Part II: Flow Control

- FOR loops
- LOGIC
- IF statements

Flow control p.12

REPEATING ORDERS:

- ‘FOR’ loops
- If you want to do something a fixed number of times:
  - you say “FOR the next ten times do this.”

script files

FOR EXAMPLE: UofI.m

ØWRITE A SCRIPT TO CALCULATE THE FIRST 10 numbers in the sequence:
u(n) = u(n-1) x (n+1)
with u(1) = 1

ØWhat we want is a vector U with 10 rows, U(i) = u(i)

script files

FOR syntax:

This is a “list” the “FOR” command will run through…

Any list will do!

```
àfor n=first:increment:final
    . . .
    . . .
  end
```

Type n=1:10:100

on your command prompt

What happened?

And k=1212:-pi:0

--> USE THE COMMAND “zeros” to get your variable U started. **REMEMBER!**

script files

A NOTE ABOUT HELP

The first continuous lines of commented text atop of your script are read as the “help” file for your program.

Etiquette dictates that you specify at least:

- What the program does and the “how to”
- Who wrote it
- When it was last updated
- Quirks & bugs

GO AHEAD AND INCLUDE HELP FOR UofI

UofI.m

```
% this is my first program, it does ...
% created on September 28, 2010 by Alejo.
% it calculates the first 10 numbers in the
% sequence:
% u(n) = u(n-1) x (n+1)     with u(1) = 1


u=zeros(10,1);
u(1)=1;
for i=2:10
    u(i,1) = u(i-1,1) .*(i+1);  %INDENT
end;
u                          %this is to show your output
                           %Here show without ; in for loop



.
```

UofI.m

Issue of first and last CASE!!

In your mind, testing these two cases is ***critical***!!!!

UofI.m

- It would be nice to add the index to U. So each row show index i and U(i).
- Go ahead…

UofI.m

```
% this is my first program, it does ...
% created on September 28, 2010 by Alejo.
% it calculates the first 10 numbers in the
% sequence:
% u(n) = u(n-1) x (n+1)     with u(1) = 1

U=zeros(10,2);
U(1,1)=1;
U(1,2)=1;
for n=2:10
    U(n,1) = U(n-1,1) .*(n+1);
    U(n,2) = n;
end;
U
```

UofI.m

It would be even nicer, if we asked the user the first value of the series.

Use ‘input’

input

Syntax:

```
VAR = input('your text here');
```

(EVALUATED INPUT)

For entering strings of characters

```
VAR = input('Your text here','s');
```

(NOT EVALUATED INPUT)

input is a "BLOCKING CALL"

UofI.m

```
% this is my first program, it does ...
% created on September 28, 2010 by Alejo.
% it calculates the first 10 numbers in the
% sequence:
% u(n) = u(n-1) x (n+1)     with u(1) = 1

U=zeros(10,2);
U(1,1)=input('first value? ');
U(1,2)=1;
for n=2:10
    U(n,1) = U(n-1,1) .*(n+1);
    U(n,2) = n;
end;
U
```

UofI.m

RUN UofI and as input put:

1220-1219

That’s why it is evaluated input.

FOR loop (part deux)

What is the variable:

myname= ‘ALEJO’;

Answer:

A list of 5 characters!

EXERCISE:

Try switching the case of myname by adding 32 to each ascii code…

Check the Help of double

What does it mean???

Well, a “character” is not a double, so double transforms it into its “number equivalent”, which is its ASCII code.

So,

asciinumberofa=double(‘a’)

How do we get the letter back?

Try “edit char”

so, now type:

```
backtoa=char(asciinumberofa)
```

FOR loop (part deux)

What is the variable:

myname= ‘ALEJO’;

Answer:

A list of 5 characters!

EXERCISE:

Try switching the case of myname by adding 32 to each ascii code…

A second look at for

```
% this scripts changes the case of UPPER CASE
% words.
   myname='ALEJO';
   i=1;
   for l=myname
      lowname(i)=double(l)+32;
      i=i+1;
   end;
   lowname=char(lowname)
```

Did the size of my name mattered???

Try running the program with: myname='Alejo'

Without the “for”

MATLAB CAN WORK ON ENTIRE MATRICES AT ONCE!
THIS IS THE COMPUTATIONAL POWER OF MATLAB

```
myname=‘ALEJO’;
asciiname=double(myname);
newname=asciiname+32;
lowname=char(newname);
```

We’ve VECTORIZED the For Loop!
An order of magnitude faster.

One more example:

WRITE Moo3.m

in which you add all the elements of a two dimensional matrix entered by the user.

USING: “input”
for loop
“size” (help size)

Moo3.m

```
matrix=input('enter two dimensional matrix ');
s=size(matrix);
tot=0;

for i=1:s(1)
        for j=1:s(2)
                tot=matrix(i,j)+tot
        end
end
```

WORK ON A PIECE OF PAPER THE STEPS OF THIS FOR LOOP.

Moo3.m

```
matrix=input('enter two dimensional matrix ');
s=size(matrix);
tot=0;

what is
sum(matrix)?     %help sum

so what is
tot=sum(sum(matrix));
```

LOGIC p.8

AND:	&
OR:	|
Complement:	~
Exclusive OR:	xor

LOGIC p.8

What is **TRUE** in Matlab?
NON ZERO ELEMENTS.

TRY:

```
a=[0 1 1 0 1];
b=[1 1 0 0 234009];
c = a&b
d = a|b
e = ~a
f = xor(a,b)
```

LOGIC p.8

Question (without typing):

```
Yourlove = - 18.345;
Mylove  = Yourlove * (-1000);

Ourlove = xor(Mylove,Yourlove);
Whose love is true?
```

LOGIC p.8

BRUSH UP ON YOUR LOGIC!!

--> YOU ***WILL*** MAKE LOCIGAL MISTAKES and you'll think your computer has gone crazy.

… humbling experience

-->TEST AND RETEST LOGICAL STATEMENTS.

-->DO NOT MAKE CONVOLUTATED logical statements in one swoop:
such as "if (this is true) or (that is wrong while x is also true) but not y".

à LANGUAGE LOGIC is NOT computer LOGIC

IF

```
IF something is true do
  {list of commands}
Else
  { list of commands2}
end
```

IF syntax

```
if (conditional statement)
    …
 elseif (conditional statement)
    …
 else
    …
end
```

IF syntax

So…

```
if (Mylove)
  Mylove=0;
else
   Yourlove=0;
end;
```

Whose love is true?

IF syntax

So…

```
if (Mylove == true)  % == is not =
  Mylove=0;
else
   yourlove=0;
end;
```

Whose love is true?

WHY????

```
  %help true
```

IF exercise

Ask user for password. If password is less than 6 characters long, return "invalid password"

IF exercise

Algorithm steps:

1. ask for user input (password)
2. check size of password
3. if size is too small, reject password

IF exercise

```
>  pwd = input('Please enter password: ‘,’s’);
   pwdsize = length(pwd);  %check size of pwdsize
   if (pwdsize < 6)
      'password invalid'
   else
      'valid password'
   end
```

Exercise (if time permits):

Write a script that asks the users for their name and returns it in scrambled fashion.

Use the commands ‘input’ and ‘randperm’.

Algorithm steps

- obtain user name
- Question: do we know the number of steps or repetitions?
- Can we obtain it?
- scramble using for loop.

Scramble.m

```
% notes on program/mer

name = input('What''s your name? ‘,’s’);

name_sz = length(name);

neworder = randperm(name_sz);

for i=1:name_sz
        newname(i) = name(neworder(i));

%Matlab dynamically adds cells to your variable in this way,
%but only inside a for loop!!!

end;

char(newname)
```

Run Scramble with a LONG name

Run it again with a smaller name.

What happened?

How truly inconvenient!!!

TOMORROW, "functions" will deal with this issue.

Write a script that asks the users for their name and returns it in scrambled fashion.

Use the commands ‘input’ and ‘randperm’.

Can you do the same without a flow control statement?

```
Scramble.m
% notes on program/mer

name = input('What''s your name? ‘,’s’);
name_sz = length(name);
neworder = randperm(name_sz);
newname=name(neworder);


%more compact:
  newname=name(randperm(name_sz));
```

BREAK!!

Be back at 1:00 p.m.