作业 Chapter four

1. 在命令窗口中，写一个for循环，输出从32到255的每个整数编码对应的字符。

for i=32:255

disp(char(i))

end

1. 写一个beautyofmath脚本，产生如下的输出,严格按照下面的格式打印。

>>beautyofmath

18+1=9

128+2=98

1238+3=987

12348+4=9876

123458+5=98765

1234568+6=987654

12345678+7=9876543

123456788+8=98765432

1234567898+9=987654321

high=1;

low=1;

result=0;

for i=1:9

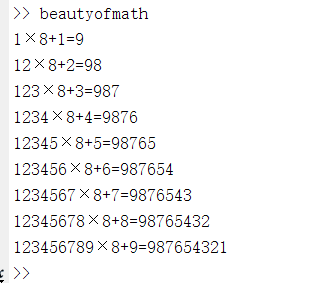
result=high\*8+low;

disp([int2str(high),'¡Á8+',int2str(low),'=',int2str(result),'\n']);

low=low+1;

high=high\*10+low;

end



1. 一个公司两个不同的部门2019年四个季度的销售额存储在向量变量中，

Div1=[4.2 3.8 3.7 3.8]

Div2=[2.5 2.7 3.1 3.3]

1. 使用subplot并排展示两个部门的销售额图形；

div1=[4.2,3.8,3.7,3.8];

div2=[2.5,2.7,3.1,3.3];

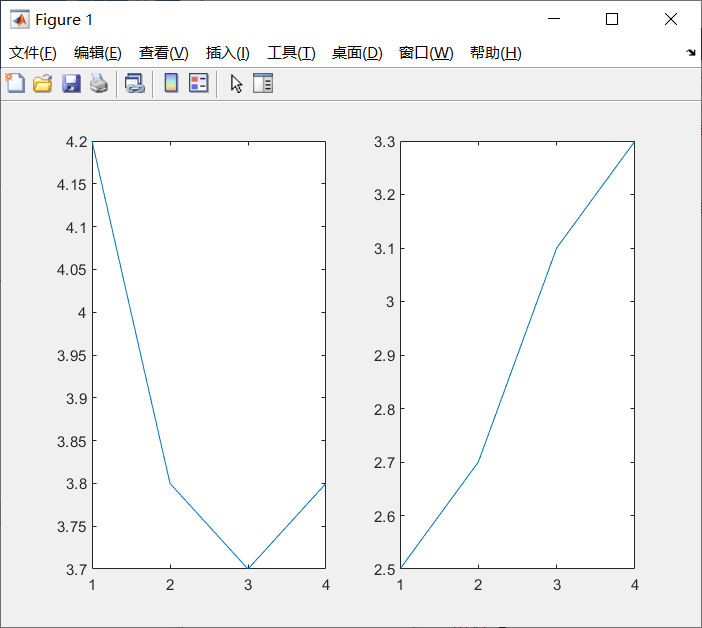
x=[1,2,3,4];

subplot(1,2,1);

plot(x,div1);

subplot(1,2,2);

plot(x,div2);



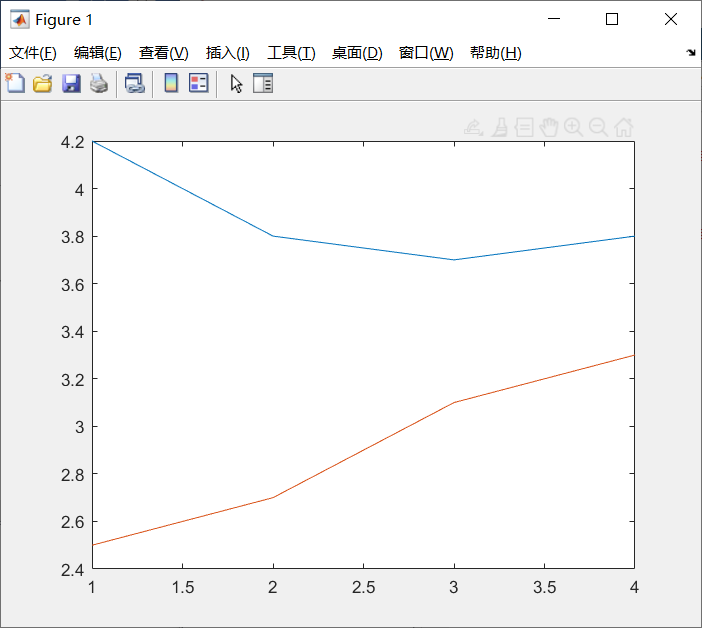
1. 在同一个图中进行比较

div1=[4.2,3.8,3.7,3.8];

div2=[2.5,2.7,3.1,3.3];

x=[1,2,3,4];

plot(x,div1,x,div2);



1. 执行下面的脚本，你有什么发现？

clear

clf

x=rand; y=rand;

plot(x,y,’r+’)

hold on

for it=1:10000

choice=round(rand\*2);

if choice==0;

x=x/2; y=y/2;

elseif choice==1

x=(x+1)/2; y=y/2;

else

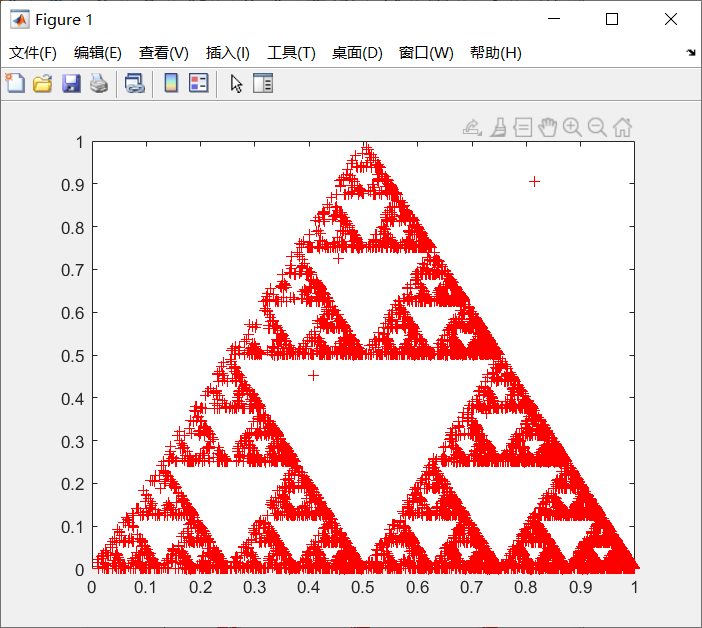
x=(x+0.5)/2; y=(y+1)/2;

end

plot(x,y,’r+’)

hold on

end



1. 数学常数e的倒数可以近似的表达为 。

写一个脚本循环计算，直至近似值和准确值的误差小于0.0001，要求脚本打印内置 的值和保留四位小数的近似值，同时打印产生此近似值所需要的n

k=0;

n=1;

ac=0.0001;

AC=1;

x=0;

while AC>ac

x=(1-1/n)^n;

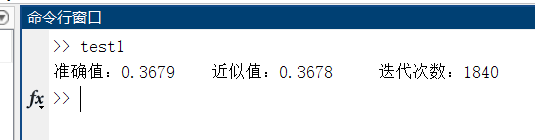
n=n+1;

AC=abs(1/exp(1)-x);

k=k+1;

end

fprintf('精确值%.4f 近似值º%.4f 迭代次数\n',1/exp(1),x,k);



1. 写一个函数计算矩阵元素的平均值，不要使用内置函数，使用循环实现。

function avg = testf(matrix)

[row,column]=size(matrix);

avg=0;

for i=1:row

for j=1:column

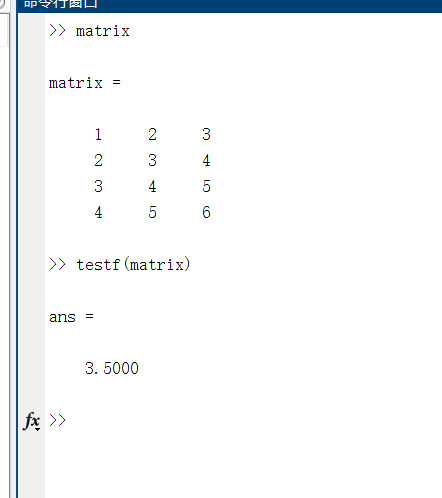
avg=avg+matrix(i,j);

end

end

avg=avg/row/column;

end



1. 向量化下面的代码
2. result=0;

for i=1:length(v)

result=result+v(i);

end

答案：

result=sum(v)

1. n=3; x=zeros(n); y=x;

for i=1:n

x(:,i)=i;

y(i,:)=I;

end

答案：

y=x’

1. 写一个“猜数字游戏的程序”。程序在特定的范围内产生一个随机整数，用户猜这个数字，程序允许玩家进行多次游戏，每次结束后，程序将会咨询玩家是否再玩一次。

基本的算法如下：

1. 程序在屏幕上打印指令开始
2. 对每次游戏

* 在给定范围内产生一个随机的正整数
* 采用循环帮助玩家猜数字直至结果正确（每次都会告诉玩家是猜低了、高了还是正确了）
* 打印猜想的正确数字
* 打印对玩家的评价信息（程序需要自己的评价方式）

disp('Gussing Number Game!');

x=input('1.Start\n2.Stop\n');

while x~=1 && x~=2

disp('Illegal input!');

x=input('1.Start\n2.Stop\n');

end

steps=0;

if x==2

disp('Bye!');

else

while(x == 1)

target=randi(100);

guess=input('Please input a number(0~100):');

while true

steps=steps+1;

if guess > target

disp('Your number is larger than the Target number!');

guess=input('Please try another number:');

elseif guess < target

disp('Your number is less than the Target number!');

guess=input('Please try another number:');

else

if steps==1

disp('Only one step you win! Amazing!')

elseif steps>1 && steps<=10

fprintf('It takes you only %d steps to win the game! Fighting!',steps);

else

fprintf('%d steps?! How low you are!',steps);

end

k=0;

break;

end

end

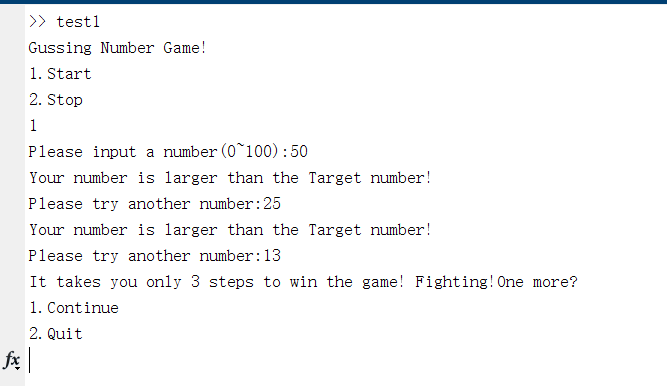
disp('One more?');

x=input('1.Continue\n2.Quit\n');

end

disp('Goodbye!');

end



1. 写一个函数演示矩阵的乘法，需要差错检测。

function matrix=testf(m1,m2);

[r1,c1]=size(m1);

[r2,c2]=size(m2);

if c1~=r2

disp('Illegal input');

return;

end

matrix=zeros([r1,c2]);

for i=1:r1

for j=1:c2

for k=1:c1

matrix(i,j)=matrix(i,j)+m1(i,k)\*m2(k,j);

end

end

end

end

