

第2讲 MATLAB基础

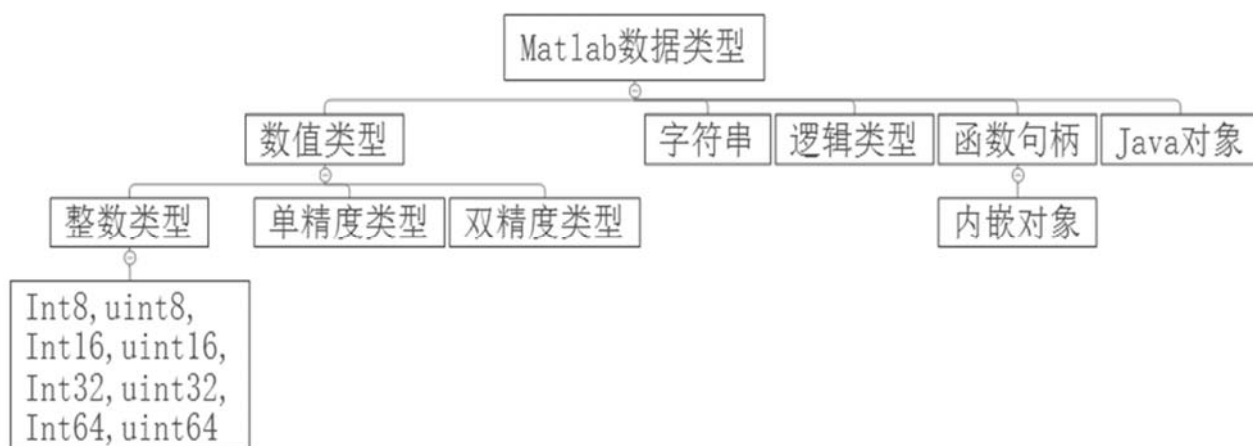
1 数据类型

2 数据结构

3 基本运算



1 数据类型



1.1 整数

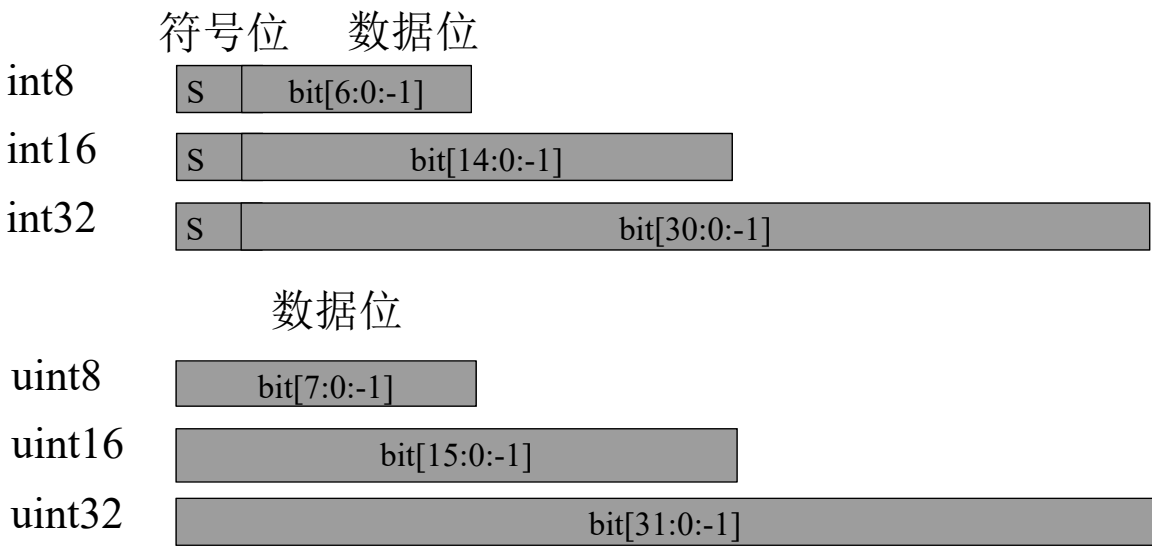
MATLAB R2019a支持8位、16位、32位和64位的有符号和无符号整数数据类型，如下表所示。

表 整数数据类型

| 数据类型 | 描 述 |
|--------|---|
| uint8 | 8 位无符号整数，范围为 0~255(即 $0\sim2^8-1$) |
| int8 | 8 位有符号整数，范围为 -128~127(即 $-2^7\sim2^7-1$) |
| uint16 | 16 位无符号整数，范围为 0~65 535(即 $0\sim2^{16}-1$) |
| int16 | 16 位有符号整数，范围为 -32 768~32 767(即 $-2^{15}\sim2^{15}-1$) |
| uint32 | 32 位无符号整数，范围为 0~4 294 967 295(即 $0\sim2^{32}-1$) |
| int32 | 32 位有符号整数，范围为 -2 147 483 648~2 147 483 647(即 $-2^{31}\sim2^{31}-1$) |
| uint64 | 64 位无符号整数，范围为 0~18 446 744 073 709 551 615(即 $0\sim2^{64}-1$) |
| int64 | 64 位有符号整数，范围为 -9 223 372 036 854 775 808~9 223 372 036 854 775 807 (即 $-2^{63}\sim2^{63}-1$) |



整形数据存储结构



例1 在整数运算中的数据溢出问题。

解 语法如下：

```
>> x=int8(100);
```

```
>> y=int8(90);
```

```
>> z=x+y
```

```
z =
```

```
127
```

结果(190)溢出上限，因此输出结果为上限(127)。

```
>> x-3*y
```

```
ans =
```

```
-27
```

3*y结果为(270)溢出上限，结果为127，继续计算(100-127)，得到最后结果 -27。

```
>> x-y-y-y
```

```
ans =
```

```
-128
```

计算x-y-y-y时，从左到右进行计算，结果溢出下限，因此结果为-128。

1.2 浮点数与精度函数

MATLAB的默认数据类型是双精度数值类型(double)。

在不同计算机系统上运行的MATLAB中，其单精度与双精度数据类型与取值范围会有所不同，这与硬件有关。具体的单精度和双精度数据类型的取值范围和精度，可以通过realmin()、realmax()、eps()函数进行查看。

(1) realmin()函数。该函数返回MATLAB语言能够表示的最小的归一化正浮点数，任何小于该数的都不是规范的IEEE标准，都会发生溢出。

(2) `realmax()`函数。该函数返回MATLAB语言中能够表示的最大的归一化正浮点数，任何大于该数的数都不是规范的IEEE标准，都会发生溢出。

类似的函数还有`intmax()`和`intmin()`：`intmax()`表示返回指定的整数数据类型能表示的最大的正整数；`intmin()`表示返回指定的整数数据类型能表示的最小的整数。



例2 举例说明在MATLAB中单精度浮点数和双精度浮点数数据类型的取值范围和精度的规定。

解 (1) 函数：

```
>> intmax('int32')
```

```
ans = 2147483647
```

```
>> intmin('int32')
```

```
ans = -2147483648
```



(2) 单精度浮点数:

```
>> realmin('single')
```

```
ans = 1.1755e-038
```

```
>> realmax('single')
```

```
ans = 3.4028e+038
```

(3) 双精度浮点数:

```
>> n = realmin
```

```
n = 2.2251e-308
```

```
>> realmax
```

```
ans = 1.7977e+308
```



eps()函数

MATLAB中还存在一个用双精度表示的浮点相对误差限`eps`，定义为1与大于1的最小数之间的步进距离，用`eps`获得。

(1) `eps`: 返回从1.0到下一个最大的双精度数的距离， $\text{eps} = 2^{(-52)}$ 。
例如:

```
>> eps
```

```
ans =
```

```
2.2204e-016
```

(2) `eps('double')`: 等同于`eps`或`eps(1.0)`。例如:

```
>> eps('double')
```

```
ans =
```

```
2.2204e-016
```



(3) `eps('single')`: 等同于`eps(single(1.0))`或`single(2^-23)`。

例如:

```
>> eps('single')  
ans =  
1.1921e-007
```



1.3 字符型

字符型数据通过单引号 ‘’ 表示，例如

```
>> str = 'I Love MATLAB';  
>> class(str)  
ans =  
'char'
```

```
>> str(1)  
ans =  
'I'  
>> str(end)  
ans =  
'B'
```

- ❖ 单引号中字符串实际上为字符（char）数组，可以通过字符位置进行索引；
- ❖ 双引号表示的字符串是string类型，不能直接通过索引获得其中某个字符，需要通过元胞数组访问方式进行访问。

```
>> Str = "I love MATLAB"  
Str =  
"I love MATLAB"  
>> Str(1)  
ans =  
"I love MATLAB"  
>> str2 = Str{1}  
str2 =  
'I love MATLAB'  
>> str2(1)  
ans =  
'I'
```



1.4 逻辑型

逻辑型数据往往产生于关系运算，表示关系运算的结果。逻辑性数据只有两个值：true，false。

可用logical（）函数将数值型数据转化为逻辑性，则所有非零数据对应true，零对应false。

```
>> a = logical(10)    >> a = logical(0)    >> re = true;
a =
    logical
         1
>> class(a)           >> a = logical(-1)   >> a == re
ans =
    'logical'          a =
    logical
         1
ans =
    logical
         1
```

1.5 复 数

我们把形如 $a+bi$ （ a, b 均为实数）的数称为复数，其中 a 称为实部， b 称为虚部， i 称为虚数单位。当虚部等于零时，这个复数可以视为实数；当 z 的虚部不等于零时，实部等于零时，常称 z 为纯虚数。

$i=j=\sqrt{-1}$ ，其值在工作间中都显示为 $0+1.0000i$ 。

在MATLAB中，可以通过两种方法创建复数：一种是直接输入法；另一种是使用complex()函数。

1. 直接输入法

直接输入法创建复数的示例如下：

```
>> c1=1-2i
```

```
c1 =
```

```
1.0000-2.0000i
```

```
>> c2=1+2j
```

```
c2 =
```

```
1.0000 + 2.0000i
```

```
>> c3=sqrt(-2)
```

```
c3 =
```

```
0 + 1.4142i
```



注意：只有数字才可以与i 或者 j连接，因此在使用表达式时，要乘以i或j来获得虚部。如：

```
>> c4=5+sin(.5)*j
```

```
c4 =
```

```
5.0000 + 0.4794i
```



2. 使用complex()函数

complex()函数的调用方法如下：

(1) $c = \text{complex}(a, b)$ 。返回结果 c 为复数，其实部为 a ，虚部为 b 。输入参数 a 和 b 可以是标量，或者是维数、大小相同矩阵。

```
>> c1=complex(1,2)
c1 =
      1.0000 + 2.0000i
```

```
>> c2=complex(1,-2)
c2 =
      1.0000-2.0000i
```



3. 复数的虚部和实部

imag()、real()函数表示分别返回复数的虚部和实部，如：

```
>> real(c2)
```

```
ans =
```

```
1
```

```
>> imag(c2)
```

```
ans =
```

```
-2
```

```
>> c2r=real(c2)
```

```
c2r =
```

```
1
```

```
>> c2i=imag(c2)
```

```
c2i =
```

```
-2
```



4. 复数的模、辐角和共轭复数

(1) 求复数的模使用abs()函数。Abs()其函数的MATLAB表达式为

$$\text{abs}(X) = \sqrt{\text{real}(X)^2 + \text{imag}(X)^2}$$

例如: $c1 = 1.0000 + 2.0000i$

```
>> abs(c1)
```

```
ans =
```

```
2.2361
```



(2) 求复数的辐角使用angle ()函数。

例如:

```
>> angle(c1)
```

```
ans =
```

```
1.1071
```

(3) 求共轭复数使用conj ()函数。例如, 复数 $Z = \text{real}(Z) + i \cdot \text{imag}(Z)$, 其共轭复数为

$$\text{conj}(Z) = \text{real}(Z) - i \cdot \text{imag}(Z)$$

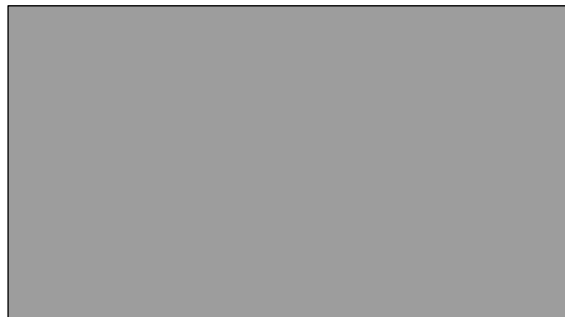
例如:

```
>> c2=conj(c1)
```

```
c2 = 1.0000-2.0000i
```



观看实操录屏视频



单选题 1分

 设置

此题未设置答案，请点击右侧设置按钮

`x = int8(10); y = 20; z = x + y;` 问：z是什么数据类型

- ☐ A int8
- ☐ B double



提交

复数运算

例1-4-1 求下列复数的实部和虚部、共轭复数、模和辐角。

$$(1) \frac{1}{3+2i}$$

$$(2) \frac{1}{i} - \frac{3i}{1-i}$$

$$(3) \frac{(3+4i)(2-5i)}{2i}$$

$$(4) i^8 - 4i^{21} + i$$

解 在MATLAB命令窗口中输入：

```
>> a=[1/(3+2i),1/i-3i/(1-i), (3+4i)*(2-5i)/2i, i^8-4*i^21+i];
```

```
>> real(a)
```

```
ans = 0.2308    1.5000   -3.5000    0
```

```
>> imag(a)
```

```
ans = -0.1538   -2.5000  -13.0000   -2.0000
```

```
>> conj(a)
```

```
ans = 0.2308 + 0.1538i    1.5000 + 2.5000i   -3.5000 +13.0000i    0 + 2.0000i
```

```
>> abs(a)
```

```
ans = 0.2774    2.9155   13.4629    2.0000
```

```
>> angle(a)
```

```
ans = -0.5880   -1.0304   -1.8338   -1.5708
```

例1-4-2 复数的指数和对数运算：

$$(1) \log(-i)$$

$$(2) \log(-1+3i)$$

解 在MATLAB命令窗口中输入：

```
>> log(-i)
```

```
ans = 0 - 1.5708i
```

```
>> log(-1+3i)
```

```
ans = 1.1513 + 1.8925i
```

2.2 数据结构

1. 矩阵

MATLAB最基本的数据结构是复数矩阵，在命令窗口输入一个复数矩阵非常简单，例如下面的语句：

```
>> B=[1+9i,2+8i,3+7j; 4+6j 5+5i,6+4i; 7+3i,8+2j 1i]
```

可输入一个矩阵B，矩阵B的各行元素由分号分隔，而同行中不同列元素由逗号或空格分隔，回车后显示的结果如下：

```
B =  
1.0000 + 9.0000i 2.0000 + 8.0000i 3.0000 + 7.0000i  
4.0000 + 6.0000i 5.0000 + 5.0000i 6.0000 + 4.0000i  
7.0000 + 3.0000i 8.0000 + 2.0000i 0 + 1.0000i
```

其中，元素 $1+9i$ 表示复数项，实矩阵、向量或标量均可更容易地以这样的表述方法输入。如果赋值表达式末尾有分号，则其结构将不显示，否则将显示出全部结果。

2. 多维数组

在MATLAB中数组、向量和矩阵的概念是经常混用的，事实上数组、向量和二维矩阵在本质上没有任何区别，都是以矩阵的形式保存的。MATLAB的数据结构只有矩阵一种形式(可细分为普通矩阵和稀疏矩阵)，但是数组与矩阵的某些运算方法是不同的。

3. 字符串与字符串矩阵

MATLAB的字符串是由单引号括起来的。如可以使用下面命令赋值：

```
>> strA='This is a string.'
```

4 单元数组 (cell array)

4.1 生成单元数组

“单元(cell)”(也称为“细胞”矩阵或者“细胞”数组)是无类型矩阵，它们中的元素可以是任何类型。

单元数组用类似矩阵的标记方法，将复杂的数据结构纳入一个变量之下，与矩阵中的圆括号表示下标类似，单元数组由大括号表示下标。单元数组中的每一个元素称为“单元”。单元中的数据可以为任何数据类型，包括数值数组、字符、符号对象、其他单元数组和结构体。

可以通过两种方式创建一个单元数组：一是通过赋值语句直接创建；二是利用`cell()`函数先为单元数组分配一个内存空间，然后再给各个单元赋值。



1. 直接生成单元数组

直接赋值法通过给每个单元逐个赋值来创建单元数组。单元数组用大括号表示，在赋值时需要将单元内容用大括号括起来。例如：

```
>> A={'中国','美国',100+200*i,[9, 8, 5; 67, 70, 102; 57, 18,  
100; -200, 89, 78]}  
A =  
    '中国'    '美国' [1.0000e+002 +2.0000e+002i] [4 × 3  
double]
```



2. 使用cell()函数生成单元数组

使用cell()函数创建单元数组的步骤为：首先用cell()函数创建一个空的单元数组，然后再为数组元素赋值。语法如下：

(1) `c = cell(n)`：生成一个 $n \times n$ 元素的空矩阵数组，如果 n 不是标量，将发出错误信息。例如：

```
>> b=cell(2)
```

```
b =
```

```
[] []
```

```
[] []
```



(2) `c = cell(m, n)`、`c = cell([m, n])`：生成一个 $m \times n$ 元素的空矩阵数组， m 、 n 必须是标量，否则将发出错误信息。例如：

```
>> B=cell(2,2)
```

```
B =
```

```
[] []
```

```
[] []
```

(3) `c = cell(m, n, p,...)`、`c = cell([m n p ...])`：生成一个 $m \times n \times p$ 元素的空矩阵数组， m 、 n 、 p 必须是标量，否则将发出错误信息。



(4) `c = cell(size(A))`: 生成一个与矩阵A大小相同的空矩阵数组。

例如:

A =

```
1  2  3
4  5  6
4  2  1
```

```
>> c = cell(size(A))
```

c =

```
[] [] []
[] [] []
[] [] []
```



4.2 单元数组的赋值

生成了空矩阵单元数组后，可以使用“按单元索引法”或“按内容索引法”为单元数组的元素赋值，这两种方法是完全等效的。

1. 按单元索引法

等号左边使用圆括号的是“按单元索引法”。例如:

```
>> b=cell(2)
```

b =

```
[] []
[] []
```

```
>> b(1,1)='OK'
```

??? Conversion to cell from char is not possible.



该语句出错，等号左边用圆括号，等号右边赋值时将值用大括号括起来，表明大括号中的表达式是单元数组元素的内容，而不是普通的数组或字符串，应使用以下形式：

$A(i,j)=\{x\}$

例如：

`>> b(1,1)={'OK!'}`

`b =`

`'OK!' []`

`[] []`

`>> b(2,2)={2+3i}`

`b =`

`'OK!' []`

`[] [2.0000 + 3.0000i]`

2. 按内容索引法

“按内容索引法”是把大括号写在等式左边，等式右边是要赋值的内容。“按内容索引法”应使用以下形式：

$A\{i,j\}=x$

例如：

`>> b{1,2}='China.'`

`b =`

`'OK!' 'China.'`

`[] [2.0000 + 3.0000i]`

`>> b{2,1}=[1 2 3;2 3 4]`

`b =`

`'OK!' 'China.'`

`[2x3 double] [2.0000 + 3.0000i]`

4.3 单元数组的内容显示

1. “按单元索引”或“按内容索引”

使用圆括号的“按单元索引”和花括号的“按内容索引”对单元数组索引是不同的。在MATLAB单元数组索引中，圆括号用于标志单元、花括号用于按单元的寻址；当采用圆括号时表示的是该单元，而采用花括号时则表示的是该单元的内容。

```
>> b{2,2}
ans =
    2.0000 + 3.0000i
>> b(2,2)
ans =
    [2.0000 + 3.0000i]
```

使用花括号的“按内容索引”可以显示完整的单元内容，而使用圆括号的“按单元索引”有时不能显示完整的单元内容。例如：

```
>> b(2,:)
ans =
    [2x3 double]    [2.0000 + 3.0000i]
>> b{2,:}
ans =
     1     2     3
     2     3     4
ans =
    2.0000 + 3.0000i
```

在显示单元数组时，MATLAB有时只显示单元的大小和数据类型，而不显示每个单元的具体内容。若要显示单元数组的内容，可以用 `celldisp()` 函数。

2. celldisp()函数

celldisp()函数是一个强制显示命令，无论单元数组有多少单元，也不论每个单元有多少内容，都将全部显示出来。celldisp()函数用于显示单元数组的全部内容，有时候只需要显示单元数组的一个单元，可以用花括号对单元进行索引。

(1) celldisp(C): 递归显示单元数组的内容。例如：

```
>> celldisp(b)
```

```
b{1,1} =
```

```
OK!
```

```
b{2,1} =
```

```
1 2 3
```

```
2 3 4
```

```
b{1,2} =
```

```
China.
```

```
b{2,2} =
```

```
2.0000 + 3.0000i
```

(2) celldisp(C, name): 使用字符串name作为名称显示单元数组的内容。例如：

```
>> celldisp(b, 'name1')
```

```
name1{1,1} =
```

```
OK!
```

```
name1{2,1} =
```

```
1 2 3
```

```
2 3 4
```

```
name1{1,2} =
```

```
China.
```

```
name1{2,2} =
```

```
2.0000 + 3.0000i
```

3. 单元数组的图形显示

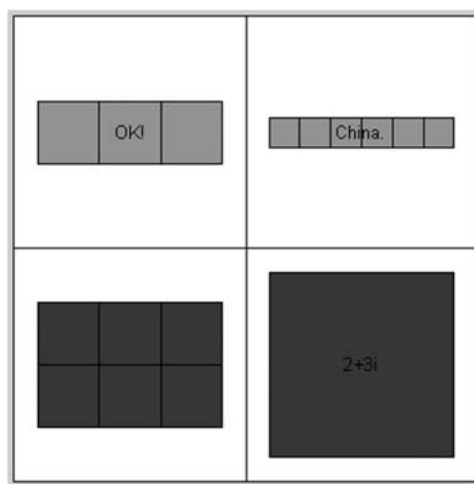
除上面的单元数组查看方式外，MATLAB还支持以图形方式查看单元数组的内容。用这种方法可

以直观地看出单元数组的结构，但需要注意的是，`cellplot()`函数只能用于显示二维单元数组的内容。

例如：

```
>> cellplot(b)
```

图形显示单元数组**b**的内容如图2-6所示。



图形显示单元数组**b**的内容



4.4 单元数组的内容获取

单元数组的内容的获取必须使用花括号的“按内容索引”对单元数组索引。例如将数组b的某单元内容赋值给x、y、z：

```
>> x=b{2,2}
```

```
x =
```

```
2.0000 + 3.0000i
```

```
>> y=b{2,1}
```

```
y =
```

```
1 2 3
```

```
2 3 4
```

```
>> z=b{1,2}
```

```
z =
```

```
China.
```



4.5 单元数组元素的删除

删除单元数组元素的方法很简单，只需将待删除的元素置为“空”。需要注意的是，在删除单元数组的元素时，采用的索引方式为一维下标，其格式为

```
A(cell_subscripts) = [ ]
```

如果操作的单元数组为多维数组，则其索引方式逐维进行，删除元素后，系统将该单元数组改变为一维单元数组，元素按照维数逐次排序。



4.6 单元数组的变维处理

改变数组的维数可以通过添加或删除数组元素完成。删除数组元素时，得到的单元数组为原数组中剩下元素排列而成为一维数组。添加数组元素时，自动添加该数组所对应的行和列，其他元素为空。

前面所述的对矩阵的变维处理同样也适用于对单元数组的变维处理。

另外，可以通过函数`reshape()`改变数组的形状。`reshape()`函数按照顺序将原单元数组的元素进行重新放置，得到新的单元数组元素个数与原数组相同。



例如：

```
>> A={ [1 2 ;3 4], 'I love you!'; '世界和平', [90, 85, 55; 67,  
70, 102; 57, 18, 100; -200, 89, 78]}
```

```
A =
```

```
 [2x2 double] 'I love you!'
```

```
 '世界和平' [4x3 double]
```

```
>> a=reshape(A,1,4)
```

```
a =
```

```
 [2x2 double] '世界和平' 'I love you!' [4x3 double]
```



5 结构体

5.1 结构体的生成

与单元数组类似，结构体也有两种生成方式：一种是直接输入；另一种是使用结构体生成函数`struct()`。

1. 直接输入法生成结构型变量

通过直接输入结构体各元素值的方法可以创建一个结构体。输入的同时定义该元素的名称，并使用“.”将变量名与元素名连接。例如：

```
>> student.test=[99 75 96 87 67 69 87 86 92];  
>> student.name='Hu Jing';  
>> student.weight=78;  
>> student.height=1.78;  
>> student.num=2010214091;
```

2. 使用`struct()`函数生成结构型变量

`Struct()`函数的最基本的使用方式是
`struct_name=struct('field1',V1,'field2',V2,...)`，其中，`fieldn` 是各成员变量名，`Vn`为对应的各成员变量的内容。例如：

```
>> truct_array=struct('d',{ '北京','上海'  
'},'strengths',[40000 1000])
```

```
truct_array =  
    d: { '北京' '上海'}  
    strengths: [40000 1000]
```

跟下面这句代码有什么不同？

```
>> truct_array=struct('d',{ '北京','上海'},'strengths',[40000 1000])
```

5.2 成员变量的操作

成员变量的操作包括添加、删除和调用。

1. 在结构体变量中添加成员变量

如果需要向结构体中添加新的成员，可以直接输入该变量的名称并赋值。例如：

```
>> student(1).gender='Male';    %在student中添加gender  
和age这两项记录
```

```
>> student(1).age=25;
```

```
>> student(2).gender='Female';
```

```
>> student(2).age=22;
```



2. 结构的查询

直接输入结构名称可进行结构的查询。例如：

```
>> student
```

```
%查询student的结构
```

```
student =
```

```
1×2 struct array with fields:
```

```
test
```

```
name
```

```
weight
```

```
height
```

```
num
```

```
gender
```

```
age
```



3. 删除成员变量

在结构体变量中删除成员变量。在MATLAB 中可以使用函数rmfield()从结构体中删除成员变量。语法如下：

(1) S=rmfield(S, 'field')：该命令将删除结构体S中的成员field，同时保留S原有的结构。

(2) S=rmfield(S,fields)：使用该命令可以一次删除多个成员，其中 fields 为字符行变量或者单元型变量。例如：



```
>> student=rmfield(student,'age');
```

```
>> student
```

```
student =
```

```
1 × 2 struct array with fields:
```

```
test
```

```
name
```

```
weight
```

```
height
```

```
num
```

```
gender
```



4. 调用成员变量

在MATLAB中调用成员变量非常简单。结构体中的任何信息可以通过“结构体变量名.成员名”的方式调用。调出成员变量后，可以利用相关函数进行调用。例如：

```
>> student(1).test      %从结构体变量中取出相关信息
```

```
ans =
```

```
99  75  96  87  67  69  87  86  92
```

```
>> student(1).test(9)
```

```
ans =
```

```
92
```



5. getfield()和setfield()函数的使用

(1) getfield()函数：取得当前存储在某个成员变量中的值。例如：

```
>> GETF=getfield(student(1),'name')
```

```
GETF =
```

```
Hu Jing
```



(2) setfield()函数: setfield(struct,'field',value)函数给某个成员变量field插入新的值value。例如:

```
>> student=setfield(student(1),'name','LiuFeng')
```

```
student =
```

```
test: [99 75 96 87 67 69 87 86 92]
```

```
name: 'LiuFeng'
```

```
weight: 78
```

```
height: 1.7800
```

```
num: 2.0102e+009
```

```
gender: 'Male'
```



MATLAB数据结构实操录屏



此题未设置答案，请点击右侧设置按钮

讨论：cell和struct的异同

- A 不同之处
- B 相同之处

提交



1.3 MATLAB的基本特性

1.3.1 数学运算

MATLAB用于数学计算的数学运算符如下表所示。

表 数学运算符

| 符 号 | 功 能 | 实 例 |
|------------|-----------|-------------|
| + | 加法 | 3+5=8 |
| - | 减法 | 3-5=-2 |
| * | 矩阵乘法 | 3*5=15 |
| .* | 点乘，即数组乘法 | |
| / | 右除 | 3/5 =0.6000 |
| ./ | 数组右除 | |
| \ | 左除 | 3\5=1.6667 |
| .\ | 数组左除 | |
| ^ | 乘方 | 3^5=243 |
| .^ | 数组乘方 | |
| ' | 矩阵共轭转置 | |
| .' | 矩阵转置 | |
| sqrt、sqrtm | 平方根、矩阵平方根 | sqrt(16)=4 |

1.3.2 关系运算

MATLAB的关系运算符包括了所有常用的比较运算，如表1-2所示。两个数通常可以用六种关系来进行描述：小于(<)、小于或等于(<=)、大于(>)、大于或等于(>=)、等于(==)和不同于(~=)。

表 1-2 关系运算符

| 运算符 | 说 明 | 运算符 | 说 明 |
|-----|-----|-----|-------|
| < | 小于 | <= | 小于或等于 |
| > | 大于 | >= | 大于或等于 |
| == | 等于 | ~= | 不等于 |



MATLAB的关系运算符可以用来比较两个维数相同的数组(矩阵)，或用来把一个数组中的每个元素与一个标量比较，结果都返回一个与原来数组同维数的数组。比较两个元素的大小时，如果关系式为“真”，则结果为1；如果关系式为“假”，则结果为0。例如关系式 $4+3 \leq 6$ (数学语言表示4与3的和小于等于6)，通过上面的叙述可知，此关系式的结果为0，标明关系式为假。



关系运算符的运算法则为：

(1) 当两个比较量是标量时，直接比较两数的大小。若关系成立，关系表达式为“真”，结果为1；否则为0。

(2) 当参与比较的两个量是维数相同的数组(矩阵)时，比较是对两数组(矩阵)相同位置的元素按标量关系运算规则逐个进行，并给出元素比较结果。最终的关系运算的结果是一个维数与原数组(矩阵)相同的数组(矩阵)，它的元素由0或1组成。

(3) 当参与比较的一个是标量，而另一个是数组(矩阵)时，则把标量与数组(矩阵)的每一个元素按标量关系运算规则逐个比较，并给出元素比较结果。最终的关系运算的结果是一个维数与原数组(矩阵)相同的数组(矩阵)，它的元素由0或1组成。

1. 数组与一个标量比较

当一个数组与一个标量比较时，首先将标量扩展成与数组同维数的数组，然后进行逐元素比较，结果返回一个与原来数组同维数的数组。例如：

```
>> m=1:9
```

```
m =
```

```
1 2 3 4 5 6 7 8 9
```

```
>> bj=m>5
```

```
bj =
```

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

从以上运行结果可以看到，在数组m中，凡是大于5的对应的结果都为“真”，返回1；其他为“假”，返回0。

2. 数组(矩阵)间的比较

数组(矩阵)间的比较，也是对应元素逐个进行比较，结果返回一个与原来数组同维数的数组(矩阵)。例如：

```
>> n=9-m
```

```
n =
```

```
8  7  6  5  4  3  2  1  0
```

```
>> tf=(m==n)
```

```
tf =
```

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



由上述可知，由于两个数组的对应元素都不相等，结果是返回一个全“假”的数组。

```
>> df=(m>n)
```

```
df =
```

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

满足条件的元素位置返回“真”，不满足条件的元素位置返回“假”。

注意：如果数组具有不同的大小，那么运行时将会产生错误。



3. 关系表达式与数学运算表达式的混合运算

关系表达式可以与数学运算表达式进行混合运算。数组中满足条件的元素位置(即为“真”)返回1, 为“假”返回0, 然后进行运算。例如:

```
>> gh=n- (m>4)
```

```
gh =
```

```
8 7 6 5 3 2 1 0 -1
```



1.3.3 逻辑运算

在MATLAB中, 有三类基本逻辑运算: “与”、“或”和“非”, 包含&、&&、|、|| 和~共五种, 如表1-3所示。

表 1-3 逻辑运算符

| 运算符 | 描 述 |
|-----|--|
| & | 与 |
| && | 标量关系表达式的避绕式(Short-Circuiting)“与”操作, 只适用于标量 a && b, 当 a 的值为假时, 则忽略 b 的值 |
| | 或 |
| | 标量关系表达式的避绕式(Short-Circuiting)“或”操作, 只适用于标量。 a b, 当 a 的值为真时, 则忽略 b 的值 |
| ~ | 非 |
| xor | 异或, 两元素不同时, 返回 1; 相同时, 返回 0 |



逻辑运算的运算法则为：

(1) 在逻辑运算中，确认非零元素为真，用1表示；零元素为假，用0表示。当运算结果为真时，返回值为1；当运算结果为假时，返回值为0。

(2) “与”、“或”操作符号可以比较两个标量或者两个通解数组(或矩阵)。设参与逻辑运算的是a和b两个标量，那么当a、b全为非零时， $a \& b$ 的运算结果为1，否则为0；a、b中只要有一个非零， $a | b$ 的运算结果都为1。

(3) 若参与逻辑运算的一个是标量、一个是矩阵，那么运算将在标量与矩阵中的每个元素之间按标量规则逐个进行。最终运算结果是一个与矩阵同维的矩阵，其元素由1或0组成。



(4) 若参与逻辑运算的是两个同维矩阵，那么运算将对矩阵相同位置上的元素按标量规则逐个进行。最终运算结果是一个与原矩阵同维的矩阵，其元素由1或0组成。

(5) 逻辑“非”是一元操作符(或叫单目运算符)，也服从矩阵运算规则。但是，对于数组(矩阵)，逻辑“非”运算是针对于数组(矩阵)中每个元素的。同样，当逻辑为真时，返回值为1；当逻辑为假时，返回值为0。例如，当a是零时， $\sim a$ 运算结果为1；当a非零时，运算结果为0。

(6) 在算术、关系、逻辑运算中，算术运算优先级最高，逻辑运算优先级最低。



1. 逻辑“与”

逻辑“与”，在数组之间进行逐元素的“与”操作。例如：

```
>> a=1:6
```

```
a =
```

```
1 2 3 4 5 6
```

```
>> b=5-a
```

```
b =
```

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

```
>> m=(a>2)&(a<5)
```

```
m =
```

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

又如：

```
>> n=(a<2)&(a>5)
```

```
n =
```

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



2. 逻辑“或”

逻辑“或”，在数组之间进行逐元素的“或”操作。例如：

```
>> b=5-a
```

```
b =
```

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

```
>> n=(b>1)|(b<0)
```

```
n =
```

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

前三个数字满足第一个条件($b>1$)，输出1；最后一个数字满足第二个条件($b<0$)，输出1。



3. 逻辑“非”

逻辑“非”，即“NOT”，是个一元操作符，对运算对象取反。凡是“真”的，在该位置输出结果就为0，其他为1。例如：

```
>> x = ~(b>2)
```

```
x =
```

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



4. 逻辑“异或”

逻辑“异或”，即“XOR”，是一个逻辑运算函数，对比较两个逻辑对象的值是否相同，若相同则结果为“假”，即为0，若不同则为“真”，即为1。例如：

```
>> b = [4  3  2  1  0 -1]
```


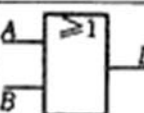
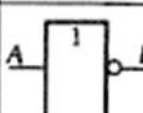
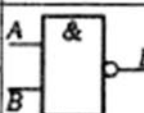
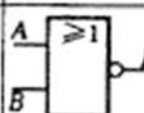
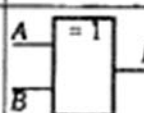
```
>> x = xor(b>2, b<4)
```

```
x =
```

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



逻辑运算规律

| 逻辑运算 逻辑门 逻辑变量 A B | | 与运算 $L = A \cdot B$ | 或运算 $L = A + B$ | 非运算 $L = \bar{A}$ | 与非运算 $L = \overline{A \cdot B}$ | 或非运算 $L = \overline{A + B}$ | 异或运算 $L = A\bar{B} + \bar{A}B$ |
|----------------------------|---|---|---|---|--|---|---|
| | |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |



1.2.4 标量关系表达式的避绕式操作

标量关系表达式的避绕式操作符(&&和||)只适用于标量关系表达式，“避绕式”(Short-Circuiting)是指MATLAB按顺序执行由这两个操作符连接的标量关系表达式，当执行到某一表达式时，就已经可以确定其结果，不再执行(绕过)后面的表达式，直接给出逻辑结果。例如：

```
>> a=0;b=pi;
>> a==0 || b~=1
ans =
1
```



第一个表达式为“真”，于是就绕过后面的表达式不再执行，直接给出逻辑结果为“真”，输出1。

```
>> b==1&&a==0
```

```
ans =
```

```
0
```

第一个表达式为“假”，于是就直接给出逻辑结果为“假”，输出0。

```
>> a==0||(1/a)<1
```

```
ans =
```

```
1
```

由于第一个表达式已经为“真”，整个操作结果必将为“真”，于是直接给出逻辑结果为“真”，输出1，绕过后面的表达式不再执行，否则将出现除数为0的警告。

1.2.5 运算符的优先级

MATLAB 中各运算符的优先级顺序如表1-4所示。

MATLAB在执行运算时，首先执行具有较高优先级的运算，然后执行具有较低优先级的运算。如果两个运算的优先级相同，则按从左到右的顺序执行。

在运算的过程中，关系运算是在所有数学运算之后进行的，所以下面两个表达式是等价的，均产生结果1。

```
>> 6 + 3 < 2 + 10
```

```
>> (6 + 3) < (2 + 10)
```

表 1-4 运算符的优先级

| 运 算 符 | 优先级 |
|---|--|
| 圆括号() | <div>最高</div> <div>↓</div> <div>最低</div> |
| 转置('.')、共轭转置(')、乘方('.')、矩阵乘方 (^) | |
| 标量加法(+)、减法(-)、取反(~) | |
| 乘法(*)、矩阵乘法(*)、右除(/)、左除(\)、矩阵右除(/)、矩阵左除(\) | |
| 加法(+)、减法(-)、逻辑非(~) | |
| 冒号运算符(:) | |
| 关系运算：小于(<)、小于等于(<=)、大于(>)、大于等于(>=)、等于(==)、不等于(~=) | |
| 数组逻辑与(&) | |
| 数组逻辑或() | |
| 避绕式逻辑与(&&) | |
| 避绕式逻辑或() | |



1.2.6 关系与逻辑函数

除了关系运算符和逻辑运算符外，MATLAB还提供了几个关系与逻辑函数。这些关系与逻辑函数及其功能如表1-5所示。

表 1-5 关系与逻辑函数及其功能

| 关系和逻辑函数 | 使用 功 能 |
|-----------|---|
| xor(s, t) | 异或运算，s 或 t 非零(真)返回 1，s 和 t 都是零(假)或都是非零(真)返回 0 |
| any(x) | 如果在一个向量 x 中，任何元素是非零，返回 1；矩阵 x 中的每一列有非零元素，返回 1 |
| all(x) | 如果在一个向量 x 中，所有元素非零，返回 1；矩阵 x 中的每一列所有元素非零，返回 1 |

MATLAB还提供了一些函数，用于检验某个特定的值是否存在或者某一条件是否成立，并返回相应的逻辑结果。由于这些函数大多以“is”开头，因此称为“is族”函数。



例 1-2-1 生成一个数组 $A = \begin{bmatrix} -4 & -2 & 0 & 2 & 4 \\ -3 & -1 & 1 & 3 & 5 \end{bmatrix}$ ，找出数

组中所有绝对值大于 3 的元素。

解 (1) 预生成一个 (2×5) 全零数组。

```
>> A=zeros(2, 5)
```

A =

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

(2) 运用“全元素”赋值法获得 A。

```
>> A(:)=-4:5
```

A =

| | | | | |
|----|----|---|---|---|
| -4 | -2 | 0 | 2 | 4 |
| -3 | -1 | 1 | 3 | 5 |

(3) 产生与 A 同维的“0、1”逻辑值数组。

```
>> L=abs(A)>3
```

L =

| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 |

(4) 用 islogical() 函数判断 L 是否为逻辑值数组。输出若为 1，则是。

```
>> islogical(L)
```

ans =

1

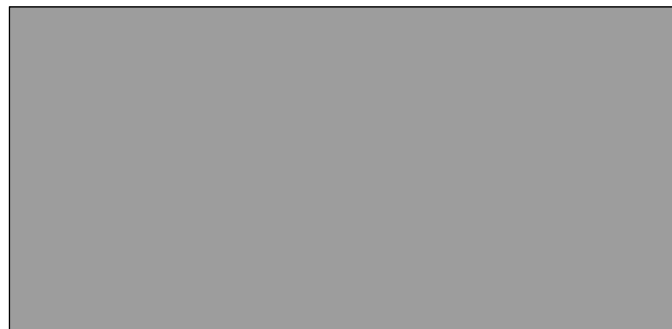
(5) 把 L 中逻辑值 1 对应的 A 元素取出。

```
>> X=A(L)
```

X =

-4
4
5

观看实操录屏



单选题 1分

 设置

此题未设置答案，请点击右侧设置按钮

$y = -1$; $y = \text{true}$; $z = x \ \&\& \ y$; 问: z 是逻辑真还是逻辑假?

- ☐ A 逻辑真
- ☐ B 逻辑假



提交

1.3.7 标点符号的使用

表 1-6 标点符号代表的意义

| 标点符号 | 定 义 | 标点符号 | 定 义 |
|----------|---------------------|----------|-------------|
| 分号(;) | 数组行分隔符；取消运行显示 | 点(.) | 小数点；结构体成员访问 |
| 逗号(,) | 数组列分隔符；函数参数分隔符 | 省略号(...) | 续行符 |
| 冒号(:) | 在数组中应用较多，如生成等差数列 | 单引号('') | 定义字符串 |
| 圆括号(()) | 指定运算优先级；函数参数调用；数组索引 | 等号(=) | 赋值语句 |
| 方括号([]) | 定义矩阵 | 感叹号(!) | 调用操作系统运算 |
| 花括号({ }) | 定义单元数组 | 百分号(%) | 注释语句的标识 |



续行符号的使用

3个点组成的省略号(...)作为续行符号。在编写程序时，往往会遇到命令行很长或一行写不下的情况。为了阅读起来方便或使程序看起来更清晰，可以将程序分成多行分别书写，使用续行符号连接。例如：

```
>> x = 5*6 ...
+8-5
x = 33
>> total= ...
5*6+8-5
total = 33
```



使用续行符号可将两行命令连接为一行，但使用续行符号的位置要注意，否则将会出错。例如：

```
>> total=...5*6+8-5
```

```
???
```

```
|Error: Incomplete or malformed expression or statement.
```

```
>> value1=10;value2=9;
```

```
>> total=value1+value...2
```

```
??? Undefined function or variable 'value'.
```

```
>> total=value1+value...
```

```
2
```

```
??? 2
```

```
|
```

```
Error: Missing MATLAB operator.
```

1.3.8 常用的操作命令和快捷键

1. 常用的操作命令

MATLAB常用的操作命令如表1-7所示。

表 1-7 常用的操作命令

| 命 令 | 功 能 | 命 令 | 功 能 |
|-------|------------|------|-------------|
| cd | 显示或改变工作目录 | hold | 图形保持命令 |
| clc | 清除工作窗口中的内容 | load | 加载指定文件的变量 |
| clear | 清除内存变量 | pack | 整理内存碎片 |
| clf | 清除图形窗口 | path | 显示搜索目录 |
| diary | 日志文件命令 | quit | 退出 MATLAB |
| dir | 显示当前目录下文件 | save | 保存内存变量到指定文件 |
| disp | 显示变量或文字内容 | type | 显示文件内容 |
| echo | 工作窗信息显示开关 | | |

2. 常用的键盘操作和快捷键

MATLAB常用的键盘操作和快捷键，如表1-8所示。

表 1-8 常用的键盘操作和快捷键

| 键盘按钮和快捷键 | 功 能 | 键盘按钮和快捷键 | 功 能 |
|------------|----------|-------------------|-----------|
| ↑ (Ctrl+p) | 调用上一行 | Home(Ctrl+a) | 光标置于当前行开头 |
| ↓ (Ctrl+n) | 调用下一行 | End(Ctrl+e) | 光标置于当前行结尾 |
| ←(Ctrl+b) | 光标左移一个字符 | Esc(Ctrl+u) | 清除当前输入行 |
| →(Ctrl+f) | 光标右移一个字符 | Del(Ctrl+d) | 删除光标处字符 |
| Ctrl+← | 光标左移一个单词 | Backspace(Ctrl+h) | 删除光标前字符 |
| Ctrl+→ | 光标右移一个单词 | Alt+BackSpace | 恢复上一次删除 |



课程小结

思维导图



工程实例1——航天器发射问题

安萨里X奖设立于1996年，这是第一个为私人航空行为设立的奖项。获得该奖项的要求是：两周内，同一航天器将三人送到外太空两次。2004年，该奖项由Burt Rutan创立的公司Tier 1获得，他们的思路是：一艘装载着宇宙飞船1号的母舰起飞并登陆在常规的跑道，将航天器发射到25000英尺的高空，再进入外太空（100千米的高空）。在一个星期内，他们重复了一次，然后赢得了大奖（1000万美元）。
问题：假设航天器消耗所有的能源，以垂直速度 u 到达25000英尺，要使航天器到达外太空， u 的值是多少？
注：1英寸 = 2.54厘米； 1英尺 = 12 英寸，重力加速度为 9.8 m/s^2



求解思路：

1、单位一致性换算；

$$\text{米} = \text{米/厘米} * \text{厘米/英寸} * \text{英寸/英尺} * \text{英尺}$$

2、写出距离计算公式，求解 u

$$v^2 - u^2 = 2as$$

3、求解过程



实例求解：一艘装载着宇宙飞船1号的母舰起飞并登陆在常规的跑道，将航天器发射到**25000英尺**的高空，再进入外太空（**100千米**的高空），假设航天器消耗所有的能源，以垂直初速度u到达**25000英尺**，要使航天器到达外太空，初速度的值是 [填空1] ？

正常使用填空题需3.0以上版本雨课堂

作答



求解思路：

1、单位一致性换算；

米= 米/厘米*厘米/英寸*英寸/英尺*英尺

2、写出距离计算公式，求解u

$$v^2 - u^2 = 2as$$

3、求解过程

```
%% 单位换算
cmPerInch = 2.54; % 1英寸=2.54cm
inchesPerFt = 12;
metersPerCm = 1/100;
metersPerFt = metersPerCm *
cmPerInch * inchesPerFt;
startFt = 25000; % 初始位置
startM = startFt * metersPerFt;
```

```
%% 使用距离加速度公式求解
g = 9.8; %重力加速度
top = 100*1000; % 千米
s = top-startM;
initialV = (2*g*s)^0.5 %最终
结果
```

实例分析-贷款问题

- 1、年利率6% = 月利率0.5%
- 2、贷款12万元，1年还清，12期
- 3、利息：12万元*6%~~X~~0.72万元

还款方式：1、等额本金；2、等额本息

一、等额本金

- 1、本金：12万元/12月=1万元/月
- 2、利息：1月-12万元*0.5%=600元
2月-11万元*0.5%=550元
.....
12月：1万元*0.5%=50元

总结：600+550+.....+50=3900元



二、等额本息

设贷款总额为A，银行月利率为 β ，总期数为m（个月），月还款额设为X，则各个月所欠银行贷款为：

第一个月 $A(1+\beta)-X$

第二个月 $(A(1+\beta)-X)(1+\beta)-X=A(1+\beta)^2-X[1+(1+\beta)]$

第三个月 $((A(1+\beta)-X)(1+\beta)-X)(1+\beta)-X=A(1+\beta)^3-X[1+(1+\beta)+(1+\beta)^2] \dots$

由此可得第n个月后所欠银行贷款为 $A(1+\beta)^n -$

$X[1+(1+\beta)+(1+\beta)^2+\dots+(1+\beta)^{(n-1)}]=A(1+\beta)^n - X[(1+\beta)^n - 1]/\beta$

由于还款总期数为m，也即第m月刚好还完银行所有贷款，

因此有 $A(1+\beta)^m - X[(1+\beta)^m - 1]/\beta = 0$

由此求得 $X = A\beta(1+\beta)^m / [(1+\beta)^m - 1]$



贷款实例-等额本息：

1、年利率6% = 月利率0.5%

2、贷款12万元，1年还清，12期

月还款额应位： $X = A\beta(1+\beta)^m / [(1+\beta)^m - 1] = 12 * 0.5\%$

$(1+0.5\%)^{12} / [(1+0.5\%)^{12} - 1] = 1.0328$ 万元

利息总额： $0.0328 * 12 = 3935.66$ 元

利息：1月： $12\text{万元} * 0.5\% = 600$ 元；

2月： $(120000 - 10328 + 600) * 0.5\% = 551.36$ 元

3月： $(120000 - 10328 * 2 + 600 + 551.36) * 0.5\% = 502.48$

元

.....



作业，用MATLAB编写代码求解下面的问题：

1、给定三角形的两条边， $a = 4.5$, $b = 6$, 这两条边的夹角为35度，计算第三条边的长度和三角形的面积

2、你想买一套300万元的房子，首付40%，当前的复利率是4.5%。

(1) 若选择等额本金方式还款。

1) 贷30年，每月的还款额是多少？

2) 贷款期间，总共付了多少利息

(2) 若选择等额本息方式还款

1) 贷30年，每月的还款额是多少？

2) 贷款期间，总共付了多少利息

