TASK-1

Write a program to assign the following expressions to a variable A and then to print out the value of A.

a) $\frac{3+4}{5+6}$

b) $2\pi^2$

c) $\sqrt{2}$

d) $(0.0000123 + 5.67\times10^{-3}) \times 0.4567\times10^{-4}$

Program:

```
A = (3+4)/(5+6);
disp(A);
A=2*pi*pi;
disp(A);
A=sqrt(2);
disp(A);
A=(0.0000123+5.67E-3)*0.4567E-4;
disp(A);
```

Results:

Command Window:

```
0.6364

19.7392

1.4142

2.5951e-07
```

TASK-2

Celsius temperatures can be converted to Fahrenheit by multiplying by 9, dividing by 5, and adding 32. Assign a variable called C the value 37, and implement this formula to assign a variable F the Fahrenheit equivalent of 37 Celsius

Program:

```
C=37;
F=9*C/5+32;
disp(F);
```

Results:

Command Window:

```
98.6000
```

TASK-3

Set up a vector called N with five elements having the values: 1, 2, 3, 4, 5. Using N, create assignment statements for a vector X which will result in X having these values:

a) 2, 4, 6, 8, 10
b) 1/2, 1, 3/2, 2, 5/2
c) 1, 1/2, 1/3, 1/4, 1/5
d) 1, 1/4, 1/9, 1/16, 1/25

Program:

```
N=[1 2 3 4 5];
X=2*N;
disp(X);
X=N/2;
disp(X);
X=N.^-1; % or X=1./N
disp(X);
X=N.^-2; % or X=1./(N.*N) or X=1./N.^2
disp(X);
```

Results:

Command Window:

```
     2     4     6     8    10

    0.5000    1.0000    1.5000    2.0000    2.5000

    1.0000    0.5000    0.3333    0.2500    0.2000

    1.0000    0.2500    0.1111    0.0625    0.0400
```

TASK-4

A supermarket conveyor belt holds an array of groceries. The price of each product (in pounds) is [0.6 1.2 0.5 1.3] ; while the numbers of each product are [3 2 1 5]. Use MATLAB to calculate the total bill

Program:

```
price=[0.6 1.2 0.5 1.3];
number=[3 2 1 5];
cost=price*number'; % or cost=sum(price.*number);
disp(cost);
```

Results:

Command Window:

```
    11.2000
```

TASK-5

The `sortrows(X)` function will sort a vector or matrix X into increasing row order. Use this function to sort a list of names into alphabetical order.

Program:

```
fable={'once';'upon';'a';'time';'there';'lived';'three';'bears'};
sorted=sortrows(fable);
disp(sorted);
```

Results:

Command Window:

```
{'a'    }
{'bears'}
{'lived'}
{'once' }
{'there'}
{'three'}
{'time' }
{'upon' }
```

TASK-6

The "identity" matrix is a square matrix that has ones on the diagonal and zeros elsewhere. You can generate one with the `eye()` function in MATLAB. Use MATLAB to find a matrix B, such that when multiplied by matrix A=[1 2; -1 0] the identity matrix I=[1 0; 0 1] is generated. That is A*B=I.

Program:

```
A=[ 2 1; -1 0];
I=eye(2);
B=I/A;
disp(B);
```

Results:

Command Window:

```
     0    -1
     1     2
```

TASK-7

Create an array of N numbers. Now find a single MATLAB statement that picks out from that array the 1,4,9,16,…,$\sqrt{N}^{th}$ entries, i.e. those numbers which have indices that are square numbers.

Program:

```
N=100;
array = 1:N;
for i=1:sqrt(N)
    y(i)= i^2;
end
disp(y)
```

Results:

Command Window:

```
Columns 1 through 10

     1     4     9    16    25    36    49    64    81   100
```

TASK-8

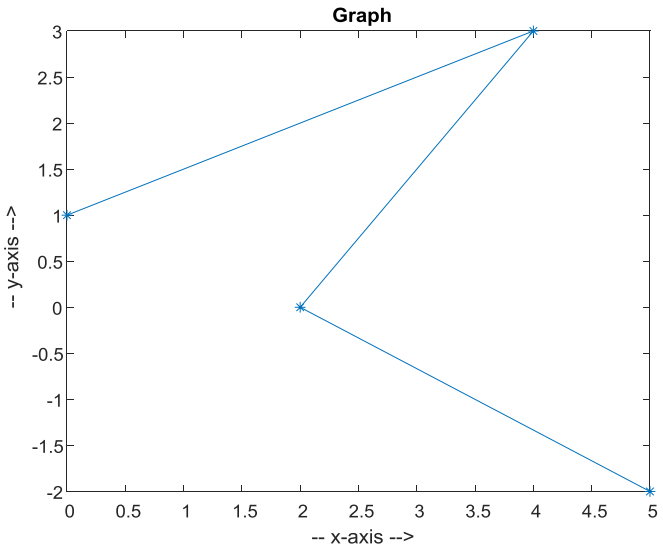
Draw a graph that joins the points (0,1), (4,3), (2,0) and (5,-2).

Program:

```
X=[0 4 2 5];
Y=[1 3 0 -2];
plot(X,Y,'*-');
xlabel('-- x-axis -->')
ylabel('-- y-axis -->')
title('Graph')
```

Results:

Figure Window:



TASK-9

The seeds on a sunflower are distributed according to the formula below. Plot a small circle at each of the first 1000 co-ordinates

$$r_n = \sqrt{n}$$

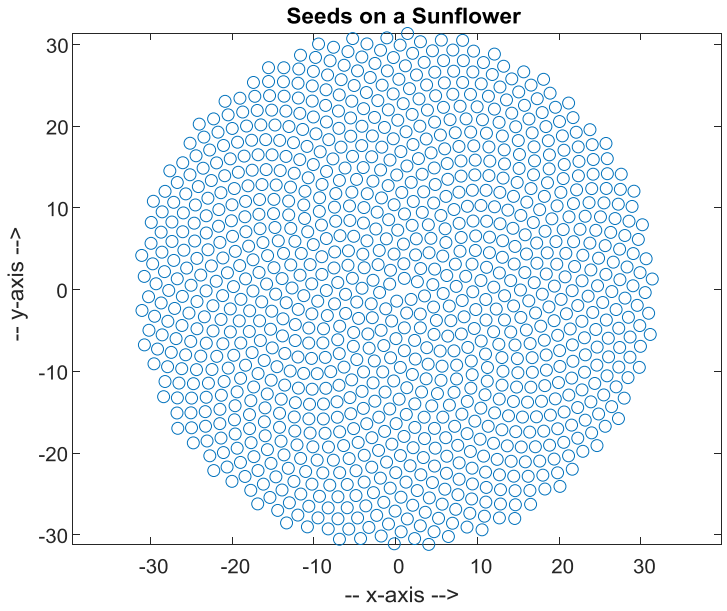
$$\theta_n = \frac{137.51}{180}\pi n$$

Program:

```
n=1:1000;
r=sqrt(n);
theta=137.51*pi*n/180;
x=r.*cos(theta);
y=r.*sin(theta);
plot(x,y,'o');
axis('equal');
xlabel('-- x-axis -->')
ylabel('-- y-axis -->')
title('Seeds on a Sunflower')
```

Results:

Figure Window:



TASK-10

Calculate 10 approximate points from the function y=2x by using the formulae:

$x_n = n$

$y_n = 2n + \text{rand} - 0.5$

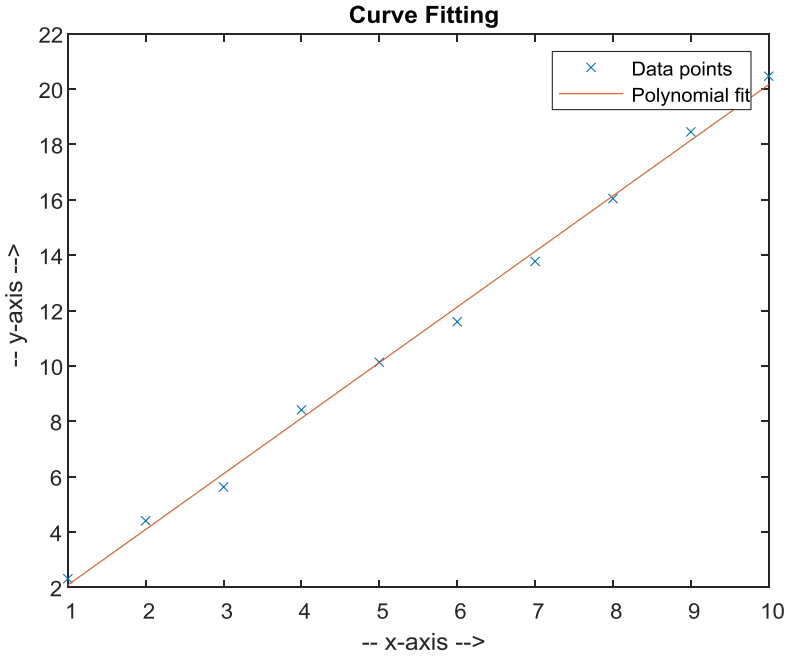
Fit a line of best fit to these points using the function `polyfit()` with `degree=1`, and generate co-ordinates from the line of best fit using `polyval()`. Use the on-line help to find out how to use these functions. Plot the raw data and the line of best fit.

Program:

```
x=1:10;
y=2*x+rand(1,10)-0.5;
p=polyfit(x,y,1);
y2=polyval(p,x);
plot(x,y,'x',x,y2,'-');
xlabel('-- x-axis -->')
ylabel('-- y-axis -->')
title('Curve Fitting')
legend('Data points','Polynomial fit')
```

Results:

Figure Window:



TASK-11

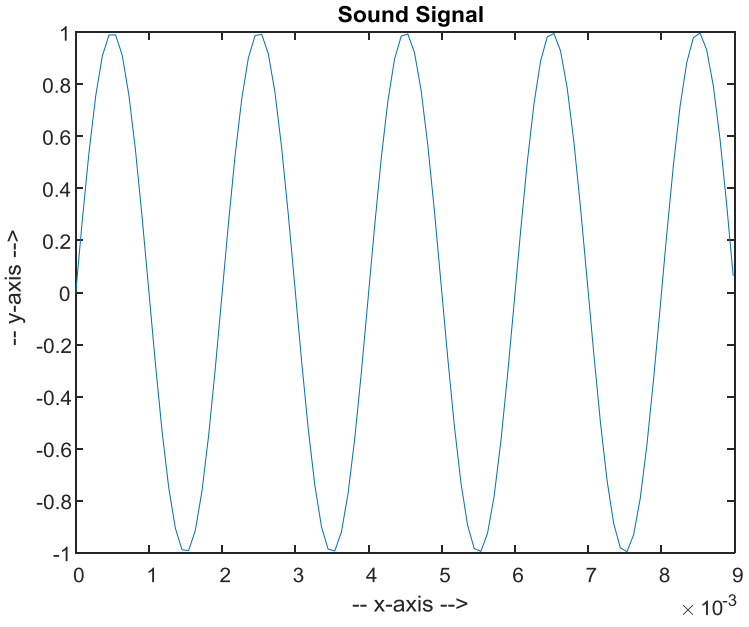
Calculate and replay 1 second of a sinewave at 500Hz with a sampling rate of 11025Hz. Save the sound to a file called "ex35.wav". Plot the first 100 samples

Program:

```
srate=11025;
t=0:1/srate:1;
s=sin(2*pi*500*t);
audiowrite('ex35.wav',s,srate);
[s,srate]=audioread('ex35.wav');
sound(s,srate)
plot(t(1:100),s(1:100),'-');
xlabel('-- x-axis -->')
ylabel('-- y-axis -->')
title('Sound Signal')
```

Results:

Figure Window:



TASK-12

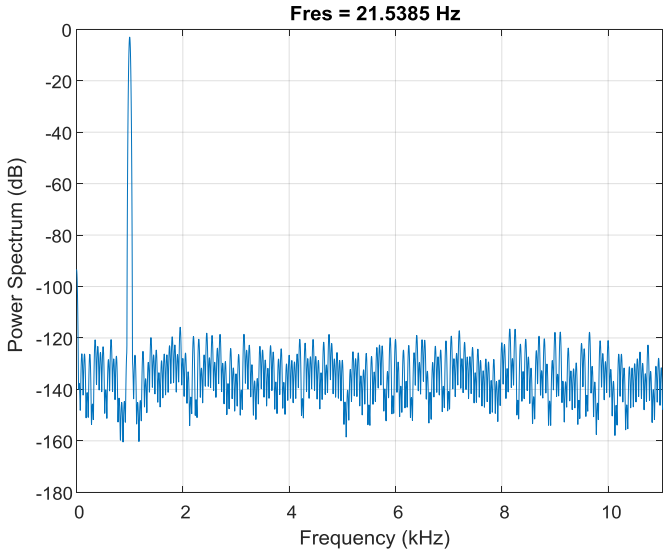
Calculate and replay a 2 second chirp. That is, a sinusoid that steadily increases in frequency with time, from say 250Hz at the start to 1000Hz at the end.

Program:

```
srate=11025;              % sample rate
nsamp=2*srate;            % number of samples
t=0:2/nsamp:2;            % sample times
f=250:750/nsamp:1000;     % required frequency at each sample time
y=sin(2*pi*f.*t);         % OK provided change in f per sample is
small.
sound(y,srate);
pspectrum(y,nsamp)
```

Results:

Figure Window:



TASK-13

Build a square wave by adding together 10 odd harmonics: 1f, 3f, 5f, etc. The amplitude of the n^{th} harmonic should be 1/n. Display a graph of one cycle of the result superimposed on the individual harmonics.

Program:

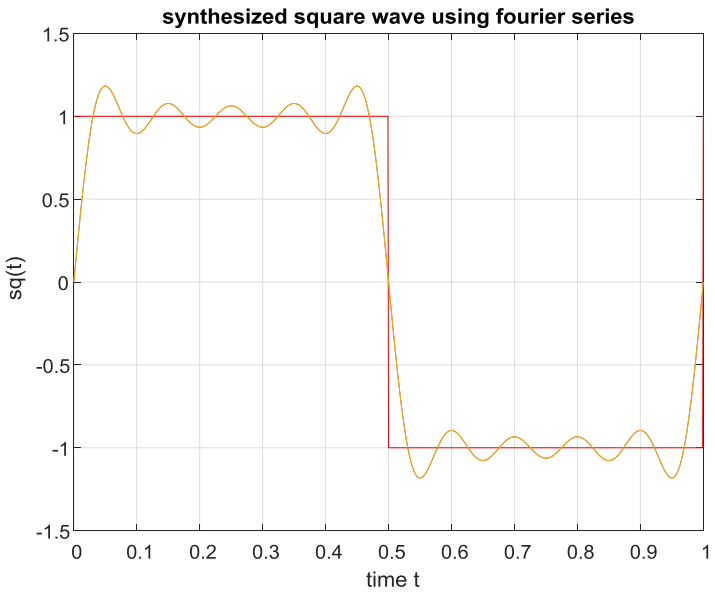
```
clc;
close all;
clear all;
N=input('type the number of harmonics');
%%to generate original square wave
t=0:0.001:1;
y=square(2*pi*t);
%to plot the original square wave signal with red line
plot(t,y,'r');
grid on
axis([0 1 -1.5 1.5]);
hold;
%generate a vector of zeros to initialize the square wave
sq=zeros(size(t));
%generate odd values beginning with 1 and incremented by 2
for n=1:2:N
    figure;
    plot(t,y,'r');
grid on
axis([0 1 -1.5 1.5]);
hold;
%equation for synthesized square wave
sq=sq+(4/(pi*n)*sin(2*pi*n*t));
plot(t,sq);
pause;
end;
%%to plot synthesized square wave
plot(t,sq);
grid on
xlabel('time t');
ylabel('sq(t)');
title('synthesized square wave using fourier series');
```

Results:

Command Window:

```
type the number of harmonics  10
```

Figure Window:



TASK-14

Write a function called FtoC to convert Fahrenheit temperatures into Celsius. Make sure the program has a title comment and a help page. Test from the command window with:

```
FtoC(96)
lookfor Fahrenheit
help FtoC
```

Program:

```
function C=FtoC(F)
% Celsius=FtoC(Fahrenheit)
% Converts Fahrenheit temperatures to Celsius
C=5*(F-32)/9;
```

Results:

Command Window:

```
FtoC(96)

ans =

    35.5556
```

TASK-15

Write a program to input 2 strings from the user and to print out (i) the concatenation of the two strings with a space between them, (ii) a line of asterisks the same length as the concatenated strings, and (iii) the reversed concatenation. For example:

```
Enter string 1: Mark
Enter string 2: Huckvale
Mark Huckvale
*************
elavkcuH kraM
```

Program:

```
s1=input('Enter string 1: ','s');
s2=input('Enter string 2: ','s');
s3=[s1 ' ' s2];
disp(s3);
disp(char('*'*ones(size(s3))));
disp(s3(length(s3):-1:1));
```

Results:

Command Window:

```
Enter string 1: Mark
Enter string 2: Huckvale
Mark Huckvale
*************
elavkcuH kraM
```