



数学软件 Matlab

——数据类型

内容提要

- **变量与常量**
- **数值型数据**
- **逻辑型数据**
- **字符与字符串**
- **细胞 (cell)**
- **结构型数据**
- **句柄**

变量

● 变量

- 变量是任何程序设计语言的**基本元素之一**
- Matlab 变量的特点：
 - 不要求事先声明（也无法声明）
 - 不需要指定变量类型：
根据所赋的值或对变量所进行的操作来确定变量的类型
 - 变量的类型可以随时改变：
在赋值过程中，如果变量已经存在，Matlab 会用新值代替旧值，并以**新的数据的类型代替旧的变量类型**

如果能够事先给**大数组**分配空间，则可以提高程序的执行效率

变量

● 变量命名要求

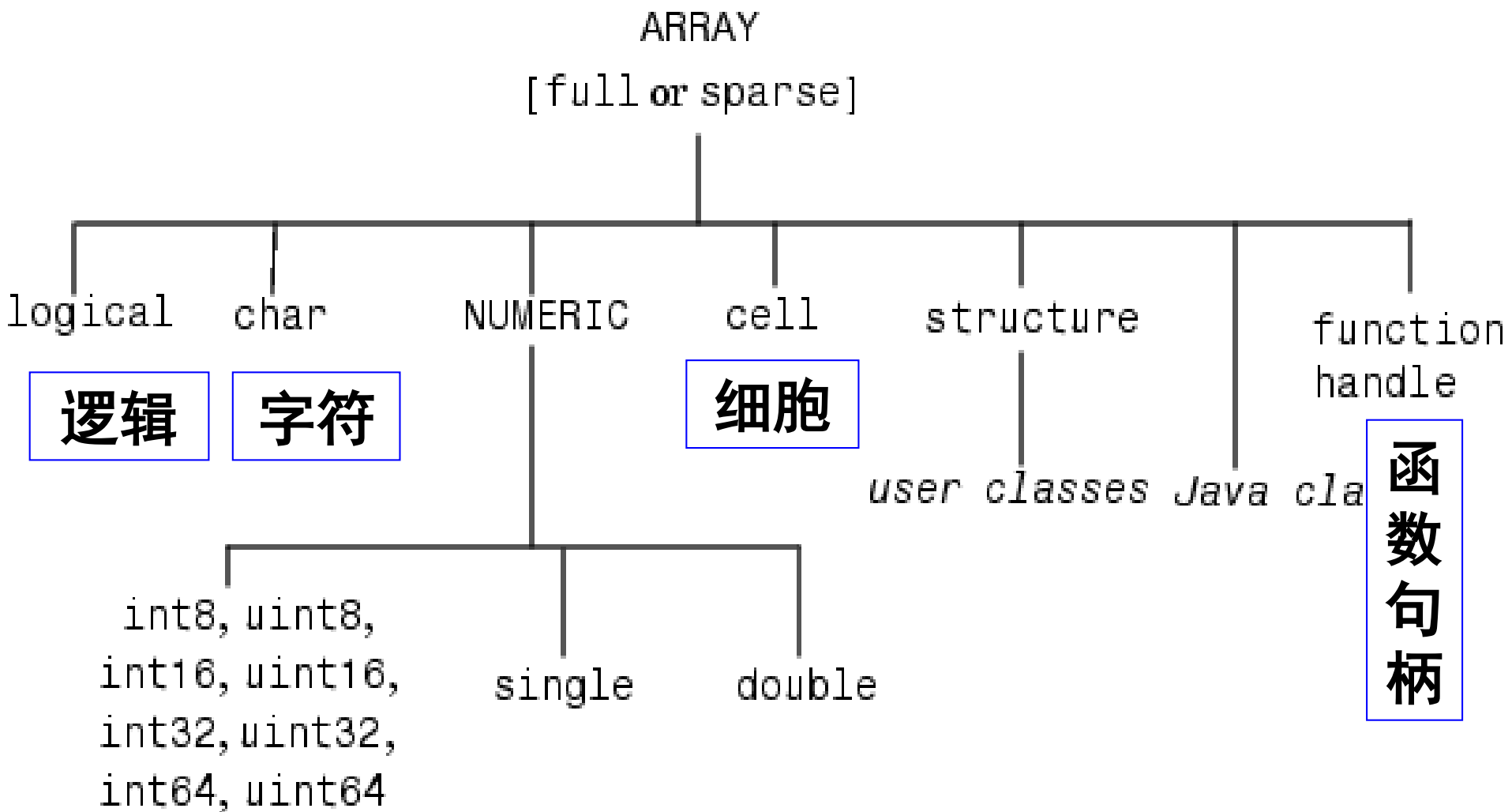
- 必须以字母开头，含字母（大小写）、数字和下划线
- 变量名长度不超过 65（Matlab6.5.1 版本以上）
- 区分大小写

● Matlab 预定义常量

<code>i, j</code>	虚数单位	<code>realmin</code>	最小正浮点数
<code>pi</code>	圆周率	<code>realmax</code>	最大正浮点数
<code>eps</code>	浮点运算相对精度	<code>intmin</code>	最小整数
<code>Inf</code>	无穷大	<code>intmax</code>	最大整数
<code>NaN/nan</code>	不定值		

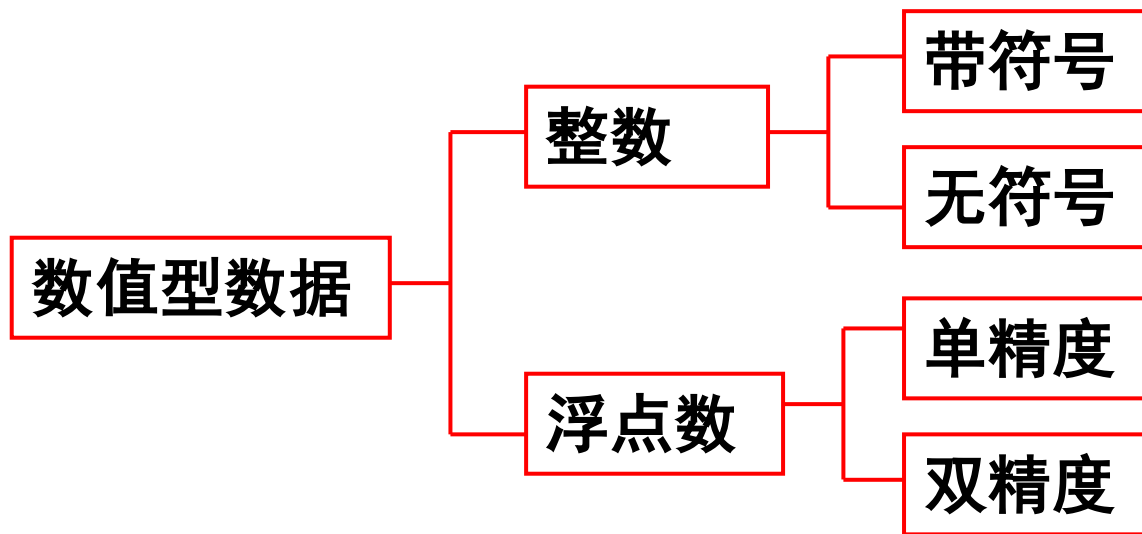
数据类型

- Matlab 常用的数据类型有：
 - `numeric`: 数值
 - `char`: 字符
 - `logical`: 逻辑
 - `cell`: 单元、细胞、元胞
 - `struct`: 结构



Matlab7.x 中的数据类型的表

数值型



- 为了提高数值计算的精度，若没有指定数据类型，则所有的数都被看作是双精度的浮点数
- 使用整型数据或单精度数据可以节约内存空间

整型数据

- Matlab 提供四种带符号整型和四种无符号整型

数据类型	取值范围	转换函数
带符号 8 位整型	$[-2^7, 2^7 - 1]$	int8
带符号 16 位整型	$[-2^{15}, 2^{15} - 1]$	int16
带符号 32 位整型	$[-2^{31}, 2^{31} - 1]$	int32
带符号 64 位整型	$[-2^{63}, 2^{63} - 1]$	int64
无符号 8 位整型	$[0, 2^8 - 1]$	uint8
无符号 16 位整型	$[0, 2^{16} - 1]$	uint16
无符号 32 位整型	$[0, 2^{32} - 1]$	uint32
无符号 64 位整型	$[0, 2^{64} - 1]$	uint64

举例

`x1=int8(11);`



`x1=11`

`x2=int8(15.49);`



`x2=15`

`x3=int8(15.5);`



`x3=16`

`x4=int8(130);`



`x4=127`

`x5=uint8(255);`



`x5=255`

`x6=uint8(256);`



`x6=255`

`x7=uint8(-1);`



`x7=0`

整型数据运算规则

- 整型数据运算规则：

- 相同整型数据之间的运算，结果为同类型的整数
- 不同整型数据之间不能进行运算！

```
x=int8(11); y=int8(5); z=x*y % OK
```

```
x=int8(11); y=int16(5); z=x*y % ERROR!
```

- 整型数据可以和双精度标量进行数值运算，结果为整数。在运算过程中保持浮点运算精度，最后将结果转换成整型。

整型数据不能与双精度数组进行运算

```
x=int8(11); y=[1,2]; z=x*y % ERROR!
```

最大整数与最小整数

- `intmax` 和 `intmin` 函数:

查询不同整型数据所能表示的最大整数和最小整数

例:

```
intmax('int8');  
intmin('int8');
```

```
intmax('uint8');  
intmin('uint8');
```

```
intmax;  
intmin;
```



```
intmax('int32');  
intmin('int32');
```

浮点数

- 浮点数：单精度（占 4 个字节）和双精度（占 8 个字节）
 - **single**: 将数据转换成单精度浮点数
 - **double**: 将数据转换成双精度浮点数

浮点数与其它类型数据运算表

operand	single	double	int / uint	char	logical
single	single	single	X	single	single
double	single	double	int / uint	double	double

最大和最小浮点数

- `realmax` 和 `realmin` :

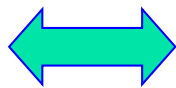
查询单精度和双精度所能表示的最大浮点数和最小浮点数

例:

```
realmax('single');  
realmin('single');
```

```
realmax('double');  
realmin('double');
```

```
realmax;  
realmin;
```



```
realmax('double');  
realmin('double');
```

复数

- 复数的输入方法

- 直接输入：

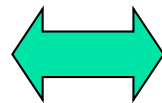
```
z=3+4i
```

虚部与虚数单位之间不能留空格

- 使用 `complex` 函数

```
z=complex(3,4)
```

```
z=complex(3)
```



```
z=complex(3,0)
```

整型数据相关函数

Integer Functions

Function	Description
int8 , int16, int32, int64	Convert to signed 1-, 2-, 4-, or 8-byte integer.
uint8 , uint16, uint32, uint64	Convert to unsigned 1-, 2-, 4-, or 8-byte integer.
ceil	Round towards plus infinity to nearest integer
class	Return the data type of an object.
fix	Round towards zero to nearest integer
floor	Round towards minus infinity to nearest integer
isa	Determine if input value has the specified data type.
isinteger	Determine if input value is an integer array.
isnumeric	Determine if input value is a numeric array.
round	Round towards the nearest integer

浮点数相关函数

Floating-Point Functions

Function	Description
<u>double</u>	Convert to double precision.
<u>single</u>	Convert to single precision.
<u>class</u>	Return the data type of an object.
<u>isa</u>	Determine if input value has the specified data type.
<u>isfloat</u>	Determine if input value is a floating-point array.
<u>isnumeric</u>	Determine if input value is a numeric array.
<u>eps</u>	Return the floating-point relative accuracy. This value is the tolerance MATLAB uses in its calculations.
<u>realmax</u>	Return the largest floating-point number your computer can represent.
<u>realmin</u>	Return the smallest floating-point number your computer can represent.

复数相关函数

Complex Number Functions

Function	Description
<u>complex</u>	Construct complex data from real and imaginary components.
<u>i</u> or <u>j</u>	Return the imaginary unit used in constructing complex data.
<u>real</u>	Return the real part of a complex number.
<u>imag</u>	Return the imaginary part of a complex number.
<u>isreal</u>	Determine if a number is real or imaginary.

其它相关函数

Infinity and NaN Functions

Function	Description
<u>inf</u>	Return the IEEE value for infinity.
<u>isnan</u>	Detect NaN elements of an array.
<u>isinf</u>	Detect infinite elements of an array.
<u>isfinite</u>	Detect finite elements of an array.
<u>nan</u>	Return the IEEE value for Not a Number.

判断数据类型相关函数

Type Identification Functions

Function	Description
<u>class</u>	Return data type (or class).
<u>isa</u>	Determine if input value is of the specified data type.
<u>isfloat</u>	Determine if input value is a floating-point array.
<u>isinteger</u>	Determine if input value is an integer array.
<u>isnumeric</u>	Determine if input value is a numeric array.
<u>isreal</u>	Determine if input value is real.
<u>whos</u>	Display the data type of input.

字符串

- 字符串和字符数组

字符与字符串运算是各种高级语言不可缺少的部分，Matlab 具有强大的字符处理能力

- 字符串必须用单引号括起来

```
str1='Matrix Laboratory'
```

- 字符串是由多个字符组成的字符行向量

```
str2=['M','a','t','r','i','x']
```

- 字符数组必须是方的，即每行的字符个数必须相等

```
str3=['M','a','t'; 'r','i','x']
```

字符串

- 字符串中的每个字符（包括空格）都是字符串的一个元素

```
str='Matrix';  
str(3)  
size(str)
```

- Matlab中字符是以 ASCII 码存储的，因而区分大小写，可用 `double` 等函数查看一个字符的 ASCII 码

```
str='Matrix';  
double(str(3));  
int16(str);
```

字符串的输入

- 直接输入

```
str1='Matrix'; str2='Laboratory';  
str=[str1, str2];  
str=[str1; str2]; // ERROR!
```

- 使用 **char** 函数

ASCII 码 → 字符串

```
str=char([77,97,116,108,97,98])
```

```
str=char(str1,str2)
```

- 生成一个字符数组，第一行为 **str1**，第二行为 **str2**
- 数组的列数= $\max(\text{length}(\text{str1}), \text{length}(\text{str2}))$

字符与数的转换

- 字符串、字符数组与数值数组之间的转换
 - 按 ASCII 码转换
 - 字符 → 数: `double`、`int8`、`int16`、...
 - 数 → 字符: `char`
 - 直接将数转换成字符
 - 字符 → 数: `str2num` (仅限于字符串由数字构成)
 - 数 → 字符: `num2str`、`int2str`、`mat2str`

```
a=[97, 97.3; 97.8, 98];  
str1=char(a);  
str2=num2str(a);  
str3=int2str(a);  
str4=mat2str(a);
```

举例

- `char` 的输入参数有效范围为 `[0,65535]`，当输入参数中有小数时，直接将小数部分舍去。
- `num2str` 直接将每个数字转换成字符
- `mat2str` 把整个矩阵转换成字符行向量，包括矩阵中的方括号、空格 和 分号（逗号自动去除）

```
a=[1,2,3];  
b=num2str(a);  
c=2*a;    % c=?  
d=2*b;    % d=?
```

- 数值数组转换为字符数组后，表面上看象一个数值数组，但此时的元素是字符而不是数！

**char、double、int8、int16 等是按
ASCII码 进行转换**

**str2num、num2str、int2str、mat2str
是直接进行转换**

字符串的合并

- 字符串水平合并

- 直接使用中括号

```
str1=['hello ', 'world!']
```

- 使用 `strcat` 函数

```
str2=strcat('hello ', 'world!')
```

- 使用 `strcat` 时，系统会自动去除原字符串**结尾处的空格**
 - 水平合并得到的是一个更长的字符串

字符串的合并

- 字符串垂直合并

- 直接使用中括号

```
str3=['Matrix      '; 'Laboratory']
```

- 使用 `strvcat` 函数

```
str4=strvcat('Matrix','Laboratory')
```

- 在中括号中用分号实现垂直合并，必须保证每个字符串的长度相等，否则需用空格补齐
- 用 `strvcat` 合并，系统会自动为较短的字符串补充空格
- 垂直合并得到的是一个字符数组

字符串的执行

- 执行字符串所表示的表达式或命令：**eval**

```
str='1+2';  
a=eval(str)
```

```
str='A=magic(3)';  
eval(str)
```

```
str1='figure(1); mesh(peaks(60))';  
str2='figure(2); waterfall(peaks(60))';  
eval(str1)  
eval(str2)
```

字符串相关函数

函 数	功 能	函 数	功 能
<code>strcmp</code>	字符串比较	<code>strrep</code>	字符串替换
<code>strcmpi</code>	字符串比较： 忽略大小写	<code>strncmp</code>	比较字符串的前n个字符
<code>upper</code>	转换为大写	<code>lower</code>	转换为小写
<code>blanks</code>	产生空字符串	<code>deblank</code>	删除字符串中的空格
<code>strmatch</code>	查找匹配字符串	<code>findstr</code>	在一个字符串中查找 另一字符串
<code>strjust</code>	对齐字符数组， 包括左对齐，右 对齐和居中	<code>strtok</code>	返回字符串中第一个 分隔符（空格，回车 和Tab键）前的部分

细胞数组/单元/元胞

- Matlab 从 5.0 版开始引入了一种新的数据类型 — **细胞** (`cell`)，该结构可以把不同类型的数据纳入到一个变量中。
- 普通数组中的每个元素都必须具有相同的数据类型，而细胞则没有此要求。
- 细胞变量的表示方法类似于带有下标的数组，但这些下标不是用圆括号括起来，而是使用**大括号**。

细胞数组举例

<p>cell 1,1</p> <table border="1"><tr><td>3</td><td>4</td><td>2</td></tr><tr><td>9</td><td>7</td><td>6</td></tr><tr><td>8</td><td>5</td><td>1</td></tr></table>	3	4	2	9	7	6	8	5	1	<p>cell 1,2</p> <table border="1"><tr><td>'Anne Smith'</td></tr><tr><td>'9/12/94'</td></tr><tr><td>'Class II'</td></tr><tr><td>'Obs. 1'</td></tr><tr><td>'Obs. 2'</td></tr></table>	'Anne Smith'	'9/12/94'	'Class II'	'Obs. 1'	'Obs. 2'	<p>cell 1,3</p> <table border="1"><tr><td>.25+3i</td><td>8-16i</td></tr><tr><td>34+5i</td><td>7+.92i</td></tr></table>	.25+3i	8-16i	34+5i	7+.92i					
3	4	2																							
9	7	6																							
8	5	1																							
'Anne Smith'																									
'9/12/94'																									
'Class II'																									
'Obs. 1'																									
'Obs. 2'																									
.25+3i	8-16i																								
34+5i	7+.92i																								
<p>cell 2,1</p> <table border="1"><tr><td>1.43 2.98 7.83 5.67</td></tr><tr><td>4.21</td></tr></table>	1.43 2.98 7.83 5.67	4.21	<p>cell 2,2</p> <table border="1"><tr><td>-7</td><td>2</td><td>-14</td></tr><tr><td>8</td><td>3</td><td>-45</td></tr><tr><td>52</td><td>-16</td><td>3</td></tr></table>	-7	2	-14	8	3	-45	52	-16	3	<p>cell 2,3</p> <table border="1"><tr><td>'text'</td><td><table border="1"><tr><td>4</td><td>2</td></tr><tr><td>1</td><td>5</td></tr></table></td></tr><tr><td><table border="1"><tr><td>7.3</td><td>2.5</td></tr><tr><td>1.4</td><td>0</td></tr></table></td><td>.02 + 8i</td></tr></table>	'text'	<table border="1"><tr><td>4</td><td>2</td></tr><tr><td>1</td><td>5</td></tr></table>	4	2	1	5	<table border="1"><tr><td>7.3</td><td>2.5</td></tr><tr><td>1.4</td><td>0</td></tr></table>	7.3	2.5	1.4	0	.02 + 8i
1.43 2.98 7.83 5.67																									
4.21																									
-7	2	-14																							
8	3	-45																							
52	-16	3																							
'text'	<table border="1"><tr><td>4</td><td>2</td></tr><tr><td>1</td><td>5</td></tr></table>	4	2	1	5																				
4	2																								
1	5																								
<table border="1"><tr><td>7.3</td><td>2.5</td></tr><tr><td>1.4</td><td>0</td></tr></table>	7.3	2.5	1.4	0	.02 + 8i																				
7.3	2.5																								
1.4	0																								

细胞变量的定义

- 可以通过以下两种方式定义一个细胞变量

- 用赋值语句直接定义

```
A=[1 2; 3 4];  
str='Matlab';  
M={1:4, A, str};    cellplot(M)
```

- 用 `cell` 函数预先分配空间，然后对细胞的每个元素逐个赋值

```
M=cell(m,n);    % 生成一个 m 行 n 列 的细胞数组
```

```
M=cell(2,3);  
M{1,1}=1; M{1,2}=rand(2); M{1,3}='hello';  
M{2,1}=1:3; M{2,2}='Matrix'; M{2,3}=5;  
cellplot(M)
```


细胞变量元素的引用

- 要引用细胞中的单个元素，必须使用**大括号**

```
A=[1 2; 3 4];  
str='Matlab';  
M={1:4, A, str};  
N={'hello', M; eye(2),magic(3)};  
N{1,1}  
N{1,2}  
N{1,2}{1,2}  
N{1,2}{1,2}(2,1)
```

相关函数

celldisp % 列出细胞变量的内容

cellplot % 用图形方式显示细胞变量的内容

iscell % 判别一个变量是否为细胞变量

```
A=[1 2; 3 4];  
str='Matlab';  
M={1:4, A, str};  
celldisp(M);  
cellplot(M);  
iscell(A)  
iscell(M)
```

上机作业

1、已知 Chebyshev 多项式定义如下：

$$T_0(x) = 1, T_1(x) = x,$$

$$T_{n+1}(x) = 2x T_n(x) - T_{n-1}(x), n = 1, 2, \dots$$

编程计算 $T_{10}(x)$ （注： $T_n(x)$ 为 n 阶多项式）

试用细胞数组实现（程序取名 m101.m）

2、阶梯问题：有一个长阶梯，若每步上 2 阶，最后剩 1 阶；若每步上 3 阶，最后剩 2 阶；若每步上 5 阶，最后剩 4 阶；若每步上 6 阶，最后剩 5 阶；若每步上 7 阶，最后刚好一阶不剩。问该阶梯至少有多少阶？编程解决这个问题。（程序取名 m102.m）

上机作业

3、哥德巴赫猜想：任何一个大于 2 的偶数都能表示成两个素数之和。试编程验证 [3, 2000] 中的偶数是否满足哥德巴赫猜想的论断，并将结果写入文本文件 m103.txt（程序取名 m103.m）

4、由 0 到 9 共 10 个数字组成 4 个**完全平方数**（即是某个正整数的平方），要求它们分别是一位、二位、三位、四位的数字，且**数位不重复**（即 0 到 9 每个数字只能在这四个数字中出现一次）。例如：

1 36 784 9025

9 16 784 3025

编程找出所有满足条件的数，并写入到文本文件 m104.txt 中。

（程序取名 m104.m）