

Basic

UNIT-2

(F) Plots : Based on scenario we have to choose appropriate plot.

UNIT-3

Functions : User defined code New top left save as .m
function name should be same as file name

Creating Input : use `input()` eg : `input("Enter: ")`

> if char or str input `input("Enter str: ", 's')`

> if `input("Enter number: ")` given i/p as 'k'
then it shows error since give arg 's' in `input()`.

> if `t = 11` is initialized Enter the number : t won't
show error since 11 is assigned to t

Output stmts : \Rightarrow `disp()` & `fprintf()`

`disp()` \Rightarrow display as it is no need for format specifiers

`fprintf()` \Rightarrow requires format specifiers

eg : `disp(6^2)`
o/p 36

eg : `fprintf('The value is %d', 4^3)`
o/p : The value is 64

Format specifiers : (integer \rightarrow %d , float \rightarrow %f , char \rightarrow %c
str \rightarrow %s)

vector in fprintf : `vec = 2:5`
`fprintf(' %d \n', vec)`

o/p :
2
3
4
5

Sample Question functions

> Factorial of a Given No

> Fibonacci Series

> Prime Numbers

1) > Ask date from the user give the day Given
start of month eg. Thursday (Then case 1: Thu)

Sln:

1) check valid F/P by logical AND date ≤ 31 .

date ≥ 1

2) Entered date > 7 . Take modulo else enter the

switch case \rightarrow case 1 : Thu 4 : Sun 0 : Wed

2 : Frid 5 : Mon

3 : Sat 6 : Tue

Code :

```
date = input('Enter the date no: '); if (date  $\geq 1$  & date  $\leq 31$ )  
n = mod(date, 7); disp('valid fp');
```

```
Switch n % Month start day is Monday %
```

```
case 1
```

```
disp('Monday')
```

```
case 2
```

```
disp('Tuesday')
```

```
case 3
```

```
disp('Wed')
```

```
case 4
```

```
disp('Thu')
```

```
case 5
```

```
disp('Fri')
```

```
case 6
```

```
disp('Sat')
```

```
case 0
```

```
disp('Sun')
```

```
otherwise
```

```
disp('invalid')
```

```
end
```

2) Demonstrate Arithmetic operator using switch case

a = input('Enter a value:')

b = input('Enter b value:')

op = input('Enter the arithmetic operator:')

switch op

case '+'

disp(a+b)

case '-'

disp(a-b)

otherwise

disp('Invalid')

end

switch op

case 'sub'

disp(a-b)

case 'add'

disp(a+b)

otherwise

disp('Invalid')

end

for

str =

'sub'

> Duplicates removal > Frequency count

C++-II

function & Module

> Plotting

> Sorting → Bubble sort

> AUF

searching - Linear, Binary

> Problem solving → sorting, Matrix i/p, nested loop, function

roots of a quadratic eqn

[function based on no. of args & return values]

function [x1, x2] = quadratic2(a, b, c)

global d;

function disc

d = sqrt(b^2 - 4*a*c);

end

disc;

x1 = (-b+d) / 2*a;

x2 = (-b-d) / 2*a;

end

Recursion func :

Factorial:

```
function res = factr(n)
if (n == 0)
    result = 1;
else
    result = n * factr(n-1);
end
end
```

Binary search :

- > need sorted array
- > Find low (=1), middle, high (length(arr))
$$\downarrow$$
$$\frac{\text{high} + \text{low}}{2}$$
- > get a search ele as input
- > Make 3 comparisons search > mid < mid = mid

function bsearch(a, n)

```
low = 1;
high = length(a); flag = 0;
while (low <= high)
    mid = floor((low + high) / 2);
    if (a[mid] == n)
        disp("element found");
        flag = 1; break;
    elseif (n < a[mid])
        high = mid - 1;
    else
        low = mid + 1;
    end
end
if flag == 0
    disp("element not found");
end
end
```

Add 1 to 100 even nos :

```
function res = sumeven(n);
if (n == 0)
    res = 0;
else
    res = n + sumeven(n-2);
end
end
```

only even
if n is odd
if odd add sum
condition

if (mod(n,2) == 0) // input
n is odd
n = n-1;
end

Duplicates Removal :

function dupremove(a)

b(1) = 1
b(2) = 2
b(3) = 3
b(4) = 4
b(5) = 5

a = [1 2 2 3 3 5 5]

b = [1 2 3 5]

c = [1 2 3 5]

	b	c
1	1	5
2	2	4
3	3	5

Statistics functions for vectors & Matrices :

vector	matrix
max	max2
mean	mean2
median	median2
min	"
mode	"
std	"
var	"

First order statistics

↳ fetching info directly from the matrix

Write a code for scaling down the image ☒ use if else

Second order statistics : GLCM

↳ Convert a matrix into another then performed using `graycomatrix()` ⇒ Gray Level Co-occurrence Matrix (GLCM)

eg : `[glcm, SI] = graycomatrix(I, 'NumLevels', 9, 'GrayLimits', [1 255])`

to extract properties or features use `stats = graycoprops(glcm, 'properties', 'all')`
 if empty values all grayscale values

O/p : stats = struct with fields :

Contrast 0.3110
Correlation 0.
Energy
Homogeneity

For scaling down images into 9 :

Let $\frac{255}{9}$ $\text{Approx} = 28$ 80 0 to 28 \rightarrow 1 1 to 9
28 6 56 \rightarrow 2

Second Order statistics features : (Using Histogram)
 $\text{imhist}(I)$

Thresholding

\hookrightarrow Ask input from user Binary or Multi

\hookrightarrow If Multi how many grps get as input
only 2 grps

To reduce the img size without losing quality & info
use

\hookrightarrow Convolution \rightarrow multiplication & addition
 \hookrightarrow Pooling

Convolution \rightarrow unnecessary info are removed

$C[A - I]$
4.10 m \leftarrow 2 programming
2 image processing

① convolution problem (X)

② conditional, control, iterative statements (X)

③ plotting methods using numeric datasets (X)

Question in whatsapp

5.a)

soln

Stride 'n' move n cols then n rows

Step 1 : Strided convolution stride = 2

1	2	3
0	4	6
2	0	1

1	0	3	0
1	0	6	1
5	5	8	1

filter

1	1	1
1	1	1
1	1	1

1x1 + 2x1 + 3x1 + 0x1
+ 4x1 + 6x1 + 2x1 + 0x1
+ 1x1

19		

Step 2 : stride 2

1	2	3	1	0	3	0
0	4	6	1	0	6	1
2	0	1	5	5	8	1
0	1	1	4	8	1	0
1	0	5	6	1	0	0
1	3	1	2	0	1	0
0	1	0	2	1	0	1

filter

1	1	1
1	1	1
1	1	1

19	22

Step 4 ←
Step 5 ←

Step 3 :

19	22	24

Step 4 : 19 22 24

11 36

Step 5 : 19 22 24
11 36

While striding filter size & image is not matching add zeros in rows/cols of image matrix

Step 6 : 19 22 24
11 36 24

Final :

19	22	24
11	36	24
12	18	4

(Total 9 steps will be there)

6) Pooling method: (Max, Min, Avg)



3	9	7	4	7	5
0	6	7	3	1	2
2	4	5	0	3	2
3	7	5	0	2	1
1	5	0	7	3	6
8	9	2	5	1	8

Min

0	3	1
2	0	1
1	0	1

Avg

4.5	5.25	3.75
-----	------	------

(but round off values)

4	2.5	2
5.75	3.5	4.5

Stride 2 Filter 2x2

Max

9	7	7
7	5	3
9	7	8

Switch:

switch $\{n\}$

case 1

② input validation()

cond stmts

Syntax

Rg: No: 45

Input validate

Rg: No: 60

Loop stmts Rg: 71

- > isLetter()
- > isKeyword()
- > isEmpty()
- > isACnum, 'FNT16')

✗ Writing a simple text file & Reading (2m)

Control stmts:

i) continue: skips the stmts below continue stmts to next % sum of even numbers %. Iteration

sum = 0;

for $i = 1 : 10$

if $(\text{mod}(i, 2) \neq 0)$

continue

end

sum = sum + i;

end

11) break : prematurely terminating the loop

eg: For $i = 1 : 10$

$n = \text{input}('Enter +ve number: ');$

$\text{if } (n < 0)$

break

end

end

⑩ Files ⑩ 10m \rightarrow write detailed fprintf or disp
Read & write nums on files : starts whenever possible

data = [1 2 3 4 5 6 7 8 9 10];

filename = 'test.txt';

fid = fopen(filename, 'w');

fprintf(fid, '%d\n', data);

fclose(fid);

fid = fopen(filename, 'r');

} write

1.4 methods to read file

1. 1 %

$y1 = \text{fscanf}(\text{fid}, '%d\n');$

$\text{fclose}(\text{fid});$

$y2 = \text{textread}(\text{filename}, '%d');$

$y3 = \text{importdata}(\text{filename});$

$y4 = \text{load}(\text{filename});$

float \rightarrow %f string \rightarrow %s

} 4 methods to read file
⑩ 2m or ⑩ 10m

⑩ Debug whether file open (validation) ⑩ 2m

file1 = fopen('text.txt', 'w');

$\text{if } \text{file1} == -1$

error('Failed to open file');

end

Read & Write Table :

Id = [101; 102; 103; 104];

Shape = {'A'; 'B'; 'C'; 'D'};

Price = [10.0; 13.5; 10.5; 0.16];

Stock = [376; 476; 500; 100];

T = table(Id, Shape, Price, Stock);

writeTable(T, 'table.txt');

tableData = readTable('table.txt');

Debugging

2 types of error

- > Syntax error
- > Runtime error

Debugging Process : 10m or 2m → Detailed exp of every steps

i) Preparing for debugging ii) Setting breakpoints

iii) Running an M-file with breakpoints } combined explanation

iv) Stepping through an M-file

v) Examining values

vi) Correcting problems vii) Ending debugging

i) Preparing for debugging :

> open file > save changes > directories

ii) Setting breakpoints : 2m

Types ⇒ • standard • conditional • error breakpoint

iii) Running with breakpoints :

(Use step given in Matlab)

ⓧ Correcting an M-File steps ⓧ 2M

- > Quit debugging
- > Don't make changes to an M-File while MATLAB is in debug mode
- > Make changes to M-File
- > Save the M-File
- > Clear breakpoints

ⓧ We can implement do while even though not supported

eg :
i = 11;
while 1
disp(i)
if (i > 10)
break;
end
end

ⓧ 2m or 5m

ⓧ For 2m Inbuilt func for prime like
isprime()

ⓧ Input validation ⓧ

- ↓
- > isa (arg1, arg2) ⇒ arg2 -> var arg2 → 'class'
(or) eg: single, logical, char, int8
 - > isnumeric(), islogical, ischar(),
isinteger(), isfloat()

ⓧ APP Building components For 10M - 7 comp 5m - 4 comp
ⓧ 2m → specific comp is asked
> For each comp 4 properties, 1 function

i) Button

- > text
- > FontName
- > FontSize
- > FontColor

} Properties func → callbackFn

ii) checkbox

vi) Label

iii) DatePicker

vii) ListBox

iv) DropDown

viii) RadioButton

v) EditField

ix) TextArea

← Numeric
Alphanumeric

> Image Filter Operations: Consider each ele as center point then do the given filter operation

1) Min

2) Max

3) Median (sort 9 elements then the middle one)

4) Mean ($\text{sum}/9 \rightarrow 3 \times 3$)

eg:

0	1	0
0	1	2
2	6	7

It shd be updated with min or max or ...

2nd iteration

0	1	0	0
1	2	3	4
6	7	8	9

3rd iteration

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

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

> Finding min / max in an array without inbuilt fun.

$a = [4 \ 3 \ 1 \ 2 \ 6]$;

$\text{min} = a(1)$; % 1st ele as min

for $i = 2$ to $\text{length}(a)$

if $\text{min} > a(i)$

$\text{min} = a(i)$

end

end

> Convolution using Matlab script:

& Pooling

Max 6×6 Input matrix

Filter 3×3

for $i = 1:3$

for $j = 1:3$

$\text{sum} = \text{sum} + (I(i,j) * F(i,j))$

end

$\text{conv}(i) = \text{sum};$

end

Input matrix

Filter matrix

CIA-3

- > Files (read & write)
- > Matrix vector operations
- > Matlab App