CHE/CEN138

COMPUTER PROGRAMMING

INTERACTIVE COMPUTING, FLOWCHARTING

References

1.Pratap, R. “Getting Started with MATLAB: A Quick Introduction for Scientists and Engineers”Oxford University Press, 2010.

2.Hunt, B.R., Lipsman, L.R. and Rosemberg J. M. “A guide to MATLAB for Beginners and ExperiencedUsers"Cambridge University Press, 2001.

3.Kubat, C. “MATLAB Yapay Zeka ve Mühendislik Uygulamaları” İkinci Baskı, Pusula Yayıncılık, 2014McGraw Hill, International Edition 2012.

MATRICES AND VECTORS

$$\text{Matrix} \begin{bmatrix} 2 & 5 & 7 & 8 \\ 5 & 6 & 8 & 3 \\ 1 & 6 & 4 & 0 \end{bmatrix}$$

MATLAB input command

```
>> A = [2 5 7 8; 5 6 8 3; 1 6 4 0]
```

MATRICES AND VECTORS

- Matrix

$$\begin{bmatrix} 2x & \ln x + \sin y \\ 5x & 3 + 2i \end{bmatrix}$$

- MATLAB input command

```
>> B= [2*x  log(x)+sin(y); 5*x  3+2i]
```

Special cases: vectors and scalars

>> u = [1 3 9] produces a row vector

>> v = [1; 3; 9] produces a column vector

>> X = [ ] produces a null Matrix

INDEXING

```
» A =[1 2 3; 4 5 6; 7 8 8]

A =

     1     2     3
     4     5     6
     7     8     8

» A(2, 3)

ans =

     6

» A(3, 3) = 9

A =

     1     2     3
     4     5     6
     7     8     9
```

INDEXING

```
» B = A(2:3, 1:3)

B =

    4    5    6
    7    8    9

» B = A(2:3, :)

B =

    4    5    6
    7    8    9

» B(:, 2) = []

B =

    4    6
    7    9
```

DIMENSIONING

B(2, 3) = 5; produces

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

C(3, 1:3) = [1 2 3]; produces

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 2 & 3 \end{bmatrix}$$

MATRIX MANIPULATION

- Transpose of matrix A
  - Obtained by typing A’

```
» A=[2 3; 6 7]

A =

     2     3
     6     7

» B = A'

B =

     2     6
     3     7
```

MATRIX MANIPULATION

- Transpose example

```
» u = [0 1 2 3 4 5 6 7 8 9];
» v = u(3:6)'

v =

     2
     3
     4
     5
```

DELETING A ROW OR COLUMN

- Any row(s) or column(s) of a matrix can be de- leted by setting the row or column to null vector
- A(2, :) = [ ] deletes the 2nd row of matrix A
- A(:, 3:5) = [ ] deletes the 3rd through 5th columns

UTILITY MATRICES

```
» eye(3)

ans =

     1     0     0
     0     1     0
     0     0     1

» B = [ones(3) zeros(3, 2); zeros(2, 3) 4*eye(2)]

B =  1     1     1     0     0
     1     1     1     0     0
     1     1     1     0     0
     0     0     0     4     0
     0     0     0     0     4
```

```
» diag(B)'

ans =

     1     1     1     4     4

» diag(B, 1)'

ans =

     1     1     0     0
```

UTİLİTY MATRİCES

```
» d = [2 4 6 8];
» d1 = [-3 -3 -3];
» d2 = [-1 -1];
» D = diag(d) + diag(d1, 1) + diag(d2, -2)

D =

     2    -3     0     0
     0     4    -3     0
    -1     0     6    -3
     0    -1     0     8
```

CREATING VECTORS

- v = InitialValue:Increment:FinalValue
  (If no increment is specified, increment is 1)

a = 0:10:100 produces a = [0 10 20 ... 100],

u = 2:10 produces a = [2 3 4 ... 10]

CREATING VECTORS

- linspace(a, b, n) generates a linearly spaced vector of length n from a to b.
- logspace(a, b, n) generates a logarithmically spaced vector of length n from 10^a to 10^b.

```
» u = linspace(0, 20, 5)

 u =

     0     5    10    15    20

» v = logspace(0, 3, 4)

v =

     1        10       100      1000
```

MATRIXAND ARRAY OPERATİONS

- Element-by-element multiplication, division and exponentiation between two matrices or vectors of the same size are done by preceding the corresponding arithmetic operators by a period (.):

.*	element-by-element multiplication
./	element-by-element left-division
.\	element-by-element right-division
.^	element-by-element exponentiation
.’	nonconjugated transpose

MATRIX AND ARRAY OPERATIONS

```
» A=[1 2 3; 4 5 6; 7 8 9];
» x = A(1, :)'

x =

     1
     2
     3

» x'*x  ans =

    14

» x*x'  ans =

     1     2     3
     2     4     6
     3     6     9

» A*x  ans =

    14
    32
    50
```

MATRIX AND ARRAY OPERATIONS

```
» v = [ 1 2 3 4 5 6]

v =

     1     2     3     4     5     6

» 1./v

ans =

    1.0000    0.5000    0.3333    0.2500    0.2000
    0.1667
```

MATRIX AND ARRAY OPERATIONS

- There is a big difference between A^2 and A.^2

```
» A = [1 2 3; 4 5 6; 7 8 9]

A =

     1     2     3
     4     5     6
     7     8     9

» A^2

ans =

    30    36    42
    66    81    96
   102   126   150

» A.^2

ans =

     1     4     9
    16    25    36
    49    64    81
```

RELATİONAL OPERATORS

- MATLAB supports six relational operators.

Less Than	<
Less Than or Equal	<=
Greater Than	>
Greater Than or Equal	>=
Equal To	==
Not Equal To	~=

LOGICAL OPERATORS

- MATLAB supports three logical operators.

not	**~**	highest precedence
and	**&**	equal precedence with **or**
or	**|**	equal precedence with **and**

a = b & c > 4 | d = 4 not the same as ...

a = b & (c > 4 | d = 4)

FLOWCHARTS

- Useful in program development
- Prepared **before** writing the program.
- Only executable statements are shown

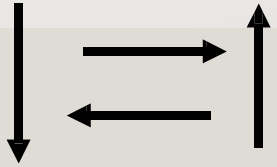
If you can flowchart it, you can program it.

OPERATION TYPES

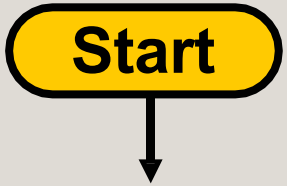
- Begin / end
- Read input, Write output
- Carry out a computation
- Decision:
  - Select one of two paths
  - Select one of multiple paths
- Proceed to the next operation

FLOWCHART SYMBOLS

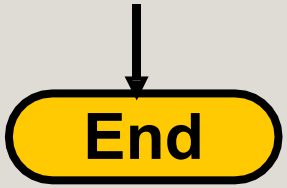
Path to follow



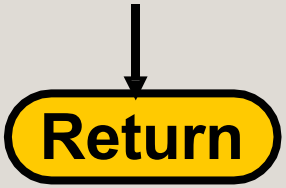
Start here



End here



Return to caller

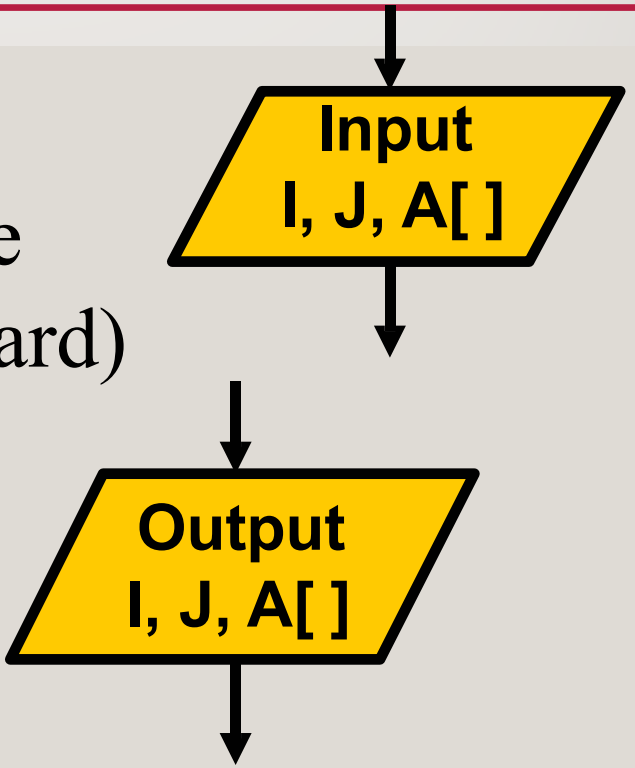


Note: One line in or out.

FLOWCHART SYMBOLS

Read input from a file
(disk/tape file, keyboard)

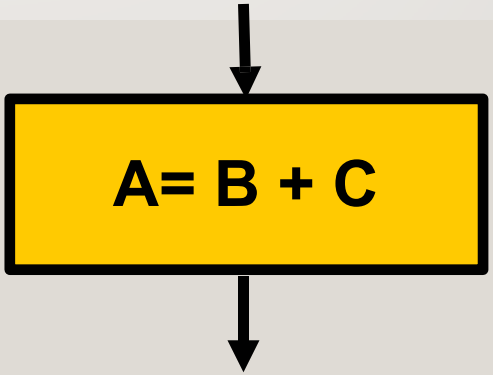
Write output to file
(disk/tape file, printer)



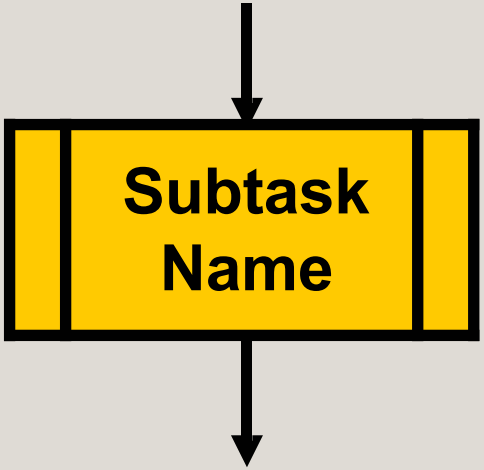
Note: One line in, one line out.

FLOWCHART SYMBOLS

Carry out a computation



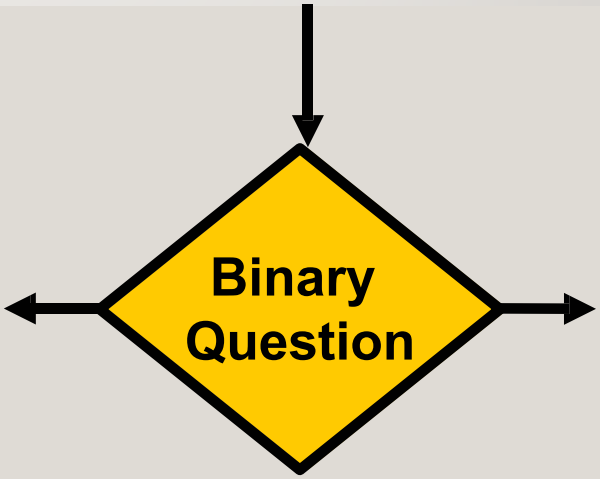
Carry out the specified subtask, then come back.



Note: One line in, one line out.

FLOWCHART SYMBOLS

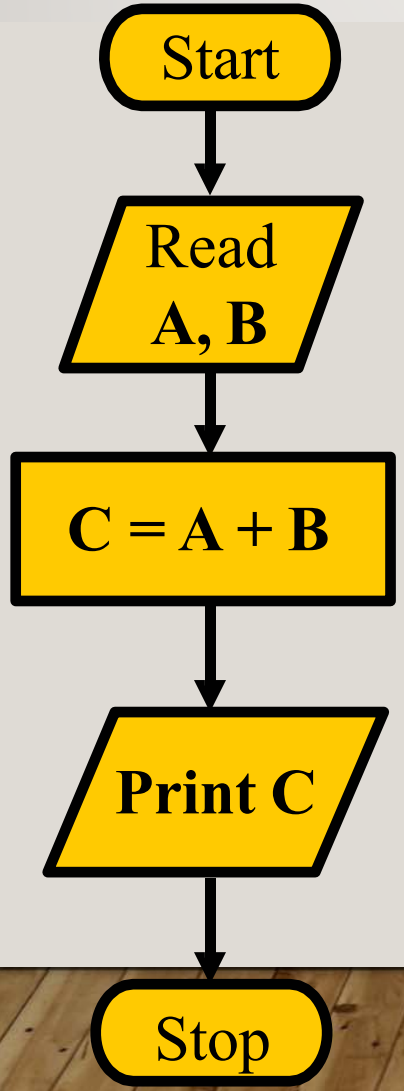
Decision Box: Select one of two paths to follow depending upon a condition being met



Must contain a binary question which can be answered by Yes/No, True/False, 0/1, etc.

Note:
One line in, two lines out.

CONNECTING FLOWCHART SYMBOLS



This chart gives a typical example of Input, Process, Output

Input two numbers,

compute a new value,

Output new value