

# matlab 工程计算课后习题

LC

2020 年 1 月 28 日

## 摘要

这是关于 matlab 课后习题的小文档

## 目录

1 入门	1
2 向量与数组	3
3 执行控制	5
4 函数	8
5 字符串	12

## 1 入门

### 编程

1. 给定三角形的两条边, $a = 4.5, b = 6$ , 这两条边的夹角是  $35^\circ$  计算第三条边的长度和三角形的面积。
2. 第九回合下半局, 满垒, 勇士队落后三分。奇珀·琼斯走向板, 挥杆两次, 都没打中。人群涌向出口。下一个球快速掉到了中间, 他挥动球杆, 完美地接触到球, 沿  $45^\circ$  角将球打到了 400 英尺以外的栅栏。
  - a. 如果忽略空气阻力, 他必须用多快的速度击中球才可以将球打到栅

- 栏的地板 b. 做一个简单的实验, 看看是否有一个更好的打球角度可以清除 12 英尺的栅栏
3. 一个冰淇淋蛋筒的长度为 6 英尺, 边缘直径为 2 英尺, 假设蛋筒上面的冰淇淋是半球形, 计算蛋筒中合适多重的冰淇淋。蛋筒的厚度可以忽略不计。假设 1 加仑冰淇淋的重量为 8 磅, 体积为 7.5 立方英尺。
  4. 为  $\theta$  选择一个合适的值, 计算方程  $\sin \theta$ 、 $\cos \theta$  和  $\tan \theta$ , 以及证明三者的关系。
  5. 我希望淋浴器的水温能保持在 100 华氏度, 但又买不起那些能控制水温的热水系统。我不在乎水流的速度。水的初始温度是 50 华氏度, 热水器的加热速度是 50000BTU/小时。要保证水温在 100 华氏度以上淋浴器的最大流速 (立方英尺/分钟) 是多少?
  6. 一辆电动车以每小时 35 英里的速度行驶, 平均需要消耗 45 马力。要使该汽车连续运行 25 英里 (只有晚上在家时才能给电池充电), 需要电池具有多大的蓄电能力? 该汽车需要多少节 D 号碱性电池?
  7. 你如果想要买一套 300000 美元的房子, 首付为 20%, 当前的复利率是 4.5%。
    - a. 计算如下数据:
      - 贷 30 年, 而每月的返款额。
      - 相当于多少的年利率。
      - 贷款期间, 总共支付了多少利率。
    - b. 如果贷 15 年, 利率为 5%, 计算一下是否更划算?
  8. 我家与公司相距 1.5 英里, 每天我都要考虑是坐公交车还是走路。公交车的速度是 25 英里/小时, 我走路的速度是 4 英里/小时。如果坐公交车比走路先到公司的概率大于 50%, 那么公交车运行的频率是多少?
  9. 有一个高为 5 英寸的圆台形玻璃杯, 顶部的直径为 3.5 英寸, 底部直径为 2 英寸。以 2 加仑/分钟, 公交车运行的频率是多少?
  10. 根据以下公式可以计算某个物体的空气阻力:

$$Drag = \frac{1}{2\rho} V^2 C_d S$$

其中, 空气密度  $\rho=1.3$  千克/平方米, 阻力面积  $C_dS$  是物体在空气中移动时受到阻力的值。一个物体从空气中坠落  $y$ , 当空气阻力等于物体的重力时, 该物体的速度为终端速度。一个 80 公斤的跳伞运动员, 如果手脚展开与地面平行,  $C_dS=0.7$ , 如果头朝下与地面垂直,  $C_dS=0.15$ 。一个跳伞运动员采用平行的姿势, 从 5000 米高的飞机上跳下, 20 秒后, 另一个跳伞运动采用垂直的姿势从飞机上跳下。请问经过多少时间, 第二个运动员可以追上第一个运动员? 他们第一次相遇时, 他们所在的高度是多少? 为了方便计算, 可以假设运动员刚跳下飞机就达到了终端速度。

11. 给定一个圆, 半径为 5, 中心坐标为  $x = 1, y = 2$ 。计算一些直线与该圆的相交点, 得到这些相交点的坐标。至少使用以下直线验证代码:

$$y = 2x - 1$$

$$y = -2x - 10$$

$$y = x + 5.9054$$

## 2 向量与数组

### 编程

- 这些练习不要使用直接输入法创建向量。编写一个完成以下任务的脚本。
  - 构建一个包含 6 33 之间的所有偶数的向量, 包含端点。结果保存在变量 *evens* 中 (注意: 33 不是偶数)。
  - 构建向量 *threes*, 初值为 8, 终值为 38, 步数为 3。
  - 构建向量 *reverse*, 初值为 20, 终值为 10, 步长为-1。
  - 构建含有 100 个元素的线性等分向量 *theta*, 初值为 0, 终值为  $2\pi$ 。
  - 构建含有 15 个元素的全零向量 *myZeros*。
  - 构建向量 *random*, 包含 15 个 1 12 之间的随机数。
- 编写一个完成以下练习脚本:
  - 定义向量 *vec* = [45 8 2 6 98 55 45 -48 75]。不想要偶数, 编写删除 *vec* 中所有偶数 (即 8、2、6、98、-48) 的脚本。你应该修改向量 *vec*, 而不是将答案保存在一个新的变量中。因为该命令需要适用于任何长度

的向量，所以不能使用直接法。

b. 新建变量 *vLength*, 用于保存 *a* 中修改后的向量 *vec* 的长度。应该使用内置函数计算基于向量本身的值。

c. 新建变量 *vSum*, 用于保存向量 *vec* 的长度。应该使用内置函数计算基于向量本身的值。

d. 使用两种方法计算向量 *vec* 中所有元素的平均值。第一种，使用内置函数找出 *vec* 的平均值。第二种，使用 b 和 c 中的结果计算 *vec* 的平均值。

e. 新建一个变量 *vProd*, 用于保存变量 *vec* 中所有元素的乘积。应该使用内置函数计算基于向量本身的值。

3. 编写一个使用向量运算解决以下问题的脚本：

a. 假设已有两个相同的长度的向量 *A1*、*B1*。创建向量 *C1*，使其包含 *A1 B1*，格式为  $C1 = [A1(1) \ B1(1) \ A1(2) \ B1(2) \dots A1(end) \ B1(end)]$ 。例如，如果  $A1=[2,4,8], B1=[3,9,27]$ ，则 *C1* 应该为  $[2,3,4,9,8,27]$ 。

b. 假设已有两个不同长度的向量 *A2*、*B2*。创建向量 *C2*，使其包含 *A2*、*B2*，格式与 a 相似。可是，其中一个向量中的元素运行完成后，*C2* 还包含较长的向量中的剩余的元素。例如，如果  $A2=[1 \ 2 \ 3 \ 4 \ 5 \ 6], B2=[10 \ 20 \ 30]$ ，则  $C2=[1 \ 10 \ 2 \ 20 \ 3 \ 30 \ 4 \ 5 \ 6]$ ， $A2=[1 \ 2 \ 3], B2=[10 \ 20 \ 30 \ 40 \ 50]$ ，则  $C2=[1 \ 10 \ 2 \ 20 \ 3 \ 30 \ 40 \ 50]$ 。

4. 编写脚本：给定数字向量 *nums*，新建向量 *newNums*，从 *nums* 的一个元素开始，一个隔一个的取出元素，放入 *newNums* 中。例如，如果  $nums=[6 \ 56 \ 8 \ 445 \ 7 \ 357 \ 4]$ 。注意，不能简单地给数字写入答案；该代码应该适应于任何数字向量。

5. 给定一个测试分数的向量 *tests*，将这些分数映射在区间 0 100 上，结果保存在新的向量 *normTest* 中。0 仍然对应 0，测试分数的最大值对应 100。例如，如果  $tests=[90 \ 45 \ 76 \ 21 \ 85 \ 97 \ 91 \ 84 \ 79 \ 67 \ 76 \ 72 \ 89 \ 95 \ 55]$ ，则 *normTests* 应该为  $[92.78 \ 46.39 \ 78.35 \ 21.65 \ 87.63 \ 100 \ 93.81 \ 86.6\dots 81.44 \ 69.07 \ 78.35 \ 74.35 \ 91.75 \ 97.94 \ 56.71]$ ;

6. 编写脚本：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，如果  $A=[1 \ 2 \ -1 \ 5 \ 6 \ 7 \ -4 \ 3 \ -2 \ 0]$ ，则 *B* 应该为  $[1 \ 8 \ 0 \ 125 \ 216 \ 343 \ 0 \ 27 \ 0 \ 0]$ 。

7. 好消息！你被《危险边缘》选中参加今年秋天的节目了。你的优势是可以建立一个对应题板上所有问题的值的数组。
- 编写代码：生成 5 行 6 列的矩阵 *jeopardy*，列都相同，但行中的值从 200 1000 等值增加。
  - 接下来，生成与 *jeopardy* 大小相同的矩阵 *doubleJeopardy*，但行中的值从 400 2000 等值增加。
  - 你觉得可以更进一步，练习一些还没有出现的问题。建立矩阵 *squaredJeopardy*，用于保存原始矩阵 *jeopardy* 平方的每一项。
8. 编写一个名称为 *arrayCollide* 的脚本，它结合了两个数组，并进行排序，然后返回一个指定大小的新数组。该脚本应该处理以下数据：
- *A*: 一个任意大小的二维数组。
  - *B*: 与 *A* 的大小不同的另一个二维数组。
  - *C*: 表示新数组列数的数字。

生成一个大小为  $N \times M$  的数组 *res*，保存 *A* 和 *B* 的前  $N \times M$  个元素，并按列排序。如果  $N \times M$  大于 *A* 和 *B* 中元素的总和，则用 0 填充空位。测试该脚本的方法：编写另一个脚本，重复设置 *A*、*B*、*M*、*N* 的值，然后调用脚本 *arrayCollide*。你可以创建尽可能多的测试案例。例如，如果  $A=[1\ 2\ 3; 5\ 4\ 6], B=[7\ 8; 9\ 10; 12\ 11], N=3, M=4$ ，则 *res* 为

$$\begin{bmatrix} 1 & 4 & 5 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}$$

如果  $N=4$ ，则 *res* 为

$$\begin{bmatrix} 1 & 5 & 9 & 0 \\ 2 & 6 & 10 & 0 \\ 3 & 7 & 11 & 0 \\ 4 & 8 & 12 & 0 \end{bmatrix}$$

### 3 执行控制

编程

1. 编写一个解决以下问题的脚本: 假设有一个名称为  $D$  的向量, 使用迭代语句 (for 或 while) 和条件语句 (if 或 switch) 将向量  $D$  分成  $posEven$ 、 $negEven$ 、 $posOdd$ 、 $negOdd$  四个向量。

- $posEven$  包含  $D$  中的所有正偶数。
- $negEven$  包含  $D$  中的所有负偶数。
- $podOdd$  包含  $D$  中所有正奇数。
- $negOdd$  包含  $D$  中所有负奇数。

例如:

$if D = [-4, -3, -2, -1, 0, 1, 2, 3, 4],$   
 $posEven = [2, 4], negEven = [-4, -2]$   
 $posOdd = [1, 3] \text{ and } negOdd = [-3, -1]$

2. 必须使用 for 或 while 来解决以下问题。
- a. 使用 for 循环遍历向量  $A$ , 返回一个元素为逻辑值的向量  $B$ 。 $B$  中的  $true$  对应  $A$  中的正值,  $false$  对应  $A$  中的其他值。例如, 如果  $A = [-300 \ 2 \ 5 \ -63 \ 4 \ 0 \ -46], B = [false \ true \ true \ false \ true \ true \ false]$ 。
  - b. 使用 while 循环遍历向量  $A$ , 返回一个新向量  $B$ 。 $B$  中的  $true$  对应  $A$  中的正值,  $false$  对应  $A$  中的其他值。
  - c. 使用 for 循环遍历逻辑数组  $N$ , 返回一个新向量  $M$ 。 $M$  中对应  $N$  中的  $true, -1$  (不是逻辑值) 对应  $N$  中的  $false$ 。例如, 如果  $N = [2 \ -1 \ -1 \ 2 \ 2 \ -1 \ 2]$ 。
  - d. 使用 while 循环遍历数组  $z$ 。在遇到一个大于 50 的数字之前, 用 3 代替  $z$  中的元素。例如, 如果  $z = [3 \ 3 \ 3 \ 3 \ 3 \ 3 \ 3 \ 3 \ 34 \ 43]$ 。
3. 你的新公司正在招聘毕业生。人力资源部让你写一个脚本, 这将帮助他们决定求职者面试后得到工作的机会。下表列出了决定求职者得到工作机会的规则。

GPA 值	被雇佣的机会	GPA 值	被雇佣的机会
$GPA \geq 3.5$	90%	$2.0 \leq GPA < 2.5$	60%
$3.0 \leq GPA < 3.5$	80%	$1.5 \leq GPA < 2.0$	40%
$2.5 \leq GPA < 3.0$	70%	$GPA < 1.5$	30%

你的脚本应该重复地要求用户提供 GPA 值, 并计算求职者被雇佣的

机会。在用户输入一个负值前，该脚本应该不断地询问 GPA 值。例如：

- GPA 输入：4，应该返回 0.9。
- GPA 输入：3.5，应该返回 0.9。
- GPA 输入：3.4，应该返回 0.8。

4. 你刚到一家本地区最好的软件公司进行暑假实习。可是，上班第一天，你知道未来三个月你唯一的工作就是将二进制数 (基数为 2) 转化为十进制数 (基数为 10)。你决定编写一个脚本，在用户输入一个无效值 (包含除了 0 和 1 以外的其他数字) 之前，他会反复要求用户提供一个二进制数，并返回对应的十进制数。输入的数字只能含有 0 和 1。最右边的数字表示的值为  $2^0$ ，其左边第  $N$  位数字表示的值为  $2^N$ 。例如，输入 110101 返回

$$53 = 2^5 + 2^4 + 2^3 + 2^0$$

必须使用迭代来求解该问题。注意：函数 `inout(...)` 提示用户输入值，根据 MATLAB 规则解析输入的字符，并返回结果。

5. 你有一个小朋友，他 (她) 小衣柜中的衣服太多了。作为好朋友，你要帮忙决定是否每一件衣服都值得保存。你决定编写一个计算各件衣服的价值值的脚本。决定衣服价值的属性有 5 个，分别是：款式 (condition)、颜色 (color)、价格 (price)、搭配衣服的数量 (matches)、舒适度 (comfort)。每个属性对应一个等级，等级的范围是 1 到 5。编写一个名称为 `clothes` 的脚本，它会要求客户输入各个属性的等级，并将结果保存在一个向量中。向量中各个属性的顺序为：`[condition color price matches comfort]`。脚本应该计算一个 0 到 100 的分值；100 表示好的衣服，而 0 表示不好的衣服。各个取值的属性如下所示：

condition: 1=>0; 2=>5; 3=>10; 4=>15; 5=>20

color: 1=>blue=>12;

2=>red=>2;

3=>pink=>15;

4=>yellow=>20;

5=>white=>12

price: 1=>8,2-3=>16,4-5=>20

matches: 1-2 => 8,3-5=> 19

comfort: 1=>6,2-3=>13,4-5=>18

注意：如果某个属性的等级不在 1 5 之间，没有对应值。

6. “yard” 是一种传统的英国容器。他的长度为 36 英寸，相当于是一个直径为 4 英寸的玻璃球连接到一个锥形物体上，锥形物体一段的直径为 1 英寸，另一端的直径为 6 英寸。编写一个完成一下练习的脚本：
  - a. 提示用户输入液体的高度。
  - b. 计算该高度对应的液体的体积。
7. 现在，你已经很熟悉迭代了，下面，你要解决一个有趣的问题。看来对方大学的数学系又一次搞砸了，他们忘记了  $\pi$  的值。你要写一个名称为 *myypi* 的函数，输入是所需的精度，并假设  $\pi$  的值约为等于该精度。你将使用以下这种基于几何概率的算法。

单位正方形中有四分之一一个单位园 (面积为  $\pi/4$ )，在正方形中随机中选择一个点，如果该点在四分之一一个单位园中，表示“命中”，如果在其他地方，表示“失误”。四分之一一个单位圆的面积近似于“命中”的次数除以选择的次数。

该函数应该重复计算命中和失误的次数，重量次数不少于 1000 次，并且连续多个  $\pi$  的估算值在指定的精度内。它将返回  $\pi$  的估算值

**提示**     该问题可以使用函数 `rand(...)`

## 4 函数

### 编程

1. 编写一个名称为 *checkFactor* 的函数，输入是两个数字，功能是检查第一个数字是否可以被第二个数字除尽。可以假设这两个数字都是正数。该函数返回逻辑值 *true* 或 *false*。

例如：

*checkFactor*(25,6) 应该返回 *false*

*checkFacor*(9,3) 应该返回 *true*

*checkFacor*(3,9) 应该返回 *false*

**提示**     *mod*(*x*,*y*) 返回 *x* 除以 *y* 的余数。



2. 编写一个名称为 *mysteryFunction* 的函数并测试其代码, 该函数的输入是向量 *V*, 输出是新向量 *W*。*W* 的长度与 *V* 相同, 元素是 *V* 中对应元素与前一个元素的总和。假定 *V*(1) 的前一个元素是 0。

例如:

*mysteryFunction*(1 : 8) 应该返回 [1 3 5 7 9 11 13 15]

*mysteryFunction*([1 : 6].<sup>2</sup>) 应该返回 [1 5 13 25 41 61]

3. 从去年 7 胜 6 负的成绩看来, 你的球队希望在这个赛季能有所进步。他们请你帮助预测一下往后的输赢情况, 希望你能编写一个名称为 *teamRecord* 的函数, 该函数的输入是参数 *wins* 和 *losses*, 输出是 *season* 和 *wPercentage*。*season* 是一个逻辑值, 如果该赛季赢了, 就等于 *true*。*wPercentage* 是获胜的百分比 (值为 0 到 100)。

例如:

[*season wPercentage*] = *teamRecord*(3,9)

应该返回 *season* = false, *wPercentage* = 25 [season

*wPercentage*] = *teamRecord*(10,2)

应该返回 *season* = true, *wPercentage* = 83.3

4. 编写一个名称为 *classAverage* 的函数, 该函数的输入是一个数字数组, 将这些数据进行标准化, 最大值对应于 100(参见第三章, 第五题), 该函数的输出是该组数据平均值的字母等级。等级范围如下:

average > 90 => A

80 <= average < 90 => B

70 <= average < 80 => C

60 <= average < 70 => F

例如:

*classAverage*([70 87 95 80 80 78 85 90 66 89 89 100]) 应该返回 B

*classAverage*([50 90 61 82 75 92 81 76 87 41 31 98]) 应该返回 C

*classAverage*([10 10 11 32 53 12 34 74 31 30 26 22]) 应该返回 F

5. 编写一个名称为 *myMin4* 的函数, 该函数的输入是四个数字, 输出的是最小值和最大值对应的索引。不能使用内置函数 *min*()。

例如:

*myMin4*(1, 3, 5, 7) 应该返回 1 和 1

*myMin4*(8, 9, 2, 4) 应该返回 2 和 3

6. 编写一个名称为 *meansAndMedian* 的函数，该函数的输入是数字向量，输出是算数平均值、几何平均值和中值。不能使用内置函数 *mean()*、*median()* 和 *geomean()*。但是，你可以通过“*help geomean*”来熟悉如何计算一组数字的几何平均值。

**提示** 内置函数 *sort()* 能帮你计算向量的中值。

7. 已知一个数字数组，该数组中可能有负值，编写函数 *postavg(a)*，计算并返回一维数组 *a* 中非负数字的平均值 (*mean*)。一种解决方法是 *mean(a(find(a > 0)))*。为了测试你对概念“类”的理解，请使用迭代重新实现函数 *posavg(a)*。在你的解决方案中，不能使用内置函数 *sum(...)*、*find(...)*、或 *mean(...)*。
8. 编写一个名称为 *subAndAverage* 的函数，该函数的输入是一个数字数组，输出是数组的总合、平均值。

例如：

*sumAndAverage*([3 2 3 2]) 应该返回 10 和 2.5

*sumAndAverage*([5 -5 2 8 0]) 应该返回 10 和 2

*sumAndAverage*([]) 应该返回 0 和 0

9. 你已经熟悉了逻辑运算符 *&&*(逻辑与)、*||*(逻辑或)、*~*(逻辑非)。在弱类型语言 (如 MATLAB) 中，二进制状态 *true* 和 *false* 可以表示为 1 和 0。假设有一个三元数系统，由状态 *true()*、*maybe(2)*、*false(0)* 组成，该系统的真假表如下。请编写实现该真值表如下。请编写实现该真值表的函数 *f = tnot(x)*、*f = tand(x,y)*、*f = tor(x,y)*。不要假设只输入有效的输入数字。

x	y	$tnot(x)$	$tand(x, y)$	$tor(x, y)$
1	1	0	1	1
1	0	0	0	1
1	2	0	2	1
0	1	1	0	1
0	0	1	0	0
0	2	1	2	0
2	1	2	2	1
2	0	2	2	0
2	2	2	2	2

10. 编写函数  $multiSum(A)$ , 该函数的输入是  $N \times M$  数组  $A$ , 输出是以下 4 个结果:

一个  $1 \times M$  向量: 列的行。

一个  $n \times 1$  向量: 行的列。

两个数字: 两个对角线的和, 主对角线在前。

例如:

$columnSum([1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9])$  应该返回  $[12\ 15\ 18]$ 、 $[6\ 15\ 24]^T$ 、15、15

$columnSum([0\ 2\ 3; 4\ 0\ 6; 7\ 8\ 0])$  应该返回  $[11\ 10\ 9]$ 、 $[5\ 10\ 15]^T$ 、0、10

$columnSum(eye[5, 5])$  应该返回  $[1\ 1\ 1\ 1\ 1]$ 、 $[1\ 1\ 1\ 1\ 1]^T$ 、5、1

$columnSum([])$  应该返回  $[]$ 、 $[]$ 、0、0

11. 你在玩一个掷骰子的游戏, 可以掷 10 次。如果你能掷七次及七次以上 5 或 6, 可以赢 2 美元; 如果你能掷四次及四次以上, 赢 1 美元; 如果掷 5 或 6 的次数等于或少于 3 次, 就赢不了钱。编写一个名称为  $diceGame$  的函数, 输入是一个向量, 表示骰子的值, 输出的是赢得金额。

例如:

$diceGame([5\ 1\ 4\ 6\ 5\ 5\ 6\ 6\ 5\ 2])$  应该返回 2

$diceGame([2\ 4\ 1\ 3\ 6\ 6\ 6\ 4\ 5\ 3])$  应该返回 1

`diceGame([1 4 3 2 5 3 4 2 6 5])` 应该返回 0

注意: 该函数应该是适用于任何长度的向量。

## 5 字符串

### 编程

1. 解答下列字符串方面的问题。
    - a. 编写函数 `dayName`, 输入的是参数 `day`, 表示 2008 年 9 月的一个日期。该函数将返回一个字符串, 表示输入的日期是星期几。例如:  
`dayNum(8) 'Monday'`。
    - b. 给定一个名为 `days` 的变量, 这是一个向量, 包含了 2008 年 9 月的多个日期。编写脚本, 将向量 `days` 中的日期转化为星期几, 并返回一个名为 `daysOfWeek` 的字符串, 字符串的各个名称用一个逗号和空格隔开。例如, 如果 `days = [8,9,10]`, `daysOfWeek` 应为 `'Monday','Tuesday','Wednesday'`。
- 注意, 第一个名称前面和最后一个名称后面都没有分隔符。

**提示** 也许应该连接名称和分隔符

2. 考虑以下, MATLAB 系统怎样解析以下字符串:  
`'V=[1 2 3 4;5,6, 7;8; 9 10]'`  
 你的任务是使用 `strtok` 解析该字符串, 并构建一个对应的数组。编写函数 `arrayParse`, 输入一个字符串, 返回的是两个变量: 一个字符串 (即, 变量名) 和一个数组。
  - a. 首先, 以 '=' 为分割符进行分词, 分割变量名和对应的表达式。将变量名返回给用户, 将表达式赋值给变量 `str1`, 以便进一步处理。可以假设字符 '[' 和 ']' 两边没有空格。
  - b. 以 '[' 和 ']' 为分隔符进行分词, 删除连接运算符, 将第一个分词保存为 `str2`。
  - c. 以 ';' 为分隔符进行分词, 生成 0 个或多个字符串, 一个字符串表示数组的一行。将他们保存在变量 `rowString` 中。暂时可以假设第一行是最长的。
  - d. 使用嵌套 `while` 循环, 以 ',' 和 ']' 为分隔符进行分词, 用 `str2num(...)`

提取每个数组项的数值，将其保存为 *rowEntry*。

e. 水平地连接各个 *rowEntry* 元素，生成数组的行。如果某行太短，用 0 填充。

f. 垂直地连接各行，生成结果数组，并将该数组返回给用户。

g. 使用以下案例测试该函数：

```
empty=[ ] row=[1 2 3 4] diag=[0 0 0 1; 0 0 1; 0 1; 1]
```

3. 编写一个名称为 *DNAcomplement* 的函数，输入是一个由一组字母组成的字符串，表示一个 *DNA* 序列，如 'gattaca'。该函数的输出是 *DNA* 序列的互补序列，即，a's 会变成 t's, g's 会变成 c's, 反之亦然。所以，字符串 'gattaca' 会变成 'ctaagt'。假设 *DNA* 序列中所有字母都是小写的，并且，他们只能是 a、t、g、c 中的一个。

注意：你可能会试着用迭代来完成该问题，但是，没有必要。

4. 函数 *rot(s,n)* 是一个简单的凯赛加密算法，他会将字符串中的各个英文字母替换为新的字母，而新的字母是原来字母按字母表顺序向前或向后移动 *n* 个位置后的字母。例如：*rot('Baz!',3)*，会将字符串中的每个英文字母替换为他们左边的第三个字母。数字、符号和非字母不能进行转换。执行以下函数：

```
funcion rotateText = rot(text,n)
```

为了更好地解答该问题，可以在 *rot.m* 文件中编写几个局部函数：*isUppercaseLetter(letter)*、*getUppercaseLetter(n)*、*getLowcaseLetter(n)*、*getPosition(letter)*。也可以得到内置函数 *isletter(...)*、*find(...)*、*mod(...)*。

5. 你遇到一个大问题。*CS* 课程的教授决定，你能通过该课程的唯一方法是编写一个能让他摆脱困境的函数。他班上的所有成绩都被意外地保存在一个很长的字符串中了，该字符串只包含有字母 A、B、C、D、F、Y。a. 你的工作是编写一个名称为 *CrazyGrade* 的函数，他能按照下面的规则处理字符串：

A 变成 F

B 变成 D

C 保持不变

D 变成 B

F 变成 A

$Y$  变成  $W$  函数的输入是一个字符串, 输入的是一个转换后的字符串。假设字符串中含有有效字母。例如,

*CrazyGrade*('BADDAD') 应该返回'DFBBFB'

*CrazyGrade*('BAYBAY') 应该返回'DFWDFW'

6. 更糟糕的是, 他还让你整理修改后的成绩。编写一个名称为 *GradeDist* 的函数, 将相同的成绩 (将所有的  $A$  放在一起将所有的  $B$  放在一起, 以此类推) 聚集到一起, 然后, 计算并返回一个成绩分布, 在该函数中输入一个字符串后, 会返回一个字符串和一个数组, 字符串将相同的成绩聚集到一起了, 数组则包含了  $A$  到  $F$  的所有百分值, 例如, 如果有 15% 的  $A$ 、16% 的  $B$ 、33% 的  $C$ 、16% 的  $D$ 、16% 的  $F$  和 4% 的  $W$ , *Grade - Dist* 应该返回 [15 16 33 16 16 4]。