασκευάσετε τον πίνακα $C = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 3 & 4 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 3 & 4 \end{bmatrix}$ χρησιμοποιώντας τον υποπίνακα:

$$D = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

A' τρόπος:

```
>> D=[1 2;3 4];
>> C=[D zeros(2) ;zeros(2) D]
C =
    1     2     0     0
    3     4     0     0
    0     0     1     2
    0     0     3     4
```

B' τρόπος: Χρήση της συνάρτησης **blkdiag** η οποία κατασκευάζει κατά μπλοκ διαγώνιους πίνακες.

```
>> C=blkdiag(D,D)
C =
    1     2     0     0
    3     4     0     0
    0     0     1     2
    0     0     3     4
```

Γ' τρόπος: Χρήση της συνάρτησης **repmat(A,m₁,n₁)** η οποία κατασκευάζει ένα σύνθετο $(m_1 \times m) \times (n_1 \times n)$ πίνακα με υποπίνακα τον $A(m,n)$ σε κάθε θέση. Η **repmat(A,n)** μας δίνει τον αντίστοιχο $(n \times m) \times (n \times n)$ πίνακα.

```
>> C=repmat(D,2,2)
```

```
C =
```

```
1  2  1  2
3  4  3  4
1  2  1  2
3  4  3  4
```

```
>> C(3:end,1:2)=0;
```

```
>> C(1:2,3:end)=0
```

```
C =
```

```
1  2  0  0
3  4  0  0
0  0  1  2
0  0  3  4
```


Η συνάρτηση **cat** συνενώνει δύο ή περισσότερους πίνακες κατά μήκος μιας διάστασης.

Έστω οι τετραγωνικοί πίνακες A, B, C:

- η **cat(1,A,B)** συνενώνει τους A και B στη διεύθυνση των y σχηματίζοντας τον σύνθετο πίνακα

$\begin{bmatrix} A \\ B \end{bmatrix}$. Άρα είναι ισοδύναμη με την $[A;B]$. Ομοίως, η **cat(1,A,B,C)** είναι ισοδύναμη με την $[A;B;C]$.

- η **cat(2,A,B)** συνενώνει τους A και B στη διεύθυνση των x σχηματίζοντας τον σύνθετο πίνακα $[A \ B]$. Άρα είναι ισοδύναμη με την $[A,B]$. Ομοίως, η **cat(2,A,B,C)** είναι ισοδύναμη με την $[A,B,C]$.

- η **cat(3,A,B)** συνενώνει τους A και B στη διεύθυνση των z σχηματίζοντας έναν $n \times n \times 2$ πίνακα. Ομοίως, η **cat(3,A,B,C)** σχηματίζει ένα $n \times n \times 3$ πίνακα.

Έστω οι πίνακες $A = \begin{bmatrix} 3 & 0 & 0 & 4 \\ 0 & 3 & 0 & 0 \\ 0 & 2 & 3 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$ και $C = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$. Θα δούμε τα αποτελέσματα των $\text{cat}(1,A,B)$, $\text{cat}(2,A,C)$ και $\text{cat}(3,A,\text{rand}(3,4))$

```
>> A=3*eye(3,4);
```

```
>> A(3,2)=2;
```

```
>> A(1,4)=4;
```

```
>> B=ones(1,4);
```

```
>> C=zeros(3,1);
```

```
>> D=cat(1,A,B)
```

```
D =
```

```
3  0  0  4
0  3  0  0
0  2  3  0
1  1  1  1
```

```
>> E=cat(2,A,C)
```

```
E =
```

```
3  0  0  4  0
0  3  0  0  0
0  2  3  0  0
```

```
>> F=cat(3,A,rand(3,4))
```

```
F(:, :, 1) =
```

```
3  0  0  4
0  3  0  0
0  2  3  0
```

```
F(:, :, 2) =
```

```
0.9572  0.1419  0.7922  0.0357
0.4854  0.4218  0.9595  0.8491
0.8003  0.9157  0.6557  0.9340
```

- ❑ Με την εντολή `whos` βλέπουμε τις διαστάσεις των πινάκων που δημιουργήσαμε και ειδικά ότι ο F είναι 3×4×2 πίνακας.

```
>> whos
```

Name	Size	Bytes	Class	Attributes
A	3x4	96	double	
B	1x4	32	double	
C	3x1	24	double	
D	4x4	128	double	
E	3x5	120	double	
F	3x4x2	192	double	

- ❑ Θα πρέπει να δοθεί προσοχή στις **διαστάσεις των πινάκων** έτσι ώστε να είναι εφικτή η συνένωση.

```
>> cat(1,A,C)
```

??? Error using ==> cat

CAT arguments dimensions are not consistent.

Η συνάρτηση **reshape(A,m,n)** ανασχηματίζει τον πίνακα A διαστάσεων $i \times j$, ώστε να έχει m γραμμές και n στήλες δημιουργώντας τον πίνακα C, δηλαδή διαβάζει ανά στήλη τα στοιχεία.

```
>> B=1:12
```

```
B =
```

```
1  2  3  4  5  6  7  8  9 10 11 12
```

```
>> reshape(B,2,6)
```

```
% μετασχηματισμός σε 2 γραμμές, 6 στήλες, συμπλήρωση κατά στήλη
```

```
ans =
```

```
1  3  5  7  9 11
2  4  6  8 10 12
```

```
>> reshape(B,3,4)
```

```
% μετασχηματισμός σε 3 γραμμές, 4 στήλες, συμπλήρωση κατά στήλη
```

```
ans =
```

```
1  4  7 10
2  5  8 11
3  6  9 12
```

```
>> reshape(B,4,[])
```

```
% μετασχηματισμός σε 4 γραμμές, το Matlab υπολογίζει πόσες στήλες χρειάζονται, συμπλήρωση κατά στήλη
```

```
ans =
```

```
1  5  9
2  6 10
3  7 11
4  8 12
```

Αναζήτηση δευτερευόντων πινάκων

Η συνάρτηση **find** επιστρέφει τους αριθμοδείκτες ή τους δείκτες των στοιχείων ενός πίνακα ο οποίος ικανοποιεί κάποια σχέση.

```
% Εφαρμογή της find σε διάνυσμα
```

```
>> x=-3:3
```

```
x =
```

```
   -3   -2   -1    0    1    2    3
```

```
>> k=find(abs(x)>1)
```

```
k =
```

```
     1     2     6     7
```

```
% Εφαρμογή της find σε πίνακες δύο διαστάσεων
```

```
>> A=[1 2 3;4 5 6;7 8 9];
```

```
>> [i,j]=find(A>5)
```

% Οι αριθμοδείκτες i και j είναι αυτοί που αφορούν τις γραμμές και τις στήλες αντίστοιχα, και για τους οποίους η παράσταση δίνει αποτέλεσμα True.

```
i =
```

```
     3
```

```
     3
```

```
     2
```

```
     3
```

```
j =
```

```
     1
```

```
     2
```

```
     3
```

```
     3
```

Αναζήτηση δευτερευόντων πινάκων

% Εφαρμογή της find σε πίνακες δύο διαστάσεων

```
>> k=find(A>5);
```

% Εναλλακτικά η find επιστρέφει ενιαίους αριθμοδείκτες για πίνακες δύο διαστάσεων. Παρατηρούμε ότι οι θέσεις που παίρνουμε προκύπτουν αν διατρέχουμε προς τα κάτω κάθε στήλη και πάμε από αριστερά προς τα δεξιά.

```
>> A(k)
```

```
ans =
```

```
3
```

```
6
```

```
8
```

```
9
```

Παράδειγμα:

Ο δείκτης μάζας σώματος ενός ενήλικα υπολογίζεται με βάση τον τύπο:
$$\Delta M\Sigma = \frac{\text{Βάρος (κιλά)}}{[\text{'Υψος (μέτρα)}]^2}$$

Έστω heights το διάνυσμα με τα ύψη 12 ενηλίκων βάρους 56 κιλών.

```
>> heights=[1.76 1.68 1.63 1.79 1.58 1.85 1.62 1.60 1.71 1.83 1.69 1.86]
```

```
heights =
```

```
1.7600 1.6800 1.6300 1.7900 1.5800 1.8500 1.6200 1.6000 1.7100 1.8300 1.6900 1.8600
```

Σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας, η κατηγορία βάρους ‘λιποβαρής’ αφορά σε $\Delta M\Sigma < 18.5$. Να κατασκευάσετε το διάνυσμα low το οποίο θα μας δείχνει ποιοι ενήλικες ανήκουν στην συγκεκριμένη κατηγορία βάρους.

Παράδειγμα:

```
>> weight=56;  
>> low=find(weight./heights.^2<18.5)  
low =  
     1     4     6    10    12
```

- Προσοχή στη χρήση της τελείας!

```
>> low=find(weight/heights.^2<18.5)  
??? Error using ==> mldivide  
Matrix dimensions must agree.
```

```
>> low=find(weight./heights^2<18.5)  
??? Error using ==> mpower  
Inputs must be a scalar and a square matrix.
```

Μπορούμε να δούμε και τα ύψη των ενηλίκων που ανήκουν στην κατηγορία βάρους ‘λιποβαρής’:

```
>> heights(low)  
ans =  
    1.7600    1.7900    1.8500    1.8300    1.8600
```

Πολυδιάστατοι πίνακες

Μέχρι τώρα ασχοληθήκαμε με το χειρισμό μονοδιάστατων και δισδιάστατων πινάκων. Από την έκδοση 5 και μετά, η Matlab υποστηρίζει την δημιουργία πινάκων οποιασδήποτε διάστασης. Εμείς θα αναφερθούμε μόνο στους πίνακες 3 διαστάσεων επειδή είναι πιο εύκολη η κατανόηση της απεικόνισής τους. Η 3^η διάσταση αριθμείται σε σελίδες (pages), έτσι ένας πίνακας τριών διαστάσεων έχει γραμμές, στήλες και σελίδες. Κάθε σελίδα περιέχει ένα πίνακα δυο διαστάσεων σε γραμμές και στήλες.

Παραδείγματα κατασκευής τρισδιάστατου πίνακα:

```
>> A=zeros(4,3,2)
```

```
A(:,:,1) =
```

```
0  0  0
0  0  0
0  0  0
0  0  0
```

```
A(:,:,2) =
```

```
0  0  0
0  0  0
0  0  0
0  0  0
```


Πολυδιάστατοι πίνακες

```
>> A=zeros(2,3)
% Αρχικά δημιουργούμε ένα δισδιάστατο πίνακα
A =
    0    0    0
    0    0    0
% Προσθέτουμε μια ακόμα σελίδα ώστε να γίνει τρισδιάστατος
>> A(:,:,2)=ones(2,3);
>> A(:,:,3)=5 % είναι ισοδύναμη με την A(:,:,3)=5 *ones(2,3)
A(:,:,1) =
    0    0    0
    0    0    0
A(:,:,2) =
    1    1    1
    1    1    1
A(:,:,3) =
    5    5    5
    5    5    5
Εναλλακτικά, χρησιμοποιούμε την συνάρτηση cat:

>> A=cat(3,zeros(2,3),ones(2,3),5*ones(2,3));

% συνένωση των 3 πινάκων κατά μήκος της τρίτης διάστασης
```

Πολυδιάστατοι πίνακες

Η εντολή **reshape** έχει εφαρμογή σε πολυδιάστατους πίνακες, μετατρέποντας οποιονδήποτε πίνακα κάποιων διαστάσεων σε οποιονδήποτε άλλο πίνακα διαφορετικών διαστάσεων.

- Έστω A ο τρισδιάστατος πίνακας που δημιουργήσαμε στο προηγούμενο παράδειγμα:

```
>> B=reshape(A,2,9) % δισδιάστατα δεδομένα, παράταξη σελίδων δίπλα-δίπλα
```

```
B =
```

```
 0  0  0  1  1  1  5  5  5
 0  0  0  1  1  1  5  5  5
```

```
>> A=reshape(B,2,3,3) % αναδημιουργία του A
```

```
A(:,:,1) =
```

```
 0  0  0
 0  0  0
```

```
A(:,:,2) =
```

```
 1  1  1
 1  1  1
```

```
A(:,:,3) =
```

```
 5  5  5
 5  5  5
```

Πολυδιάστατοι πίνακες

Η εντολή **repmat** έχει επίσης εφαρμογή σε πολυδιάστατους πίνακες.

```
>> C=ones(2,3)
```

```
C =
```

```
 1  1  1
 1  1  1
```

```
>> repmat(C,1,1,3)
```

% μορφή που δεν επιτρέπεται σε περισσότερες από δύο διαστάσεις!

??? Error using ==> repmat

Too many input arguments.

```
>> repmat(C,[1,1,3])
```

% ο σωστός τρόπος χρήσης της συνάρτησης

```
ans(:,:,1) =
```

```
 1  1  1
 1  1  1
```

```
ans(:,:,2) =
```

```
 1  1  1
 1  1  1
```

```
ans(:,:,3) =
```

```
 1  1  1
 1  1  1
```

➤ Έχουμε ήδη αναφερθεί στην εντολή `format`, ακόμα δυο εντολές που έχουν να κάνουν με την απεικόνιση των μεταβλητών και των τιμών τους στο παράθυρο εντολών (command window) του Matlab παρουσιάζονται στον πιο κάτω πίνακα:

Εντολή	Ερμηνεία
<code>disp</code>	απεικόνιση μεταβλητών στην οθόνη
<code>input</code>	υποβολέας για είσοδο δεδομένων

- `disp(var)`: απεικονίζει την τιμή της μεταβλητής `var` (χωρίς το όνομα) στην οθόνη.
- `disp var`: απεικονίζει το όνομα της μεταβλητής `var` (χωρίς την τιμή) στην οθόνη.
- `disp(['μήνυμα κειμένου', num2str(όνομα μεταβλητής)])`: απεικονίζει ένα μήνυμα κειμένου μαζί με την τιμή της μεταβλητής.

❑ Η τιμή της `var` μπορεί να είναι αριθμός ή πίνακας ή αλφαριθμητικό.

Παράδειγμα 1:

```
>> a=[1 2 3]
a =
     1     2     3
>> disp(a)
     1     2     3
>> disp a
a
>> >> disp(['Το διάνυσμα a είναι το ακόλουθο: ',num2str(a)])
Το διάνυσμα a είναι το ακόλουθο: 1  2  3
```

Παράδειγμα 2:

Θεωρούμε εδώ το αλφαριθμητικό

```
>> x='R2020a'
x =
R2020a
>> disp(x)
R2020a
>> disp x
x
>> disp(['Η τελευταία έκδοση του Matlab είναι η: ',num2str(x)])
Η τελευταία έκδοση του Matlab είναι η: R2020a
```

- Το βασικό πλεονέκτημα της **disp** είναι το ότι μας επιτρέπει την εκτύπωση χρήσιμων ενημερωτικών αλφαριθμητικών/κειμένου για την επεξήγηση των μεταβλητών:

```
>> disp('Ο ακόλουθος είναι ένας μοναδιαίος πίνακας διαστάσεων 3x3: '), eye(3)
```

Ο ακόλουθος είναι ένας μοναδιαίος πίνακας διαστάσεων 3x3 :

ans =

```
1    0    0
0    1    0
0    0    1
```

- Κάτι που δεν μπορούμε να κάνουμε με τη **disp** είναι να τυπώσουμε διάφορες μεταβλητές σε μια γραμμή:

```
>> a=130; % ανώτατο όριο ταχύτητας επιβατικού αυτοκινήτου σε αυτοκινητόδρομο
```

```
>> disp('Maximum speed:'), disp(a), disp('km/h')
```

Maximum speed:

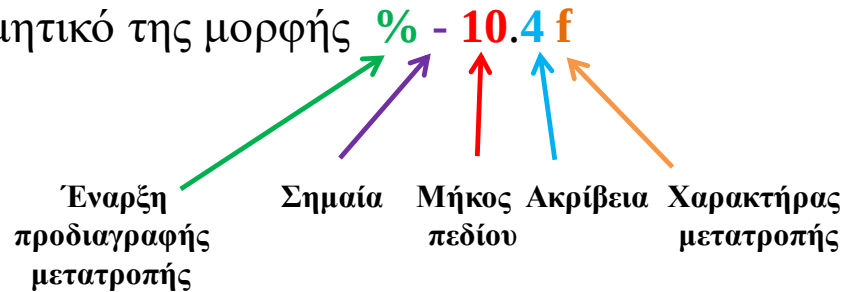
130

km/h

- Αυτή η δυνατότητα δίνεται με την εντολή **sprintf**

- Η εντολή **sprintf** χρησιμοποιείται αντί της **disp** στην περίπτωση που είναι επιθυμητή η εκτύπωση δεδομένων σε προκαθορισμένη **μορφή** (format). Η γενική σύνταξη της εντολής είναι η εξής: **s = sprintf(format, A, B,)**

- s: αλφαριθμητικό το οποίο τυπώνεται στην οθόνη
- format: αλφαριθμητικό της μορφής **% - 10.4 f**



Σημαία: Ελέγχει την ευθυγράμμιση της εκτύπωσης.

-	Αριστερή στοίχιση
+	Εκτύπωση πρόσημου (+ ή -)
κενό	Εισαγωγή κενού πριν από την τιμή
0	Κάλυψη κενού χώρου με μηδενικά αντί με κενά

Χαρακτήρας μετατροπής: Δηλώνει τον συμβολισμό της εκτύπωσης.

f	Συμβολισμός σταθερής υποδιαστολής
e	Εκθετικός συμβολισμός
s	Αλφαριθμητικό

Μήκος πεδίου: Προσδιορίζει τον ελάχιστο αριθμό ψηφίων που θα εκτυπωθούν.

Ακρίβεια: Προσδιορίζει το πλήθος των ψηφίων που θα τυπωθούν δεξιά της υποδιαστολής (default value=6).

Για να μην εκτυπωθεί το αλφαριθμητικό s στην οθόνη $\Longrightarrow$ **disp(sprintf(format, A, B,))**

Είσοδος και έξοδος δεδομένων

```
>> r=pi/2
r =
    1.5708
>> s=sprintf('Το μέτρο της ορθής γωνίας είναι %9.2f ακτίνια',r) %εκτύπωση σε 9 θέσεις, 2 δεκαδικά ψηφία και συμβολισμός σταθ. υποδιαστολής
s =
Το μέτρο της ορθής γωνίας είναι    1.57 ακτίνια
>> s=sprintf('Το μέτρο της ορθής γωνίας είναι %-9.2f ακτίνια',r) %αριστερή στοίχιση, εκτύπωση σε 9 θέσεις, 2 δεκαδικά ψηφία και συμβολισμός σταθ. υποδιαστολής
s =
Το μέτρο της ορθής γωνίας είναι 1.57    ακτίνια
>> s=sprintf('Το μέτρο της ορθής γωνίας είναι %+9.2f ακτίνια',r) %πρόσημο, εκτύπωση σε 9 θέσεις, 2 δεκαδικά ψηφία και συμβολισμός σταθ. υποδιαστολής
s =
Το μέτρο της ορθής γωνίας είναι  +1.57 ακτίνια
>> s=sprintf('Το μέτρο της ορθής γωνίας είναι % 9.2f ακτίνια',r) %κενό, εκτύπωση σε 9 θέσεις, 2 δεκαδικά ψηφία και συμβολισμός σταθ. υποδιαστολής
s =
Το μέτρο της ορθής γωνίας είναι    1.57 ακτίνια
>> s=sprintf('Το μέτρο της ορθής γωνίας είναι %09.2f ακτίνια',r) %0 στις κενές θέσεις, εκτύπωση σε 9 θέσεις, 2 δεκαδικά ψηφία και συμβολισμός σταθ. υποδιαστολής
s =
Το μέτρο της ορθής γωνίας είναι 000001.57 ακτίνια
>> s=sprintf('Το μέτρο της ορθής γωνίας είναι %04.2f ακτίνια',r) %0 στις κενές θέσεις, εκτύπωση σε 4 θέσεις, 2 δεκαδικά ψηφία και συμβολισμός σταθ. υποδιαστολής
s =
Το μέτρο της ορθής γωνίας είναι 1.57 ακτίνια
>> s=sprintf('Το μέτρο της ορθής γωνίας είναι %9.4f ακτίνια',r) %εκτύπωση σε 9 θέσεις, 4 δεκαδικά ψηφία και συμβολισμός σταθ. υποδιαστολής
s =
Το μέτρο της ορθής γωνίας είναι    1.5708 ακτίνια
>> s=sprintf('Το μέτρο της ορθής γωνίας είναι %9.4e ακτίνια',r) %εκτύπωση σε 9 θέσεις, 4 δεκαδικά ψηφία και εκθετικός συμβολισμός
s =
Το μέτρο της ορθής γωνίας είναι 1.5708e+000 ακτίνια
>> d=90
d =
    90
>> s=sprintf('Το μέτρο της ορθής γωνίας είναι %4.2f ακτίνια ή %2.0f μοίρες',r,d)
s =
Το μέτρο της ορθής γωνίας είναι 1.57 ακτίνια ή 90 μοίρες
```


- Γενική μορφή εντολής input: **R=input('prompt')** , όπου prompt ένα αλφαριθμητικό.

Με την εντολή αυτή εμφανίζεται στην οθόνη η προτροπή prompt και το σύστημα αναμένει από το χρήστη να εισάγει την τιμή της μεταβλητής R. Αν η R είναι αλφαριθμητικό τότε η τιμή της πρέπει να δοθεί μέσα σε τόνους '.....'.

Παράδειγμα 1:

```
>> a=input('Enter variable: ')
```

```
Enter variable:12
```

```
a =
```

```
12
```

```
>> B=input('Enter a 3x2 random array: ')
```

```
Enter a 3x2 random array: rand(3,2)
```

```
B =
```

```
0.1869 0.6463
```

```
0.4898 0.7094
```

```
0.4456 0.7547
```

```
>> c=input('Enter complex variable: ')
```

```
Enter complex variable: 2-5i
```

```
c =
```

```
2.0000 - 5.0000i
```

```
>> first_name=input('Enter first name: ')
```

```
Enter first name: 'Anna'
```

```
first_name =
```

```
Anna
```

```
>> whos
```

Name	Size	Bytes	Class	Attributes
B	3x2	48	double	
a	1x1	8	double	
c	1x1	16	double	complex
first_name	1x4	8	char	

- ❑ Ένας τρόπος να αποφύγουμε τους τόνους είναι να δηλώσουμε μέσω της εντολής ότι η μεταβλητή είναι αλφαριθμητικό. Αυτό γίνεται ως εξής:

R=input('prompt', 's')

Παράδειγμα 2:

```
>> first_name=input('Enter first name: ', 's')
```

```
Enter first name: Anna
```

```
first_name=
```

```
Anna
```

- ❑ Ένας τρόπος να αλλάξουμε γραμμή είναι να χρησιμοποιήσουμε το σύμβολο `\n`. Αν θέλουμε να γράψουμε μια πρόταση σε 3 γραμμές τότε η εντολή διαμορφώνεται ως εξής:

R=input('line 1 \n line 2 \n line3')

Παράδειγμα 3:

```
>> SUM=input('x=1 \n y=-1 \n Το άθροισμα των x και y είναι : ')
```

```
x=1
```

```
y=-1
```

```
Το άθροισμα των x και y είναι : 0
```

```
SUM =
```

```
0
```

- Το Matlab παρέχει την δυνατότητα αποθήκευσης μιας ακολουθίας εντολών σε ένα αρχείο και στη συνέχεια την εκτέλεση αυτών με μια μόνο εντολή. Η ακολουθία αυτή των εντολών γράφεται στον διορθωτή κειμένου με την επέκταση .m, δηλαδή Όνομα_Αρχείου.m.

Τα m-files διακρίνονται σε:

- **Αρχεία script ή αρχεία εντολών** τα οποία δεν έχουν ορίσματα εισόδου και εξόδου αλλά εκτελούν μια ακολουθία εντολών σε μεταβλητές του χώρου εργασίας. Οι μεταβλητές που υπολογίζονται προστίθενται στο χώρο εργασίας με αποτέλεσμα να μπορούν να χρησιμοποιηθούν και από άλλα αρχεία script ή από το παράθυρο εντολών. Πρόκειται δηλαδή για μεταβλητές καθολικές.
- **Αρχεία συναρτήσεων** τα οποία περιλαμβάνουν μια γραμμή ορισμού συνάρτησης, δέχονται ορίσματα εισόδου και επιστρέφουν μεταβλητές εξόδου. Οι μεταβλητές που ορίζονται μέσα στη συνάρτηση ισχύουν μόνο για τη συνάρτηση, είναι δηλαδή τοπικές.

Για να δημιουργήσουμε ένα νέο αρχείο script:

- επιλέγουμε από το μενού File → New → Script
- επιλέγουμε από τη γραμμή εργαλείων το εικονίδιο με το λευκό χαρτί
- στο παράθυρο εντολών πληκτρολογούμε edit (>> edit)

Για να επεξεργαστούμε ένα υπάρχον αρχείο script:

- επιλέγουμε από το μενού File → Open

Για να αποθηκεύσουμε ένα αρχείο script:

- επιλέγουμε από το μενού File → Save as

Για την ονοματολογία των αρχείων script ισχύουν οι κανόνες που αναφέραμε για την ονοματολογία των μεταβλητών του Matlab.

Για να εκτελεστεί ένα αρχείο script:

- πληκτρολογούμε το όνομα του (χωρίς την επέκταση .m) στο παράθυρο εντολών
- ενώ έχουμε ανοιχτό το αρχείο, επιλέγουμε το εικονίδιο από τη γραμμή εργαλείων με την ονομασία Save and Run 
- επιλέγουμε όλο τον κώδικα ή μέρος του κώδικα που θέλουμε να εκτελεστεί, πατάμε δεξί κλικ και επιλέγουμε Evaluate Selection

Παράδειγμα:

Έστω ότι θέλουμε να υπολογίσουμε την ετήσια απόσβεση ενός περιουσιακού στοιχείου με τη μέθοδο της σταθερής ή γραμμικής απόσβεσης. Το ποσό της ετήσιας απόσβεσης υπολογίζεται από τον τύπο:

$$\text{Ετήσια απόσβεση} = \frac{\text{Κόστος κτήσης} - \text{Υπολειμματική αξία}}{\text{αριθμός ετών ωφελίμου ζωής}}$$

Στον επεξεργαστή κειμένου του Matlab πληκτρολογούμε τις ακόλουθες εντολές:

```
clear all
```

```
KK=500000;
```

```
YA=150000;
```

```
n=5;
```

```
AA=KK-YA; %Αποσβεστέα αξία
```

```
Straight_line_depreciation=AA/n %Ποσό ετήσιας απόσβεσης
```

Στη συνέχεια αποθηκεύουμε το αρχείο **script** με το όνομα **depreciation.m**.

Αν στο παράθυρο εντολών πληκτρολογήσουμε το όνομα depreciation τότε θα εμφανιστεί στην οθόνη το αποτέλεσμα:

```
>> depreciation  
Straight_line_depreciation =  
70000
```

Το αρχείο depreciation.m μπορεί να γίνει πιο λειτουργικό με χρήση των συναρτήσεων input και disp που αναφέραμε παραπάνω. Δημιουργούμε το αρχείο **depreciation_new.m** το οποίο περιλαμβάνει την ακολουθία εντολών:

```
clear all  
  
disp('Το πρόγραμμα αυτό υπολογίζει το ποσό της ετήσιας απόσβεσης ενός περιουσιακού στοιχείου με τη μέθοδο της σταθερής απόσβεσης');  
KK=input('Πληκτρολογείτε το κόστος κτήσης:');  
YA=input('Πληκτρολογείτε την υπολειμματική αξία:');  
n=input('Πληκτρολογείτε τον αριθμό των ετών διάρκειας της υπολειμματικής ζωής:');  
disp('Πατήστε το πλήκτρο Enter για να υπολογιστεί το ποσό της ετήσιας απόσβεσης');  
pause  
AA=KK-YA; %Αποσβεστέα αξία  
  
Straight_line_depreciation=AA/n; %Ποσό ετήσιας απόσβεσης  
  
disp(['Το ποσό της ετήσιας απόσβεσης είναι:',num2str(Straight_line_depreciation)]);
```

Στο αρχείο `depreciation_new.m` έχει χρησιμοποιηθεί η συνάρτηση `pause`. Η εν λόγω συνάρτηση προκαλεί προσωρινή παύση μέχρι ο χρήστης να πατήσει κάποιο πλήκτρο ή για `n` δευτερόλεπτα (`pause(n)`) και μετά συνέχεια.

Παράδειγμα για εξάσκηση:

Να δημιουργήσετε ένα αρχείο `script` το οποίο θα υπολογίζει το μέσο όρο της βαθμολογίας ενός φοιτητή βάσει των βαθμών του σε 3 μαθήματα. Μέσω της εντολής `input` να δίνεται η δυνατότητα εισαγωγής της βαθμολογίας σε κάθε ένα από τα 3 μαθήματα μέσω του παραθύρου εντολών του Matlab.

Στο Matlab υπάρχουν ενσωματωμένες συναρτήσεις για μια πληθώρα εφαρμογών αλλά δίνεται η δυνατότητα στο χρήστη να δημιουργήσει τις δικές του μέσω του διορθωτή κειμένου. Τα αρχεία συναρτήσεων περιλαμβάνουν ορίσματα εισόδου και εξόδου τα οποία μπορεί να αποτελούνται από μία ή περισσότερες μεταβλητές.

□ Η γραμμή ορισμού της συνάρτησης έχει τη μορφή:

`function [μεταβλητές εξόδου] = όνομα_συνάρτησης (μεταβλητές εισόδου)`

Στην περίπτωση που υπάρχουν περισσότερες από μια μεταβλητές εισόδου, αυτές χωρίζονται μεταξύ τους με κόμμα. Το ίδιο ισχύει και για τις μεταβλητές εξόδου. Όταν υπάρχει μια μεταβλητή εξόδου οι αγκύλες μπορούν να παραλειφθούν.

`function μεταβλητή εξόδου= όνομα_συνάρτησης (μεταβλητές εισόδου)`

Το όνομα της συνάρτησης πρέπει να είναι το ίδιο με το όνομα με το οποίο αποθηκεύεται το αρχείο της συνάρτησης και δεν πρέπει να συμπίπτει με όνομα συνάρτησης βιβλιοθήκης. Η συνάρτηση καλείται με το όνομα με το οποίο είναι αποθηκευμένο το αρχείο συνάρτησης. Το Matlab από προεπιλογή σας προτρέπει να αποθηκεύσετε το αρχείο της συνάρτησης με το όνομα της.

□ Μετά τη γραμμή ορισμού της συνάρτησης ακολουθούν συνήθως κάποια σχόλια τα οποία είναι προαιρετικά αλλά χρησιμοποιούνται για την καλύτερη κατανόηση της λειτουργίας της συνάρτησης. Τα σχόλια αυτά εμφανίζονται στο παράθυρο εντολών με την εντολή:

```
>> help όνομα_συνάρτησης
```

Παράδειγμα:

Έστω ότι θέλουμε να δημιουργήσουμε μια συνάρτηση η οποία θα υπολογίζει την μηνιαία δόση δανείου σταθερού επιτοκίου με δεδομένα το κεφάλαιο του δανείου, το ετήσιο επιτόκιο και τον αριθμό των ετών του δανείου.

Γνωρίζουμε ότι η δόση δανείου δεδομένης περιόδου υπολογίζεται από τη σχέση:

$$R = K_0 \frac{r}{1 - (r + 1)^{-T}}$$

,όπου K_0 είναι το αρχικό κεφάλαιο, r το επιτόκιο περιόδου και T ο αριθμός των περιόδων της διάρκειας του δανείου.

```
function [mpay] = loan(amount,rate,years)
%Η συνάρτηση loan υπολογίζει τη μηνιαία δόση δανείου
%Μεταβλητές εισόδου:
%amount: το κεφάλαιο του δανείου σε ευρώ
%rate: το ετήσιο επιτόκιο σε μονάδες %
%years: αριθμός των ετών του δανείου
%Μεταβλητή εξόδου:
%mpay: μηνιαία δόση του δανείου

format bank
ratem=rate*0.01/12;
a=1+ratem;
mpay=amount*(ratem/(1-(a)^(-12*years)));
```

```
>> loan(100000,8,20)
```

```
ans =
```

```
836.44
```

```
>> help loan
```

Η συνάρτηση loan υπολογίζει τη μηνιαία δόση δανείου

Μεταβλητές εισόδου:

amount: το κεφάλαιο του δανείου σε ευρώ

rate: το ετήσιο επιτόκιο σε μονάδες %

years: αριθμός των ετών του δανείου

Μεταβλητή εξόδου:

mpay: μηνιαία δόση του δανείου

Η συνάρτηση `loan` αποτελείται από τα εξής επιμέρους τμήματα:

- ❑ τη γραμμή ορισμού της συνάρτησης (γραμμή 1 του κώδικα): Ορίζει ότι το `m-file` είναι αρχείο συνάρτησης, ορίζει το όνομα της συνάρτησης καθώς και το όνομα και την ακολουθία των μεταβλητών εισόδου και εξόδου.
- ❑ τη γραμμή `H1` και τις λοιπές βοηθητικές γραμμές (γραμμές 2-8 του κώδικα): Προαιρετικές γραμμές σχολίων οι οποίες παρέχουν πληροφορίες σχετικά με τη λειτουργία της συνάρτησης. Εμφανίζονται στο παράθυρο εντολών (`command window`) όταν ο χρήστης εισάγει την εντολή `help loan`.
- ❑ το κύριο σώμα της συνάρτησης (γραμμές 9-12 του κώδικα): Η ακολουθία των εντολών που εκτελούνται με δεδομένο ότι οι μεταβλητές εισόδου `amount`, `rate`, `years` είναι γνωστές (έχουν ήδη εισαχθεί από το χρήστη).

Παράδειγμα για εξάσκηση:

Το διάσημο μοντέλο των Black-Scholes για την αποτίμηση δικαιωμάτων προαίρεσης είναι μια μη γραμμική συνάρτηση που μπορεί εύκολα να προγραμματιστεί στο Matlab. Πιο συγκεκριμένα, ο τύπος των Black-Scholes υπολογίζει τις τιμές ενός δικαιώματος αγοράς, c , ή ενός δικαιώματος πώλησης, p , ως ακολούθως:

$$c = SN(d_1) - Ke^{-rT}N(d_2)$$

$$p = Ke^{-rT}N(-d_2) - SN(-d_1)$$

όπου $d_1 = \frac{\ln(S/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$, $d_2 = d_1 - \sigma\sqrt{T}$, S είναι η τρέχουσα τιμή του υποκείμενου τίτλου π.χ. της μετοχής, K η τιμή εξάσκησης, σ η μεταβλητότητα της τιμής της μετοχής, T ο χρόνος έως τη λήξη, r το επιτόκιο χωρίς κίνδυνο, και $N(d)$ η αθροιστική συνάρτηση κατανομής της τυπικής κανονικής κατανομής.

- ✓ Σας ζητείται να κατασκευάσετε ένα αρχείο συνάρτησης με όνομα `BSMy` και μεταβλητές εισόδου τις S , K , r , T , σ το οποίο να υπολογίζει μέσω του μοντέλου των Black-Scholes τις τιμές δικαιωμάτων αγοράς, c , και πώλησης, p , με τα ανωτέρω χαρακτηριστικά. Για την αθροιστική συνάρτηση κατανομής να χρησιμοποιήσετε τη συνάρτηση `normcdf`.

Παράδειγμα για εξάσκηση (συνέχεια):

Έστω Ευρωπαϊκά δικαιώματα με τιμή εξάσκησης 95\$ και ημερομηνία λήξης σε 3 μήνες. Αν η υποκείμενη μετοχή δεν πληρώνει μερίσματα, έχει τρέχουσα τιμή 100\$, ετήσια μεταβλητότητα 50% και το ετήσιο επιτόκιο χωρίς κίνδυνο είναι 10%, να χρησιμοποιήσετε τη συνάρτηση που κατασκευάσατε για να υπολογίσετε την τιμή των δικαιωμάτων αγοράς και πώλησης.

□ Επαληθεύστε τα αποτελέσματα, με τη χρήση της συνάρτησης:

$[Call, Put] = blsprice(Price, Strike, Rate, Time, Volatility)$