



中山大學

**ISE128 Artificial Intelligence Programming
Language Introduction Assignment**

人工智能编程语言考核作业

姓名：方桂安

学号：20354027

邮箱：2639135175@qq.com

所有作业中的 MATLAB 脚本的命名格式：

Assignment1_题号_小题号（如果存在）_小题的小题号（如果存在）

Problem 1 (10 分)

Problem 1.1. 请对比命令 *who* 和 *whos*, 列出他们的功能、工作对象等的异同 (2.5 分)

对于未知命令的使用可以在[官方帮助中心](#)了解或者在命令行窗口输入 `help name`, `name` 可以是函数、方法、类、工具箱或变量。

相同点:

1. 二者均可列出工作区中的变量
2. 语法类似, 都可通过相应命令列出活动工作区, MAT 文件, 全局工作区等的变量

不同点:

1. *whos* 更为全面, 可列出变量的大小, 类型, 属性等
2. `S = whos(____)` 将变量的信息存储在结构体数组 `S` 中而 `C = who(____)` 将变量的名称存储在元胞数组 `C` 中

```
>> who

您的变量为:

c g p s

>> whos

Name      Size      Bytes  Class      Attributes
-----
c         1x3         48  double    complex
g         1x1          8  double    global
p         1x1          8  double
s         5x5        128  double    sparse
```

效果如图:

Problem 1.2. 请先给变量 `a`, `b`, `c` 赋值后再清除变量 `a` 和 `b`。(2.5 分)

执行操作如图:



工作区中变量仅存 `c`, 注意清除时不能使用 `clear`, 否则会将所有变量清除。

Problem 1.3. 以下哪种描述是正确的，以及原因 (5 points)

- (a) 变量名可以为“12a”;
- (b) 变量“A”和变量“a”代表的是同一个变量;
- (c) 变量名可以为“a_bcd”;
- (d) 变量名可以为“.abcd”;

(a) 不正确，变量名以数字开头，会报错并被提醒检查缺失的乘法运算符、缺失或不对称的分隔符或者其他语法错误等。

(b) 不正确，“A”与“a”是互相独立的两个变量，没有关系（通过 $a+A$ 的结果验证）

(c) 正确，变量名可以包含数字，字母，下划线（不以数字开头）

(d) 不正确，变量名不可以包含小数点（经过尝试，放在字母前中后都导致相应的报错）

验证如下：

```
>> 12a=1
```

```
12a=1
```

```
↑
```

错误：表达式无效。请检查缺失的乘法运算符、缺失或不对称的分隔符或者其他语法错误。要构造矩阵，请使用方括号而不是圆括号。

是不是想输入：

```
>> a=1;A=2;
```

```
>> a+A
```

```
ans =
```

```
3
```

```
>> a_bcd=1;
```

```
>> .abcd=2
```

```
.abcd=2
```

```
↑
```

错误：运算符的使用无效。

Problem 2 (10 分)

Problem 2.1. 使用 MATLAB 的 *help* 功能用以下方式搜索函数 *exp* : (a) 在命令行窗口, (b) 在 Matlab Help 窗口. (5 分)

效果分别如下：

exp - 指数

此 MATLAB 函数 为数组 X 中的每个元素返回指数 e^x 。对于复数元素 $z = x + iy$ ，它返回以下复指数

$Y = \exp(X)$

另请参阅 [expint](#), [expm](#), [expm1](#), [log](#), [log10](#), [mpower](#), [power](#)

[exp 的文档](#)

[名为 exp 的其他函数](#)

exp

指数

R2020a

全页折叠

语法

$Y = \exp(X)$

说明

$Y = \exp(X)$ 为数组 X 中的每个元素返回指数 e^x 。对于复数元素 $z = x + iy$ ，它返回以下复指数

示例

全部折叠

e 的数字表示形式

计算 1 的指数，它是欧拉数 e 。

尝试此示例

View MATLAB Command

exp(1)

ans = 2.7183

Problem 2.2. 使用 **lookfor** 命令确定以 10 为底的 log 函数用法, 并计算 $\log_{10}(2)$.
(5 分)

先使用 lookfor 命令搜索 log，点击 log10 观看函数用法

命令行窗口

[countType](#)
[findAll](#)
[findCat](#)
[findType](#)
[getDialogSchema](#)
[getDialogSummaryList](#)
[getDialogTypeCat](#)
[MessageLog](#)
[showDlgForMsgItem](#)
[showDlgForMsgLog](#)
[updateForLinkedLog](#)
[isOptionsDialogUsingStaticMethods](#)
[logicalToOnOff](#)
[onOffToLogical](#)
[RegisterDDGMethods](#)

- Count all messages of chosen type from message log.
- Find all messages in message log.
- Find all messages of chosen category from message log.
- Find all messages of chosen type from message log.
- Construct MessageLog dialog.
- Return header and listbox for message summaries.
- Return type and category selected in dialog.
- MessageLog Define the MessageLog class.
- Open dialog, based on MessageItem type and AutoOpenMode.
- Open dialog, based on content of MessageLog and AutoOpenMode.
- Update message log dialog, if open, due to change
- True if the object is
- Convert a logical to an on/off value.
- Convert on/off to a logical value.
- Register basic dialog methods.

log10 - 常用对数（以 10 为底）

此 MATLAB 函数 返回数组 X 中每个元素的常用对数。该函数同时接受实数和复数输入。对于 X 在区间 $(0, \text{Inf})$ 内的实数值，**log10** $(-\text{Inf}, \text{Inf})$ 内的实数值。对于 X 的复数值和负实数值，**log10** 函数返回复数值。

$Y = \log_{10}(X)$

另请参阅 [exp](#), [log](#), [loglp](#), [log2](#), [loglog](#), [logm](#), [reallog](#), [semilogx](#), [semilogy](#)

[log10 的文档](#)
[名为 log10 的其他函数](#)

计算获得答案：

```
>> a=log10(2)
```

```
a =
```

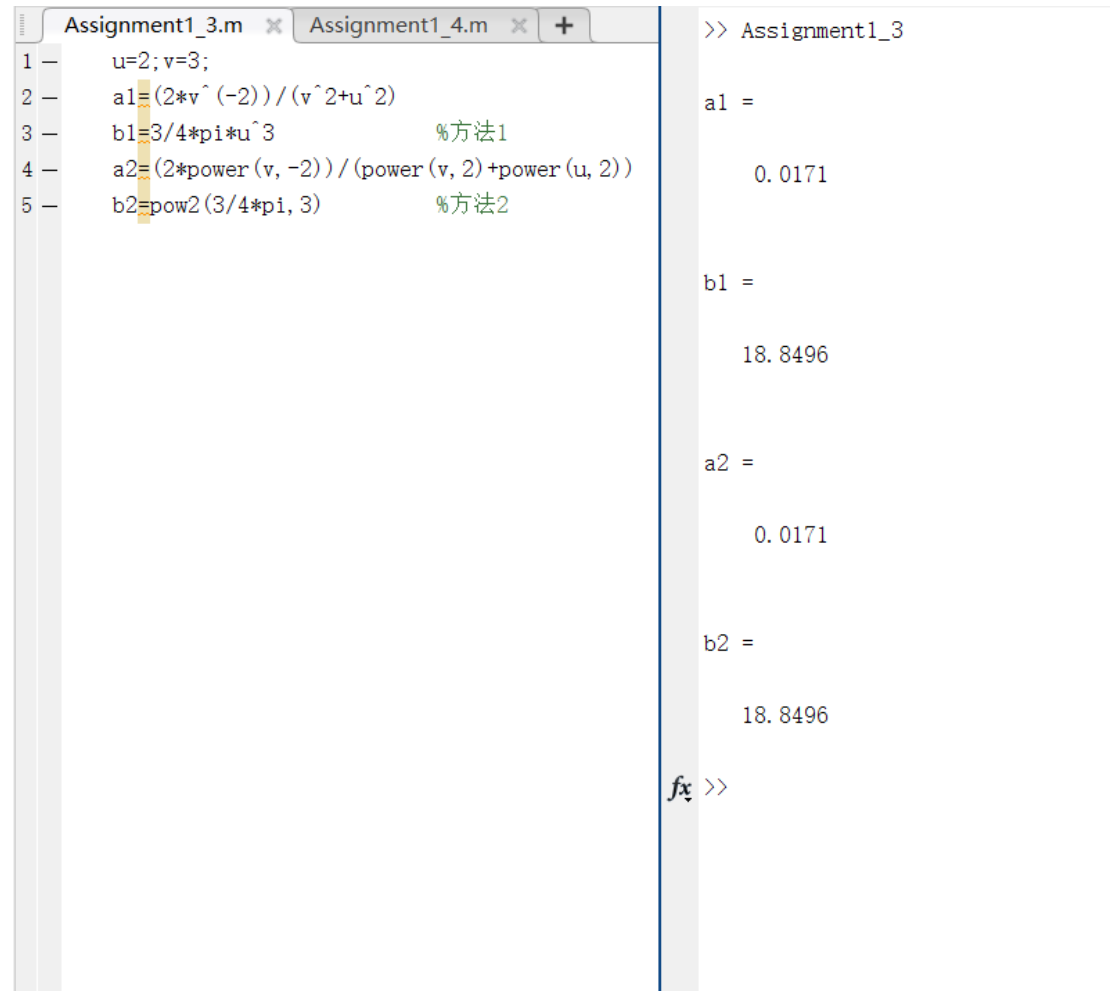
```
0.3010
```

Problem 3 (10 points)

已知 $u = 2$, $v = 3$. 请在 MATLAB 中计算以下表达式的值，并用不同表达方式计算。

(a) $\frac{2v^{-2}}{v^2+u^2}$ (5 points) (b) $\frac{3}{4}\pi u^3$ (5 points)

题目要求用不同表达方式计算，所以在进行幂的运算考虑使用 power 函数和 ^ 运算符两种方式，且 $u=2$ ，还可以使用以 2 为底的 pow2 函数。结果如图：



The image shows a MATLAB script editor with two tabs: 'Assignment1_3.m' and 'Assignment1_4.m'. The script in 'Assignment1_3.m' contains the following code:

```
1 - u=2;v=3;
2 - a1=(2*v^(-2))/(v^2+u^2)
3 - b1=3/4*pi*u^3 %方法1
4 - a2=(2*power(v,-2))/(power(v,2)+power(u,2))
5 - b2=pow2(3/4*pi,3) %方法2
```

The command window on the right shows the results of running the script:

```
>> Assignment1_3
a1 =
    0.0171
b1 =
    18.8496
a2 =
    0.0171
b2 =
    18.8496
fx >>
```

Problem 4 (10 points)

$$\text{已知} A = \begin{bmatrix} 3 & 7 & -4 & 2 \\ -5 & 9 & 10 & 2 \\ 6 & 13 & 8 & 11 \\ 15 & 5 & 4 & 1 \end{bmatrix}$$

(a) 构造一个矩阵 B,使得 A+B 形成的新矩阵各元素的值为 20. (5 分)

(b) 求解 A*B. (5 分)

(a) 为了使 A+B 各元素均为 20, 先构造各元素都为 20 的 A 的同型矩阵 C, 由此可得 B=C-A。

(b) 直接计算 A*B, 也就是两个矩阵的位乘。

结果如下图所示:

```

1 - A=[3 7 -4 2;-5 9 10 2;6 13 8
2 - C=repmat(20,4);
3 - B=C-A
4 - D=A*B

>> Assignment1_4

C =

    20    20    20    20
    20    20    20    20
    20    20    20    20
    20    20    20    20

>> Assignment1_4

B =

    17    13    24    18
    25    11    10    18
    14     7    12     9
     5    15    16    19

D =

    180    118    126    182
    290    134    122    200
    594    442    546    623
    441    293    474    415

fx >>

```

Problem 5 (20 points)

编写可以实现下列操作的命令 (不允许手动逐一输入矩阵各元素, 以完成下列要求)

- 构造一个 20×40 矩阵 **A**, 并使得矩阵 **A** 的第 1 行到第 10 行所有元素值为 1, 矩阵 **A** 从第 11 行到第 20 行所有元素值为 2; (5 分)
- 构造一个 20×40 新矩阵 **B**, 矩阵 **B** 与矩阵 **A** 相同, 除了矩阵 **B** 的第 11 行满足 $\mathbf{B}(11, j) = 1/j$, for $1 \leq j \leq 40$; (5 分)
- 构造一个 20×41 矩阵 **C**, 矩阵 **C** 与矩阵 **B** 相同前 40 列相同, 第 41 列的所有值为 3; (5 分)
- 构造一个 20×41 矩阵 **D**, 矩阵 **D** 与矩阵 **C** 一样, 除了主对角线上的前 10 个元素值为 $\mathbf{D}(i, i) = i * \mathbf{C}(i, i)$, for $1 \leq i \leq 10$. (5 分)

(a) 矩阵 A 由 10x40 的全 1 矩阵和 10x40 的全 2 矩阵按列连接，需要使用到 repmat 函数和 cat 函数，为了增加可读性，前者可以用 ones 函数替代。

(b) 可以通过矩阵 A 来构造矩阵 B，使用 for 循环实现这一过程。

(c) 可以通过矩阵 B 来构造矩阵 C，构造全 3 列向量 c，再用方括号运算符 [] 进行水平串联

(d) 可以通过矩阵 C 来构造矩阵 D，改用 while 循环实现这一过程。

代码与结果如图：

Assignment1_5.m

```

1 A=cat(1,ones(10,40),repmat(2,10,40)); %构造矩阵A
2 B=A;
3 for j=1:40
4     B(11,j)=1/j;
5 end %构造矩阵B
6 c=repmat(3,20,1); %构造矩阵C
7 C=[B c]; %构造矩阵C
8 D=C; i=1;
9 while i<=10
10     D(i,i)=i*C(i,i);
11     i=i+1;
12 end %构造矩阵D

```

Output:

名称	值
A	20x40 double
B	20x40 double
C	20x1 double
D	20x41 double
i	11
j	40

B

20x40 double

	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1
11	1	0.5000	0.3333	0.2500	0.2000	0.1667	0.1429	0.1250	0.1111	0.1000
12	2	2	2	2	2	2	2	2	2	2
13	2	2	2	2	2	2	2	2	2	2
14	2	2	2	2	2	2	2	2	2	2
15	2	2	2	2	2	2	2	2	2	2
16	2	2	2	2	2	2	2	2	2	2
17	2	2	2	2	2	2	2	2	2	2
18	2	2	2	2	2	2	2	2	2	2
19	2	2	2	2	2	2	2	2	2	2
20	2	2	2	2	2	2	2	2	2	2

C

20x41 double

	37	38	39	40	41
1	1	1	1	1	3
2	1	1	1	1	3
3	1	1	1	1	3
4	1	1	1	1	3
5	1	1	1	1	3
6	1	1	1	1	3
7	1	1	1	1	3
8	1	1	1	1	3
9	1	1	1	1	3
10	1	1	1	1	3
11	0.0270	0.0263	0.0256	0.0250	3
12	2	2	2	2	3
13	2	2	2	2	3
14	2	2	2	2	3
15	2	2	2	2	3
16	2	2	2	2	3
17	2	2	2	2	3
18	2	2	2	2	3
19	2	2	2	2	3
20	2	2	2	2	3

	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	1	2	1	1	1	1	1	1	1	1
3	1	1	3	1	1	1	1	1	1	1
4	1	1	1	4	1	1	1	1	1	1
5	1	1	1	1	5	1	1	1	1	1
6	1	1	1	1	1	6	1	1	1	1
7	1	1	1	1	1	1	7	1	1	1
8	1	1	1	1	1	1	1	8	1	1
9	1	1	1	1	1	1	1	1	9	1
10	1	1	1	1	1	1	1	1	1	10
11	1	0.5000	0.3333	0.2500	0.2000	0.1667	0.1429	0.1250	0.1111	0.1000
12	2	2	2	2	2	2	2	2	2	2
13	2	2	2	2	2	2	2	2	2	2
14	2	2	2	2	2	2	2	2	2	2
15	2	2	2	2	2	2	2	2	2	2
16	2	2	2	2	2	2	2	2	2	2
17	2	2	2	2	2	2	2	2	2	2
18	2	2	2	2	2	2	2	2	2	2
19	2	2	2	2	2	2	2	2	2	2
20	2	2	2	2	2	2	2	2	2	2

Problem 6 (10 points)

请计算下列集合序列当 $N = 10$ and $r = 1/3$ 的 $N + 1$ 项之和。

$$S = \sum_{i=0}^N r^i = 1 + r + r^2 + r^3 + \dots + r^N,$$

不允许使用 **for** or **while** 循环。

由于不能使用循环累加，改用多项式解决这个问题，构造 1 到 $N+1$ 的系数向量 p ，令 $r=1/3$ 使用 `polyval` 函数求得总和 S

```

Assignment1_6.m x +
1 - p=ones(1,11);r=1/3;
2 - S=polyval(p,r)
3
>> Assignment1_6

S =

    1.5000

fx >> |

```

Problem 7 (30 points)

已知 Fibonacci（斐波那契）序列定义如下，

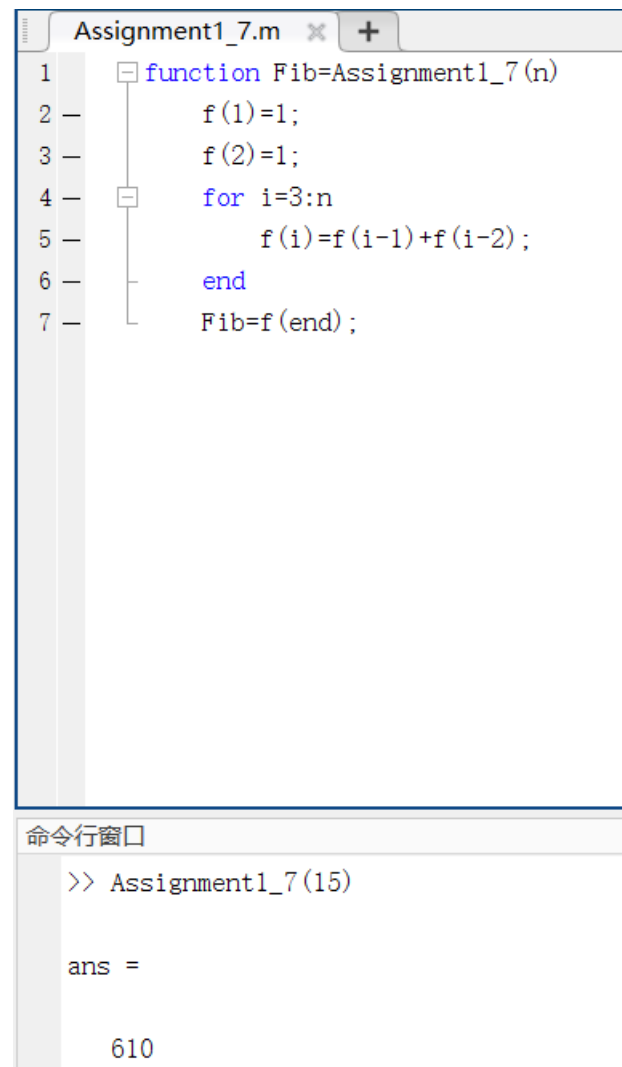
$$F_n = \begin{cases} 1, & n = 1; \\ 1, & n = 2; \\ F_{n-1} + F_{n-2}, & n \geq 3. \end{cases}$$

- (a) 使用 **for** 循环计算 F_{15} 。注意，在计算过程中，整个序列只有三个数值能被保留 F_{curr} , F_{old} , 和 F_{older} . (10 分)
- (b) 使用 **while** 循环找到 N , 使得 N 满足 $F_N < 1000$ 并且 $F_{N+1} \geq 1000$ 。再打印出列出相关斐波那契数的单索引数组 $[F_1, F_2, \dots, F_N]$. (10 分)
- (c) 计算前 40 个能被 2 或者 5 整除的斐波那契数 F_n 的和; $1 \leq n \leq 40$,

$$\sum_{n=1}^{40} \begin{cases} F_n, & \text{if } F_n \text{ is divisible by 2 or 5} \\ 0, & \text{otherwise} \end{cases}$$

(例如，第一个符合要求的斐波那契数为 $F_3 = 2$, 第二个为 $F_5 = 5$.) (10 分)

读完题写出函数，只需键入对应的 n 就可求得结果。



The image shows a MATLAB script editor window titled "Assignment1_7.m". The script defines a function `Fib=Assignment1_7(n)`. The function initializes `f(1)=1` and `f(2)=1`, then uses a `for` loop from `i=3:n` to calculate `f(i)=f(i-1)+f(i-2)`. Finally, it returns `Fib=f(end)`. Below the script editor, the command window shows the execution of `>> Assignment1_7(15)`, which returns `ans = 610`.

```

1  function Fib=Assignment1_7(n)
2      f(1)=1;
3      f(2)=1;
4      for i=3:n
5          f(i)=f(i-1)+f(i-2);
6      end
7      Fib=f(end);

```

命令行窗口

```

>> Assignment1_7(15)

ans =

    610

```

(a) 题目里提到 `for` 循环和保留三个数值，决定换一种做法，通过不断地交换数值，确保工作区内只有 `Fcurr`, `Fold` 和 `Folder` 三个变量存在

```

Assignment1_7_a.m
1  Folder=1;Fold=1;
2  for i=3:15
3      Fcurr=Folder+Fold;
4      Folder=Fold;
5      Fold=Fcurr;
6  end
7  disp(['F15=', num2str(Fcurr)])

命令行窗口
>> Assignment1_7_a
F15=610

```

(b) 设置空数组 array，循环判定 N 的过程中将前 N 项不断存入，每一项都有对应的 j，最终打印出单索引数组 $[F_1, F_2, \dots, F_N]$ 。

```

Assignment1_7_b.m
1  Folder = 1;Fold = 1; %给n=1和2的两种特殊情况赋值
2  j=0;array=zeros; %设置计数变量j和空数组array，便于给array(j)赋值;预分配内存加快计算速率
3  while i < 10000 %i的上限设置足够大，确保在这其中符合条件的N
4      Fcurr = Folder+Fold;
5      if Fold<1000&&Fcurr>=1000 %判定是否到达N，若是则跳出循环
6          break
7      end
8      Folder = Fold;
9      Fold = Fcurr; %循环计算出最新的项，并不断替换掉旧的两项
10     j = j+1;
11     array(j)=Fcurr; %将前N项放入数组array中
12 end
13 disp(array) %打印出单索引数组

命令行窗口
>> Assignment1_7_b
2 3 5 8 13 21 34 55 89 144 233 377 610 987

```

(c) 计算斐波那契数列的思路不变，增设总和 sum 和计数变量 n，判断并累加符合条件的前 40 项，达到 40 时跳出循环输出结果。

```
Assignment1_7_c.m
1 - sum = 0;n = 0; %初始化总和sum的值并设置一个计数的变量n
2 - Folder = 1;Fold = 1;
3 - for i = 3:10000 %i的上限设置足够大，确保在这其中找出40项
4 -     Fcurr = Folder+Fold;
5 -     if mod(Fcurr,2)==0||mod(Fcurr,5)==0 %判断是否符合条件，将符合的项累加
6 -         sum = sum+Fcurr
7 -         n = n+1;
8 -     end
9 -     if n==40 %计数变量n达到40时跳出循环，输出总和
10 -         break
11 -     end
12 -     Folder = Fold;
13 -     Fold = Fcurr;
14 - end
15 - disp(['Sum=',num2str(sum)]) %输出总和的值

命令行窗口
8.7694e+16

sum =

1.4900e+17

Sum=1.489999920947949e+17
```

Findings: 完成作业过程中我发现 matlab 会自动帮我在可以改进的地方下标波浪线，将光标悬停在上面可以学到很多新东西并改进我的代码。如 problem 5 的 (a) 在创建重复数组时就提示我可以改用 ones 来替代 repmat，以增加可读性。且在使用 for 循环的过程中，由于以往的习惯我设置了 i 和 j 变量，提示中建议我改换成 1i，刚开始不理解，自行搜索研究后才发现，i 和 j 是虚数单位，所以为了防止和变量混淆，应该改用 1i。综上，matlab 的提示不仅是纠正代码错误的工具，也是优化代码，学习新知识的助手。