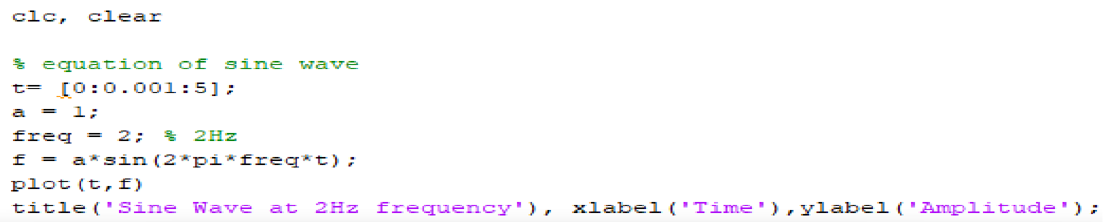
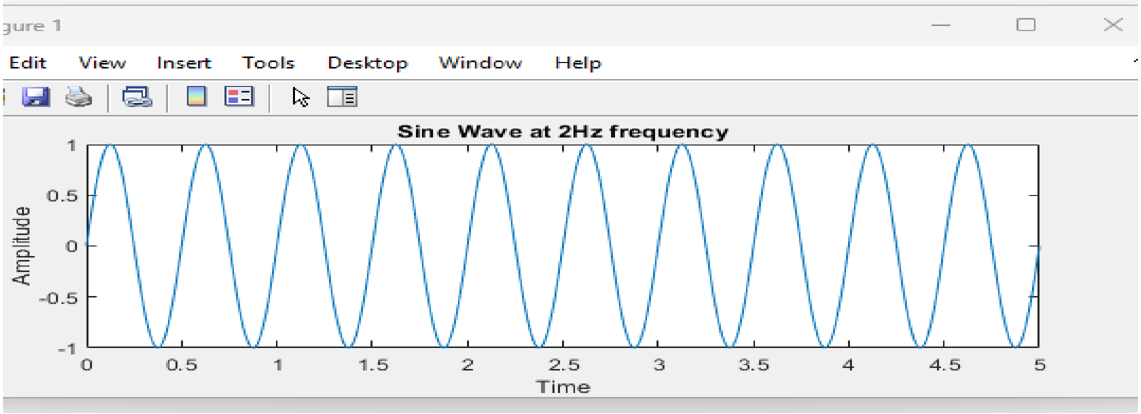
Lab 1 Exercises

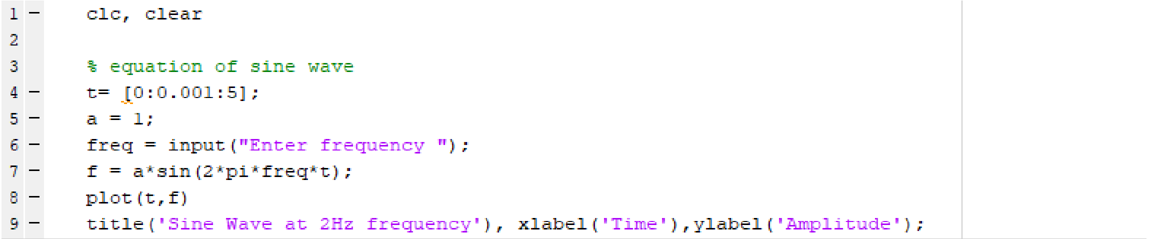
Task # 1: Generate a script file which produces a sine wave with 2 Hz frequency.



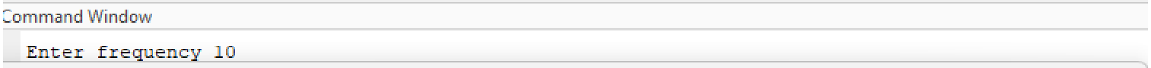
```
clc, clear

% equation of sine wave
t= [0:0.001:5];
a = 1;
freq = 2; % 2Hz
f = a*sin(2*pi*freq*t);
plot(t,f)
title('Sine Wave at 2Hz frequency'), xlabel('Time'),ylabel('Amplitude');
```


gure 1
Edit View Insert Tools Desktop Window Help
Sine Wave at 2Hz frequency
Amplitude
1
0.5
0
-0.5
-1
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
Time

Task # 2: Generate a script file which produces a sine wave by taking frequency as an input at run time.

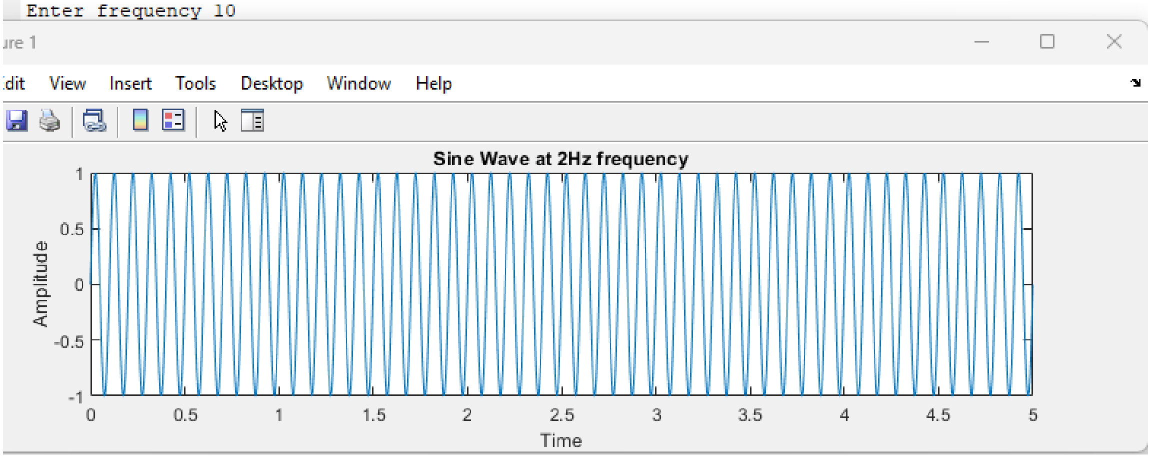


```
clc, clear

% equation of sine wave
t= [0:0.001:5];
a = 1;
freq = input("Enter frequency ");
f = a*sin(2*pi*freq*t);
plot(t,f)
title('Sine Wave at 2Hz frequency'), xlabel('Time'),ylabel('Amplitude');
```

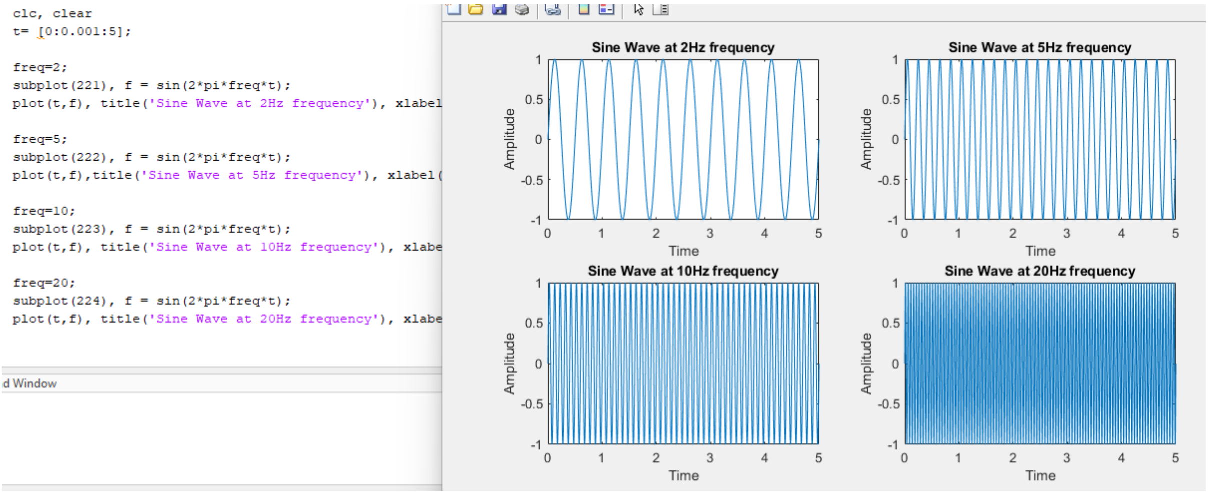


Command Window

```
Enter frequency 10
```


ure 1
dit View Insert Tools Desktop Window Help
Sine Wave at 2Hz frequency
Amplitude
1
0.5
0
-0.5
-1
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
Time

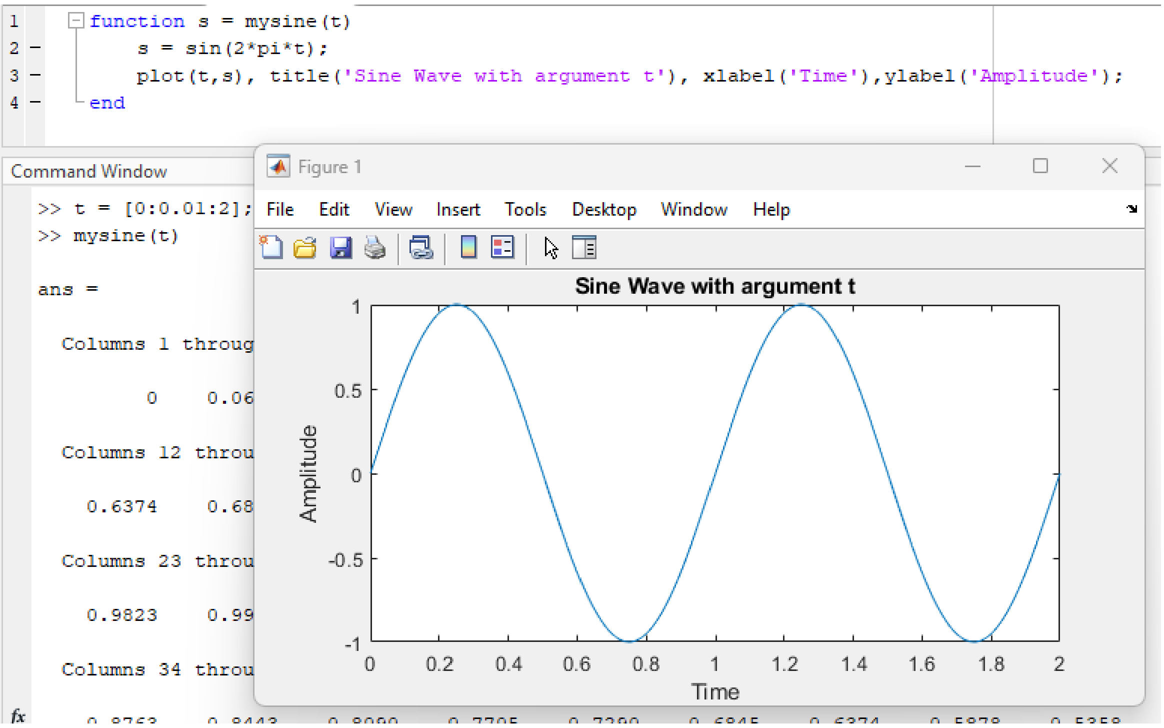
Task # 3: Generate a script file which produces a sine wave with 2Hz, 5Hz, 10Hz and 20Hz frequencies on a single figure by using subplot command. Label the subplots with the relevant frequency to distinguish the signals clearly.



```
clc, clear
t= [0:0.001:5];

freq=2;
subplot(221), f = sin(2*pi*freq*t);
plot(t,f), title('Sine Wave at 2Hz frequency'), xlabel

freq=5;
subplot(222), f = sin(2*pi*freq*t);
plot(t,f),title('Sine Wave at 5Hz frequency'), xlabel(

freq=10;
subplot(223), f = sin(2*pi*freq*t);
plot(t,f), title('Sine Wave at 10Hz frequency'), xlabe

freq=20;
subplot(224), f = sin(2*pi*freq*t);
plot(t,f), title('Sine Wave at 20Hz frequency'), xlabe
```

d Window
Sine Wave at 2Hz frequency
Sine Wave at 5Hz frequency
Sine Wave at 10Hz frequency
Sine Wave at 20Hz frequency
Amplitude
Time
1
0.5
0
-0.5
-1
0
1
2
3
4
5

Task # 4: Implement a function as defined below, which produces a sine wave accepting time as argument.

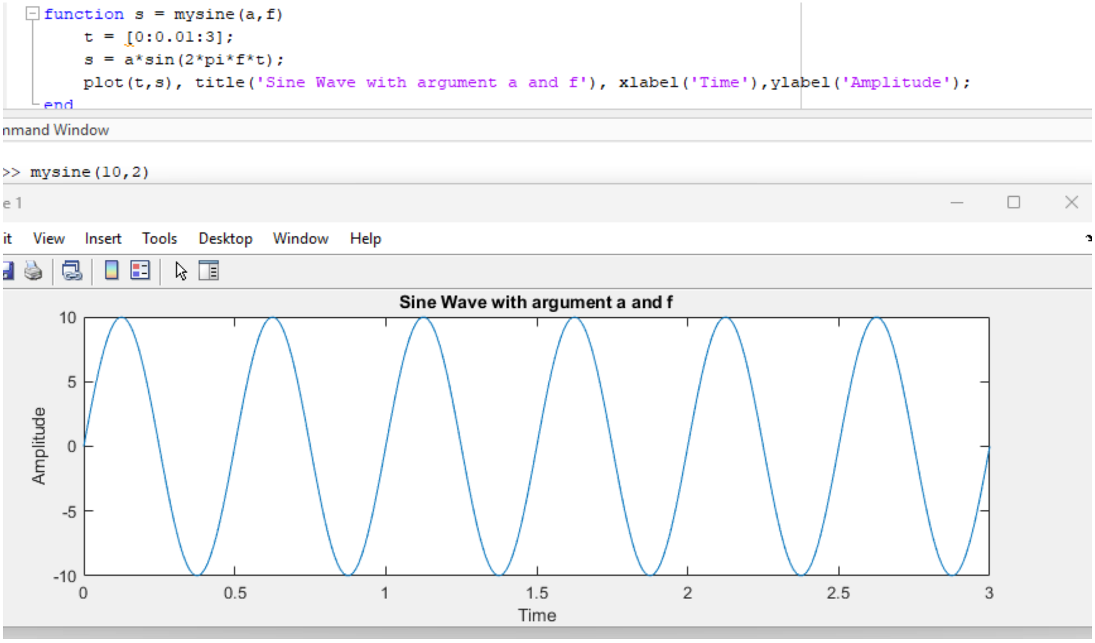


```
function s = mysine(t)
    s = sin(2*pi*t);
    plot(t,s), title('Sine Wave with argument t'), xlabel('Time'),ylabel('Amplitude');
end
```

Command Window

```
>> t = [0:0.01:2];
>> mysine(t)

ans =

  Columns 1 throug

         0    0.06

  Columns 12 throu

    0.6374    0.68

  Columns 23 throu

    0.9823    0.99

  Columns 34 throu

    0.8763    0.8443    0.8090    0.7705    0.7290    0.6845    0.6374    0.5878    0.5358
```

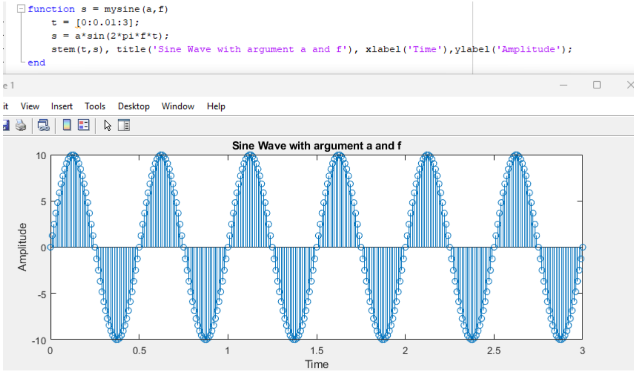
Task # 5: Generate a function which produces a sine wave by accepting frequency and amplitude as input arguments and return a plot of sine wave for the inputted parameters.



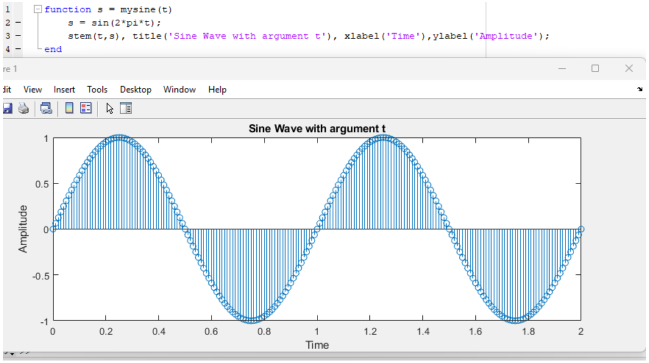
```
function s = mysine(a,f)
    t = [0:0.01:3];
    s = a*sin(2*pi*f*t);
    plot(t,s), title('Sine Wave with argument a and f'), xlabel('Time'),ylabel('Amplitude');
end
```

```
>> mysine(10,2)
```

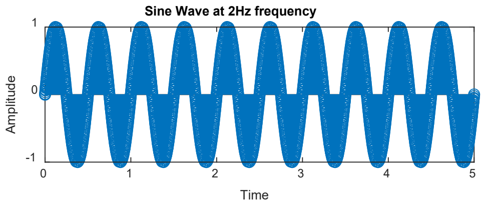
Task # 6: Repeat all above tasks by using stem command rather than plot command.

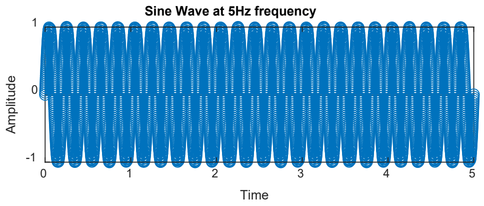


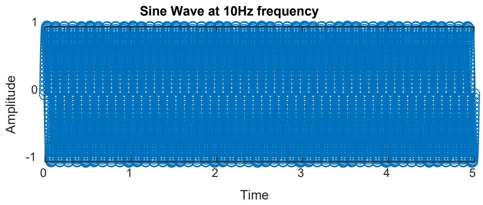
```
function s = mysine(a,f)
    t = [0:0.01:3];
    s = a*sin(2*pi*f*t);
    stem(t,s), title('Sine Wave with argument a and f'), xlabel('Time'),ylabel('Amplitude');
end
```

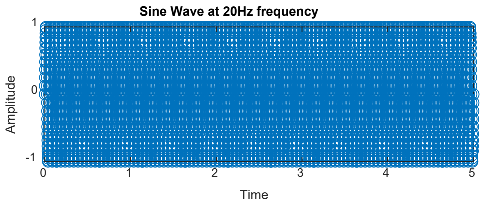


```
function s = mysine(t)
    s = sin(2*pi*t);
    stem(t,s), title('Sine Wave with argument t'), xlabel('Time'),ylabel('Amplitude');
end
```

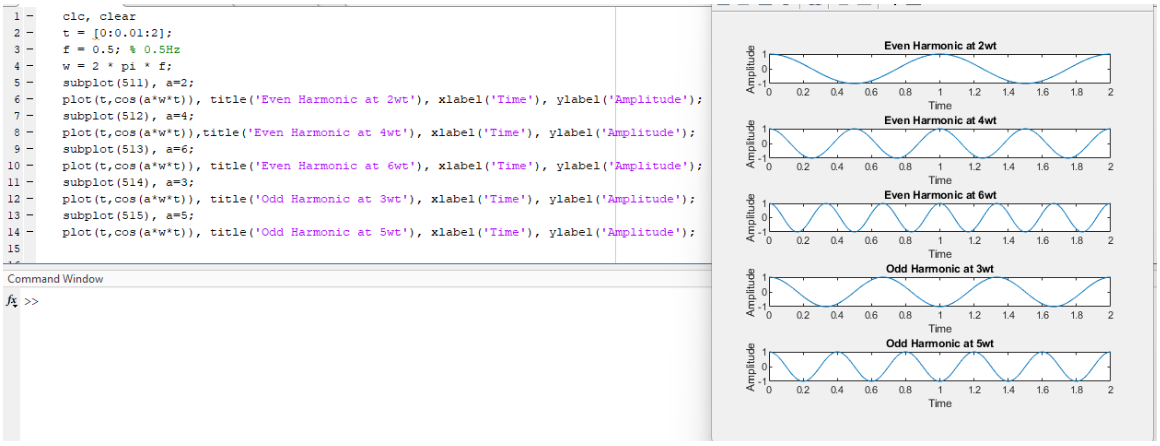








Task # 7: Write MATLAB function for plotting the 5 harmonics of a signal x(t)=cos(wt) with a fundamental frequency of f = 0.5 Hz. Take t = 0:0.01:2

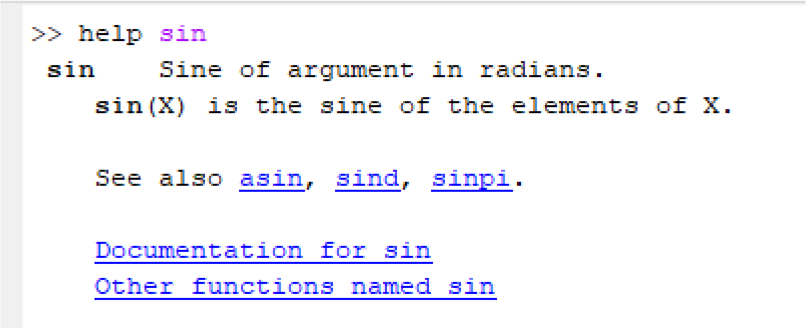


```
clc, clear
t = [0:0.01:2];
f = 0.5; % 0.5Hz
w = 2 * pi * f;
subplot(511), a=2;
plot(t,cos(a*w*t)), title('Even Harmonic at 2wt'), xlabel('Time'), ylabel('Amplitude');
subplot(512), a=4;
plot(t,cos(a*w*t)),title('Even Harmonic at 4wt'), xlabel('Time'), ylabel('Amplitude');
subplot(513), a=6;
plot(t,cos(a*w*t)), title('Even Harmonic at 6wt'), xlabel('Time'), ylabel('Amplitude');
subplot(514), a=3;
plot(t,cos(a*w*t)), title('Odd Harmonic at 3wt'), xlabel('Time'), ylabel('Amplitude');
subplot(515), a=5;
plot(t,cos(a*w*t)), title('Odd Harmonic at 5wt'), xlabel('Time'), ylabel('Amplitude');
```

```
Command Window
>>
```

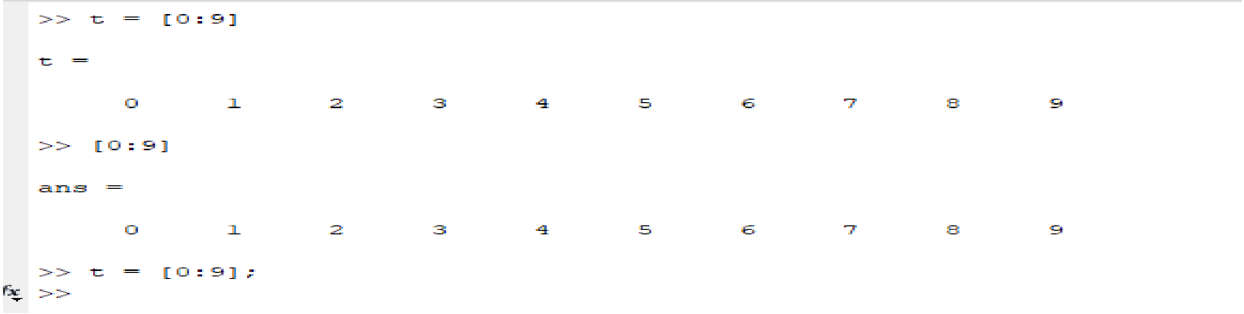
	Harmonic #1	Harmonic #2	Harmonic #3	Harmonic #4	Harmonic #5
Time period:	1	0.5	0.3333	0.6667	0.4
Freq:	1	2	3	1.5	2.5

Task # 8: Try all help methods to get help of any function.



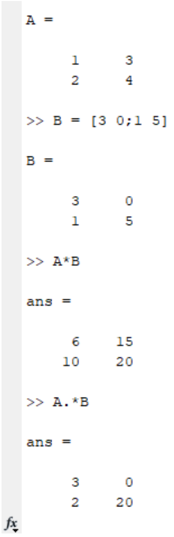
```
>> help sin
 sin    Sine of argument in radians.
    sin(X) is the sine of the elements of X.

    See also asin, sind, sinpi.

    Documentation for sin
    Other functions named sin
```

Task # 9: (a) Assign 10 samples, from 0 to 9, of time to the vector t.
(b) Assign a vector of samples without assigning it to a variable.
(c) Assign 10 samples, from 0 to 9, of time to any vector without printing it to screen.



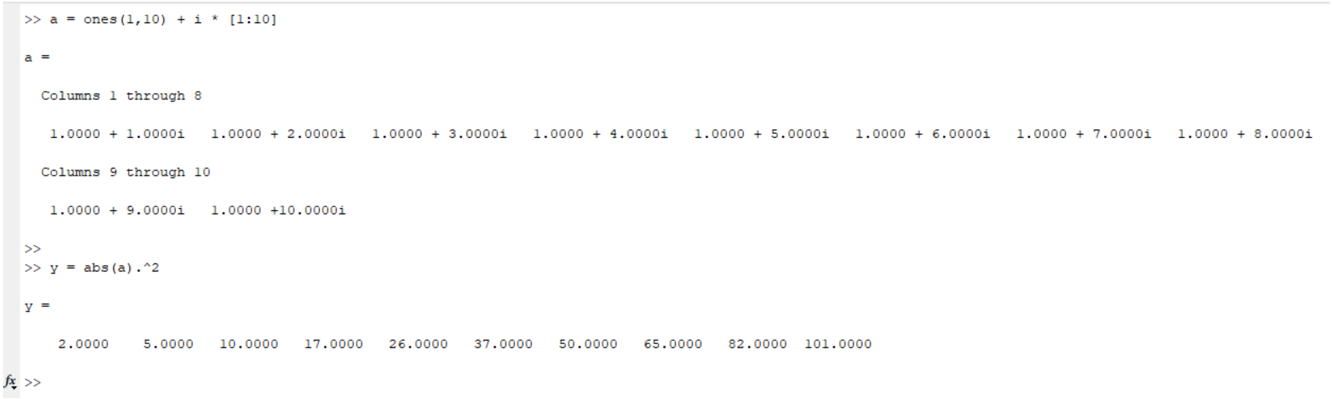
```
>> t = [0:9]

t =

     0     1     2     3     4     5     6     7     8     9

>> [0:9]

ans =

     0     1     2     3     4     5     6     7     8     9

>> t = [0:9];
>>
```

Task # 10: Investigate the difference between multiplication*and element-wise multiplication.* of vectors/matrices.



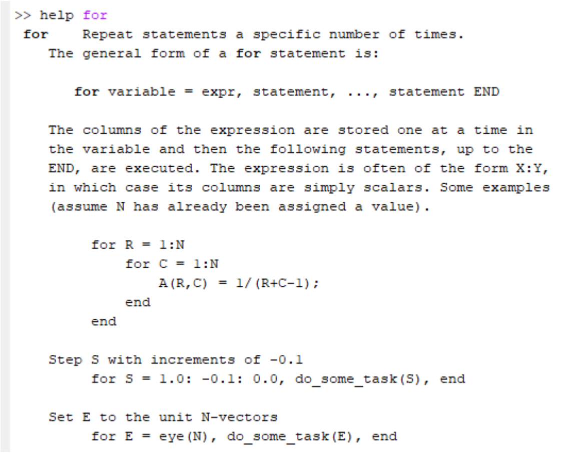
```
A =

     1     3
     2     4

>> B = [3 0;1 5]

B =

     3     0
     1     5

>> A*B

ans =

     6    15
    10    20

>> A.*B

ans =

     3     0
     2    20
```

Task # 11: Generate a complex-valued matrix a = ones (1,10)+ i* (1 : 10) and calculate the absolute square of all elements of this matrix.

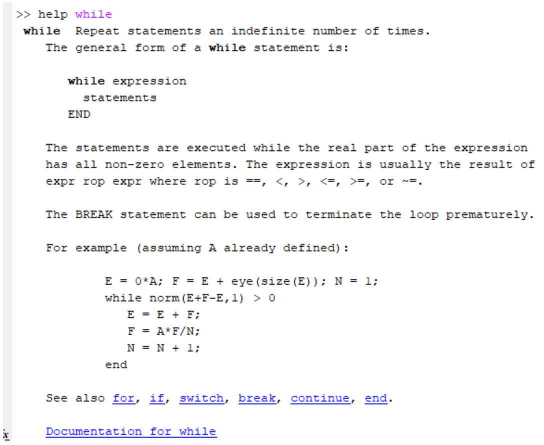


```
>> a = ones(1,10) + i * [1:10]

a =

  Columns 1 through 8

   1.0000 + 1.0000i   1.0000 + 2.0000i   1.0000 + 3.0000i   1.0000 + 4.0000i   1.0000 + 5.0000i   1.0000 + 6.0000i   1.0000 + 7.0000i   1.0000 + 8.0000i

  Columns 9 through 10

   1.0000 + 9.0000i   1.0000 +10.0000i

>>
>> y = abs(a).^2

y =

    2.0000    5.0000   10.0000   17.0000   26.0000   37.0000   50.0000   65.0000   82.0000  101.0000

>>
```

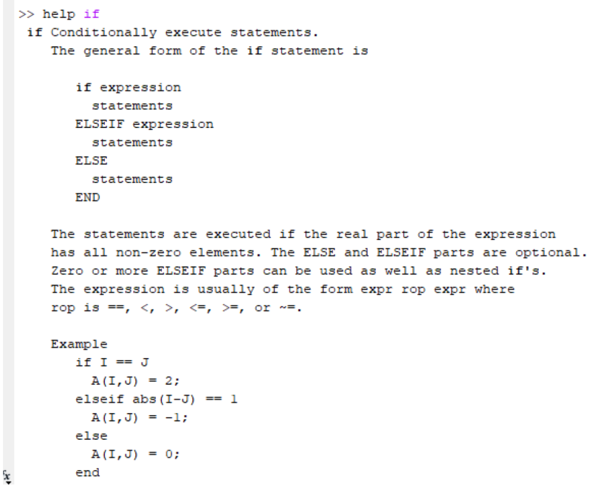
Task # 13: Use MATLAB help to get familiar with the syntax of FOR, WHILE and IF-ELSE statement.



```
>> help for
 for    Repeat statements a specific number of times.
    The general form of a for statement is:
 
       for variable = expr, statement, ..., statement END
 
    The columns of the expression are stored one at a time in
    the variable and then the following statements, up to the
    END, are executed. The expression is often of the form X:Y,
    in which case its columns are simply scalars. Some examples
    (assume N has already been assigned a value).
 
         for R = 1:N
             for C = 1:N
                 A(R,C) = 1/(R+C-1);
             end
         end
 
    Step S with increments of -0.1
         for S = 1.0: -0.1: 0.0, do_some_task(S), end
 
    Set E to the unit N-vectors
         for E = eye(N), do_some_task(E), end
```

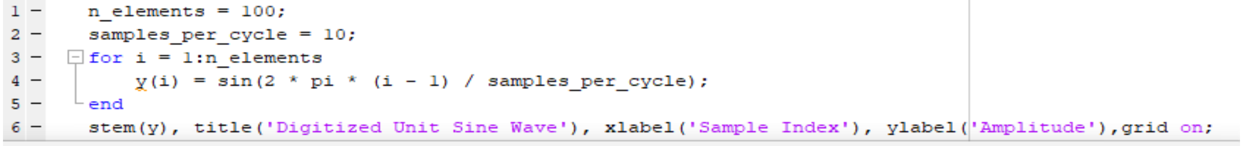


```
>> help while
 while  Repeat statements an indefinite number of times.
    The general form of a while statement is:
 
       while expression
         statements
       END
 
    The statements are executed while the real part of the expression
    has all non-zero elements. The expression is usually the result of
    expr rop expr where rop is ==, <, >, <=, >=, or ~=.
 
    The BREAK statement can be used to terminate the loop prematurely.
 
    For example (assuming A already defined):
 
            E = 0*A; F = E + eye(size(E)); N = 1;
            while norm(E+F-E,1) > 0
               E = E + F;
               F = A*F/N;
               N = N + 1;
            end
 
    See also for, if, switch, break, continue, end.

    Documentation for while
```



```
>> help if
 if Conditionally execute statements.
    The general form of the if statement is
 
       if expression
         statements
       ELSEIF expression
         statements
       ELSE
         statements
       END
 
    The statements are executed if the real part of the expression
    has all non-zero elements. The ELSE and ELSEIF parts are optional.
    Zero or more ELSEIF parts can be used as well as nested if's.
    The expression is usually of the form expr rop expr where
    rop is ==, <, >, <=, >=, or ~=.
 
    Example
       if I == J
         A(I,J) = 2;
       elseif abs(I-J) == 1
         A(I,J) = -1;
       else
         A(I,J) = 0;
       end
```

Task # 14: Use FOR loop to generate a single vector y, which is a digitized unit sine wave with ten samples per cycles, with 100 elements.



```
n_elements = 100;
samples_per_cycle = 10;
for i = 1:n_elements
    y(i) = sin(2 * pi * (i - 1) / samples_per_cycle);
end
stem(y), title('Digitized Unit Sine Wave'), xlabel('Sample Index'), ylabel('Amplitude'),grid on;
```

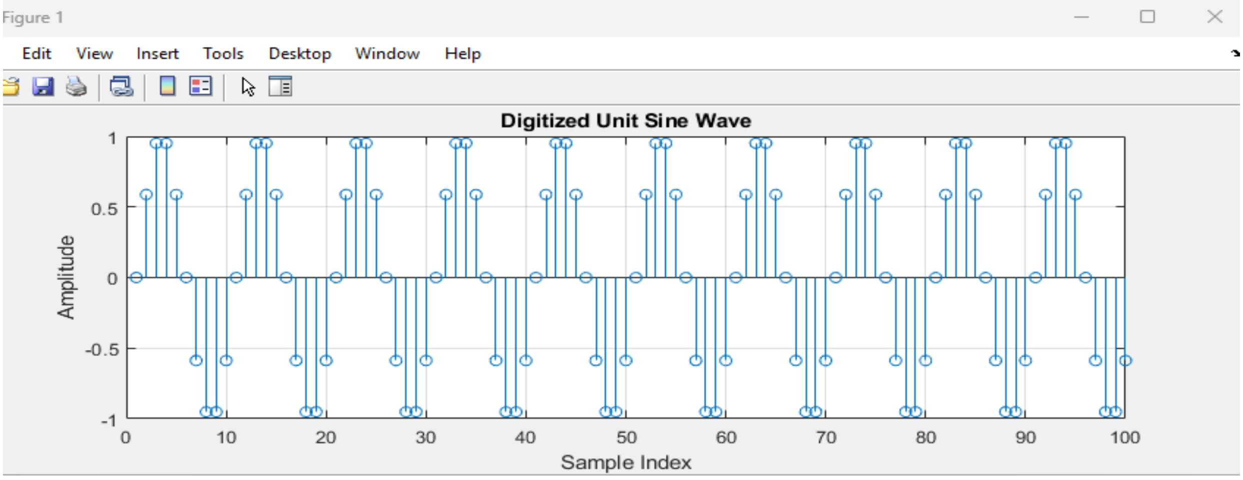


Figure 1
Edit
View
Insert
Tools
Desktop
Window
Help
Digitized Unit Sine Wave
Amplitude
Sample Index