第3讲　向量、数组和矩阵

北京邮電大學

1 向量、数组与矩阵的创建

1.1 向量的创建

1．简单向量的创建

直接输入法构建向量：向量元素用“[ ]”括起来，元素之间用**空格**、**逗号**或者**分号**相隔。需要注意的是，用它们相隔生成的向量形式是不相同的。

(1) 用空格或逗号生成不同列的元素，即行向量。

北京邮电大学

(2) 用分号生成不同行的元素，即列向量。例如：

```
>> a1=[15;21;27;93;101];
>> a1
a1 =
    15
    21
    27
    93
   101
>>  a2=[15,21,27,93,101];
>>  a2
a2 =
    15    21    27    93   101
>> a3=[1 2 3 4]
a3 =
     1     2     3     4
```

2．冒号表达式创建等差数组

语法： • a=i:k:j

这一语句可以生成一个行向量，其中，**i为向量的起始值，k为增量步距，而j为向量的终止值**。

生成的向量为：

[i i+k*1 i+k*2 … i+k*n]，其中 i+k*n<=j<i+k*（n+1）

- 当k == 0、k >0且i>j或k<0且i<j时，返回一个空向量。
- 语句可以简写为a = i:j，此时**j的缺省值为1**

例如：

```
>>  vec1=10:5:60
vec1 =
    10    15    20    25    30    35    40    45    50    55    60
```

3．linspace()函数与等差数组的创建

y = linspace(a，b，n)。在a、b之间(包括a、b)生成n点线性间隔分布的行向量y，即向量y有n个元素。如果n小于2，linspace返回b。 若n忽略，则默认值位100个。

既 y=linspace(a,b) 与y=linspace(a,b,100)等效。

$$y_i = a + \frac{b-a}{n-1} * (i-1)$$

问：求10~60之间差值为5的等差数列，应该用哪条语句：

A. linspace(10, 60,10)　　B. linspace(10,60,11)

>> vec2=linspace (10,60,11)

vec2 =

10　15　20　25　30　35　40　45　50　55　60

>> vec3=linspace (10,60,10)

vec3 =

10.0000　15.5556　21.1111　26.6667　32.2222　37.7778　43.3333　48.8889　54.4444　60.0000

4．等比数组的创建

(1) X=logspace(a,b,n)。在10^a和10^b之间生成n个对数间隔等分数据的行向量。构成等比数列，数列的第一项X(1)=10^a，最后一项X(n)=10^b。

(2) X=logspace(a,b)。在10^a和10^b之间生成**50个**以对数间隔等分数据的行向量。构成等比数列，数列的第一项X(1)=10^a，最后一项X(50)=10^b。

$$X_i = 10^{a+\frac{b-a}{n-1}*(i-1)}$$

思考: 怎么用MATLAB创建1~10之间10个元素的等比数组?

1.2 向量的转置与操作

1．普通转置

使用转置符号(‘)可以将行向量转成列向量，反之亦然，b=a'，即b是a的转置向量。例如：

```
>> f=1:4
f =
    1    2    3    4
>> F=f '
F =
    1
    2
    3
    4
```

再次使用转置符号(')可将列向量转回成行向量。

2．点转置

MATLAB还提供了点转置(.')符号。对实数而言，(.')与(')操作是等效的；对于复数，(')操作结果是复数共轭转置。也就是说，在转置过程中，虚部的符号也改变了，而(.')操作只转置，不进行共轭操作。例如：

```
>> f=1:3
f =
   1   2   3
>> x=complex(f,f)
x =
   1.0000 + 1.0000i   2.0000 + 2.0000i   3.0000 + 3.0000i
```

```
>> y=x'
y =
  1.0000 - 1.0000i
  2.0000 - 2.0000i
  3.0000 - 3.0000i
>> z=x.'
z =
  1.0000 + 1.0000i
  2.0000 + 2.0000i
  3.0000 + 3.0000i
```

3．适用于向量的常用函数

适用于向量的常用函数有以下几种：

(1) min(x)、max(x)：向量x的元素的最小值、最大值。

(2) mean(x)：向量x的元素的平均值。

(3) median(x)：向量x的元素的中位数。

(4) std(x)：向量x的元素的标准差。

(5) diff(x)：向量x的相邻元素的差。

(6) sort(x)：对向量x的元素进行排序(Sorting)。

(7) length(x)：向量x的长度(元素个数)。

(8) sum(x)、prod(x)：向量x的元素总和、总乘积。

(9) cumsum(x)、cumprod(x)：向量x元素的累计总和、累计总乘积。

(10) dot(x, y)、cross(x, y)：向量x和y的内积、外积。

```
clear all;clc;close all;
A=[10 13 7 1 5 9 6 2 3]
>> [B,k]=min(A)
B =
   1
k =
   4
>> [B,k]=max(A)
B =
  13
k =
   2
>> B=mean(A)
B =
6.2222
>> B=median(A)
B =
   6
>> B=std(A)
B =
3.9616
>> B=diff(A)
B =
3   -6   -6    4    4   -3   -4    1
>> [B,k]=sort(A)
B =
1    2    3    5    6    7    9   10   13
k =
4    8    9    5    7    3    6    1    2
>> length(A)
ans =
   9
```

A=[10 13 7 1 5 9 6 2 3]

\>> sum(A)

ans =

56

\>> prod(A)

ans =

1474200

\>> cumsum(A)

ans =

10 23 30 31 36 45 51

53 56

\>> cumprod(A)

ans =

1 至 6 列

10 130 910 910

4550 40950

7 至 9 列

245700 491400 1474200

\>> dot(B,k)

ans =

225

\>>cross([1 2 3],[3,2,1])

ans=-4 8 -4

1.3 矩阵的创建方法

在MATLAB中创建矩阵，同样遵循行向量和列向量的生成规则：

(1) 矩阵元素必须在“[]”内；

(2) 矩阵的同行元素之间用空格或逗号(,)隔开；

(3) 矩阵的行与行之间用分号(;)或回车符隔开；

(4) 矩阵的元素既可以是数值、变量、表达式或函数，也可以是实数，甚至是复数。

(5) 矩阵的尺寸不必预先定义。

矩阵要求：
1、矩阵元素类型必须一致
2、矩阵每一个维度的尺寸必须相等

1、直接输入法

2、数列生成法

matTwo = [1:2:5;2:2:6];

matTwo =

1	3	5
2	4	6

3、矩阵合成法

mat1 = [1 3];

mat2 = [2 4];

mat = [mat1 5;mat2 6]

4、矩阵重构法

vec1 = 1:6； mat = reshape(vec1,2,3)

5、函数生成法(zeros,ones, eye,rand,randn,diag,triu,tril)

6、文件载入法（load, xlsread, csvread, …）

diag(A)：抽取矩阵A的主对角线元素。

```
>> A =[1 4 7 10; 2 5 8 11;3 6 9 12]
A =
 1   4   7   10
 2   5   8   11
 3   6   9   12
```

```
>> diag(A)
ans =
1
5
9
```

tril(A)：抽取矩阵A的主下三角元素

```
>> tril(A)
   ans =
      1   0   0   0
      2   5   0   0
      3   6   9   0
```

triu(A)：抽取矩阵A的主上三角元素

```
>> triu(A)
   ans =
      1   4   7   10
      0   5   8   11
      0   0   9   12
```

repmat(A,m,n)或repmat(A,[m n])：用A的拷贝m × n份的重构一个大矩阵B。

```
>> >> B = repmat(A(1:2,1:2),2,3)
B =
  1   4   1   4   1   4
  2   5   2   5   2   5
  1   4   1   4   1   4
  2   5   2   5   2   5
```

flipud (A)：将A矩阵的元素按列进行上下反转。

```
>> B = flipud(A)
B =
    3    6    9   12
    2    5    8   11
    1    4    7   10
```

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

fliplr (A)：将A矩阵的元素按列进行左右反转。

```
>> B = fliplr(A)
B =
   10    7    4    1
   11    8    5    2
   12    9    6    3
```

rot90(A,k)：将矩阵A的元素逆时针旋转k × 90°，k是一个整数。

```
>> B = rot90(A,2)
B =
   12    9    6    3
   11    8    5    2
   10    7    4    1
```

2 矩阵的寻址与赋值

2.1 向量的寻址与赋值

向量中各元素可以用单下标来寻址。

A(j)：向量A的第j个元素，首元素的索引值为**1**。

例如：

```
>>  vec1=10:5:60
vec1 =
   10   15   20   25   30   35   40   45   50   55   60
>> vec1(3)      → ans =   20
vec1(3:5) = 1:3
vec1 =
   10   15    1    2    3   35   40   45   50   55   60
```

2.2 矩阵的寻址与赋值

1 矩阵(数组)的下标索引

对于二维数组，其下标可以是按列排序的单下标A(k)，如图1所示；也可以是按行、列顺序编号的双下标A(i,j)，如图2所示。

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

图1 单下标表示

1, 1	1, 2	1, 3	1, 4
2, 1	2, 2	2, 3	2, 4
3, 1	3, 2	3, 3	3, 4
4, 1	4, 2	4, 3	4, 4

图2 双下标表示

(1) 使用双下标来进行矩阵的索引

在矩阵A 中，位于第i行、第j列的元素可表示为 A(i, j)，i与j即是此元素的下标(Subscript)或索引(Index)。例如：

```
>> A=[4 10 1 6 2;8 2 9 4 7;7 5 7 1 5;0 3 4 5 4;23 13 13 0 3]
   A =
      4   10    1    6    2
      8    2    9    4    7
      7    5    7    1    5
      0    3    4    5    4
     23   13   13    0    3
   >> A(2,2)
   ans = 2
   >> A(4:5,2:3)：取出矩阵A的第4、5行与2、3列所形成的部分矩阵。
   ans =
      3    4
     13   13
```

(2) 使用单下标进行矩阵的索引

用一维下标的方式可达到同样目的。对于某一个元素A(i, j)，其对应的单下标表示为A(k)，其中k = i+(j-1)*m，m为矩阵A的行数。例如：

```
>> A(7)
ans = 2
>> A([9 14; 10 15])
ans =
    3    4
   13   13
```

```
A =
    4   10    1    6    2
    8    2    9    4    7
    7    5    7    1    5
    0    3    4    5    4
   23   13   13    0    3
```


（3）使用冒号表达式选择行、列或数组元素

冒号表达式可以用来寻访、提取向量、数组或矩阵元素。

1) A(i:j)：是寻访A 的第i～j个元素，从i开始、以1作为增量，单下标寻访直到j。

例如：

```
>> vec1(1:5)  %返回向量vec1的第1到第5个元素。
ans =
   10   15   20   25   30
>>  A(1:7)
ans =
    4    8    7    0   23   10    2
```

A(i:k:j)：从i开始寻访，以k作为增量，直到j。

2) 使用冒号可取出一整列或一整行

A(i,:)：是寻访A 的第i行。例如：

>> A(3,:)

ans = 7　5　7　1　5

A(:,j)：是寻访A 的第j列。例如：

>> A(:, 5)：取出矩阵 A 的第5列。

ans =

2

7

5

4

3

3) A(:)：依次提取矩阵A的每一列，按单下标次序将A拉伸为一个列向量，即把A的所有元素视为单一列。不论原数组A是多少维的，A(:)将返回一个列向量。例如：

>> A(:)

ans =

4

8

7

……

5

4

3

A =

4	10	1	6	2
8	2	9	4	7
7	5	7	1	5
0	3	4	5	4
23	13	13	0	3

4) 取矩阵A的第i1～i2行、第j1～j2列构成新矩阵：A(i1:i2, j1:j2)。

```
>> A(2:3,1:3)
ans =
    8    2    9
    7    5    7
```

```
A =
     4    10     1     6     2
     8     2     9     4     7
     7     5     7     1     5
     0     3     4     5     4
    23    13    13     0     3
```

A(:,:)相当于二维数组，等同于A。

例如：A(:,1)将提取A矩阵的第1列，而A(1:2,1:2:5)将提取A的前2行与1,3,5列组成的子矩阵(起始值s1=1、步距s2=2、终止值s3=5)。

```
>>  A(:,1)
ans =
    4
    8
    7
    0
   23
>> A(1:2,1:2:5)
ans =
    4    1    2
    8    9    7
```

```
A =
     4    10     1     6     2
     8     2     9     4     7
     7     5     7     1     5
     0     3     4     5     4
    23    13    13     0     3
```

5) A(k: -i: j)是指按逆序返回A的各元素值。例如：以逆序提取矩阵A的第i1～i2行，构成新矩阵：A(i2: -1:i1，:)。

```
>>  A(3: -1:2,1:3)
ans =
    7    5    7
    8    2    9
>> A(3:-1:2,:)
ans =
    7    5    7    1    5
    8    2    9    4    7
```


（4）使用end关键字

关键字end表示数组的最后一个元素，代表某一维度的最大值，在矩阵元素提取时还可以使用end这个关键字。

A(:, end)：矩阵A的最后一列。例如：

```
>> B=[1 2 3;4 5 6]
B =
    1    2    3
    4    5    6
>> B(:, end)
ans =
    3
    6
```

B(i:end,:)将提取B的第i行到最后一行的所有列构成的子矩阵。例如寻访向量vec1的除前4个之外的所有元素，即从第5个元素开始到最后：

```
>> vec1(5:end)
ans =
   30   35   40   45   50   55   60
>>  A(2:end,:)
ans =
    8    2    9    4    7
    7    5    7    1    5
    0    3    4    5    4
   23   13   13   0    3
```


（5）矩阵元素的删除

可以直接删除矩阵的某一整个列或行，具体方法如下：

(1) A(2, :) = []：删除A矩阵的第2行。

(2) A(:, [2 4 5]) = []：删除 A 矩阵的第2、4、5列。

(3) 删除A的第i1～i2行，构成新矩阵:A(i1:i2, :)=[ ]。

(4) 删除A的第j1～j2列，构成新矩阵:A(:, j1:j2)=[ ]。

2.3 矩阵元素的赋值

1．全元素赋值方式

对矩阵(数组)中所有元素进行赋值。

例 创建一个(2*4)的全零数组，然后从1～8给其赋值。

解 (1) 创建一个(2*4)的全零数组。

```
>> A=zeros(2,4)
A =
    0    0    0    0
    0    0    0    0
```

(2) 从1～8给其赋值。

```
>> A(:)=1:8
A =
    1    3    5    7
    2    4    6    8
```


2．单下标方式赋值

例 将上例中下标为2、3、5的元素分别赋值为10、20、30。

解 该例当然可以使用下标寻址的方式，逐个赋值，例如：

```
>> A(2)=10
A =
    1    3    5    7
   10    4    6    8
```

或 用矩阵下标进行索引赋值

```
>>A([2 3 5])=[10 20 30];
A =
    1   20   30    7
   10    4    6    8
```

3．双下标方式赋值

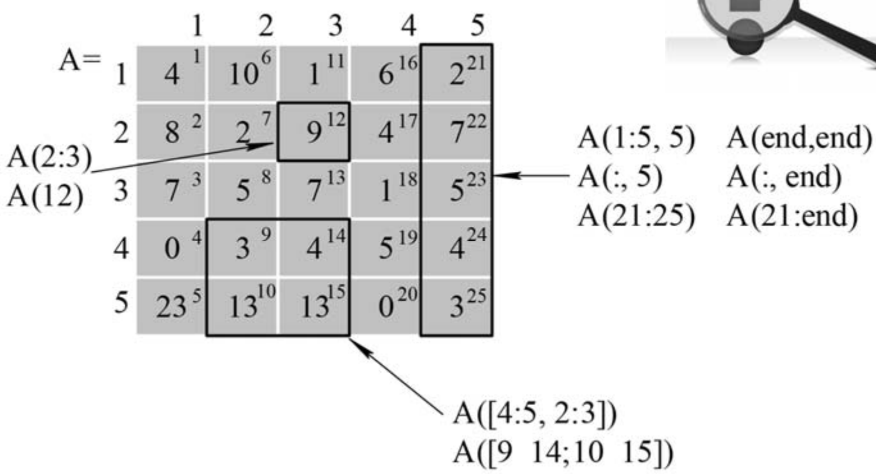
把A的第2、3列元素全赋为1。

```
>> A(:,[2 3])=ones(2)
A =
     1     1     1     7
    10     1     1     8
```

或者

```
>> A(:,[2 3])=[1 1;1 1]
```


下图中有**3处**错误，请指出。



主观题　10分

设置

问题：怎么创建任意数为底的等比数列？
例如：3，3*4, 3*4^2,….,3*4^10

作答

正常使用主观题需2.0以上版本雨课堂

3 标准矩阵与特殊矩阵

表 2-1　常用的标准矩阵和特殊矩阵

函　数	说　明
zeros(m, n)	产生维度为 m × n，构成元素全为 0 的矩阵
ones(m, n)	产生维度为 m × n，构成元素全为 1 的矩阵
eye(n)	产生维度为 n × n，对角线的各元素全为 1，其他各元素全为 0 的单位矩阵
pascal(m, n)	产生维度为 m × n 的 Pascal 矩阵
vander(m, n)	产生维度为 m × n 的 Vandermonde 矩阵
hilb(n)	产生维度为 n × n 的 Hilbert 矩阵
rand(m, n)	产生[0, 1] 均匀分布的随机数矩阵，其维度为 m×n
randn(m, n)	产生 $\mu = 0, \sigma = 1$ 的正规分布随机数矩阵，其维度为 m×n
magic(n)	产生维度为 n × n 的魔方阵，其各个直行、横列及两对角线的元素和都相等
diag()	生成对角矩阵
triu()、tril()	生成上、下三角矩阵
compan()	伴随矩阵

3.1 标准矩阵

由于标准矩阵具有通用性，MATLAB提供了一些专用矩阵函数来创建它们，标准矩阵一般包括全1矩阵、全0矩阵、单位矩阵、随机矩阵及对角矩阵等。

1．全1矩阵

ones()函数：产生全为1的矩阵。

(1) ones(n)：产生n × n维的全1矩阵。

(2) ones(m,n)、ones([m n])：产生m × n维的全1矩阵。例如：

```
>> ones(2,3)
ans =
    1    1    1
    1    1    1
```


2．全0矩阵

zeros()函数：与ones()函数类似，产生全为0的矩阵。

3．随机矩阵

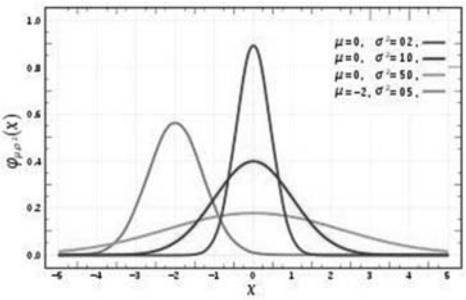
(1) rand()函数：产生在(0,1)区间均匀分布的随机矩阵。例如：

```
>> rand(2,3)
ans =
   0.9058    0.9134    0.0975
   0.1270    0.6324    0.2785
```

(2) randn()函数：产生均值为0，方差为1的标准正态分布随机矩阵。例如：

```
>> randn()
ans =
   0.3426
>> randn(2,3)
ans =
   3.5784   -1.3499    0.7254
   2.7694    3.0349   -0.0631
```

$$p(y)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(y-\mu)^2}{2\sigma^2}}$$



（3）随机整数序列的生成

方法一

rand(size1,size2)产生的随机数进行取整- fix,floor, ceil, round

例如

产生-5~5之间随机分布的5个整数：

```
>>round(rand(1,5)*10-5,0)
ans =
    0    -4    -3     4    -3
```

方法二

使用unidrnd（n,size1,size2）生成随机整数矩阵

例：

```
>> unidrnd(11,1,5)-6
ans =
    -4     5    -5     3     3
```

方法三

使用randperm(N,k)生成不大于N的k个不重复的随机整数向量

例：

```
>> randperm(11,5)-6
ans =
     5     1    -3    -2     4
```

4．单位矩阵

对角元素为1，其余元素为零的n阶方阵称为n阶单位矩阵，记为In或简写为I。

eye()函数：产生单位矩阵。使用为eye(m,n)或eye(n)。例如：

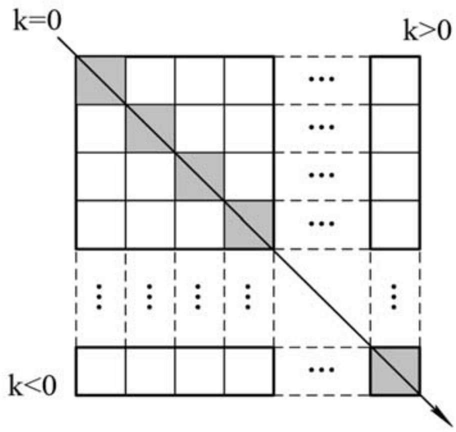
```
>> I5= eye(5)
ans =
     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1
```

5．对角矩阵

diag()函数：产生对角矩阵。例如：

X = diag(v,k)

当v是一个n元素的向量时，返回n+abs(k)阶的X方阵，v的元素排列在与主对角线平行的第k个元素的对角线上，如图所示。



当k = 0时，各元素出现在主对角线上。

当k > 0时，各元素位于对角线上方。

当k < 0时，各元素位于对角线下方。

例如：

```
>>v=[1 2 4 7 9];
>> X = diag(v,0)
X =
    1    0    0    0    0
    0    2    0    0    0
    0    0    4    0    0
    0    0    0    7    0
    0    0    0    0    9
```

```
>> X = diag(v, -2)
X =
    0    0    0    0    0    0    0
    0    0    0    0    0    0    0
    1    0    0    0    0    0    0
    0    2    0    0    0    0    0
    0    0    4    0    0    0    0
    0    0    0    7    0    0    0
    0    0    0    0    9    0    0
```

填空题 9分

设置

此题未设置答案，请点击右侧设置按钮

现有矩阵：

A=[17 24 1 8 15
23 5 7 14 16
4 6 13 20 22
10 12 19 21 3
11 18 25 2 9]

请给出对以上矩阵进行寻址的结果，

A(2,3) = [填空1] ; A(12)= [填空2] ;
A([1,3])= [填空3] ;A(1:3)= [填空4] ;
A(1:5,5)= [填空5] ;A(:,5)= [填空6] ;
A(23:end)= [填空7] ;A(4:5,2:3)= [填空8] ;
A([9 10;14 15])= [填空9]

作答

正常使用填空题需3.0以上版本雨课堂

4 基本的四则运算

向量、数组的四则运算法则总结如表1所示，而矩阵的四则算术运算有些与此不同。

表 1 向量、数组的运算法则

元素对元素的运算	例 A=[a1 a2 …an] B=[b1 b2 …bn] c(标量)
标量加减法	A±c=[a1±c a2±c … an±c]
标量乘法	A*c=[a1*c a2*c … an*c]
标量除法	A/c= [a1/c a2/c … an/c]
数组加减法	A±B=[a1±b1 a2±b2 … an±bn]
数组乘法	A.*B=[a1*b1 a2*b2 … an*bn]
数组右除法	A./ B=[a1/b1 a2/b2 … an/bn]
数组左除法	A.\ B=[b1/a1 b2/a2 …bn/an]
数组乘方	A.^c=[a1^c a2^c … an^c] c.^ A =[c^a1 c^a2 … c^an] A.^B=[a1^b1 a2^b2 … an^bn]

4.1 向量、数组与数的四则运算

1．向量与数的加法(减法)

对向量中的每个元素与数进行加法(减法)运算。例如：

```
>> v1=80: -9:10
v1 =
   80   71   62   53   44   35   26   17
>> v1+101
ans =
  181  172  163  154  145  136  127  118
```


北京邮电大学

2．向量与数的乘法(除法)

对向量中的每个元素与数进行乘法(除法)运算。 例如：

```
v1 =
   80   71   62   53   44   35   26   17

>> v1*2
ans =
  160  142  124  106   88   70   52   34
```

北京邮电大学

3．数组与数之间的四则运算

数组与数之间的运算(或叫标量、数组运算)，与向量运算规则相同，即数组的每个元素分别与数进行运算。例如：

```
>> s=[1 2 3;8 5 2]
s =
     1     2     3
     8     5     2
>> S=s-2
S =
    -1     0     1
     6     3     0
>> H=2*s/3+1
H =
   1.6667    2.3333    3.0000
   6.3333    4.3333    2.3333
```


4.2 向量、数组之间的四则运算

向量中的每个元素与另一个向量中相对应的元素进行四则运算，**两个向量的长度必须相同**。例如：

```
>> ve1=linspace(200,500,7)
ve1 =
   200   250   300   350   400   450   500
>> ve2=linspace(90,60,7)
ve2 =
    90    85    80    75    70    65    60
>> ve3=ve1+ve2
```

ve3 =

290 335 380 425 470 515 560

```
>> ve4=ve1.*ve2
ve4 =
   18000   21250   24000   26250   28000   29250   30000
>> ve5=ve1./ve2
ve5 =
   2.2222   2.9412   3.7500   4.6667   5.7143   6.9231   8.3333
>> ve6=ve1.\ve2
ve6 =
   0.4500   0.3400   0.2667   0.2143   0.1750   0.1444   0.1200
```

4.3 矩阵的乘法

假定有两个矩阵A和B，若A为m × n矩阵，B为p × q矩阵。当n=p时，B为n × q矩阵，则两个矩阵可以相乘，即后面矩阵B的行数必须与前面矩阵A的列数相同，二者可以进行乘法运算，否则是错误的。结果矩阵C=A × B为m × q矩阵。

矩阵乘法不可逆，在MATLAB中，矩阵乘法由(*)实现。

根据线性代数知识，矩阵乘法规则为：

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}\begin{pmatrix} j & m \\ k & n \\ l & o \end{pmatrix} = \begin{pmatrix} aj+bk+cl & am+bn+co \\ dj+ek+fl & dm+bn+co \\ gj+hk+il & gm+hn+io \end{pmatrix}$$

例如：

A=[1 2 3; 4 5 6]; B=[1 2 ; 3 4;5 6]; C=A × B，结果为

$$C=\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}\times\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}=\begin{bmatrix} 1\times1+2\times3+3\times5 & 1\times2+2\times4+3\times6 \\ 4\times1+5\times3+6\times5 & 4\times2+5\times4+6\times6 \end{bmatrix}=\begin{bmatrix} 22 & 28 \\ 49 & 64 \end{bmatrix}$$

1) 标量与矩阵相乘

与数组一样，标量与矩阵相乘，即把标量与每个元素相乘。

上例中，如果A或B是标量，则A × B返回标量A(或B)乘上矩阵B(或A)的每一个元素所得的矩阵。例如：

```
>> 6*A
ans =
   6   12
  18   24
```

2) 矩阵之间的乘法

矩阵之间的乘法与数组的点乘法不同，主要区别如下：

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

矩阵乘法结果：

```
>> C=A*B
C =
   12   15   18
   24   30   36
   36   45   54
```

数组乘法结果：

```
>> D=A.*B
D =
    1    2    3
    8   10   12
   21   24   27
```

4.4 矩阵的除法

在MATLAB中，有两种矩阵除法符号，即左除“\”和右除“/”。如果A矩阵是非奇异方阵，则A\B和B/A运算可以实现：

(1) A\B：等效于A的逆左乘B矩阵，也就是A\B = inv(A)*B。

(2) B/A：等效于A矩阵的逆右乘B矩阵，也就是B/A = B*inv(A)。

对于矩阵来说，左除和右除表示两种不同的除数矩阵和被除数矩阵的关系，一般A\B≠B/A。但对于含有标量的运算，两种除法运算的结果相同。

通常：

x=A\B就是A*x=B的解；

x=B/A就是x*A=B的解。

例

$$\begin{cases} 2x_1+x_2+x_3=0 \\ x_1+2x_2+x_3=3 \\ x_1+x_2+2x_3=2 \end{cases}$$

$$\begin{cases} a_{11}x_1+\cdots+a_{1n}x_n=b_1 \\ \vdots \\ a_{n1}x_1+\cdots+a_{nn}x_n=b_n \end{cases}$$

$$x=\frac{D_i}{D}, \quad D=\det(A)$$

$$D_i=\det\begin{pmatrix} a_{11} & \cdots & a_{1i-1} & b_1 & a_{1i+1} & \cdots & a_{1n} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & \cdots & a_{ni-1} & b_n & a_{ni+1} & \cdots & a_{nn} \end{pmatrix}$$

5 向量、数组和矩阵的其他运算

5.1 乘方、开方运算

1．向量、数组的乘方运算与power()函数

符号(.^)或power()函数是数组用来执行元素对元素的乘方运算的，当乘方指数是一个标量时，该标量对数组的所有元素进行取乘方操作。数组乘方运算的语法如下：

c = a.^k或 c = power(a,k)：计算$c=a^k$，k是实数。

思考：开方运算怎么表示？

例如：

```
>> g=[1 2 3;4 5 6]
>> g.^2
ans =
     1     4     9
    16    25    36
>> g.^-2
ans =
    1.0000    0.2500    0.1111
    0.0625    0.0400    0.0278
```

2．矩阵的乘方与mpower()函数

与数组的指数运算不同，一个矩阵的乘方运算可以表示成A^P，即A自乘P次。要求A必须为方阵，P为标量。语法如下：

C=A^P或C=mpower(A,P)。

```
>> A =[1    2    3
     4    0    6
     7    8    9];
>> A^2  %等于A*A
ans =
      30    26    42
      46    56    66
     102    86   150
```

5.2 求极小值与极大值

1．min()函数

min()函数用于求极小值，其用法如下：

(1) C = min(A)：返回数组A中的不同维度的最小元素。

- 如果A是一个向量，返回A中的最小元素。
- 如果A是一个矩阵(数组)，按**列**返回该列向量中的最小元素。

(2) C = min(A,B)：返回数组A、B中的相同维度的最小元素，A、B的维数必须相同。

(3) C = min(A,[],dim)：返回数组A中dim指定的维数(列)中的最小元素。

- dim=1，生成行向量，每个元素为按列返回该列向量中的最小元素。
- dim=2，生成列向量，每个元素为按行返回该行向量中的最小元素。

(4) [C,I] = min(...)：C中是返回的最小元素值，I向量是返回的最小元素的位置号。

2．max()函数

max()函数用于求极大值，用法与min()函数相同。

例如：

```
>> B=[1 2 3 0;2 2 2 1;3 0 5 8]
B =
    1    2    3    0
    2    2    2    1
    3    0    5    8
>> min(B,[],1)
ans =
    1    0    2    0
>> min(B,[],2)
ans =
    0
    1
    0
```

```
>> [C,I] = min(B)
C = 1    0    2    0
 I = 1    3    2    1
>> [C,I] = min(B,[],2)
C =
      0
      1
    0
I =
      4
      4
      2
>> [C,I] = max(B)
C = 3    2    5    8
 I = 3    1    3    3
```

5.3 mean()函数求平均值

mean()函数用于求平均值，其用法如下：

(1) C = mean(A)：返回数组A中的不同维度的平均值。

- 如果A是一个向量，返回A中所有元素的平均值。
- 如果A是一个矩阵(数组)，按列返回该列向量中所有元素的平均值。

(2) C = mean(A,dim)：根据dim指定的维数，返回数组A中所有元素的平均值。

- dim=1，按列求平均值，生成行向量C。
- dim=2，按行求平均值，生成列向量C。

例如：


```
>> h=[1 1 1;2 2 2;3 3 3]
h =
    1    1    1
    2    2    2
    3    3    3
>> mean(h,1)
ans =
    2    2    2
>> mean(h,2)
ans =
    1
    2
    3
```

5.4 求和

sum()函数用于求和，其语法如下：

(1) C = sum(A)：返回数组A中的不同维度元素的和。

- 如果A是一个向量，则返回向量A中所有元素的和B。
- 如果A是一个矩阵(数组)，按**列**返回该列向量中所有元素的和B。B是一个行向量，元素是A的列元素的和。

例如：

```
>> a=[1 2 3 4];
>>  B = sum(a)
B = 10
>> b=[1 2 3 4;1 2 3 4 ]
b =
    1     2     3     4
    1     2     3     4
>> sum(b)
ans = 2     4     6     8
        9
```

(2) B = sum(A,dim)：按标量dim指定的维数，返回数组A中所有的元素的和B。

- dim=1：按列求和，即sum(A,1)沿列累加，生成行向量C。
- dim=2：按行求和，即sum(A,2)沿行累加，生成列向量C。

例如：

```
>> A = [1 2 3; 4 5 6]
A =
    1    2    3
    4    5    6
>> sum(A,1)
ans =
    5    7    9
>> sum(A,2)
ans =
    6
   15
```


5.5 求积

prod()函数用于求积，其语法如下：

(1) B = prod(A)：返回数组A的元素的积B。

- 如果A是一个向量，返回A中所有元素的积。
- 如果A是一个矩阵(数组)，则按列求积，B是一个行向量。

(2) B = prod(A,dim)：返回数组A的元素的积B。标量dim指定累积的方向。

- dim=1，即prod (A,1)按列求积，生成行向量C。
- dim=2，即prod (A,2)按行求积，生成列向量C。

例如：

```
>> M = magic(3)
M =
     8     1     6
     3     5     7
     4     9     2
>> prod(M)
ans =
    96    45    84
>> prod(M,2)
ans =
    48
   105
    72
```

填空题 6分

设置

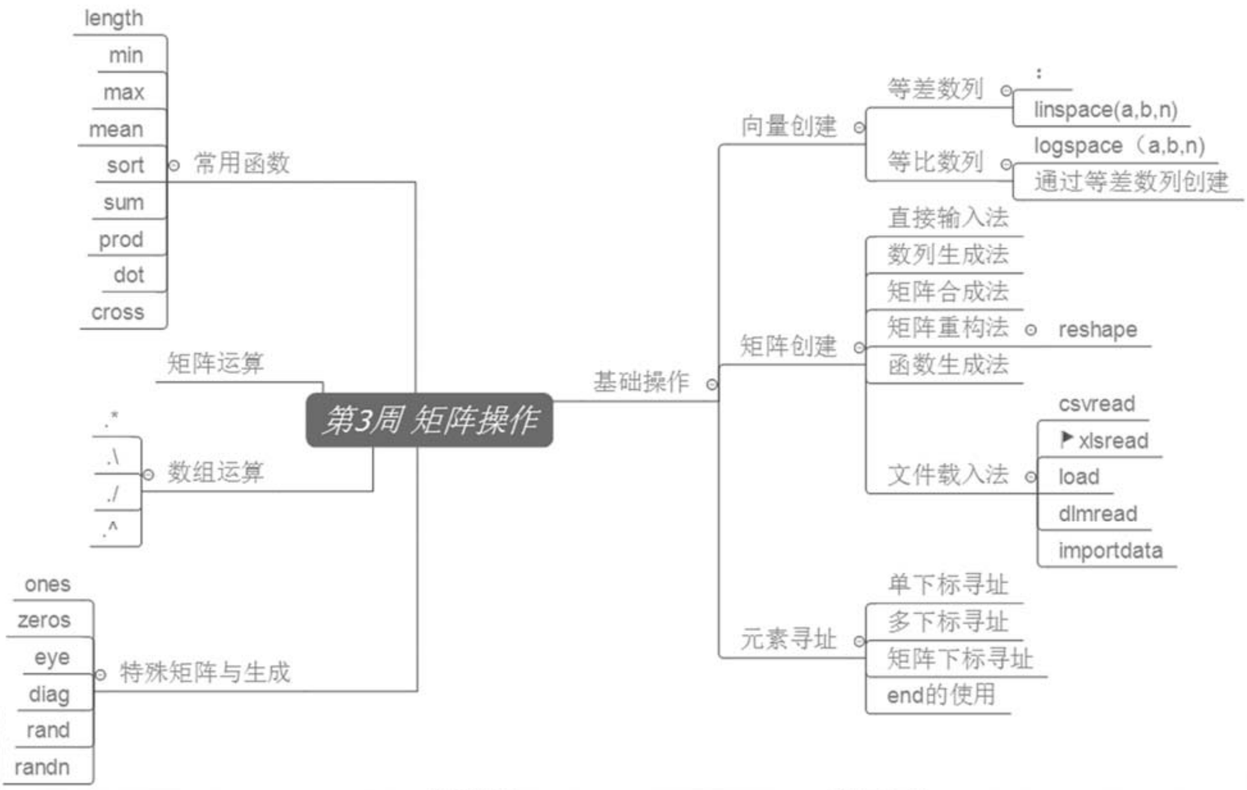
填入下列语句输出的解雇

```
clc
clear
A = [2 5 7 1 3 4];
odds = 1:2:length(A);
A(odds)                      [填空1]
B(odds) = A(2:2:end)         [填空2]
B(2:2:end) = 9               [填空3]
small = A <4                 [填空4]
A(small) = A(small)+10       [填空5]
A(A<10) = A(A<10) + 10       [填空6]
```

正常使用填空题需3.0以上版本雨课堂

作答

课程小结



练习：

1、生成全零矩阵

z =

$$\begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix}$$

2、生成全1矩阵

A =

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

3、生成矩阵

B=

$$\begin{matrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{matrix}$$

4、生成C矩阵

C =

$$\begin{matrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{matrix}$$

5、获取C的对角向量

D =

$$\begin{matrix} 0 \\ 1 \\ 1 \end{matrix}$$

6、利用D生成如下矩阵

E =

$$\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{matrix}$$

7、生成0-1之间均匀分布的6个随机数组成的向量

F =

0.8147 0.9058 0.1270 0.9134 0.6324 0.0975

8、生成2-6之间均匀分布的4个随机数组成的向量

F =

3.1140 4.1875 5.8300 5.8596

9、两个维数一致的矩阵，分别提取对应位置较大的元素和较小的元素，组成两个新矩阵。

A =

$$\begin{matrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{matrix}$$

B =

$$\begin{matrix} 9 & 6 & 3 \\ 8 & 5 & 2 \\ 7 & 4 & 1 \end{matrix}$$

作业：

1、编写代码构建向量randNum，包含10个1~12之间的随机数(随机数服从均匀分布)

2、假设已有两个相同长度的向量A和B，

A=round(rand(1,10)*10,0);

B=round(rand(1,10)*10,0);

要求不使用直接输入法，通过编写脚本实现创建新向量C，使其包含A和B的所有元素，格式为C=[A(1) B(1) A(2) B(2)…A(end) B(end)]。

3、用尽量简单的代码完成下题：已有数字向量A，计算该向量中所有正数的立方，并将结果保存在新的向量B中。如果A的元素是负值，则用0表示其立方。例如，A= [1 2 -1 5 6 7 -4 3]，则B= [1 8 0 125 216 343 0 27]