Jonathan Hess
Section 3.2

Week 9 Lab Report: Arrays

Lab Report Rubric

Category	Student Score	Grader Score
Organization		
Appropriate sections	1/1	/1
Appearance and formatting	2/2	/2
Spelling, grammar, sentence structure	1/1	/1
Work		
Experimental procedure	2/2	/2
Results (data, code, figure, graph, tables, etc.)	1.5/2	/2
Conclusion	1.5/2	/2
Total	**9/10**	**/10**

Introduction

Today we learned about arrays in Matlab (and Arduino). We also learned about plotting in matlab and how to make the data look more pleasing to the eye.

Procedure

Procedure

1. What are the values of the array? How many elements are there in 'arr'?

   The values are 1,3,5,7 and it has 4 elements

2. Try arr(1) and arr(3). What does this do?

   Gets the 1st element of the array and then the 3rd. This gets the values in the array. This is different from C or many other

3. What if you try arr(0) or arr(10)?

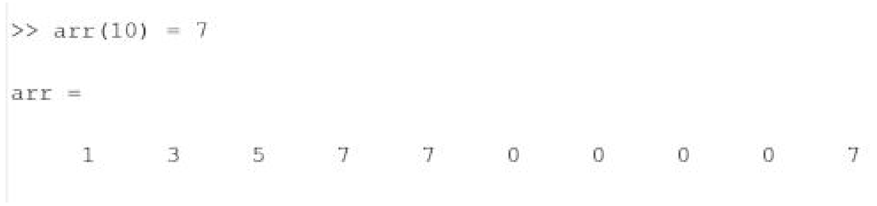
   It throws an error because it is not in the bounds of the array.

4. How about 'arr(2) = 10;' and 'arr(5) = 7;'?

   It sets the values of the array

5. How about 'arr(10) = 1;'? What is created and what are the values from first to last element?

   It will create empty values between the last known array value and 10. The default value is 0.



```
>> arr(10) = 7

arr =

     1     3     5     7     7     0     0     0     0     7
```

6. Go to the language reference page for MATLAB arrays (here) and look at some of the functions for arrays. Try 'length', 'sort', 'flip', 'max', and 'sum'. Discuss.

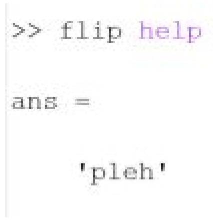
   Length- returns length of array

   sort - puts the array in ascending order but also has other features with more parameters

   flip - flips the array, I tried looking at help and accidentally flipped the string help

   max- finds the max in the array

   sum - adds all values of array



```
>> flip help

ans =

    'pleh'
```

7. Try ‘b = [1 3; 5 7];’. How is this different from the array a above?

Now the array is two dimensional

1. Without looking at the output on MATLAB, can you describe what is the function of this script?

   It prints out the values of the array

2. What changes do you make to print only the even number elements?

   Change the for loop to start at 2 then go to array_length/2 and print out arr(2i) instead or just change the incrementation of the for loop (still changing the start though)

Challenge #1: Notice each element in the array above is a power of two value. Create a script that will calculate the exponent of each value. Use a combination of array, for and while loop.

Well all the values are power of 2 so you can use log with base 2.

I went to the matlab help and found this page https://www.mathworks.com/help/matlab/ref/log.html

so the code “ log(arr)/log(2):” gives all the values in the array because log does the natural log of the values and then divides by the natural log of 2 (getting the log of 2 of the values)

This isn’t probably how it wanted you to solve the problem though so here is the other solution

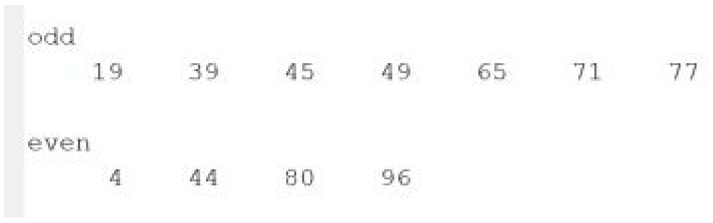
```
arr = [1 2 4 8 16 32 64 128 256 512 1024];
powers;
temp = 0;
for i = 1:1:length(arr)
  power =0;
  temp = arr(i);
  while temp > 1
    temp = temp/2;
    power = power + 1;
  end
    powers(i) = power;
end
disp(powers);
```

It works because the for loop should go through the array and then find how many powers of 2 can fit into it, you do this by dividing by 2 and then adding one to the power (annoyingly you cannot use power++; or temp=/2;)

Also a note here is that if you wanted you could do it all with the single array and without a temp value but I usually remove variables after the fact and it can make it easier to understand if they are still in there.

Challenge #2:Have MATLAB create an array with 10 random elements ranging from 1-100. First, rearrange the 10 elements in increasing numbers and print out the array. Next, have MATLAB sort the 10 elements into odd and even numbers.

```
array = [];
for i = 1:1:10
  array(i) = randi(100);
end
array = sort(array);
disp("in order array" + array);

even =[];
odd = [];
for i = 1:1:10
   if(mod(array(i),2))
      odd(length(odd)+1) = array(i);

   else
      even(length(even)+1) = array(i);
   end
end
disp("odd");
disp(odd);
disp("even");
disp(even);
```



```
odd
    19    39    45    49    65    71    77

even
     4    44    80    96
```

I first had it go through 1-11 in the for loop but that caused there to be 11 values instead (matlab for loop includes the last one. It is “<=” rather than just “<”)

It first assigns random numbers to the array then goes into the second loop.

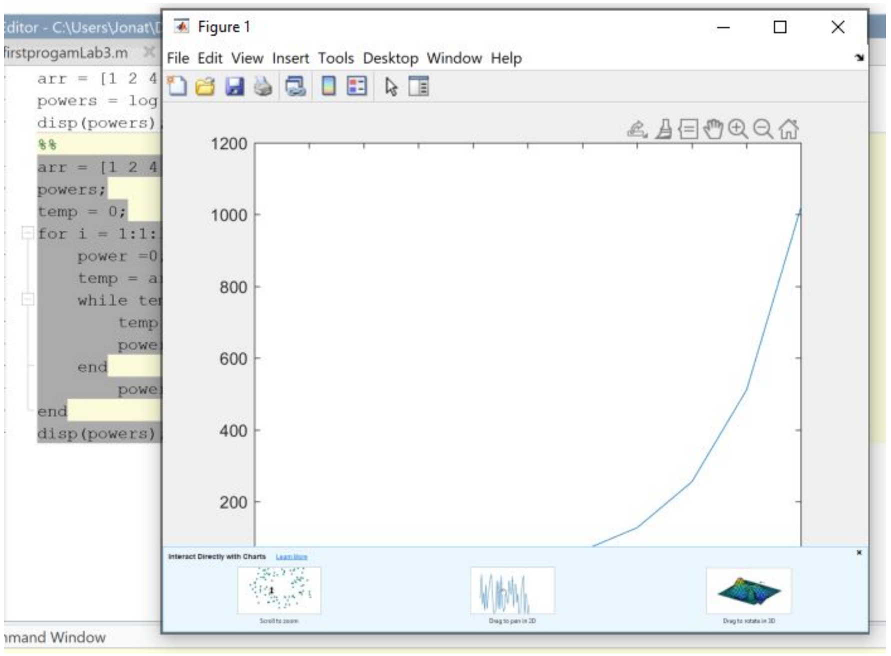
In the second loop it checks if the remainder of i value in the array and 2. This is because it will return 1 if there is a remainder (true so it is odd) or 0 if it is even (because there isn’t a remainder). It then adds it to the correct array and prints them out.

One neat thing I learned was that doing `disp("in order array" + array);` led it to print “in order array” + the value of all the parts. (as can be seen in the image below)

```
Columns 1 through 5

  "in order array4"    "in order array19"    "in order array39"    "in order array44"    "in order array45"

Columns 6 through 10

  "in order array49"    "in order array65"    "in order array71"    "in order array77"    "in order array80"

Column 11

  "in order array96"
```

Arrays and Plots

First Plot without any labels



```
arr = [1 2 4
powers = log
disp(powers)
%%
arr = [1 2 4
powers;
temp = 0;
for i = 1:1:
    power =0
    temp = a
    while te
        temp
        powe
    end
        powe
end
disp(powers)
```

Editor - C:\Users\Jonat\D
firstprogamLab3.m
Figure 1
File Edit View Insert Tools Desktop Window Help
1200
1000
800
600
400
200
Interact Directly with Charts
mmand Window

plot(arr,'Color',[0,0.7,0.9])

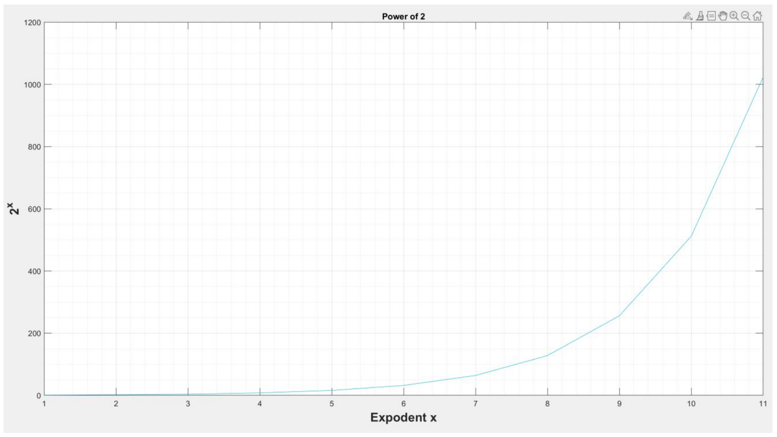
title('Power of 2');

grid on

grid minor

xlabel('Expodent x','FontSize', 16,'FontWeight','bold');

ylabel('2^x','FontSize', 16,'FontWeight','bold');



NOTE: I did this wrong because I didn’t use arr_exp, I learned while doing challenge 3

Challenge #3: Create an array (each) for the following: power of 3, power of 4, power of 5, and power of 10. Plot your results.

I wanted to have an array of arrays that would go through plotting all of them automatically but it wouldn't plot anything. This is because it only sets the first value equal to the array (I tried a 2d array as well).

```
allArrays = [array3,array4,array5,array10];
for i = 1:1:4
  plot(allArrays(i),'Color',[0,0.7,0.9])
...
```

The code bellow checks the first value in the array, it is a single number rather than the full array

```
disp(allArrays(1));
```

```
3
```

Here is the function that creates the arrays for the plotting

```
function [array] = arrayCreator(x)
  for i = 1:1:10
    array(i) = power(x,i);
  end
end
```

I could not get the plotting to work right, I used the code bellow

```
plot(array3,arr_exp,array4,arr_exp, array5,arr_exp, array10,arr_exp)
```

Results

Results

I learned a lot about how arrays work in matlab, they are very similar to how they work in other languages. A couple differences is that they start at 1 and they are broken into 2d by ; signs rather than nested brackets. One thing I really like about them is that it acts more like an ArrayList than it does a typical array, you can allocate more space for it as the program continues. The main thing I learned was how plots worked in matlab because I had almost no experience in the type of plotting except in my python class.

Conclusions and Reflection

I liked the plotting because it could be useful for showing lots of data but I don't know what I would personally use it for. I was still annoyed by matlab because of its weird differences. Because they use % for comments and not modulus I had a bit of trouble, also the inability to use "+=", "++", ect. is quite annoying. When it comes to plots, I could see how this would be useful for easily computing large amounts of data and then plotting them very similarly to python. I also found it frustrating that you couldn't put arrays inside of arrays. I need to work on them because I had a lot of trouble with the last problem because of the ploting.

Arrays Start at 1