Question 3

```
b=ones(3,2) %b creates and stores an 3*2 matrix of ones
```

```
b = 3×2
     1     1
     1     1
     1     1
```

```
c=size(b); %c Stores the row then column (3 then 2) in
%a row vector and supresses the value from the output

%Takes the absolute value of the 1*2 row vector [5.2 3]
%prints it to the output
abs([-5.2 , 3])
```

```
ans = 1×2
    5.2000    3.0000
```

```


floor(3.6) %Rounds 3.6 down to 3 and prints it to the output.
```

```
ans = 3
```

```
d=[1:-3.5:-9]; %Assign, creates, and supress the output
%of d to be a row vector from 1 to 9 decrementing by 3.5
f=d(2); g=sin(pi/2); %Assign, creates, and suppress the
%output of f to be the second element of d and g to be sin(pi/2)
K=[1.4, 2.3; 5.1, 7.8]; %Assign, creates, and suppress the
%output of K to be a 2*2 matrix of elements 1.4, 2.3, 5.1, 7.8
m=K(1,2); %Assign, creates, and suppress the output of m to be the
%element in row 1 column 2 of the matrix
n=K(:,2); %Assign, creates, and suppress the output of n to be the
%2nd column vector at K
comp = 3+4i; %Assign, creates, and suppress the output of comp
%to be the complex rectangular codinates of 3 + 4i
real(comp) %Outputs the real value the complex rectangular cordinate (3)
```

```
ans = 3
```

```
imag(comp) %Outputs the imaginary value of the complex rectangular cordinates (4)
```

```
ans = 4
```

```
%Outputs the magnitude of the complex rectangular codinates
%from the origin [squrt(3^2+4^2)]
abs(comp)
```

```
ans = 5
```

```
%Outputs the phase angle of the complex rectangular cordinate phasor in radians
angle(comp)
```

```
ans = 0.9273
```

```
disp('haha, MATLAB is fun'); %Prints the string 'haha, MATLAB is fun'
```

```
haha, MATLAB is fun
```

```
3^2 %Outputs 3*3, n "to the power of" (^) m
```

```
ans = 9
```

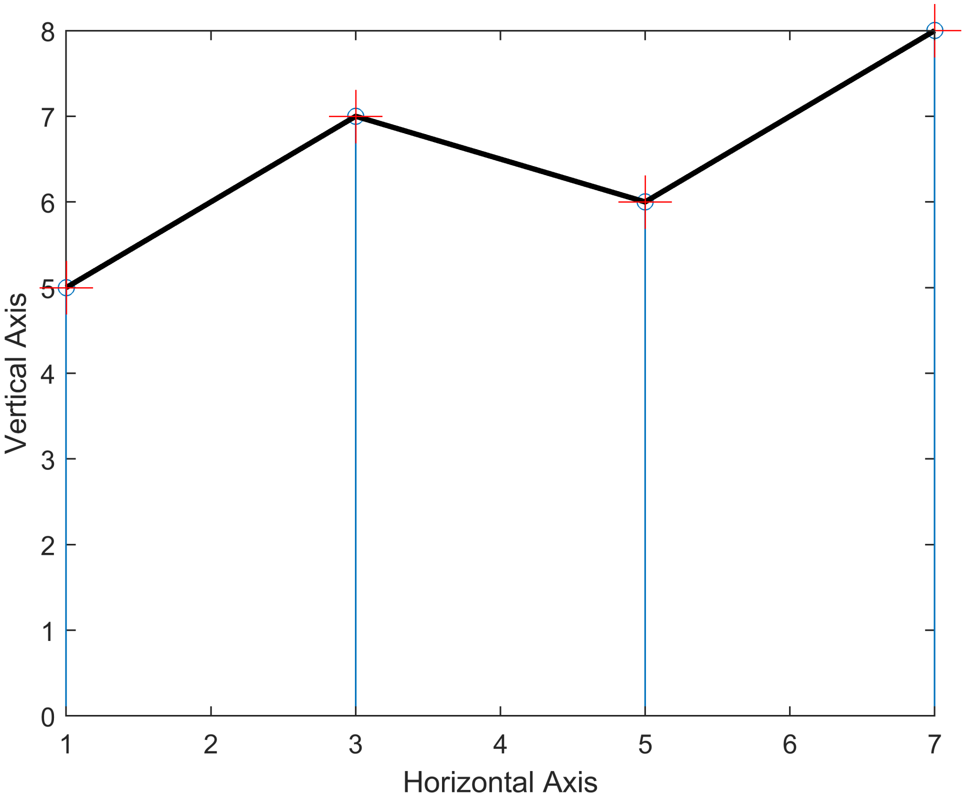
```
%Outputs 1, or true, if two values are the same when using "==",
%else the output is false. 4 is equal to 4.
4==4
```

```
ans = logical
   1
```

```


%Outputs a row vector of booleans element 1 is to check
%if something is equl %(==) while the other (~=) checks if something is not
[2==8 3 ~= 5]
```

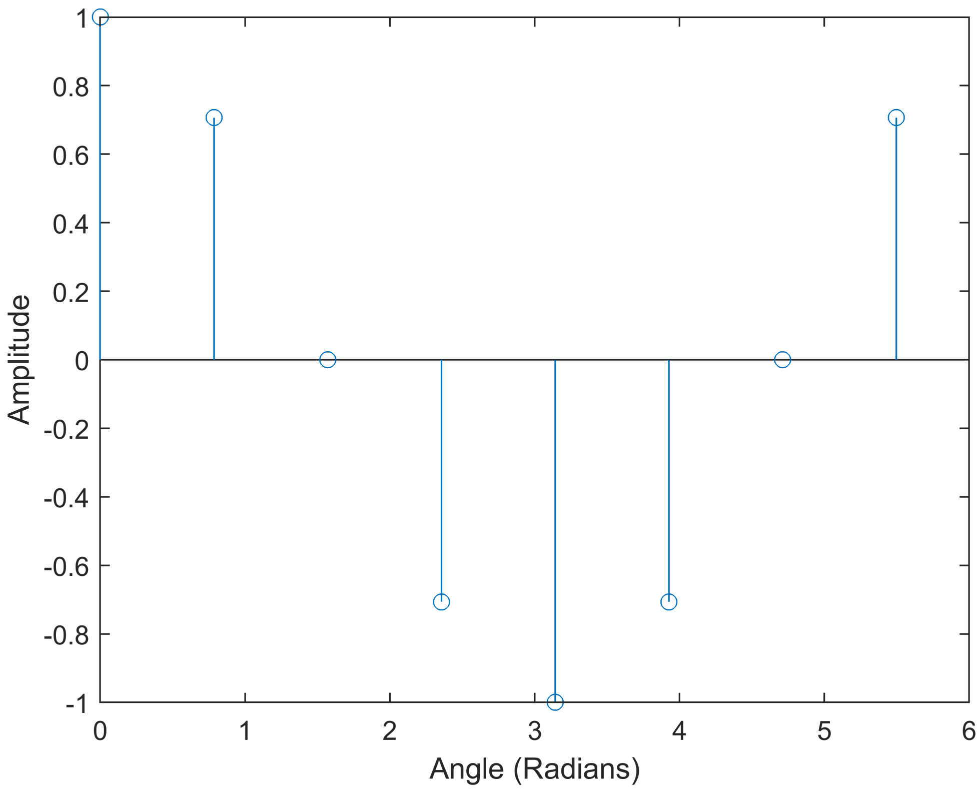
```
ans = 1×2 logical array
   0   1
```

```

x=[1:2:8]; %Assign, creates, and supress the output of x to be a row
%vector from 1 to 8 incrementing by 2
y=[5 7 6 8]; %Assign, creates, and supress the output of y to be a row vector
%containing the elements [5,7,6,8]
q = zeros(10,1); %Assign, creates, and supress the output of q to be a vector
%of zeroes with 10 zero elements in a column
for ii = 1:10 %Creates a loop that will iterate n:m, or n to m times, or 1 to
%10 times.
q(ii) = ii^2; %Changes the element in q at element ii in the loop to be
%the square of ii
end %ends for loop
figure(1021); %creates a figure window with it being the nth or 1021 figure
stem(x,y) % plots the data of y at the values specified at x
hold on; %sets the next plot point of the current figure and axes to add

%Plots x and y, Add a black line to the plot width of 2 points
plot(x,y, 'k', 'linewidth', 2)
%Plots x and y, Adds a red plus shaped marker to the plot
plot(x,y,'+r', 'markersize', 20);

hold off; %sets the next plot property of the current axes to "replace"
xlabel('Horizontal Axis') %labels the x axis
ylabel('Vertical Axis') %labels the y axis
```

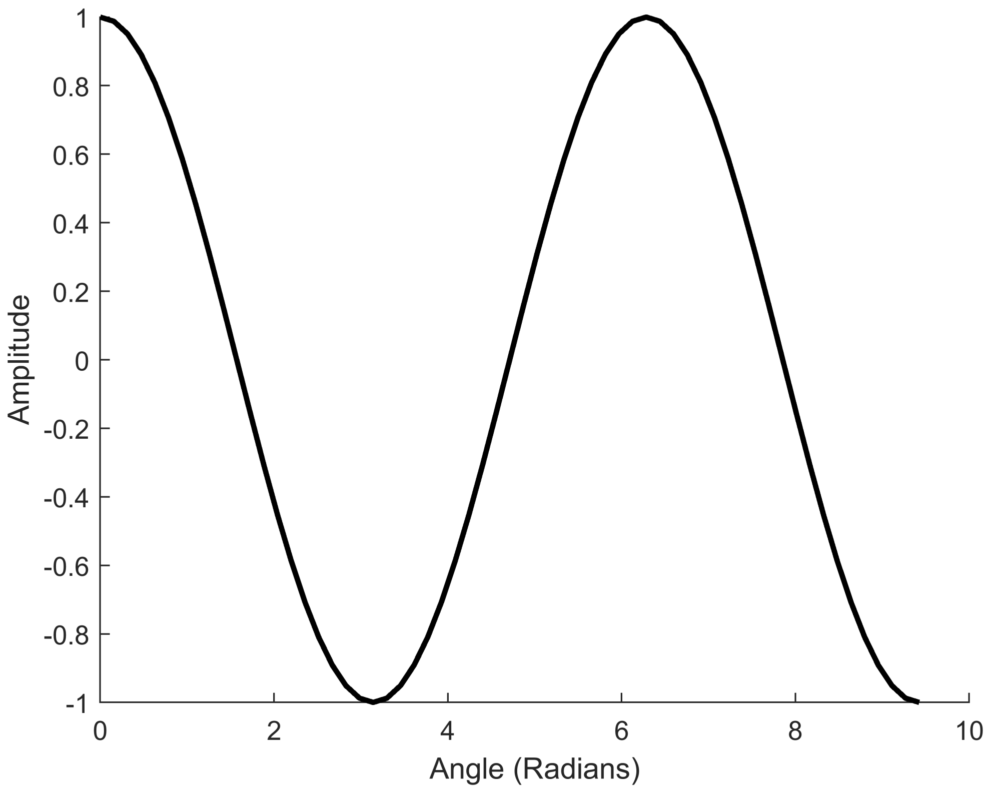


Question 4

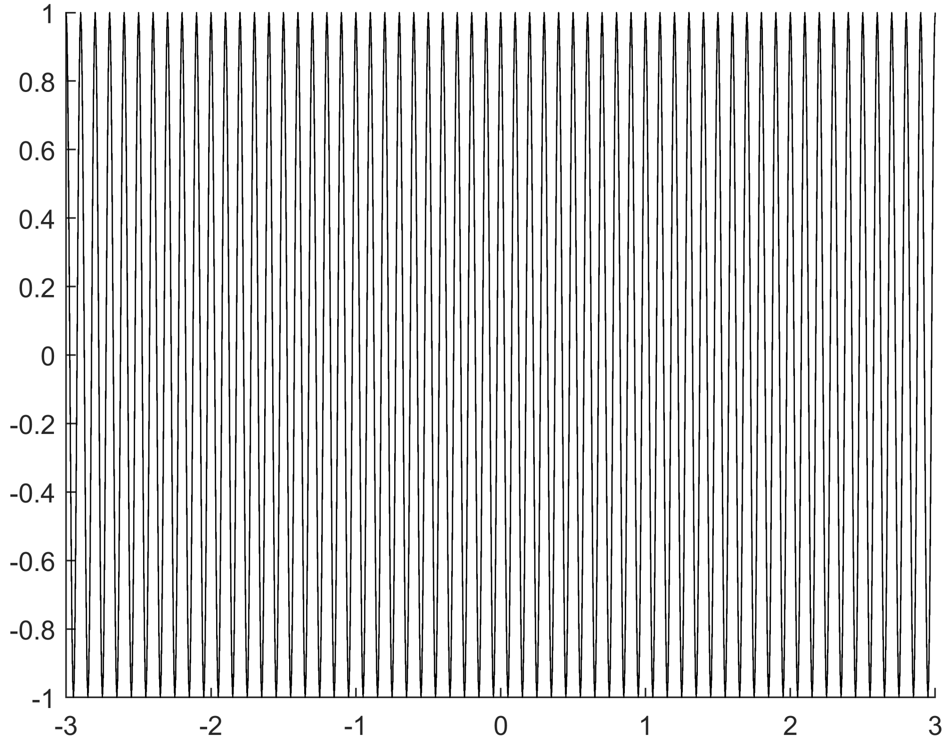
```
%a
vect1 = [0 pi/4 2*pi/4 3*pi/4 4*pi/4 5*pi/4 6*pi/4 7*pi/4];
vect2 = cos(vect1);
figure(1);
stem(vect1,vect2);
xlabel('Angle (Radians)')
ylabel('Amplitude')
```



```
%b
theta = 0:pi/20:3*pi;
y = cos(theta);
figure(5);
hold on
plot(theta, y, 'k', 'linewidth', 2);
hold off
xlabel('Angle (Radians)');
ylabel('Amplitude');
```



```
%c
t = -3:0.005:3;
x = cos(20*pi*t);
figure(19);
hold on
plot(t,x,'k', 'linewidth', 0.5);
axis([-3 3 -1 1]);
hold off
```



Question 5

```
%a
type myroots.m
```

```
function r = myroots(n,a)
%myroots: Find all the nth roots of a complex number
%
%Input Args:
%   n: a positive integer specifying the nth root
%   a: a complex number whose nth roots are to be returned
%
%Outout
%   r: 1xn vector containing all the nth roots of a

r = zeros(1, n);%Intializes 1xn matrix of zeros
%converts the real and imaginary parts of a to polar
[phi, A] = cart2pol(real(a), imag(a));
aPolar = (A^(1/n))*exp((1i*phi)/n);

%solves each root of a up to the nth root adding 2*pi each iteration
for k = 1:n
    r(1,k) = ((aPolar))*exp((1i*2*pi*(k-1))/n);
end

end
```

```


%b
help myroots
```

```
myroots: Find all the nth roots of a complex number

Input Args:
   n: a positive integer specifying the nth root
   a: a complex number whose nth roots are to be returned

Outout
   r: 1xn vector containing all the nth roots of a
```

```
%The help command gives a synopsis of the function; for "help myroots", it states that it is a
% and puts the pseudo code for the function that is commented before the first line of code.
%For the built in functions in matlab like "help cart2pol", it does the same output as in lists
%a synopsis for the functions:it includes the input arguments, outputs, overloads/overrides,
%and what the function does.

%c
disp('9th roots of 2: ')
```

```
9th roots of 2:
```

```
disp(myroots(9,2))
```

```
  Columns 1 through 5

   1.0801 + 0.0000i   0.8274 + 0.6942i   0.1876 + 1.0637i  -0.5400 + 0.9354i  -1.0149 + 0.3694i

  Columns 6 through 9

  -1.0149 - 0.3694i  -0.5400 - 0.9354i   0.1876 - 1.0637i   0.8274 - 0.6942i
```

```
disp('23rd roots of -j: ')
```

```
23rd roots of -j:
```

```
disp(myroots(23,-1i))
```

```
  Columns 1 through 5

   0.9977 - 0.0682i   0.9791 + 0.2035i   0.8879 + 0.4601i   0.7308 + 0.6826i   0.5196 + 0.8544i

  Columns 6 through 10

   0.2698 + 0.9629i   0.0000 + 1.0000i  -0.2698 + 0.9629i  -0.5196 + 0.8544i  -0.7308 + 0.6826i

  Columns 11 through 15

  -0.8879 + 0.4601i  -0.9791 + 0.2035i  -0.9977 - 0.0682i  -0.9423 - 0.3349i  -0.8170 - 0.5767i

  Columns 16 through 20

  -0.6311 - 0.7757i  -0.3984 - 0.9172i  -0.1362 - 0.9907i   0.1362 - 0.9907i   0.3984 - 0.9172i

  Columns 21 through 23

   0.6311 - 0.7757i   0.8170 - 0.5767i   0.9423 - 0.3349i
```