Πίνακας (*matrix*) είναι μια ορθογώνια διάταξη αριθμών, που καθορίζεται από τον αριθμό των στηλών και σειρών, που ονομάζονται διαστάσεις του πίνακα. Έτσι, ένας πίνακας διαστάσεων MxN αποτελείται από M σειρές και N στήλες. Ένας ιδιαίτερα σημαντικός πίνακας είναι ο τετράγωνος πίνακας, δηλαδή αυτός με MxN διαστάσεις όπου (M=N), δηλαδή ο αριθμός των σειρών είναι ίδιος με αυτών των στηλών του πίνακα.

Ένας για παράδειγμα πίνακας 2x4 διαστάσεων είναι:

$$A = \begin{bmatrix} 3 & 5 & 2 & 6 \\ 10 & 14 & 12 & 15 \end{bmatrix}$$

Σε Matlab μπορεί να γραφεί ως εξής:	Έξοδος Matlab:
`clc; clear all;` `A = [3,5,2,6; 10,14,12,15];` `disp('A matrix:'); disp(A)`	`A matrix:` `     3     5     2     6` `    10    14    12    15` `>>`

Ανάστροφος πίνακας είναι ο πίνακας που προκύπτει αν οι σειρές γίνουν στήλες και οι στήλες σειρές με την ίδια ακολουθία (δηλαδή η πρώτη σειρά να γίνει πρώτη στήλη, η δεύτερη σειρά δεύτερη στήλη κτλ.).

Σε Matlab μπορεί να γραφεί ως εξής:	Έξοδος Matlab:
`% Transpose matrix` `clc; clear all;` `A = [3,5,2,6; 10,14,12,15];` `AT = A';` `disp('Transpose A matrix:'); disp(AT)`	`Transpose A matrix:` `     3    10` `     5    14` `     2    12` `     6    15` `>>`

Διαγώνιος πίνακας είναι ο μονο-διάστατος πίνακας (μία γραμμή ή μία στήλη) που περιέχει τους αριθμούς της διαγωνίου του πίνακα A.

`% Diagonal matrix` `clc; clear all;` `A=[ 3 5 2 6; 10 14 12 15; 12 18 20 21; 25 29 30 35];` `D = diag(A);` `disp('Diagonal of A matrix:');` `disp(D)`	`A =` `     3     5     2     6` `    10    14    12    15` `    12    18    20    21` `    25    29    30    35` `Diagonal of A matrix:` `     3    14    20    35` `>>`

Άθροισμα κατά μήκος σειρών / στηλών πίνακα A:

```
%Sum of columns/rows

clc; clear all;

A = [ 3 5 2 6; 10 14 12 15; 12 18 20 21; 25 29 30 35]

Sum_across_rows=sum(A');
Sum_across_columns=sum(A);
Sum_ALL_1=sum(sum(A));
Sum_ALL_2=sum(sum(A'));

disp('Sum across rows/columns of A matrix:');
disp('Sum of rows'); disp(Sum_across_rows)
disp('Sum of columns'); disp(Sum_across_columns)
disp('Sum of ALL MATRIX'); disp(Sum_ALL_1)
disp('Sum of ALL MATRIX'); disp(Sum_ALL_2)
```

```
A =

     3     5     2     6
    10    14    12    15
    12    18    20    21
    25    29    30    35

Sum across rows/columns
of A matrix:
Sum of rows
    16    51    71   119

Sum of columns
    50    66    64    77

Sum of ALL MATRIX
   257

Sum of ALL MATRIX
   257

>>
```

Αντιστροφή ενός τετράγωνου πίνακα είναι ένας τετράγωνος πίνακας:

Αν $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ τότε $A^{-1} = \dfrac{1}{a \times d - b \times c} \times \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

Δηλαδή το πηλίκο της διαίρεσης με αριθμητή τον πίνακα $\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

και παρανομαστή την ορίζουσα του A, det(A) = $a \times d - b \times c$

- Για να ορίζεται ο αντίστροφος πίνακας θα πρέπει **det(A)** $\neq$ **0**

Ο αντίστροφος πίνακας επίσης έχει την εξής ιδιότητα:

$\boxed{\mathbf{A^{-1}A=AA^{-1}=I}}$, όπου I είναι το μοναδιαίο διάνυσμα:

με: $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Ο αντίστροφος πίνακας δηλώνεται και με την εντολή `invA` ή `A^-1`

Εναλλαγή των στοιχείων της κυρίας διαγωνίου και στην αντιδιαγώνιο βάζω μείον. Εν συνεχεία διαιρώ όλα τα στοιχεία του πίνακα με την ορίζουσα.

```
%Inverse matrix

clc; clear all;

A=[  3, 5, 2, 6
    10,14,12,15
    12,18,20,21
    25,29,30,35];
```

```
matrix A:
     3     5     2     6
    10    14    12    15
    12    18    20    21
    25    29    30    35

inverse matrix A:
   -0.2025    0.1076   -0.2722    0.1519
   -0.2215    0.6646   -0.0633   -0.2089
   -0.2373   -0.0380    0.1108   -0.0095
    0.5316   -0.5949    0.1519    0.1013
```

`disp('matrix A:'); disp(A)` `A_INV=inv(A);` `disp('inverse matrix A:');` `disp(A_INV);` `X1=round(A*inv(A));` `X2=round(inv(A)*A);` `disp('A^-1*A:'); disp(X1)` `disp('A*A^-1:'); disp(X2)`	`A^-1*A:` `1 0 0 0` `0 1 0 0` `0 0 1 0` `0 0 0 1` `A*A^-1:` `1 0 0 0` `0 1 0 0` `0 0 1 0` `0 0 0 1` `>>`

Πράξεις μεταξύ πινάκων:

Πρόσθεση: Για να προστεθούν 2 πίνακες πρέπει να έχουν ίδιες διαστάσεις MxN.

```
clc; clear all;

A=[  3, 5, 2, 6
    10,14,12,15
    12,18,20,21
    25,29,30,35]

B=[  4, 6, 8, 9
    21,23,24,25
    12,14,19,11
    17,18,14,12
   ]

C=A+B;
disp('matrix addition C=');
disp(C)
```

```
A =

   3   5   2   6
  10  14  12  15
  12  18  20  21
  25  29  30  35


B =

   4   6   8   9
  21  23  24  25
  12  14  19  11
  17  18  14  12

matrix addition C=
   7  11  10  15
  31  37  36  40
  24  32  39  32
  42  47  44  47

>>
```

Πολλαπλασιασμός Πινάκων:

Για τον πολλαπλασιασμό δύο πινάκων πρέπει οι πίνακες να έχουν διαστάσεις MxN και NxM δηλαδή όσες σειρές / στήλες έχει ο πρώτος πρέπει να έχει στήλες / σειρές ο δεύτερος πίνακας.

```
clc; clear all;

C =[
     3     5     2     6
    10    14    12    15
    12    18    20    21
]

D =[
     3     5     2
    10    14    12
    12    18    20
    25    29    30
]

E=C*D;
disp('matrix Multiplication E=');
disp(E)
```

```
C =

     3     5     2     6
    10    14    12    15
    12    18    20    21


D =

     3     5     2
    10    14    12
    12    18    20
    25    29    30

matrix Multiplication E=
         233          295          286
         689          897          878
         981         1281         1270

>>
```

Διαφορά μεταξύ πολλαπλασιασμού πινάκων και πολλαπλασιασμού στοιχείων πινάκων: **A*A** ≠ **A.*A**=A^2

```
clc; clear all;

A=[  3, 5, 2, 6
    10,14,12,15
    12,18,20,21
    25,29,30,35];

B1 = A
B2 = 2*A
F1=B1*B2;

disp('matrix Multiplication, F1= ');
disp(F1)

F2=B1.*B2;
disp('matrix element multiplication, F2= ');
disp(F2)
```

```
B1 =

     3     5     2     6
    10    14    12    15
    12    18    20    21
    25    29    30    35


B2 =

     6    10     4    12
    20    28    24    30
    24    36    40    42
    50    58    60    70

matrix Multiplication, F1=
         466         590         572         690
        1378        1794        1756        2094
        1962        2562        2540        2994
        3200        4172        4096        4880

matrix element multiplication, F2=
          18          50           8          72
         200         392         288         450
         288         648         800         882
        1250        1682        1800        2450

>>
```

Άθροισμα στοιχείων μονοδιάστατου πίνακα γραμμή ή στήλη και δυσδιάστατου πίνακα **(sum)**

```
clc; clear all;

A=[  3, 5, 2, 6
    10,14,12,15
    12,18,20,21
    25,29,30,35];
B= [1 2 3 4];
C= [1;2;3;4];

X0 = sum(A);
X1 = sum(A,1);
X2 = sum(A,2);
X3 = sum(B);
X4 = sum(C);

disp('Διάνυσμα - γραμμή του οποίου κάθε στοιχείο ισούται με το άθροισμα των στοιχείων της στήλης που ανήκει', X0= ');
disp(X0)
disp('1ης διάστασης διάνυσμα του οποίου κάθε στοιχείο ισούται με το άθροισμα των στοιχείων της στήλης που ανήκει', X1= ');
disp(X1)
disp('2ης διάστασης διάνυσμα του οποίου κάθε στοιχείο ισούται με το άθροισμα των στοιχείων της γραμμής που ανήκει', X2= ');
disp(X2)
disp('Επιστρέφει το άθροισμα όλων των στοιχείων του μονοδιάστατου πίνακα γραμμή', X3= ');
disp(X3)
disp('Επιστρέφει το άθροισμα όλων των στοιχείων του μονοδιάστατου πίνακα στήλη', X4= ');
disp(X4)
```

```
Διάνυσμα-γραμμή του οποίου κάθε στοιχείο ισούται με το άθροισμα των στοιχείων της στήλης που ανήκει', X0=
    50    66    64    77

1ης διάστασης διάνυσμα του οποίου κάθε στοιχείο ισούται με το άθροισμα των στοιχείων της στήλης που ανήκει', X1=
    50    66    64    77

2ης διάστασης διάνυσμα του οποίου κάθε στοιχείο ισούται με το άθροισμα των στοιχείων της γραμμής που ανήκει', X2=
    16
    51
    71
   119

Επιστρέφει το άθροισμα όλων των στοιχείων του μονοδιάστατου πίνακα γραμμή', X3=
    10

Επιστρέφει το άθροισμα όλων των στοιχείων του μονοδιάστατου πίνακα στήλη', X4=
    10
>>
```

Μιγαδικοί αριθμοί

Οι μιγαδικοί αριθμοί αποτελούνται από το πραγματικό και το φανταστικό μέρος και έχουν την μορφή **a+bi** ή **a+bj**, όπου a και b είναι πραγματικοί αριθμοί (π.χ. 4, 10, 13.2, κλπ) και i ή j είναι η ρίζα $\sqrt{-1}$ και χρησιμοποιείται για την δημιουργία του φανταστικού μέρους του μιγαδικού αριθμού.

Μπορούμε να ορίσουμε έναν μιγαδικό αριθμό στο Matlab χρησιμοποιώντας τις μορφές a+jb ή a+ib ή a+j*b ή a+i*b, όπως φαίνεται στο παρακάτω παράδειγμα:

`clc; clear;` `a=sqrt(-1); disp(a);` `b=3+4j;     disp(b);` `c=3+4i;     disp(c);` `d=3+4*i;    disp(d);` `e=3+4*j;    disp(e);`	`0 + 1.0000i` `3.0000 + 4.0000i` `3.0000 + 4.0000i` `3.0000 + 4.0000i` `3.0000 + 4.0000i` `>>`

Υπάρχουν συναρτήσεις στο Matlab που σχετίζονται με τους μιγαδικούς αριθμούς **real**(d), **imag**(d), **abs**(d), **conj**(d) για την εύρεση του πραγματικού μέρους, του φανταστικού μέρους, του πλάτους και του συζυγή μιγαδικού του d.

`clc; clear;` `complex=3+4i; disp(complex);` `complex_real=real(complex); disp(complex_real);` `complex_imag=imag(complex); disp(complex_imag);` `complex_ampl=abs(complex);  disp(complex_ampl);` `complex_conjugate=conj(complex); disp(complex_conjugate);`	`3.0000 + 4.0000i` `3` `4` `5` `3.0000 - 4.0000i` `>>`

Οι περισσότεροι τελεστές και συναρτήσεις του Matlab όπως +, -, *, /, sqrt, exp, δουλεύουν με μιγαδικούς αριθμούς.

`clc; clear;` `a=3+4i` `b=4+5i` `c=a-b` `d=a*b` `e=a/b` `a_power=a^2` `a_exponential=exp(a)` `a_square_root=sqrt(a)`	`c =` `-1.0000 - 1.0000i` `d =` `-8.0000 +31.0000i` `e =` `0.7805 + 0.0244i`

	a_power = -7.0000 +24.0000i a_exponential = -13.1288 -15.2008i a_square_root = 2.0000 + 1.0000i >>

Σε πίνακες με μιγαδικούς αριθμούς υπάρχουν δύο τελεστές, ο τελεστής (**.'**) που χρησιμοποιείται για την εύρεση του ανάστροφου του πίνακα και ο τελεστής (**'**) που χρησιμοποιείται για να ευρεθεί ο ανάστροφος συζυγής μιγαδικός πίνακας.

<table>
<tr>
<td>

```
clc; clear;

d=[1+3i 4-2i; 2+5i 4+6i; 5+6i 7+8i]

d_transpose=d.'

d_transpose_conjugate=d'
```

</td>
<td>

```
d =

   1.0000 + 3.0000i   4.0000 - 2.0000i
   2.0000 + 5.0000i   4.0000 + 6.0000i
   5.0000 + 6.0000i   7.0000 + 8.0000i

d_transpose =

   1.0000 + 3.0000i   2.0000 + 5.0000i   5.0000 + 6.0000i
   4.0000 - 2.0000i   4.0000 + 6.0000i   7.0000 + 8.0000i

d_transpose_conjugate =

   1.0000 - 3.0000i   2.0000 - 5.0000i   5.0000 - 6.0000i
   4.0000 + 2.0000i   4.0000 - 6.0000i   7.0000 - 8.0000i
```

</td>
</tr>
</table>

ΚΑΤΑΝΟΗΣΗ ΑΝΑΣΤΡΟΦΟΥ ΣΥΖΥΓΗ ΜΙΓΑΔΙΚΟΥ ΠΙΝΑΚΑ ΜΕ ΠΡΑΓΜΑΤΙΚΕΣ ΚΑΙ ΜΙΓΑΔΙΚΕΣ ΤΙΜΕΣ

```
>> (1+i)'                    {ΜΙΓΑΔΙΚΟΣ - ΠΡΑΓΜΑΤΙΚΟ & ΦΑΝΤΑΣΤΙΚΟ ΜΕΡΟΣ}

ans =

   1.0000 - 1.0000i

>> D=[5 1+i; 1-i 10]         {ΠΙΝΑΚΑΣ 2Χ2 ΜΕ ΠΡΑΓΜΑΤΙΚΕΣ ΚΑΙ ΜΙΓΑΔΙΚΕΣ ΤΙΜΕΣ]

D =

   5.0000             1.0000 + 1.0000i
   1.0000 - 1.0000i  10.0000
```

```
>> D.'                              {ΕΝΔΙΑΜΕΣΟ ΣΤΑΔΙΟ - ΟΙ ΓΡΑΜΜΕΣ ΓΙΝΟΝΤΑΙ ΣΤΗΛΕΣ}

ans =

   5.0000             1.0000 - 1.0000i
   1.0000 + 1.0000i  10.0000

>> D' {ΤΕΛΙΚΟ ΣΤΑΔΙΟ - ΑΛΛΑΓΗ ΠΡΟΣΗΜΟΥ ΣΤΟ ΦΑΝΤΑΣΤΙΚΟ ΜΕΡΟΣ Ή ΆΛΛΩΣ ΠΑΙΡΝΟΥΜΕ ΤΟΥΣ ΣΥΖΥΓΕΙΣ ΜΙΓΑΔΙΚΟΥΣ}

ans =

   5.0000             1.0000 + 1.0000i
   1.0000 - 1.0000i  10.0000
```

Προσπέλαση στοιχείων πίνακα

Προσπέλαση και αλλαγή στοιχείου πίνακα. Τα στοιχεία ενός αριθμητικού πίνακα μπορούν να εκτυπωθούν ή να μεταβληθούν με τον προσδιορισμό των συντεταγμένων του στοιχείου και την αντίστοιχη εντολή Matlab όπως για παράδειγμα `disp(A(x,y))` ή `A(x,y)=z`, όπου z ένας αριθμός παρόμοιου format με τα υπόλοιπα στοιχεία του πίνακα. Ακολουθεί ένα σχετικό παράδειγμα εκτύπωσης και αλλαγής ενός στοιχείου αριθμητικού πίνακα.

```
clc; clear;
A =[ 3  5  2  6
    10 14 12 15
    12 18 20 21
    25 29 30 35];
disp('DISPLAY MATRIX A ELEMENT [4,2]');
disp(A(4,2));
disp('CHANGE CONTENTS OF A(4,2) TO 12');
A(4,2)=12;
disp('DISPLAY NEW MATRIX ELEMENT [4,2]');
disp(A(4,2));
disp('DISPLAY NEW MATRIX A: ');
disp(A);
```

```
DISPLAY MATRIX A ELEMENT [4,2]
    29

CHANGE CONTENTS OF A(4,2) TO 12
DISPLAY NEW MATRIX ELEMENT [4,2]
    12

DISPLAY NEW MATRIX A:
     3     5     2     6
    10    14    12    15
    12    18    20    21
    25    12    30    35

>>
```

Ακολουθεί παράδειγμα προσπέλασης και αλλαγής κάθε στοιχείου πίνακα μέσω διπλού βρόχου `for`:

```
clc; clear;
M=4;N=4;
B=ones(M,N);  %Αρχικοποίηση του B
C=zeros(M,N); %Αρχικοποίηση του C

A=round(B.*rand(M,N)*10); %Πίνακας 4X4 με τιμές από 0 έως 10
disp('Table A:'); disp(A);

for i=1:size(A,1)      %σειρές (1η διάσταση)
    for j=1:size(A,2) %στήλες (2η διάσταση)
        C(i,j)= round(A(i,j)*rand()*5); %Πίνακας C 4X4 από A
    end
end
disp('Table C:'); disp(C);
```

```
Table A:
     8     6    10    10
     9     1    10     5
     1     3     2     8
     9     5    10     1

Table C:
    17    27    40    48
    30     0    42    23
     3    11     7    16
    29     4    35     0

>>
```

Αρχικοποίηση Πίνακα

Πολλές φορές σε χρονοβόρα απαιτητικά προγράμματα είναι καλό να αρχικοποιούμε τους πίνακες πριν τους χρησιμοποιήσουμε, ώστε να βελτιστοποιήσουμε τον χρόνο εκτέλεσης χρονοβόρων προγραμμάτων. Έτσι, όπως γίνεται σε άλλα προγράμματα (π.χ. C, Java, κ.α.) εκτιμούμε το μέγεθος του πίνακα που πρέπει να χρησιμοποιήσουμε και δημιουργούμε τον πίνακα γεμίζοντάς τον με μηδενικά (ή μονάδες εφόσον κρίνεται αναγκαίο για το πρόγραμμα).

`clc; clear;` `M=4; N=5;` `fprintf('MATRIX DIMENSIONS M=%d, N=%d\n',M,N);` `B=zeros(M,N);` `fprintf('MATRIX initialization with zeros:\n');` `disp(B);` `fprintf('MATRIX initialization with ones:\n');` `C=ones(N,M);` `disp(C);`	MATRIX DIMENSIONS M=4, N=5 MATRIX initialization with zeros: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 MATRIX initialization with ones: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 >>

Επέκταση πίνακα από συνδυασμό πινάκων

Μερικές φορές είναι χρήσιμο να δημιουργούμε ένα καινούργιο πίνακα από συνδυασμό (οριζόντια ή κάθετα) δύο ή περισσοτέρων πινάκων. Προς αυτό χρησιμοποιούμε τις εντολές `horzcat` και `vertcat`, για **οριζόντια** και **κάθετη** επέκταση. Για τον κάθετο συνδυασμό πινάκων σε έναν επεκταμένο πίνακα, πρέπει ο αριθμός των στηλών και των δύο πινάκων να είναι ίσος. Έτσι για κάθετο συνδυασμό 2 πινάκων B(n,$\boldsymbol{m}$) και C(k,$\boldsymbol{l}$) είναι απαραίτητο $\boldsymbol{m=l}$ αλλά το n μπορεί να είναι διάφορο του k. Αντίστοιχα για τον οριζόντιο συνδυασμό πινάκων σε έναν επεκταμένο πίνακα, πρέπει ο αριθμός των γραμμών και των δύο πινάκων να είναι ίσος.

`` clc; clear; M=4; N=5; B=ones(M,N); C=ones(M-2,N)*5; fprintf('Matrix B=\n'); disp(B); fprintf('Matrix C=\n'); disp(C); VERT_Matrix=vertcat(B,C); fprintf('Matrix by B and C vertical combination=\n'); disp(VERT_Matrix); D=ones(M,N-2)*5; fprintf('Matrix D=\n'); disp(D); HORZ_Matrix=horzcat(B,D); fprintf('Matrix by B and D horizontal combination=\n'); disp(HORZ_Matrix); ```	Matrix B= 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 Matrix C= 5 5 5 5 5 5 5 5 5 5 Matrix by B and C vertical combination= 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 5 5 5 5 5 5 5 5 5 5 Matrix D= 5 5 5 5 5 5 5 5 5 5 5 5 Matrix by B and D horizontal combination= 1 1 1 1 1 5 5 5 1 1 1 1 1 5 5 5 1 1 1 1 1 5 5 5 1 1 1 1 1 5 5 5 >>

Ταξινόμηση τιμών πίνακα κατά αύξουσα ή φθίνουσα σειρά

Κατά την ταξινόμηση των τιμών ενός πίνακα, οι τιμές του πίνακα δύναται να ιεραρχηθούν κατά αύξουσα ή φθίνουσα αριθμητική τιμή με τις εντολές:

`sort(A,'ascend')` ή `sort(A,'descent')`, αντίστοιχα.

Έστω ένας μονοδιάστατος πίνακας A=[4, 3, 6, 9, 7]. Η ταξινόμηση των τιμών κατά **αύξουσα** τιμή είναι ο πίνακας B=[3, 4, 6, 7, 9] και κατά **φθίνουσα** τιμή ο πίνακας C=[9, 7, 6, 4, 3]. Ακολουθεί σχετικό πρόγραμμα σε Matlab με την χρήση της εντολής `sort`. Στην τελευταία σειρά του προγράμματος εκτυπώνονται και οι **αρχικοί δείκτες** των στοιχείων του πίνακα A.

clc; clear; A=[4, 3, 6, 9, 7]; disp('matrix A:'); disp(A); B=sort(A,'ascend'); disp('matrix A in ascending order:'); disp(B); C=sort(A,'descend'); disp('matrix A in descending order:'); disp(C); [D,index]=sort(A,'ascend') fprintf('matrixA--sortedA--index\n'); [A' D' index']	matrix A: 4 3 6 9 7 matrix A in ascending order: 3 4 6 7 9 matrix A in descending order: 9 7 6 4 3 D = 3 4 6 7 9 index = 2 1 3 5 4 matrixA--sortedA--index ans = 4 3 2 3 4 1 6 6 3 9 7 5 7 9 4 >>

Για την περίπτωση πινάκων δύο διαστάσεων, είναι δυνατόν να ιεραρχηθούν οι τιμές του πίνακα κατά στήλες ή κατά σειρές, σύμφωνα με τις εντολές `sort(A,1)` και `sort(A,2)`, αντίστοιχα. Ακολουθεί σχετικό πρόγραμμα σε Matlab.

clc; clear; M=3; N=3; B=ones(M,N); A=round(B.*rand(M,N)*10); disp('Table A:'); disp(A); B=sort(A,1); disp('Table A sorted along columns:'); disp(B); C=sort(A,2); disp('Table A sorted along rows:'); disp(C);	Table A: 4 2 6 8 5 7 8 4 8 Table A sorted along columns: 4 2 6 8 4 7 8 5 8 Table A sorted along rows: 2 4 6 5 7 8 4 8 8 >>