close all; clear all;

%% A simple example using MATLAB

load score\_data % input N score

N=input(' number of student: ');

score=zeros(2,N);

% input the name and score of the student evaluate the average score

for i=1:N

str1= input('student name:','s');

eval(['name',int2str(i),'=str1;']);

% if (i==1)

% name=str1;

% else

% name=char(name,str1); % Create a character array.

% end

score(1,i)=input('math score: ');

score(2,i)=input('english score: ');

avg(i)=(score(1,i)+score(2,i))/2; % avg(i) = sum(score(:,i))/2;

end

% output value

for i=1:N

eval(['str1=name',int2str(i),';']);

fprintf('the average score of %s is %3.2f \n',str1,avg(i));

end

save score\_data N score

%% prog 2.3-1 Variable stored as array

% Array (scalar, vector, matrix0 in matlab

% vectors, and matrices...

N = 5 % a scalar

v = [1 0 0] % a row vector

v = [1;2;3] % a column vector

v = v' % transpose a vector (row to column or column to row)

v = [1:.5:3] % a vector in a specified range:

v = pi\*[-4:4]/4 % [start:stepsize:end]

v = [] % empty vector

m = [1 2 3; 4 5 6] % a matrix: 1ST parameter is ROWS

% 2ND parameter is COLS

m = zeros(2,3) % a matrix of zeros

v = ones(1,3) % a matrix of ones

m = eye(3) % identity matrix

v = rand(3,1) % rand matrix (see also randn)

save matrix\_data m % save the variable m to a file named matrix\_data.mat

clear all % clear all variables currently used by MATLAB

load matrix\_data % read data from the saved file

m % display it - it is still there!

v = [1 2 3]; % access a vector element

length(v) % length of a vector

v(2:3) % vector(number)

% 2.3.5 subscripts

a=[ 1 2 3 4 5 6 7 8];

a(2:6)

a(2:2:6)

m = [1 2 3; 4 5 6;7 8 9]

m(1,3) % access a matrix element

% matrix(rownumber, columnnumber)

m(2,:) % access a matrix row (2nd row)

m(:,1) % access a matrix column (1st row)

size(m) % size of a matrix

size(m,1) % number rows

size(m,2) % number of columns

% to chane the value by finding the subscript

[i j]=find(m>=3);

disp([i j]);

pp=find(m>=3);

m(pp)=0

m1 = zeros(size(m)) % create a new matrix with size of m

who % list of variables

whos % list/size/type of variables

%% chap2.5-1 Array operations

% (A) Pointwise (element by element) Operations:

% addition of vectors/matrices and multiplication by a scalar

% are done "element by element"

a = [1 2 3 4]; % vector

2 \* a % scalar multiplication

a / 4 % scalar multiplication

b = [5 6 7 8]; % vector

a + b % pointwise vector addition

a - b % pointwise vector addition

a .^ 2 % pointise vector squaring (note .)

a .\* b % pointwise vector multiply (note .)

a ./ b % pointwise vector multiply (note .)

log( [1 2 3 4] ) % pointwise arithmetic operation

round( [1.5 2; 2.2 3.1] ) % pointwise arithmetic operation

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% (B) Vector Operations (no for loops needed)

% Built-in matlab functions operate on vectors, if a matrix is given,

% then the function operates on each column of the matrix

a = [1 4 6 3] % vector

sum(a) % sum of vector elements

mean(a) % mean of vector elements

var(a) % variance (sigma^{2})

std(a) % standard deviation (sigma)

max(a) % maximum

a = [1 2 3; 4 5 6] % matrix

mean(a) % mean of each column

max(a) % max of each column

max(max(a)) % to obtain max of matrix

max(a(:)) % another way to obtain max of matrix

xx = linspace(0, pi/2, 10)

yy = logspace(0, 2, 10)

% ddy=diff(yy);

% yy1=yy(1:end-1+ddy./2);

% figure(1);plot(yy1,ddy)

%%%%%%%%%%%%%%%%%%%%%%%%

% (C) Matrix Operations:

[1 2 3] \* [4 5 6]' % row vector 1x3 times column vector 3x1

% results in single number, also

% known as dot product or inner product

[1 2 3]' \* [4 5 6] % column vector 3x1 times row vector 1x3

% results in 3x3 matrix, also

% known as outer product

a = rand(3,2) % 3x2 matrix

b = rand(2,4) % 2x4 matrix

c = a \* b % 3x4 matrix

a = [1 2; 3 4; 5 6] % 3 x 2 matrix

b = [5 6 7]; % 3 x 1 vector

b \* a % matrix multiply

a' \* b' % matrix multiply

%%

%(D) Saving your work

save mysession % creates session.mat with all variables

save mysession a b % save only variables a and b

clear a b % clear variables a and b

clear all % clear all variables

load mysession % load session

a

b

%% Prog 2.6 format , disp statement

format long % (1) short e ; (2) bank (3) compact

x=[ 1e3 1 1e-4]

% 2.7-1 p. 58 square roots with newton's method

a = 2;

% a=input('input a number for the computation:');

x = a/2;

% display a variable

disp(['The approach to sqrt(a) for a =', num2str(a)]) % an str variable

for i = 1:6

x = (x + a / x) / 2;

disp( x )

end

disp( 'Matlab''s value: ' )

disp( sqrt(a) )

aa=[1:4] % row vector

bb=[5:8]

disp([aa' bb']) % a matrix variable of 4\*2 matrix

aa=[-pi:0.25\*pi:pi]

disp([ aa' sin(aa)'] ) % math. expression

format

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 2.7 Prog 2.7-1 Repeating with for statements

% Example: given a vector v, create a new vector with values equal to

% v if they are greater than 0, and equal to 0 if they less than or

% equal to 0.

v = [3 5 -2 5 -1 0]; % 1: FOR LOOPS

% initialize; generate zero matrix with same dimension

u = zeros( size(v) );

for i = 1:length(v)

disp([i v(i)]); % i=1 then [i v(i)]=[ 1 3 ]

if( v(i) > 0 )

u(i) = v(i);

end

end

u

v = [3 5 -2 5 -1 0] % 2: NO FOR LOOPS

u2 = zeros( size(v) ); % initialize

ind = find( v>0 ) % index into >0 elements

u2(ind) = v( ind )

%% Exercise For loop p.78 translate between Celsius and Fahrenheit

% input

% the initial value of the temperture in degree C : 20

% the final value of the temperture in degree C : 30

% the step of the temperture in degree C f: 2

% output using fprintf

% Celsius 20.00 Fahrenheit 68.00

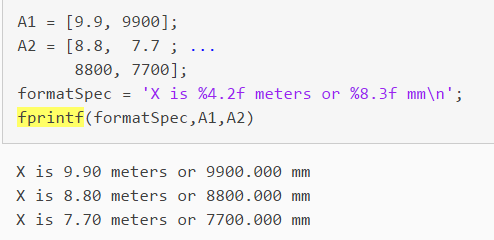
% Celsius 22.00 Fahrenheit 71.60

% Celsius 24.00 Fahrenheit 75.20

% Celsius 26.00 Fahrenheit 78.80

% Celsius 28.00 Fahrenheit 82.40

% Celsius 30.00 Fahrenheit 86.00



Some other format for the fprintf , check the table

%%=========================================================================

Avoid "for" loops by vectorizing

%%=========================================================================

t0 = clock;

s = 0;

for n = 1:100000

s = s + n;

end

etime(clock, t0)

t0 = clock;

n = 1:100000;

s = sum( n );

etime(clock, t0)

%%=========================================================================

%%=========================================================================

% pp. 62 : sum(1/n^2) for n=1:100000

tic

s = 0;

for n = 1:100000

s = s + 1/n^2;

end

toc

% n is a vector

tic

n = 1:100000;

s = sum( 1./n.^2 );

toc

%%=========================================================================

%%=========================================================================

% p.63

sign = -1;

s = 0;

for n = 1:9999

sign = -sign;

s = s + sign / n;

end

display(s);

% n is a vector

n = 1:2:9999;

s = sum( 1./n - 1./(n+1) )

%% Exercise

% input the number of the student

clear all;close all;

N=input(' number of student: ');

score=zeros(2,N);

% input the name and score of the student evaluate the average score

MAXN=10;

name=zeros(MAXN,10);

for i=1:N

name(i,:)= input('student name:','s');

score(1,i)=input('math score: ');

score(2,i)=input('english score: ');

avg(i)=(score(1,i)+score(2,i))/2; % avg(i) = sum(score(:,i))/2;

end

% output value

for i=1:N

fprintf('the average score of %s is %3.2f \n',name(i,:),avg(i));

end

%%=========================================================================

% 2.8.2 p. 66 if-lese statement

%%=========================================================================

% Relational operations p.65 table 2,4

x= (3>2)

x= (2>3)

x= (3==3)

bal = 10000 \* rand;

if bal < 5000 % relational

rate = 0.09;

else

rate = 0.12;

end

newbal = bal + rate \* bal;

disp( 'New balance after interest compounded is:' )

format bank

disp( newbal )

%%=========================================================================

% 2.8.4 p. 67 elseif statement

%%=========================================================================

bal = 15000 \* rand;

if bal < 5000

rate = 0.09;

elseif bal < 10000

rate = 0.12;

else

rate = 0.15;

end

newbal = bal + rate \* bal;

format bank

disp( 'New balance is:' )

disp( newbal )

%%=========================================================================

% multiple logical condition

%%=========================================================================

bal=7000;

rate=0;

if ((5000 < bal) & (bal< 10000)) % if 5000 < bal < 10000 (wrong)

rate = 0.12;

end

newbal = bal + rate \* bal;

format bank

disp( 'New balance is:' )

disp( newbal )

%%=========================================================================

% 2.8.9 p. 71 switsh elseif statement

%%=========================================================================

d = floor(3\*rand) + 1

switch d

case 1

disp('That''s a 1!' );

case 2

disp( 'That''s a 2!');

otherwise

disp( 'Must be 3!');

end

d = floor(10\*rand);

switch d

case {2, 4, 6, 8}

disp( 'Even' );

case {1, 3, 5, 7, 9}

disp( 'Odd' );

otherwise

disp( 'Zero' );

end

%% Exercise

% (1) score case.

% (2) Hw. 1

% (3) To write the code for the root of quadratic equation in p. 94

%%=========================================================================

% complex number p. 72

%%=========================================================================

i=sqrt(-1);

circle = exp( 2\*i\*[1:360]\*pi/360 );

figure, plot(circle)

% axis([-1 1 -2 2])

axis('equal')

axis([-2 2 -2 2])

a=3;

b=5;

a=[a b];

b=a(1);

a(1)=[]

%%=========================================================================

%%=========================================================================

a = [1+i 2+2i; 3+3i 4+4i]

a'

a.'

%%=========================================================================

%%=========================================================================

tic

k=1:40000;

s=sum(1./k.^2);

disp(sqrt(6\*s))

toc

clc;clear;close all

money=50;%%¥»ª÷

newBalance = zeros(1,12);

for k=1:12 %% ¤ë¥÷

money = money \*1.01;%%¥»ª÷¥[§Q®§

newBalance(k)=money;%%¨C¤ë¦s´Úµ²ºâ

money=money+50;%%¨C¤ë©w¦s

end

display(['Month' ' New Balance']);

display(num2str([ (1:12)' newBalance' ]));

%% Exercise answer

close all;clear all;

aa = floor(100\*rand(20,1)) + 1

for i=1:length(aa)

bb(i)='n';

if (aa (i) >= 60)

bb(i)='p';

end

disp([ ' ' num2str(aa(i)) ' ' bb(i) ]);

end