

Part V

`fprintf`

`textread` and `cell` arrays

`reading` code

FPRINTF: weird. p16-17

- fprintf allows us to write to a file.
- fprintf(fileID, 'PRINTING FORMAT', variable).

Start of conversion specification — %o-12.5e — Conversion character

Flags —

Field width — Precision

The diagram illustrates the components of the conversion specification `%o-12.5e`. It shows the following parts from left to right: a flag character `-`, a field width `12`, a precision `.5`, and a conversion character `e`. Brackets and lines connect these parts to their respective labels: 'Start of conversion specification' points to the beginning of the string, 'Flags' points to the `-`, 'Field width' points to `12`, 'Precision' points to `.5`, and 'Conversion character' points to `e`.

The 'printing format' is where you manage your CURSOR.

Start of conversion specification — % — 12.5e — Conversion character

Flags —

Field width —

Precision —

FPRINTF: weird p16-17

- OPEN PAGE 17.
 - flags page 17
 - Field width: MINIMUM number of digits (spaces) to be printed
 - Precision: decimals after period.
 - CURSOR: RESERVES Leftward space
Overruns rightwards.

Start of conversion specification — % — 12.5e — Conversion character

Flags —

Field width —

Precision —

FPRINTF: weird p16-17

- Most common conversion characters.
 - %c single character
 - %d decimal notation
 - %s string of characters
 - INSIDE PRINTING area:
 - \n new line
 - \t horizontal tab

WRITING AND READING FORMATTED DATA

- DATA files are formatted. We can write formatted data with `fprintf` and read them into variables with `fscanf`.
 - It's all about cursor placement.

The cursor:

- DON'T SPECIFY FID: (works same)
prints to command window.
- `fprintf('I print this.\n');`
- `fprintf('I print this.');`
- Now, let's work on cursor placement.

The cursor:

```
fprintf('I print this.') %shift  
enter
```

```
fprintf('and this.');
```

-->cursor continues printing where
it left off.

--> Add a space inside the ' ' at
end of first fprintf command.

The cursor:

RESERVING LEFTWARDS SPACE:

- `myname = 'alejo';`
- `fprintf('%s%s%s', myname, myname, myname);` %compare to
- `fprintf('%6s%6s%6s', myname, myname, myname);`
- THERE IS ONE UNUSED "LEFT" SPACE

The cursor:

FORMATTED DATA

- `trial=1:10;`
- `condition = mod(trial,2);`
- `fprintf(' %2d %d',trial,`
 `condition);`
- `fprintf(' %2d %d\n',trial,`
 `condition);`
 - What happened?

The cursor:

FORMATTED DATA

- ```
for count=1:10
 fprintf('%2d %d\n',trial(count),...
 condition(count));
end;
```

– What happened?

# Intermixing text and variables

- `fprintf('This is trial %2d.\n', trial);`
- `for count=1:10`
  - `fprintf('This is trial %2d, and condition %d\n.', trial(count), condition(count));`
- `end;`

# Example

- Create a three column matrix with:
- first column: numbers from 1-10.
- second column: alternating 0-1.
- third column: random number between 150 and 1000.
- WRITE TO screen:
  - think trial number, condition, RT.

# Example

- `data = zeros(10,3);`
- `data(:,1)=1:10;`
- `data(:,2)=mod(data(:,1),2);`
- `data(:,3)=rand(1,10)*850 +150;`

# Example

```
%with for loop:
```

```
for count=1:10
```

```
 fprintf('%2d %d %3.1f\n',data(count,1),...
```

```
 data(count,2),data(count,3));
```

```
end;
```

# Example

`%WITHOUT FOR LOOP`

```
> fprintf('%2d %d %3.1f\n',data);
```

- writes data column-wise.
- Treats matrix as comma-delimited list.
- CONTINUES EXECUTION until all the specified variables HAVE BEEN PRINTED.

IOW, it goes through data as a stacked list of values and prints those in the order it finds them (all the first column first, followed by all the second column, etc...)

- what we want is:  
data': the transposition of data

```
> fprintf('%2d %d %3.1f\n',data');
```

# Last issue.

- How do you print a ' or % or \ with fprintf?

ex: it's a beautiful day!

ex: I'm 100% certain 2\4=2.

- Answer: you double the escape character to make it printable (page17)

```
> fprintf('I'm 100%% certain 2\\4=2.')
```



Questions?

# Exercise fprintf

Ask user to enter N student names.

Ask user to enter a specific student's 3 grades and SS#.

Save everything in one file `stdata.txt` of the sort:

|          |     |        |    |    |
|----------|-----|--------|----|----|
| Student1 | SS# | grade1 | g2 | g3 |
| Student2 | SS# | grade1 | g2 | g3 |
| ...      |     |        |    |    |

# Exercise fprintf

You will use

input

fprintf

for loop

# Exercise fprintf

[illegible]

# Exercise fprintf

```
for n=1:nu
 fprintf('What ' 's the name of
student number %d? ',n);
name=input(' ', 's'); %reads string
fprintf('What ' 's %s ' 's social
security number? ', name);
ssnum=input(' '); %reads number
%NEW LOOP FOR GRADES!
for gr=1:3
```

# Exercise fprintf

```
%NEW LOOP FOR GRADES!
```

```
for gr =1:3
```

```
 fprintf('What was %s''s grade in...
 exam %d?', name, gr);
```

```
 grade(gr) = input(' ');
```

```
end;
```

```
fprintf('\n');
```

```
 fprintf(fid, ' %-10s %11d %3d %3d...
 %3d\n', name, ssn, grade(1), ...
 grade(2), grade(3));
```

```
%note the minus sign: left justify.
```

```
end; %of for nu loop
```

```
end; %of else loop
```

# textread

Now we have a formatted data file. We want to read in the values to perform calculations on them (average grade) --> *textread* function.

SYNTAX:

`A = textread('filename')` transforms data in filename into Matrix A.

ONLY WORKS WITH HOMOGENEOUS Matrices.

# textread

SYNTAX:

```
[A,B,C] = textread('filename','%s%d%f')
```

reads each column into a variable, of specified type.

GROOVY!

Note: strings are saved in "cell" arrays (**multidimensional arrays whose elements are copies of other arrays**, here a table of strings of different sizes). clarity will come...



# textread

So, let's do it! Read your student data.

```
[names,ssnum,gr1,gr2,gr3]=...
 textread('stdata.txt','%s%d%d%d%d');
```

check out names: what's new?

try:

```
> names(1)
```

```
> names{1}
```

```
> names{1}(1)
```

Anyone wants to comment?

# Parenthesis: cells

`names(1)` is the cell itself

so `trash = names(1)` makes trash a cell

`names{1}` refers to the value in the cell

so `trash = names{1}` makes trash a character array

`names{1}(j)` is the jth element in the character array stored in the cell 1.

CHECK OUT THE ARRAY EDITOR... SWEET!

In a cell array, every cell can have whatever you want in it! No matter size or type of variable!!

# textread

We have the values, now we can calculate the averages.

```
numstu = size(names,1); %number of rows
average = zeros(numstu,1);
for n = 1:numstu
 average(n) = (gr1(n)+gr2(n)+gr3(n))/3;
end;
```

# Exercise: `querystdata.m`

Write program `querystdata.m`, that asks user for name of student and calculate that student's grade. Use **`strcmp`**

```
name2f = input('what student?','s');
for findex=1:numstu
 if (strcmp(name2f,names{findex}))
 whichisit =findex;
 end;
end;
%You can finish from here.
```

# Reading code

Take a piece of paper and

1. write in English what the following program is doing.
2. write the output of the program for **all** possible user inputs.

YOU SHOULD USE "HELP" to understand functions you don't recognize.

# Exercise 1: reading code

```
%CosmoSexQuiz.m written by Susi Bloggs June 2000
score=0;
gender=questdlg('Are you male or female?','Question', 'Male', 'Female', 'Male');

like=questdlg('Do you like your partner?', 'Question', 'Yes', 'No', 'Yes');

if strcmp(like, 'Yes')
 score=score+2;
else
 score=score-2;
end;

unfaithful=questdlg('Are you sleeping with anyone else?', 'Question', 'Yes', 'No', 'No');
if (strcmp(unfaithful, 'Yes') & strcmp(gender, 'Male'))
 score=score-2;
elseif (strcmp(unfaithful, 'Yes') & strcmp(gender, 'Female'))
 score=score-5;
else
 score=score+1;
end
if(score >0)
 str=(['Your score = ', num2str(score), '. Happiness awaits!']);
elseif(score ==0)
 str =(['Your score = ', num2str(score), '. You''re very very confused.']);
else
 str =(['Your score = ', num2str(score), '. Get real. Move on.']);
end;
questdlg(str, 'Answer', 'Quit', 'Quit');
```

# Exercise: reading code

`help questdlg`

*ButtonName = QUESTDLG(Question) creates a modal dialog box that automatically wraps the cell array or string (vector or matrix) Question to fit an appropriately sized window.*

*The name of the button that is pressed is returned in ButtonName.*

*The Title of the figure may be specified by adding a second string argument:*

*ButtonName = questdlg(Question, Title)*

# Exercise: reading code

`help questdlg`

*Up to 3 custom button names may be specified by entering the button string name(s) as additional arguments to the function call. If custom button names are entered, the default button must be specified by adding an extra argument, DEFAULT, and setting DEFAULT to the same string name as the button you want to use as the default button:*

```
ButtonName = questdlg(Question,...
 Title, Btn1, Btn2, DEFAULT);
```



# Exercise: reading code

1. Asks for gender (default is male)

2. Asks for likeness of partner  
(default is yes).

if you like--> positive score

if you don't --> negative

3. Asks for fidelity

same, but more negative points for  
females.

4. if positive score: happy

if negative score: break up

if null score: undecided

# Exercise 2: reading code

Take a piece of paper and

1. write in English what the following program is doing.
2. draw what the output will be.

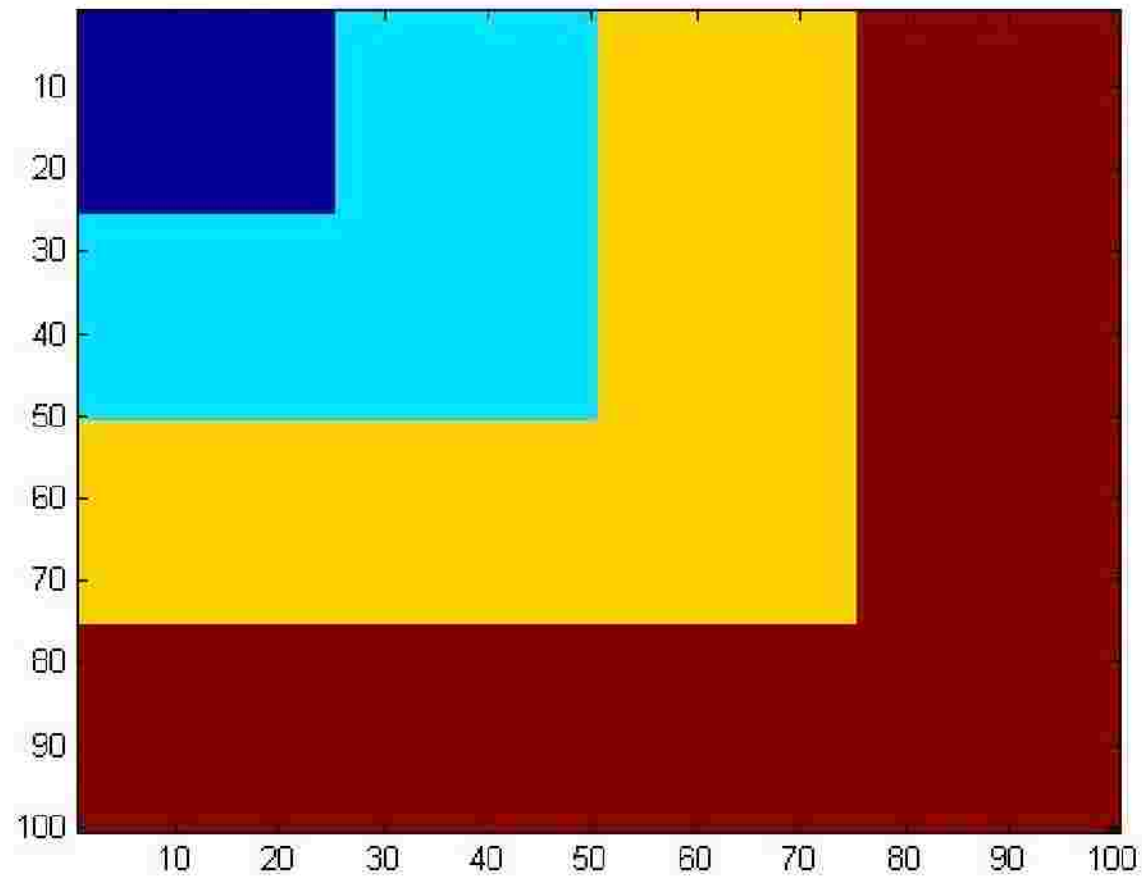
YOU SHOULD USE "HELP" to understand functions you don't recognize.

# Exercise 2: reading code

```
> clear;
> im=zeros(100, 100);
> for square=[100 75 50 25]
> for trows= linspace(1, 100, 100)
> for tcols=linspace(1, 100, 100)
> if((trows<=square) & (tcols<=square))
> color=square./25;
> im(trows, tcols)= color;
> end;
> end;
> end;
> end;
> imagesc(im);
```

# Exercise 2: reading code

- Answer:



# Exercise 3: reading code

Take a piece of paper and

1. write in English what the following program is doing.
2. what will be the output of the program?

YOU SHOULD USE "HELP" to understand functions you don't recognize.

## Exercise 4: reading code

```
fid = fopen('UofI.m','r');
count = 0;
while ~feof(fid)
 line = fgetl(fid);
 if isempty(line) | strncmp(line,'% ',1)
 continue
 end
 count = count + 1;
end
disp(sprintf('%d lines',count));
```