Numbers: 1 2 3 4

Logicals: 1<3 1==1

Assignment Statements: x = 1

If Statements

Relational Operators

== ~= < > <= >=

[2 3 4] <= 2 🡪 [1 0 0]

[2 3 4] \* 2 = [4 6 8]

[2 3 4] \* [2; 3; 4] = 29

2 \* 2 = 4

2 \* [2 3 4] = [4 6 8]

Logical Operators && ||

Plots

b blue . point - solid

g green o circle : dotted

r red x x-mark -. dashdot

c cyan + plus -- dashed

m magenta \* star (none) no line

y yellow s square

k black d diamond

w white v triangle (down)

^ triangle (up)

< triangle (left)

> triangle (right)

p pentagram

h hexagram

P’ – invert rows and coloumns of vector P

Sequences

(Colon Notation) a:n:b = from a to b increments of n

Vector

Vector indexing using scalar, vector, logicals

Ans = [0 2 4 6 8]

X = [1 4]

Ans(1:3) = [0 2 4]

Ans(Ans<4) = [0 2]

Ans(X) = [0 6]

Strings

double('abcdef')

ans = 97 98 99 100 101 102

char([72 101 108 108 111])

ans =Hello

char('abcdefg' - 32)

ans =ABCDEFG

double('amnzAMNZ')

ans = 97 109 110 122 65 77 78 90

ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^\_`abcdefghijklmnopqrstuvwxyz

For loops

While loops

User defined functions

Global variables

global G

G = 21039

Built-In Functions

Sum

sum(X) sum of elements

S = sum(X,DIM) Dimensions

Linspace

linspace(0,7.5,4)

ans = 0 2.5000 5.0000 7.5000

Rand(2,3)

ans =

0.7577 0.3922 0.1712

0.7431 0.6555 0.7060

Sqrt

Randperm

randperm(6) might be [2 4 5 6 1 3]

Strcmp

Compares two strings, returns 1 if identical 0 if not.

Length

It is equivalent to MAX(SIZE(X)) for non-empty arrays and 0 for empty ones.

Size

D = size(X), for M-by-N matrix X, returns the two-element row vector

D = [M,N] containing the number of rows and columns in the matrix.

Find

[x j k] = find(ans)

x = column number

j = row number

k = value of position

Normally outputs the indicies

corresponding to X in

I = find(X,K,'first'/‘last’)

K = 8 3 4 1 5 9 6 7 2

find(K<5)

ans = 2 3 4 9

find(k<5,3)

ans = 2 3 4

find(k<5,3,'last')

ans = 3 4 9

Disp

disp(X) displays the array, without printing the array name

Zeros

zeros(M,N) or zeros([M,N]) is an M-by-N matrix of zeros.

Ones

See zeoros

Double

double(X) returns the double precision value for X.

Char

S = char(X) converts the array X that contains nonnegative integers

representing character codes into a MATLAB character array (the first

127 codes are ASCII).

Class

See classes

Num2str

Converts [2 3 4] to [‘2’ ‘3’ ‘4’]

Str2num

Opposite of num2str

Floor

Closest to -inf

Ceil

Closest to +inf

Round

Round to int

Fix

Round to 0

Rng

rng(SD) seeds the random number generator using the non-negative

integer SD so that RAND, RANDI, and RANDN produce a predictable

sequence of numbers.

rng(‘default’) sets the rng back to start of

matlab

No i,j,k no sin, cos, tan

Classes:

double -- Double precision floating point number

(this is the traditional MATLAB matrix or

single -- Single precision floating point

logical -- Logical array

char -- Character array

cell -- Cell array

struct -- Structure array

function\_handle -- Function Handle

int8 -- 8-bit signed integer array

uint8 -- 8-bit unsigned integer array

int16 -- 16-bit signed integer array

uint16 -- 16-bit unsigned integer array

int32 -- 32-bit signed integer array

uint32 -- 32-bit unsigned integer array

int64 -- 64-bit signed integer array

uint64 -- 64-bit unsigned integer array

<class\_name> -- MATLAB class name for

MATLAB objects

<java\_class> -- Java class name for java objects

Format + (plus and minus signs only)

Bank – 2 dec

Compact - compact

Long – 14 dec

Long e – 14 dec exp

Loose – default

Short – 4 dec

Short e -4 exp

abs - absolute value

cumprod – product of elements in vector

cumsum

log

log10

log2

mean

median

min

prod

rand

randn (Gaussian. Normally distributed)

rem

sign

sort (ascending order)

std (standard deviation)

pause (until key is hit)

peaks (sample matrix to demonstrate graphing functions

pie (pie chart)

pie3 (3d)

plot

plot3

polar

semilogx

semilogy

xlabel

ylabel

clock

else

elseif

end

find

for

if

tic

toc (displays elapsed time after toc)

while

***2nd Part***

fid = fopen(‘filename.txt’)

fgetl(fid) (gets each line of the file, outputs a string for each time called with each line then the value -1)

Example:

In file:

223

344

44

>> fgetl(fid)

Ans =

223

Ans =

344

Ans =

44

Ans =

-1

Common loops:

function readLines(filename)

fid = fopen(filename);

while true

lineText = fgetl(fid);

if lineText == -1

break

end

disp(lineText)

end

fclose(fid);

end

Can also be:

if feof == 1

Break

end

fprintf(fid,'%i\n',item\_to\_be\_printed);

function saveArray(array,filename)

fid = fopen(filename,'w');

[r c] = size(array);

fprintf(fid,'%i\n',r);

fprintf(fid,'%i\n',c);

for i=1:r

for j=1:c

fprintf(fid,'%i\n',array(i,j));

end

end

fclose(fid);

end

function array = loadArray(file)

fid = fopen(file);

for i=1:2

dimensions(i) = fgetl(fid);

end

r = dimensions(1) - '0';

c = dimensions(2) - '0';

array = zeros(r,c);

for j=1:r

for k=1:c

array(j,k) = fgetl(fid) - '0';

end

end

fclose('all');

end

function vector = countLetters(string)

vector = zeros(1,26);

for i=1:length(string)

if double(string(i)) >= 65 && double(string(i)) <= 90

string(i) = string(i) + 32;

count(i) = string(i) - 96;

elseif double(string(i)) >= 97 && double(string(i)) <= 122

count(i) = string(i) - 96;

end

end

count = count(find(count));

for j=1:length(count)

vector(count(j)) = vector(count(j)) + 1;

end

end

Relational operators < <= > >= == ~= ~

And, or (&&, ||)

find

if, if/else, if/elseif/else

switch statement

for loops

while loops

break statement, but not continue

>> a = {1 [1 2] 'hello'}

a =

[1] [1x2 double] 'hello'

>> class(a{1})

ans =

double

>> class(a{3})

ans =

char

>> char(65:122)

ans =

ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^\_`abcdefghijklmnopqrstuvwxyz

Get a character:

c = input('Enter a character: ', 's');

c = c(1);

function readLines(filename)

fid = fopen(filename);

while true

lineText = fgetl(fid);

if lineText == -1

break

end

disp(lineText)

end

fclose(fid);

end

function writeLoop(filename)

fid = fopen(filename,'w');

while true

lineText = input('Enter a line of text: ','s');

if isempty(lineText) == 1

break

end

fprintf(fid,'%s\n',lineText);

end

fclose(fid);

end