ECE-103 LAB 1 CADEN ROBERTS

1. Create the vector `x = [1,2,...,100]`. Assign the even numbers of `x` to a new vector `y`.

Summary:

We create a vector x with values 1 - 100. We create a true/false array with true values corresponding to even indices and false values corresponding to odd indices. A vector y is created with the true/false array used to select only the even numbers from the vector x. We print the resulting vector y.

Results

```
>> lab1problem1
Values in y:
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52
 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98 100
```

Code:

```
x = 1:100; % creates a vector x with the numbers 1
- 100
y = x(mod(x,2)==0); % creates a t/f array where
true corresponds to even numbers and false
corresponds to odd numbers. x is then selected from
according to the true values, putting only even
numbers into vector y.
fprintf('Values in y:\n');
for i=1:1:50
   fprintf('%0.f ', y(i)); % print out vector y
   if i == 26
       fprintf('\n '); % newline for formatting
   end
end
fprintf('\n');
```

2. Use **for** loop to find the values of $x(t) = 3\cos(2\pi ft + 0.1)$ for `t = 0, 01, 0.2, 0.3, 0.4 s` when `f =10, 15,` and `20 Hz`. Use one set of statements to compute the values for all three frequencies and store the results in a two-dimensional array. Use two nested for loops and double indexing.

Summary:

We create a 2D vector and store results of the function for each t and f value combination inside it using a double for loop. We then use more for loops and logic to properly format the printed results.

Results:

```
>> lab1problem2
Results f vs t:
  t:       0.0         0.1         0.2         0.3         0.4
f:
10      2.9850      2.9850      2.9850      2.9850      2.9850
15      2.9850     -2.9850      2.9850     -2.9850      2.9850
20      2.9850      2.9850      2.9850      2.9850      2.9850
```

Code:

```
tval = 0:0.1:0.4; % Define time vals
fval = [10, 15, 20]; % Define freq vals
results = zeros(length(fval), length(tval)); % 2D array to store the results
for i = 1:length(fval)             % Compute for each freq, put in results
  f = fval(i);                     % load f value
  for j = 1:length(tval)
      t = tval(j);                 % load t value
      x = 3*cos(2*pi*f*t + 0.1);   % Compute x(t) for given f and t
      results(i, j) = x;           % Store the result in the results array
  end
end
fprintf('Results f vs t:\n  t:'); % use for loops/if's to print formatted
for i = 1:5
  fprintf('      %.1f ', tval(i));
end
fprintf('\nf:\n');
n = 1;
while n < 4
   fprintf('%.0f   ', fval(n));
   for i = n:3:(n+12)
      if results(i) > 0
          fprintf(' ');
      end
      fprintf(' %.4f  ', results(i));
   end
   n=n+1;
   fprintf('\n');
end
```

3. Use **while** loop to find the largest value of positive `t` for which $e^{1.2} \cos(\omega t)$ and t^3 are both less than `10`. Make the computation for ω`=35, 40,` and `45`. Find your answers to the nearest `0.01`.

Summary:
We find the max value of t by rounding the cube root of 10 to the nearest 0.01. We then find the max t-value for which the function is less than 10 for w = 35, 40, and 45. We store the t value and function value. Then, we display the W vals, max t vals, and max function vals.

Results:

```
>> lab1problem3
W vals:
   35   40   45

Max t vals:
  2.15  2.15  2.15

Corresponding func vals:
  3.28  -1.27  -2.66
```

Code:

```
wvals = [35, 40, 45]; % Define the values of w
funcvals = zeros(size(wvals)); % Initialize array to store the results
maxtvals = zeros(size(wvals)); % initialize array to store max t vals
for i = 1:length(wvals) % Iterate over each value of w
  w = wvals(i); % select value of w
  t = round((10^(1/3)) * 100) / 100; % Initialize t
  while t^3 >= 0 % loop until t is negative
      if (exp(1.2)*cos(w*t)) < 10 % if func is less than 10
          maxtvals(i)=t; % update max t val
          funcvals(i)=exp(1.2)*cos(w*t); % update func val
          break % exit while loop
      end
      t = t - 0.01; % decrement t if func is > than 10, loop again
  end
end
fprintf('W vals:\n   %.0f   %.0f   %.0f\n\nMax t vals:\n  %.2f  %.2f
%.2f\n\nCorresponding func vals:\n  %.2f  %.2f  %.2f\n', wvals(1),
wvals(2), wvals(3), maxtvals(1), maxtvals(2), maxtvals(3),
funcvals(1), funcvals(2), funcvals(3)); % print results formatted
```

4. Create a 15-element vector with values of $x(t) = 4\cos(2\pi t + 0.2) + 3\sin(\pi^2 t)$ at equally spaced interval $0 \le t \le 1$. Find the maximum element value, the minimum element value, the average of the element values, and the indices of the elements for which the element magnitude is greater than 4.

Summary:
We create a vector starting at 0 and ending at 1 spaced by 1/14's, and calculate all x(t) for the values. We find the max, min, and average as we do so, and we call find(abs()>4) to find the elements with magnitudes greater than 4. We display the results.

Results:

```
>> lab1problem4
Max element value: 5.5319
Min element value: -6.8464
Ave of element values: 0.7356
Indices with mag > 4: 2, 3, 6, 7, 8, 9, 12, 13, 14
```

Code:

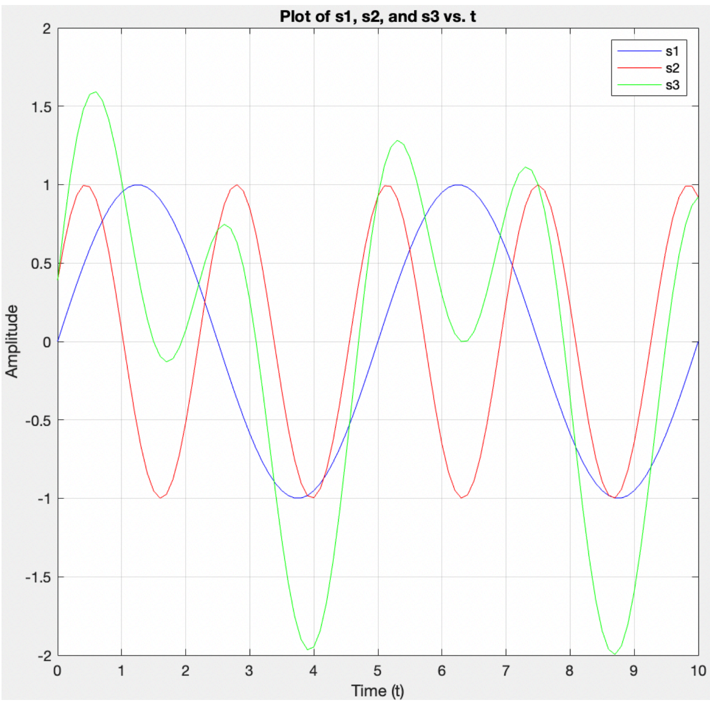
```
t = 0:(1/14):1; % initialize 15 even spaced t vals 0<=t<=1
x = 4*cos(2*pi*t + 0.2) + 3*sin(pi^2*t); % Compute vals of x(t)
xmax = -1000; % initialize vars for min max and ave
xmin = 1000;
xave = 0;
for i=1:length(x) % Calculate results
  if xmax < x(i) % if xmax is less than x(i), update
      xmax = x(i);
  end
  if xmin > x(i) % if xmin is greater than x(i), update
      xmin = x(i);
  end
  xave = xave + x(i); % add each x(i) to xave
end
xave = xave / 15; % divide sum of x(i)'s by 15 to get average
fprintf('Max element value: %.4f\nMin element value: %.4f\nAve of
element values: %.4f\nIndices with mag > 4:', xmax, xmin, xave); %
format results
indices=find(abs(x)>4); % find indices of elements with mag > 4
for i=1:1:(length(indices)-1) % print each index
   fprintf(' %.0f,', indices(i))
end
fprintf(' %.0f\n', indices(length(indices))); % last index and newline
```

5. Assume $s_1 = \sin(2\pi f_1 t)$, $s_2 = \sin(2\pi f_2 t + 0.4)$ and $s_3 = s_1 + s_2$, where $f_1 = 0.2$ and $f_2 = 0.425$. Plot s_1, s_2 and s_3 v/s t with $t = 0 : 0.1 : 10$ on the same graph (you have to use `hold on` command). Label the axes and create legends for each graph.

Summary:

Using our $f_1 = 0.2$ and $f_2 = 0.425$, we iterate t from 0 to 10 in increments of 0.1 and graph s1, s2, and s3 on the same graph in blue, red, and green. We create labels, a title, and legend for the graph as well.

Results:



Code:

```
f1 = 0.2; % Define the vals f1, f2, t
f2 = 0.425;
t = 0:0.1:10;
s1 = sin(2*pi*f1*t); % Calculate s1, s2, and s3
s2 = sin(2*pi*f2*t + 0.4);
s3 = s1 + s2;
plot(t, s1, 'b'); % plot s1 in blue
hold on; % Allow multiple plots on the same graph
plot(t, s2, 'r'); % plot s2 in red
plot(t, s3, 'g'); % plot s3 in green
xlabel('Time (t)'); % label x axis
ylabel('Amplitude'); % label y axis
title('Plot of s1, s2, and s3 vs. t'); % create title
legend('s1', 's2', 's3'); % create legend
grid on; % add a grid
hold off; % stop allowing multiple plots on the same graph
```

6. Sinc function is a function that arises frequently in our course. It is defined as

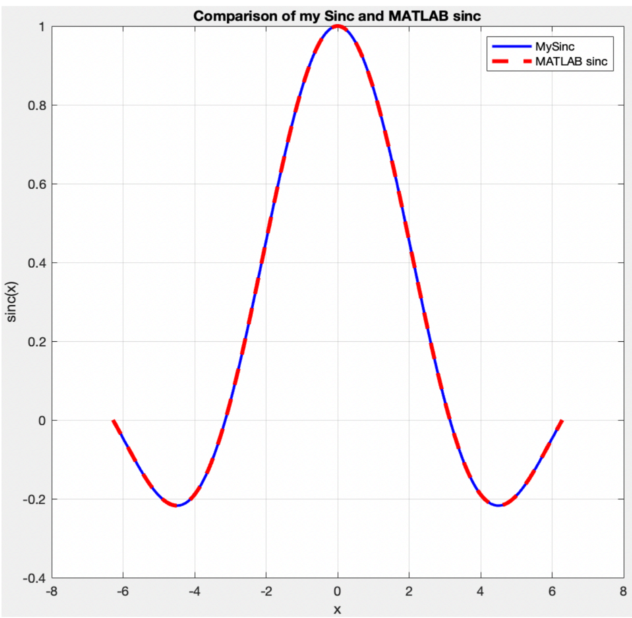
$$\text{sinc}(x) = \begin{cases} 1 & \text{for } x=0 \\ \sin(x)/x & \text{otherwise} \end{cases}$$

Create a Matlab function `MySinc ()` that defines sinc(x) function following the above definition. Plot the value sinc(x) in the interval $[-2\pi \;\; 2\pi]$ using `MySinc ()` function and Matlab inbuilt `sinc()` function on the same graph.

Summary:

We define the function as described above and plot through the interval $-2\pi \text{ and } 2\pi$, and plot the Matlab sinc() as well. Matlab's is in dashed red and my user defined function is in solid blue. We create labels, a title, and legend for the graph as well.

Results:



Code:

```
function y = Sinc(x)
   y = sin(x) ./ x; % Define the sinc function using the given definition
   y(x == 0) = 1;   % Replace with 1 for x = 0
end
x = linspace(-2*pi, 2*pi, 1000); % Define the range of x values
mysinc = Sinc(x); % Calculate the values of my Sinc(x)
matlabsinc = sinc(x/pi); % MATLAB's function is defined sin(pi*x) / (pi*x)
plot(x, mysinc, 'b', 'LineWidth', 2); % mysinc blue width 2
hold on; % allow multiple plots on same graph
plot(x, matlabsinc, 'r--', 'LineWidth', 3); % matlabsinc dashed red width 3
xlabel('x'); % label x axis x
ylabel('sinc(x)'); % label y axis sinc(x)
title('Comparison of my Sinc and MATLAB sinc'); % create title
legend('MySinc', 'MATLAB sinc'); % create legend
grid on; % add grid
hold off; % don't allow more plots on same graph
```