

Name: Andrew Brown	Lab Time: T 12:00
People Worked With: Cailin Moore	Websites Used: n/a
Time spent on zyBooks (hrs):2	Time spent on lab (hrs):2
Submission Instructions	
Turn all work in to Lab 3 on Gradescope (PDF) and Canvas (.zip file), even if it is not complete yet. If you are not finished, complete the assignment outside of lab and re-submit to Lab 3 on Gradescope and Canvas. All labs are typically due at the same time on Monday every week, but check Canvas if in doubt.	

Learning objectives:

- Write several simple for loops.
- Write several simple if statements to control program flow.
- Edit a variable inside of a for loop.
- Using the debugger to step through the for loops and if statements.

New MATLAB commands

These are highlighted in **bold** in the instructions below.

- `length(array)` – Length of an array
- `subplot(nrows, ncols, position)` – split the plotting window
- `zeros(nrows, ncols)` – make matrix of zeros. Use 1 for nrows to make an array
- `for k = (array)`
 % things to do in for loop
`end`
- `if (condition 1)`
 % things to do if condition 1 is true
`elseif (condition 2)`
 % things to do if condition 2 is true
`else`
 % things to do if neither conditions are true
`end`

Formatting Needs

Properly indent all loops and if statements. The easy way is to select all the code in your script and hit this green button in the editor:



DO THIS FREQUENTLY AND BEFORE TURNING IN YOUR SCRIPT.

Indenting is one of life's essentials.

Lab Problems

Files to download to your Lab 3 Folder

- *Lec3_2ForVsWhile.m*

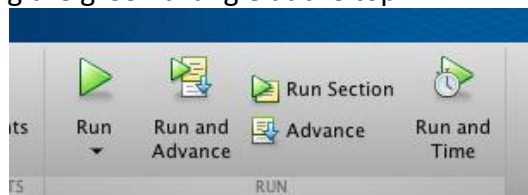
Getting Started

Read before you write, practice on paper before you code, and practice with the debugger... I highly recommend you walk through the following, even if you're 100% sure you can jump to the problems and do them. If you do skip this, come back and do it when you're stuck.

- Open up the Lec3_2ForVsWhile script
- Read the description of Fibonacci numbers at the top – if you don't understand it, check out the [Wikipedia page](#).
- Write out what you expect the first 4 elements to be (i.e., pretend you're the computer and calculate the values).
- Put break points at lines 25, 27, 44 and 48
 - Click on the dashed line next to the number – a red dot should appear

```
21 % Fibonacci series - we know we want the first two elements to be 1
22 % (which they already are)
23 % So start the loop at 3
24 % k will be 3,4,5,6, then 7
25 for k = 3:nElements
26     % Calculate the value at the kth place from the previous two
27     t(k) = t(k-1) + t(k-2);
28     % Somewhere between here and starting the for loop again matlab will
29     % set k to be the next value in the array 3:nElements
30 end
31 % Each element of t should be the sum of the previous two elements
32 fprintf('t from a for loop is\n');
33 disp(t)
34
```

- Run the script by clicking the green triangle at the top



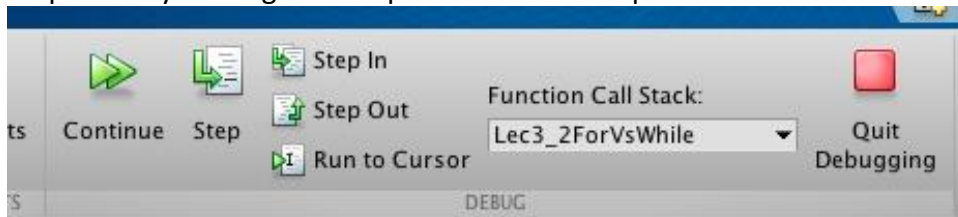
- Matlab will stop at line 25 – you'll see a green arrow where it stopped. Put your cursor over `t` and it will show you what the value of `t` is
 - If you put your cursor over `k`, what happens? Why? Does `k` exist yet? Check the variable window

```

24 % k will be 3,4,5,6, then 7
25 for k = 3:nElements
26 % Calculate the value at the kth place from the previous
27 t(k) = t(k-1) + t(k-2);
28
29 t: 1x7 double =
30 1 1 1 1 1 1 1
31 % Each element of t should be the sum of the previous
32 fprintf('t from a for loop is\n');

```

- Step once by clicking the “step” button at the top



- Put your cursor over k – what is the value?
- Go to the command line and type t(k-1). What do you get? Type t(k-2). What do you get? BTW, you can cut and paste: copy t(k-1) + t(k-2) from the script and paste it into the command line. What do you get? Finally, type t(k).
- Take one step. Watch where the green arrow goes. What is k?
- Take another step. What is k?
 - Your cursor should now be back at line 27. k should now be 4 – matlab set it to be the next element in the array 3:nElements.
 - What is t(k)? What is t(3)?
- This is called “stepping through the for loop” and “watching the variables change”. It is KEY to writing loops – every time you write a for loop you should put a breakpoint at the for statement and the first line of the loop and check what it does. This single thing will save HOURS of wasted time.
- When you are ready, take out the breakpoint at line 27 by clicking on the red dot
- Hit the Continue button
- This should take you to the **while** loop.
- Repeat what you did for the for loop – step through the loop and check the values of s and k
 - Notice that s “grows” each time through the loop
 - s(k) = blah;
 - if k is bigger than the length of s then matlab will make s bigger
 - Notice that we have to make our own k this time AND increment it at the bottom of the loop
- Take the break points out
- Try taking the k = k+1; line out of the while loop and re-run. What happens?
 - Use Control-c in the command window to stop the script
 - This is called an infinite loop.

Problem 1

Demonstrate your bottle counting skills and your ability to count backwards.

Write a for loop to print out the following:

- 1 bottles of (favorite beverage) on the wall
- 2 bottles of (favorite beverage) on the wall
- ... counting up to 10

Print out the total number of bottles on the wall.

Then, do the same, but count backwards.

Extra credit: Be grammatically correct: “1 bottle of...” instead of “1 bottles of...”

Step by Step Instructions:

- Create a **for** loop that starts at 1 and ends at 10 and prints out ‘bottles of favorite beverage’ each time through the loop
 - If you aren’t sure how to start, copy the **for** loop from Lec3_2ForVsWhile, change $k=3:nElements$ to $k = 1:10$ and replaces the $t(k)$ equation with an **fprintf**.
- Now alter the **fprintf** so that it prints out your loop variable (k) every time through the loop.
- To do the sum: Create a variable **before** the for loop (do NOT call it **sum**, since that’s a MATLAB function; call it **totalSum**) and set it to zero.
- Add the current number of bottles (k) to **totalSum** every time you go through the loop
 - $totalSum = totalSum + (k);$
 - Note that this gets the value in totalSum, adds (k), and puts it back into totalSum – use the debugger to check how totalSum changes every step of the loop
- Add a print statement **after** the **for** loop to print out the answer.
- Copy the **for** loop (don’t forget to copy the $totalSum = 0$ and print statements)
- Change $k = \dots$ in the for loop to count *backwards*
 - Start:step:End – what should step be to go backwards?
- EXTRA credit:
 - You will need an **if-else** statement in the **for** loop
 - If the number of bottles is 1, print bottle instead of bottles

Self-check:

```
1 cups of tea on the shelf
2 cups of tea on the shelf
3 cups of tea on the shelf
4 cups of tea on the shelf
5 cups of tea on the shelf
6 cups of tea on the shelf
7 cups of tea on the shelf
8 cups of tea on the shelf
9 cups of tea on the shelf
10 cups of tea on the shelf
Total number of cups is: 55
10 cups of tea on the shelf
9 cups of tea on the shelf
8 cups of tea on the shelf
7 cups of tea on the shelf
6 cups of tea on the shelf
5 cups of tea on the shelf
4 cups of tea on the shelf
3 cups of tea on the shelf
2 cups of tea on the shelf
1 cups of tea on the shelf
0 cups of tea on the shelf
Total number of cups is: 55
```

Grading Criteria:

[20 pts] Sufficient comments
[20 pts] for loop correct
[10 pts] fprintf in for loop correct
[10 pts] Sum computed in the for loop
[10 pts] Sum printed out at the end
[20 pts] Reverse the for loop
[10 pts] Sum for second for loop correct
[15 pts EC] if-else to print 1 bottle

Answer script here:

```
% Andrew Brown Lab 3 Problem 1

clc
clear

%Practice with for loops and if statements

totalSum=0;
%For loop counting from 1 to 10 going up
for i=1:10
    if i==1 %use proper grammar for the first bottle
        fprintf('%0.0f bottle of beer on the wall\n',i)
    else %use proper grammar for the plural "bottles"
```

```
        fprintf('%0.0f bottles of beer on the wall\n',i)
    end %end if statement
    totalSum=totalSum+i;%add current bottles to toalSum
end %end for loop

%print out the total number of bottles from the totalSum variable
fprintf('Total number of bottles is: %0.0f\n',totalSum)

%Reset the variable totalSum to 0
totalSum=0;

%For loop conting from 10 to 1 going down
for k=10:-1:0
    if k==1 %use proper grammar for the first bottle
        fprintf('%0.0f bottle of beer on the wall\n',k)
    else %use proper grammar for the plural "bottles"
        fprintf('%0.0f bottles of beer on the wall\n',k)
    end %end if statement
    totalSum=totalSum+k; %add current bottles to toalSum
end %end for loop

%print out the total number of bottles from the totalSum variable
fprintf('Total number of bottles is: %0.0f\n',totalSum)
```

Command window output

```
1 bottle of beer on the wall
2 bottles of beer on the wall
3 bottles of beer on the wall
4 bottles of beer on the wall
5 bottles of beer on the wall
6 bottles of beer on the wall
7 bottles of beer on the wall
8 bottles of beer on the wall
9 bottles of beer on the wall
10 bottles of beer on the wall
Total number of bottles is: 55
10 bottles of beer on the wall
9 bottles of beer on the wall
8 bottles of beer on the wall
7 bottles of beer on the wall
6 bottles of beer on the wall
5 bottles of beer on the wall
4 bottles of beer on the wall
3 bottles of beer on the wall
2 bottles of beer on the wall
1 bottle of beer on the wall
0 bottles of beer on the wall
Total number of bottles is: 55
```

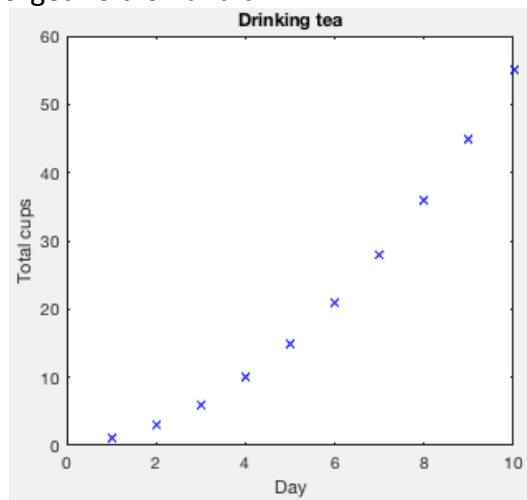
Problem 2

Plot the growth of total number of bottles on the wall.

Note: This is about how to save values you calculate in your for loop so you can plot them later.

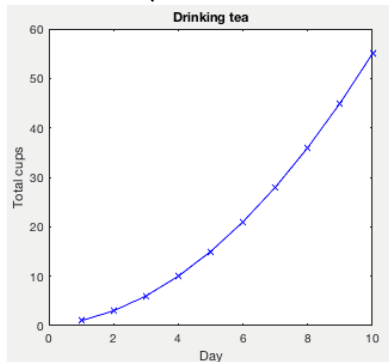
Step by Step Instructions:

- Copy your script from problem 1 and delete the second half (counting backwards)
- Add a line in the for loop to plot a marker for each k, totalSum pair
 - Note – make sure to use markers, not lines, or nothing will show up
 - Don't forget hold on and clf



- It's your choice if you count bottles at the *start* of the for loop (first point would be at 1,0) or at the *end* of the loop (as is shown above). This **will** matter for Euler/Epidemic, btw.
- How do you plot a line instead of markers?
 - Answer: you need to save the values you calculate as an array so you can plot them afterwards
- To do this cleanly, first declare a variable n = 10 that is the total number of times you want to go through the loop
 - Change your **for** loop indexing to use n instead of 10
- Now declare **two** array variables (**saveSums** and **saveDays**) that are of length n
 - Don't hard-wire 10 in – it's better to set their size to be the number of times you are going through the loop.
- In the loop, set the kth value of **saveSums** to be **totalSums** and the kth value of **saveDays** to be k.
- Plot **saveDays** versus **saveSum**
- Change n to 15 and make sure your script still works WITHOUT having to change anything but n.
- Plot with n = 10 or n = 15.

Self-check (either $n = 10$ or $n = 15$ is fine, but must have line connecting markers):



Grading Criteria:

[20 pts] Has comments

[10 pts ea] Declare the two array variables (correct size, based on n)

[10 pts ea] Sets the array variable in the for loop

[20 pts] for loop depends on n

[20 pts] plot with labels and title

Answer script here:

```
% Andrew Brown Lab 3 Problem 2

clc
clear
clf

%How to save values you calculate in your for loops so you can plot them
%later

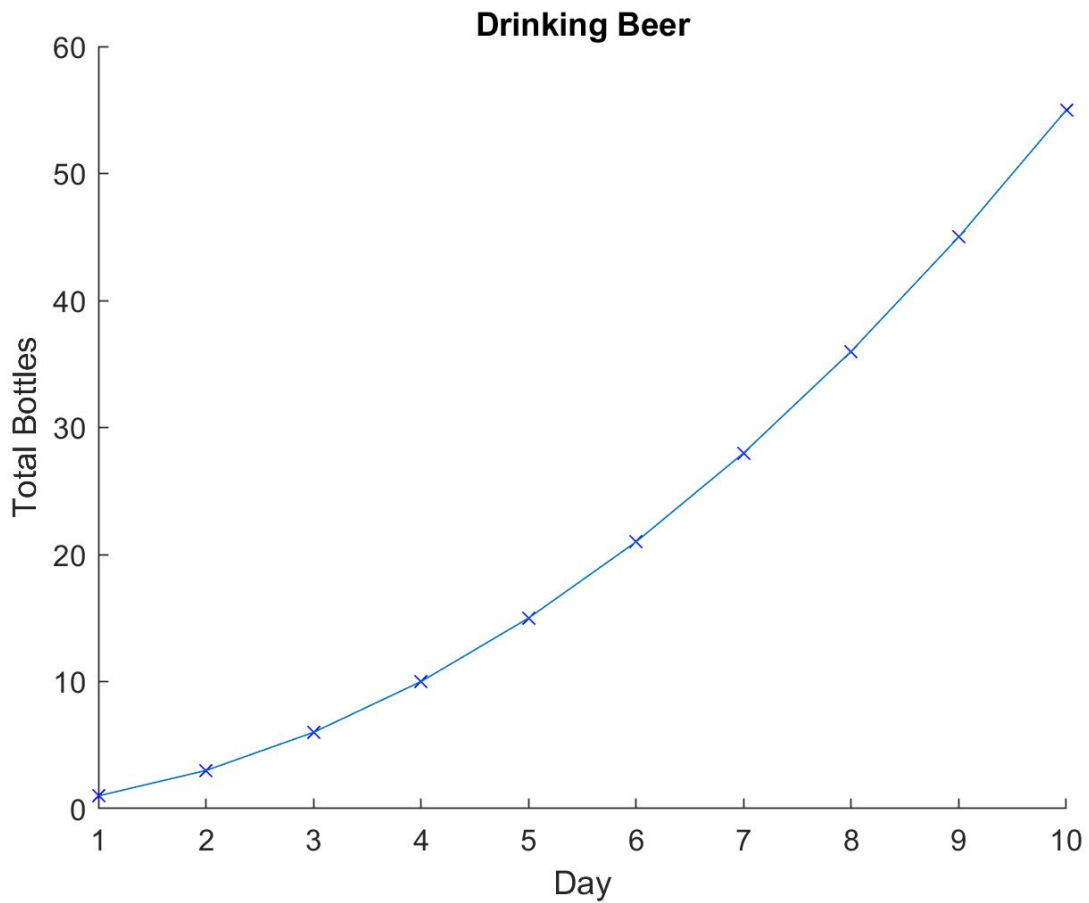
%Define given variables
totalSum=0;
n=10;
saveSums=zeros(1,length(n));
saveDays=zeros(1,length(n));

%For loop counting from 1 to 10 going up
for k=1:n
    if k==1 %use proper grammar for the first bottle
        fprintf('%0.0f bottle of beer on the wall\n',k)
    else %use proper grammar for the plural "bottles"
        fprintf('%0.0f bottles of beer on the wall\n',k)
    end %end if statement
    totalSum=totalSum+k;%add current bottles to totalSum
    saveSums(k)=totalSum; %set kth term of saveSums to the current total sum
    saveDays(k)=k; %save an array that counts k
    hold on
    plot(k,totalSum,'xb')
    title('Drinking Beer')
    xlabel('Day')
    ylabel('Total Bottles')
end %end for loop
```

```
%Plot the line on the same graph
hold on
plot(saveDays,saveSums)

%print out the total number of bottles from the totalSum variable
fprintf('Total number of bottles is: %0.0f\n',totalSum)
```

Plot here:



Problem 3

Sum up bottles until you are over 75. How many days/loops are needed? Print out bottles on the wall and total until you reach at least 75.

Step by Step Instructions:

- Look at Lec3_2ForVsWhile.m to see an example of changing a **for** loop into a **while** loop
- Starting with your problem one lab script (no plotting) will probably be easier than starting with your lab script from problem two.
- Replace the **for k = ...** part of the loop with a while loop
 - What is your stopping condition?
- You will need to make your own k variable, and increment it at the bottom of the loop as shown in Lec3_2ForVsWhile.m
- Remember control-c in the command window if your program doesn't stop

Self-check:

```
1 cups of tea on the shelf, total 1
2 cups of tea on the shelf, total 3
3 cups of tea on the shelf, total 6
4 cups of tea on the shelf, total 10
5 cups of tea on the shelf, total 15
6 cups of tea on the shelf, total 21
7 cups of tea on the shelf, total 28
8 cups of tea on the shelf, total 36
9 cups of tea on the shelf, total 45
10 cups of tea on the shelf, total 55
11 cups of tea on the shelf, total 66
12 cups of tea on the shelf, total 78
Total number of cups is: 78 after 12 loops
```

Grading Criteria:

[20 pts] Comments

[30 pts] while loop

[20 pts] Correct stopping condition

[10 pts] print statement in the while loop (should include total and loop number as shown)

[10 pts] Total print statement after the loop

[10 pts] Setting and incrementing the loop variable (k)

Answer script here:

```
% Andrew Brown Lab 3 Problem 3
```

```
clc
clear
```

```
%Print out bottles on the wall and total until you reach at least 75
```

```
clc
clear
```

```
clf

%Practice with a while loop

%Define given variables
totalSum=0;
n=10;
saveSums=zeros(1,length(n));
saveDays=zeros(1,length(n));
k=0;

%For loop counting from 1 to 10 going up
while totalSum<75
    k=k+1;
    if k==1 %use proper grammar for the first bottle
        fprintf('%0.0f bottle of beer on the wall, ',k)
    else %use proper grammar for the plural "bottles"
        fprintf('%0.0f bottles of beer on the wall, ',k)
    end %end if statement
    totalSum=totalSum+k;%add current bottles to totalSum
    fprintf('total %0.0f\n',totalSum)
    saveSums(k)=totalSum; %set kth term of saveSums to the current total sum
    saveDays(k)=k; %save an array that counts k
    hold on
    plot(k,totalSum,'xb')
    title('Drinking Beer')
    xlabel('Day')
    ylabel('Total Bottles')
end %end for loop

%Plot the line on the same graph
hold on
plot(saveDays,saveSums)

%print out the total number of bottles from the totalSum variable
fprintf('Total number of bottles is: %0.0f after %0.0f loops\n',totalSum,k)
```

Command window output

```
1 bottle of beer on the wall, total 1
2 bottles of beer on the wall, total 3
3 bottles of beer on the wall, total 6
4 bottles of beer on the wall, total 10
5 bottles of beer on the wall, total 15
6 bottles of beer on the wall, total 21
7 bottles of beer on the wall, total 28
8 bottles of beer on the wall, total 36
9 bottles of beer on the wall, total 45
10 bottles of beer on the wall, total 55
11 bottles of beer on the wall, total 66
12 bottles of beer on the wall, total 78
Total number of bottles is: 78 after 12 loops
```

Extra Credit: Problem 1

Write a script that keeps getting a number from the user. If that number is positive, print positive; negative, print negative. If it is zero, print zero and then stop. Demonstrate with any sequence that mixes positives and negatives before giving it zero.

Partial credit for doing this for just one input number (ie, no loop)

Step by Step Instructions:

- Start with the **if** statement. Get a number from the user and print out if it is positive. Check that it works by inputting a positive, then a negative, then a zero. It should only print something if the number is positive
- Now add the first **elseif** – in this case, check to see if the number is negative.
 - Check again with positive, zero, and negative – nothing should be printed for the zero
- Now add the final **elseif**. You could use **else** with no condition; not incorrect, just doesn't catch not a number (see below)
 - BTW **NaN** stands for not a number – if you see this, you've gone and divided zero by zero or something like that.
- Now wrap the whole thing in a **while** loop.
 - What should the stopping criteria be?
 - There are two ways to do this:
 - Either put an
 - `x = input()`
 - outside of the loop AND inside of the loop to set x
 - OR
 - take advantage of the fact that you know what you're check for and just set x to be any other number before the loop
 - I prefer this because it's "safer" if you change the input statement

Self-check:

```
Number, please: -1
-1.000000 is negative
Number, please: 10.2
10.200000 is positive
Number, please: 3.0
3.000000 is positive
Number, please: -.00001
-0.000010 is negative
Number, please: 0
x is zero 0.000000
>>
```

If you want to be really, really correct:

```
Number, please: 3
3.000000 is positive
Number, please: -5
-5.000000 is negative
Number, please: 2
2.000000 is positive
Number, please: 1/0
Inf is positive
Number, please: -1/0
-Inf is negative
Number, please: 0/0
Neither negative or positive; must be not a number NaN
Number, please: 0
Neither negative or positive; must be zero 0.000000
>>
```

Grading Criteria:

[20 pts] Comments

[10 pts] input

[15 pts ea] if for negative, positive, zero

[20 pts] while

[5 pts] Catching nan

Answer script here:

```
% Andrew Brown Lab 3 Problem 4

clc
clear

%Write a script that keeps getting a number from the user. If that number
%is positive, print positive; negative, print negative. If it is zero,
%print zero and then stop. Demonstrate with any sequence that mixes
%positives and negatives before giving it zero.

x=input('Number, please: ');
while x~=0
    if x<0
        fprintf('%0.6f is negative\n',x)
    elseif x>0
        fprintf('%0.6f is positive\n',x)
    elseif x == isNaN
        fprintf('Neither negative or positive; must be not a number NaN')
    end
    x=input('Number, please: ');
end

%If x is 0
fprintf('Neither negative or positive; must be %0.0f\n',x)
```

Command window output

Number, please: -1

-1.000000 is negative

Number, please: 5

5.000000 is positive

Number, please: 0

Neither negative or positive; must be 0

zyBooks Challenge Exercises

Do the challenge activities for the following in Week 3

1. *Logical variables*
 - a. *Logic Variables*
2. *Relational operators*
 - a. *Equality check: Number of bricks*
 - b. *Small aircraft weight limit*
3. *Logical operators*
 - a. *Check bounds*
4. *If-else*
 - a. *Bridge toll*
5. *Multiple branches*
 - a. *Medication dosage by weight*
 - b. *[optional] Multiple if statements*
6. *For loops*
 - a. *For loops: Savings account*
 - b. *For loops: Odds sum*
7. *While loops*
 - a. *Writing a while loop*
 - b. *While loop with multiple conditions*
8. *String comparisons*
 - a. *Lie detector output converter*
 - b. *String comparison: Floor directory*