```
 1 close all; clear all;
 2
 3 %% A simple example using MATLAB
 4 load score_data % input N score
 5
 6 N=input('   number of student:   ');
 7 score=zeros(2,N);
 8
 9 % input the name and score of the student evaluate the average score
10 for i=1:N
11     str1= input('student name:','s');
12     eval(['name',int2str(i),'=str1;']);
13 %     if (i==1)
14 %         name=str1;
15 %     else
16 %         name=char(name,str1); % Create a character array.
17 %     end
18 score(1,i)=input('math score:   ');
19 score(2,i)=input('english score:   ');
20 avg(i)=(score(1,i)+score(2,i))/2;% avg(i) = sum(score(:,i))/2;
21 end
22
23 % output value
24 for i=1:N
25     eval(['str1=name',int2str(i),';']);
26  fprintf('the average score of %s is %3.2f \n',str1,avg(i));
27 end
28 save score_data N score
29
30 %% prog 2.3-1   Variable stored as array
31 % Array (scalar, vector, matrix0 in matlab
32 % vectors, and matrices...
33
34
35 N   = 5            % a scalar
36 v   = [1 0 0]          % a row vector
37 v   = [1;2;3]          % a column vector
38 v   = v'               % transpose a vector (row to column or column to row)
39 v   = [1:.5:3]         % a vector in a specified range:
40 v   = pi*[-4:4]/4          %   [start:stepsize:end]
41 v   = []               % empty vector
42
43
44 m   = [1 2 3; 4 5 6]       % a matrix: 1ST parameter is ROWS
45                     %       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);
```

```
136 % yy1=yy(1:end-1+ddy./2);
137 % figure(1);plot(yy1,ddy)
138
139 %%%%%%%%%%%%%%%%%%%%%%%%%
140 % (C) Matrix Operations:
141
142 [1 2 3] * [4 5 6]'             % row vector 1x3 times column vector 3x1
143                                 % results in single number, also
144                               % known as dot product or inner product
145
146 [1 2 3]' * [4 5 6]             % column vector 3x1 times row vector 1x3
147                                 % results in 3x3 matrix, also
148                                 % known as outer product
149
150 a   = rand(3,2)          % 3x2 matrix
151 b   = rand(2,4)          % 2x4 matrix
152 c   = a * b              % 3x4 matrix
153
154 a   = [1 2; 3 4; 5 6]       % 3 x 2 matrix
155 b   = [5 6 7];           % 3 x 1 vector
156 b * a                    % matrix multiply
157 a' * b'                  % matrix multiply
158
159 %%
160 %(D) Saving your work
161
162 save mysession                 % creates session.mat with all variables
163 save mysession a b              % save only variables a and b
164
165
166 clear a b                      % clear variables a and b
167 clear all               % clear all variables
168
169 load mysession              % load session
170 a
171 b
172
173
174 %%  Prog 2.6 format , disp statement
175
176 format long % (1) short e ; (2) bank (3) compact
177 x=[ 1e3 1 1e-4]
178
179 %   2.7-1  p. 58 square roots with newton's method
180 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
```

```
226
227 % input
228 %   the initial value of the temperture in degree C : 20
229 %   the final value of the temperture in degree C : 30
230 %   the step of the temperture in degree C f: 2
231
232 % output using fprintf
233 %    Celsius 20.00 Fahrenheit 68.00
234 %    Celsius 22.00 Fahrenheit 71.60
235 %    Celsius 24.00 Fahrenheit 75.20
236 %    Celsius 26.00 Fahrenheit 78.80
237 %    Celsius 28.00 Fahrenheit 82.40
238 %    Celsius 30.00 Fahrenheit 86.00
239
240
241 %%=========================================================================
242 Avoid "for" loops by vectorizing
243 %%=========================================================================
244
245 t0 = clock;
246 s = 0;
247 for n = 1:100000
248 s = s + n;
249 end
250 etime(clock, t0)
251
252 t0 = clock;
253 n = 1:100000;
254 s = sum( n );
255 etime(clock, t0)
256
257 %%=========================================================================
258 %%=========================================================================
259 % pp. 62   :  sum(1/n^2) for n=1:100000
260 tic
261 s = 0;
262 for n = 1:100000
263 s = s + 1/n^2;
264 end
265 toc
266
267 % n is a vector
268 tic
269 n = 1:100000;
270 s = sum( 1./n.^2 );
```

```
271 toc
272
273
274 %%=====================================================================
275 %%=====================================================================
276
277 % p.63
278 sign = -1;
279 s = 0;
280 for n = 1:9999
281 sign = -sign;
282 s = s + sign / n;
283 end
284 display(s);
285
286 % n is a vector
287 n = 1:2:9999;
288 s = sum( 1./n - 1./(n+1) )
289
290 %% Exercise
291
292  % input the number of the student
293 clear all;close all;
294 N=input('   number of student:   ');
295 score=zeros(2,N);
296 % input the name and score of the student evaluate the average score
297 MAXN=10;
298 name=zeros(MAXN,10);
299 for i=1:N
300 name(i,:)= input('student name:','s');
301 score(1,i)=input('math score:   ');
302 score(2,i)=input('english score:   ');
303 avg(i)=(score(1,i)+score(2,i))/2;% avg(i) = sum(score(:,i))/2;
304 end
305 % output value
306 for i=1:N
307  fprintf('the average score of %s is %3.2f \n',name(i,:),avg(i));
308 end
309
310
311 %%=====================================================================
312 % 2.8.2 p. 66 if-lese statement
313 %%=====================================================================
314 % Relational operations p.65 table 2,4
315 x= (3>2)
```

```
316 x= (2>3)
317 x= (3==3)
318
319 bal = 10000 * rand;
320 if bal < 5000  % relational
321     rate = 0.09;
322 else
323     rate = 0.12;
324 end
325 newbal = bal + rate * bal;
326 disp( 'New balance after interest compounded is:')
327 format bank
328 disp( newbal )
329
330
331 %%=========================================================================
332 % 2.8.4 p. 67 elseif  statement
333 %%=========================================================================
334
335
336 bal = 15000 * rand;
337 if bal < 5000
338     rate = 0.09;
339 elseif bal < 10000
340     rate = 0.12;
341 else
342     rate = 0.15;
343 end
344 newbal = bal + rate * bal;
345 format bank
346 disp( 'New balance is:' )
347 disp( newbal )
348
349
350
351
352 %%=========================================================================
353 % multiple logical condition
354 %%=========================================================================
355 bal=7000;
356 rate=0;
357 if ((5000 < bal) & (bal< 10000))% if 5000 < bal < 10000 (wrong)
358 rate = 0.12;
359 end
360 newbal = bal + rate * bal;
```

```
361 format bank
362 disp( 'New balance is:' )
363 disp( newbal )
364
365
366
367 %%=====================================================================
368 % 2.8.9 p. 71 switsh elseif  statement
369 %%=====================================================================
370
371 d = floor(3*rand) + 1
372 switch d
373 case 1
374 disp('That''s a 1!' );
375 case 2
376 disp( 'That''s a 2!');
377 otherwise
378 disp( 'Must be 3!');
379 end
380
381
382
383 d = floor(10*rand);
384 switch d
385 case {2, 4, 6, 8}
386 disp( 'Even' );
387 case {1, 3, 5, 7, 9}
388 disp( 'Odd' );
389 otherwise
390 disp( 'Zero' );
391 end
392
393 %% Exercise
394 % (1) score case.
395 % (2) Hw. 1
396 % (3) To write the code for the root of quadratic equation in p. 94
397 %%=====================================================================
398 % complex number p. 72
399 %%=====================================================================
400
401 i=sqrt(-1);
402 circle = exp( 2*i*[1:360]*pi/360 );
403 figure, plot(circle)
404 % axis([-1 1 -2 2])
405 axis('equal')
```

```
406 axis([-2 2 -2 2])
407
408 a=3;
409 b=5;
410 a=[a b];
411 b=a(1);
412 a(1)=[]
413 %%=========================================================================
414 %%=========================================================================
415
416 a = [1+i 2+2i; 3+3i 4+4i]
417 a'
418 a.'
419
420
421 %%=========================================================================
422 %%=========================================================================
423 tic
424 k=1:40000;
425 s=sum(1./k.^2);
426 disp(sqrt(6*s))
427 toc
428
429
430 clc;clear;close all
431
432 money=50;%%本金
433 newBalance = zeros(1,12);
434
435 for k=1:12 %% 月份
436    money = money *1.01;%%本金加利息
437     newBalance(k)=money;%%每月存款結算
438     money=money+50;%%每月定存
439 end
440
441 display(['Month' ' New Balance']);
442 display(num2str([ (1:12)' newBalance' ]));
443
444
445 %% Exercise answer
446 close all;clear all;
447
448 aa = floor(100*rand(20,1)) + 1
449 for i=1:length(aa)
450     bb(i)='n';
```

```
451     if (aa (i) >= 60)
452         bb(i)='p';
453     end
454     disp([ '       ' num2str(aa(i)) '    ' bb(i) ]);
455 end
456
457
458
```