Ex. 7 Arrays

a = [3 6 7];

b = [1 9 4];

c = a + b

Output:

4 15 11

Remarks: (1) Both a and b are given as a three-element array. In the third line, the

operation of "a+b" results in element-by-element addition

Ex. 8 Extracting an individual element of an array

a = [3 6 7];

b = [1 9 4 5];

c = a(2) + b(4)

Output:

c = 11

Remark: If b is a one-dimensional array, b(n) is the n-th element of that array. Since

a(2) is 6 and b(4) is 5, the 3rd statement leads to c = 6+5 = 11.

Ex. 9 Comment

%

% This program demonstrates how to "comment out"

% a segment of code

%

A = 3;

B = A\*A;

%

% B = 2\*B <--- This statement is not executed

%

C = A+B

Output:

c = 12  
---------------------

Ex. 11 "Clear" a variable

c1 = 3;

c2 = c1+5;

clear c1

c1

Output:

??? Undefined function or variable 'c1'.

Ex. 14 Loop: Using for command

b = 3;

for k = 1:5

b

end

Output:

3

3

3

3

3

Remark: The blue-colored segment in lines 2-4 forms a "for-loop". The statement

sandwiched between "for k = 1:5" and "end" is repeated 5 times, with the "k" index

going from 1 to 5 step 1.

Ex. 15 For loop: Utility of the dummy index

b = 3;

for k = 1:5

b^k

end

Output:

3

9

27

81

243

Remark: The outputs are 3^1, 3^2, 3^3, 3^4, and 3^5. the value of "k" keeps changing as

we go through the loop  
---------

Ex. 16 For loop: More on the dummy index

sum1 = 0;

for k = 1:9

sum1 = sum1+k;

end

sum1

Output:

45

Remark: this program performs the summation of 1+2+3+4+5+6+7+8+9 (= 45).

Ex. 17 For loop: More on the dummy index

sum1 = 0;

for k = 1:2:9

sum1 = sum1+k;

end

sum1

Output:

25

Remark: this program performs the summation of 1+3+5+7+9 (= 25). The command

"for k = 1:2:9" means we go through the loop only 5 times. First time with k = 1, second

time with k = 1+2 (=3), third time with k = 1+2+2 (=5), and so on. The looping stops

once k reaches 9.

9

Ex. 18 Treatment of array within a loop

b = [3 8 9 4 7 5];

sum1 = 0;

for k = 1:4

sum1 = sum1+b(k);

end

sum1

Output:

24

Remark: This program performs the summation of sum1 = b(1)+b(2)+b(3)+b(4) =

3+8+9+4 = 24

Ex. 19 Treatment of array within a loop

b = [3 8 9 4 7 5];

sum1 = 0;

for k = 1:2:5

sum1 = sum1+b(k);

end

sum1

Output:

19

Remark: This program performs the summation of sum1 = b(1)+b(3)+b(5) = 3+9+7 = 19

10

Ex. 20 Double loop

sum1 = 0;

for n = 1:2

for m = 1:3

sum1 = sum1+n\*m;

end

end

sum1

Output:

18

Remark: this program performs the summation of

Sum1 = 1\*1+1\*2+1\*3 +2\*1+2\*2+2\*3 = 18  
------

Ex. 22 More complicated use of loop and index

b = [2 5 7 4 9 8 3];

c = [2 3 5 7];

sum1 = 0;

for k = 1:4

sum1 = sum1+b(c(k));

end

sum1

Output:

24

Remark: This program performs the summation of

sum1 = b(c(1))+b(c(2))+b(c(3))+b(c(4))

= b(2)+b(3)+b(5)+b(7)

= 5+7+9+3

= 24  
---

Ex. 23 The if command

num1 = 7;

if (num1 > 5)

fprintf('%4u is greater than 5 \r', num1)

else

fprintf('%4u is less than or equal to 5 \r', num1)

end

Output:

7 is greater than 5

Same program, but change first line to "num1 = 3;"

Output:

3 is less than or equal to 5

Remark: In this program, if (num1 > 5) (num1 is greater than 5) is true, the statement

"fprintf('%4u is greater than 5 \r', num1)" is executed. Otherwise, the statement

"fprintf('%4u is less than or equal to 5 \r', num1)" is executed.

Ex 25 An application - determine whether a given year is a leap year (try to change the

given value of nyear and observe the outcome)

nyear = 1975;

if (mod(nyear, 400) == 0)

fprintf('%6u is a leap year', nyear)

elseif (mod(nyear,4) == 0) & (mod(nyear,100) ~= 0)

fprintf('%6u is a leap year', nyear)

else

fprintf('%6u is not a leap year', nyear)

end

Output:

1975 is not a leap year

Remarks:

(1) In the elseif command (4th line), "&" means "AND". Both statements

"(mod(nyaer,4) == 0)" and "(mod(nyear,100) ~= 0)" have to be true for Matlab to

execute the command, "fprintf('%6u is a leap year', nyear)". Also commonly used in an

if statement is "|" (a vertical line), which means "OR".

(2) The symbols "~=" in line 4 means "NOT EQUAL TO". There are 6 commonly used

expressions to compare two numbers in an if command:

A > B A is greater than B

A < B A is less than B

A >= B A is greater than or equal to B

A <= B A is less than or equal to B

A == B A equals B

A ~= B A does not equal B

(3) The "mod(A,B)" function returns the remainder of A divided by B. For example,

mod(7,2) = 1, mod(10,4) = 2, and mod(25,5) = 0. If A is divisible by B, mod(A,B) = 0.

This is a very useful function in many applications related to numerical methods.

Ex. 41 A quick example of plot command: Draw a curve

a = [0:0.5:5];

b = 2\*a.^2 + 3\*a -5;

plot(a,b)

--

a = [0:0.5:5];

b = 2\*a.^2 + 3\*a -5;

plot(a,b,'-or','MarkerFaceColor','g','LineWidth',2)

xlabel('X'); ylabel('Y'); legend('Test','Location','NorthWest')

--