

## Matrix Calculator

【产生背景】众所周知，MATLAB 是一款非常强大的基于矩阵运算的数学软件。但是其庞大的体积、高昂的价格令我们这些只需要最基本矩阵运算功能的大一新生们望而却步。幸好，有学习 C 程序设计的你们！动手帮未来的学弟学妹们写一个矩阵计算器吧！

【功能描述】对这个矩阵计算器（以下简称 MatCalc）的功能描述如下：

### (一) 指令格式

- a) MatCalc 的所有指令格式均为“指令(操作数 1, 操作数 2...)”或“指令 操作数 1 操作数 2 ...”。
- b) 指令与操作数之间以空格作为分隔
- c) 作为分隔符的空格或回车换行符可能会连续出现多个，整条命令的开头或结尾可能有多余的空格
- d) 除了定义矩阵时，均把回车换行符（'\n'）视作命令的结束
- e) 指令不区分大小写
- f) MatCalc 中使用的均为英文标点

### (二) 读入与输出

- a) 基本：
  - i. MatCalc 应始终可以直接从键盘读入指令
  - ii. MatCalc 可以将运算结果输出到屏幕上
  - iii. MatCalc 需要在每条命令执行后输出对应的返回值，除非上一条命令结尾有分号抑制输出
- b) 中级：
  - i. MatCalc 支持用  
`readfrom filename`  
指定一个文件作为读入文件，并以命令  
`read`  
表示从该文件中读入所有指令并立即执行。在任何时候都可以用  
“`readfrom filename`”来指定一个新的读入文件
  - ii. MatCalc 支持用  
`writeto filename`  
指定一个文件作为输出文件，并从这一步开始将以后所有的运算结果以追加的形式输出到该文件。在任何时候都可以用“`writeto filename`”来指定一个新的输出文件，若不跟文件名则表示指定屏幕为输出终端

### (三) 基本功能

- a) 矩阵 ans  
MatCalc 中有一个名为 ans 的变量，存放最后一条指令的返回值。
- b) 定义一个新矩阵（返回值即该矩阵）
  - i. 用

`MatName = [data]`

定义一个新矩阵。其中矩阵的命名规范符合 C 语言中变量命名，且不能与 MatCalc 中指令重名，也不能叫“ans”。若“MatName”已存在，则将它覆盖。数据由一对方括号包含。数据中同一行上的数字以逗号或空格隔开。行与行用分号或回车换行符隔开。数据应该保证每行有相同的列数。矩阵名与数据间有一个等号，等号两边可能有空格。



如

```
_MatrixA = [ 1 3, 4; 5 3 1
```

```
1 8,1
```

```
]
```

是合法的，定义了名为 “\_MatrixA” 的 3\*3 矩阵，且 *ans*=\_MatrixA，同时立即输出

```
ans=
```

```
1 3 4
```

```
5 3 1
```

```
1 8 1
```

ii. 用 “:” 算子

```
MatName = [N:x:M]
```



表示产生一个行向量，元素依次为 “N, N+x, N+2x, ..., M”，若只有 [N:M] 而不指定 x 表示 x=1

iii. 用

```
MatName = rand(N,M)
```



定义一个名为 “MatName” 的 N\*M 的随机矩阵，矩阵中每个元素为一个 [0,1] 的随机数。若不指定 M 则表示 M=N。若不加 “MatName = ” 则仅有 *ans* 等于这个随机矩阵。

iv. 用

```
MatName = zeros(N, M)
```



定义一个名为 “MatName” 的 N\*N 的零矩阵。若不指定 M 则表示 M=N。若不加 “Matname = ” 则仅有 *ans* 等于这个零矩阵。

v. 用

```
MatName = eye(N)
```



定义一个名为 “MatName” 的 N\*N 的单位矩阵，若不加 “Matname = ” 则仅有 *ans* 等于这个零矩阵。

vi. 用

```
MatName = ans
```

定义一个名为 “MatName” 且等于 *ans* 的矩阵

c) 矩阵某行某列的元素（返回一个数）



```
MatA(N, M)
```

表示 MatA 的 N 行 M 列上的数。可以把它赋值给其它变量，或者通过对它赋值来改变 MatA。

d) 子矩阵（返回值即该子矩阵）

```
MatA(N:M, :)
```



```
MatA(VectorA, VectorB)
```

“:” 算子意义同上，不加参数表示所有列（或行）。也可以用行向量来指定从 MatA 中取出的行或列。如 *VectorA* = [1, 5], *VectorB* = [2, 4] 表示从 MatA 中取出第 1 行、第 5 行的第 2 列、第 4 列构成的子矩阵。可将子矩阵赋值给其它矩阵，也可将其它矩阵赋值给子矩阵来改变 MatA。



e) 四则运算、取余（返回值为计算结果）

MatCalc 应具有二元普通运算和取余 (+、-、\*、\、%) 功能。其中加、减操作数可以是常数和常数或矩阵和矩阵；乘的操作数是常数和常数、常数和矩阵或矩阵和矩阵。除法和取余只要支持：常数除以常数或矩阵除以常数。允许以已经定义的矩阵和 **ans** 作为操作数，允许直接以 “[data]” 表示一个矩阵作为操作数，也允许以所有返回值合法的指令加上一对小括号作为指令。但为了不引起混淆，用 “[data]” 来表示一个矩阵时不允许用回车换行符作为行与行间的分隔。



- f) 矩阵求和、最大值最小值（返回值为一个数）

MatCalc 可以对矩阵所有元素求和、求出矩阵中的最大值和最小值。格式如：

*sum(MatA)*



*max(MatA)*

*min(MatA)*

- g) 四舍五入和取整（返回值为计算结果）

MatCalc 包含各种取整的命令，如

*round(MatName)*



*upper(MatName)*

*lower(MatName)*

分别表示让 MatName 中的元素四舍五入、向上取整、向下取整

#### (四) 中级功能

- a) 嵌套定义（返回定义的矩阵）

MatCalc 应支持诸如

**MatName = [ eye(3), zeros(3); 1:3, 2:2:6 ]**

将定义一个 MatName =

1	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
1	2	3	2	4	6

- b) 矩阵的幂（返回值为计算结果）

MatCalc 应具有快速计算矩阵的幂的功能。格式为

*MatName^N*



其中 N 应为一整数。若 N 是负数则表示其逆的 |N| 次。“^” 两端可能有空格。若矩阵名后面有一个 “.” 表示对矩阵中每一个元素分别求幂。

- c) 矩阵的转置（返回值为计算结果）

以命令



*MatName'*

表示矩阵的转置。单引号和矩阵名之间不允许有分隔符

- d) 方阵的行列式（返回值为计算结果）

MatCalc 应能够计算一个方阵的行列式。格式为



*det(MatName)*

此时应将返回值视作常数，而非一个 1\*1 的矩阵

- e) 矩阵的秩（返回值为计算结果）

MatCalc 应能够计算一个矩阵的秩。格式为



*rank(MatName)*

此时应将返回值视作常数，而非一个 1\*1 的矩阵

- f) 化为行最简形（返回值为计算结果）

MatCalc 应能通过基本行运算将一个矩阵化为行最简形。格式为



*rref(MatName)*

rref 即 reduced row echelon form

- g) 矩阵的逆、除法（返回值为计算结果）

MatCalc 应能求方阵的逆, 除法运算中若一个除数是方阵即乘以它的逆, 同样, 一个方阵的  $n$  次幂, 若  $n < 0$  即其逆的  $|n|$  次幂。格式为



*reverse(MatA)*

*MatA^n*

*MatA/MatB*

- h) 向量的内积、模和夹角（返回值为计算结果）

MatCalc 应能计算向量的内积、模和夹角。格式为



*dot(MatA, MatB)*

*norm(MatA)*

*angle(MatA, MatB)*

其中 MatA 和 MatB 应是同一空间中的向量（行数相同列数为 1 的矩阵）

- i) 命令追溯

MatCalc 应具有对尽可能多过去命令的追溯的功能。并能追溯从文件中读入的命令。在输入命令时按“↑”键追溯

## (五) 高级功能

- a) 表达式

MatCalc 要能识别并计算任意的中缀表达式。并且, 以上提到的所有指令中的操作数（除了文件名）均可以是一个任意复杂的表达式

- b) 特征向量与特征值

MatCalc 用以下命令计算特征向量和特征值



*eig(MatA)*

*[X D] = eig(MatA)*

第一条指令表示求 MatA 的特征值, 第二条指令表示 X 为 MatA 所有特征向量构成的矩阵, D 为特征值构成的对角矩阵。

- c) 命令合法性判断

MatCalc 要能对命令格式、中间结果等的合法性以及解的个数进行判断。出现任何不合法或无解的命令应在屏幕上（无论有没有指定输出文件）输出相应的信息

- d) 帮助文档

编写一个能在 MatCalc 中实时查阅的帮助文档, 形式可参考 DOS 系统中的 help 命令

## (六) 可选的指令

你们可以给 MatCalc 加上一些其他的功能。如进制转换、输出每条指令运算时间、格拉姆-施密特正交化、各种矩阵分解、前后缀表达式的计算、符号表达式的计算、方程根的近似求解、函数图像绘制等等。

【数据范围】以下数据范围供参考:

矩阵的个数  $\leq 10^3$

矩阵行数、列数  $\leq 10^4$

矩阵的行数 \* 列数  $\leq 10^6$

矩阵中元素的范围不超过 long double, 当值落在 $[-2^{31}, 2^{31} - 1]$ 中时需区分浮点数和整数, 超过这个范围可均视作浮点数

计算矩阵的幂次 $\leq 2^{31} - 1$

说明: 若用 TC 开发, 受到内存申请的限制, 则矩阵个数可适当减少。

**【考察知识】**结合线性代数的基础知识和简单算法, 综合考察数据处理、结构程序设计、函数程序设计与模块思想、指针与链表、文件输入输出等 C 语言相关知识。