KARPAGAM ACADEMY OF HIGHER EDUCATION

Coimbatore - 641021.

(For the candidates admitted from 2016 onwards)

DEPARTMENT OF COMPUTER SCIENCE, CA & IT

SUBJECT : PROGRAMMING IN MATLAB-PRACTICAL
SEMESTER : III
SUBJECT CODE : 16CSU314B CLASS : II B.Sc CS

1. Write a program to assign the following expressions to a variable A and then to print out the value of A.
a. (3+4)/(5+6)
b. $2\Pi^2$
c. √2
d. (0.0000123 + 5.67×10-3) × 0.4567×10-4

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.

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

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.

5. The sort rows(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.

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.

7. Create an array of N numbers. Now find a single MATLAB statement that picks out from that array the 1,4,9,16,…,√Nth entries, i.e. those numbers which have indices that are square numbers.

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

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 :

$$r_n = \sqrt{n}$$

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

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

i. xn = n

ii. yn = 2n + 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.

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.

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.

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

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

i. FtoC(96)

ii. look for Fahrenheit

iii. help FtoC

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:
i. Enter string 1: Mark

ii. Enter string 2: Huckvale

iii. Mark Huckvale

iv. *************

v. elavkcuH kraM

Ex.No: 01

ARITHMETIC CALCULATION

Date:

Aim:

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

a. (3+4)/(5+6)
b. $2\Pi^2$
c. √2
d. (0.0000123 + 5.67×10-3) × 0.4567×10-4

Algorithm:

Step 1: Start the process.
Step 2: Open the new window.
Step 3: Enter the command as A= (3+4)/(5+6) and enter disp(A) to display the result.
Step 4: Enter the command A=2*pi*pi and enter disp(A) to display the result.
Step 5: Enter the command A=**√2** and enter disp(A) to display the result.
Step 6: Enter the command A=(0.0000123+5.67*10-3)*0.4567*10-4 and enter disp(A) to display the result.
Step 7: Stop the process

Program:

```
a)   A = (3+4)/(5+6);
disp(A);


b) A=2*pi*pi;
disp(A);


c) A=sqrt(2);
disp(A);


d) A=(0.0000123+5.67E-3)*0.4567E-4;
disp(A);
```

Output:

```
a)  A=0.6364

b)  A=19.7392

c)  A=1.4142

d)  A=2.5951e-07
```

Result:

The above program has been executed successfully and output is verified.

Ex. No: 02

CONVERT CELCIUS TO FAHRENHEIT

Date:

Aim:

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.

Algorithm:

Step 1: Start the process.
Step 2: Open the new window.
Step 3: Initialize the variable C with the value 37.
Step 4: To calculate F by using formula F=9*C/5+32.
Step 5: Enter disp(F) to display the result.
Step 6: Stop the process.

Program:

```
C=37;

F=9*C/5+32;

disp(F);
```

Output:

```
F=98.6000
```

Result:

The above program has been executed successfully and output is verified.

Ex. No: 03

CREATE ASSIGNMENT STATEMENTS FOR A VECTOR

Date:

Aim:

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

Algorithm:

Step 1: Start the process.
Step 2: Open the new window.
Step 3: Declare the variable N and assign the values as 1,2,3,4 and 5
Step 4: Calculate X= 2*N.
Step 5: Calculate X= N/2.
Step 6: Calculate X= N^.-1.
Step 7: Calculate X= N^.-2.
Step 8: Display the result using disp(X) command.
Step 9: Stop the process.

Program:

a)

```
N=[1 2 3 4 5];
 X=2*N;
 disp(X);
```

b)

```
X=N/2;
 disp(X);
```

c)

```
X=N.^-1; % or X=1./N
 disp(X);
```

d)

```
X=N.^-2; % or X=1./(N.*N) or X=1./N.^2
 disp(X);
```

Output:

```
a)  2       4       6       8       10

b)  0.5000      1.0000      1.5000      2.000 2.5000

c)  1.0000      0.5000      0.3333      0.2500      0.2000

d)  1.0000      0.2500      0.1111      0.0625      0.0400
```

Result:

The above program has been executed successfully and output is verified.

Ex.No: 04

CALCULATE TOTAL BILL

Date:

Aim:

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.

Algorithm:

Step 1: Start the process.
Step 2: Open the new window.

Step 3: Assign the values for price and number of product .

Step 4: Calculate the cost by using formula cost=price*number.

Step 5: Display the result using disp(cost) command.

Step 6: Stop the process.

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);
```

Output:

```
Cost= 11.2000
```

Result:

The above program has been executed successfully and output is verified.

Ex.No: 05

SORTING A LIST OF NAMES INTO ALPHABETICAL ORDER

Date:

Aim:

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.

Algorithm:

Step 1: Start the process.

Step 2: Open the new window.

Step 3: Assign a list of values for the variable fable.

Step 4: Sort the list of names into alphabetical order by using sorted=gsort(fable,’lr’,’i’).

Step 5: Display the result using disp(sorted) command.

Step 6: Stop the process.

Program:

```
fable=['once';'upon';'a';'time';'there';'lived';'three';'bears'];

sorted=gsort(fable,'lr','i');

disp(sorted);
```

Output:

```
!    a        !
!             !
! bears !
!             !
! lived      !
!             !
! once       !
!             !
!  there !
!             !
!  three !
!             !
!  time  !
!             !
!  upon !
```

Result:

The above program has been executed successfully and output is verified.

Ex.No: 06

MATRIX OPERATION

Date:

Aim:

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.

Algorithm:

Step 1: Start the process

Step 2: Open the new window

Step 3: Assign the matrix values [2 1 ; -1 0] to the variable A

Step 4: Use eye() function to calculate I.

Step 5: Calculate matrix B value by using B=I/A.

Step 6: Display the result using disp(B) command

Step 7: Stop the process

Program:

```
A=[ 2 1; -1 0];

I=eye(2);

B=I/A;

disp(B);
```

Output:

```
0          -1

1           2
```

Result:

The above program has been executed successfully and output is verified.

Ex.No: 07

TO SQUARE THE NUMBERS

Date:

Aim:

Create an array of N numbers. Now find a single MATLAB statement that picks out from that array the 1,4,9,16,…,√Nth entries, i.e. those numbers which have indices that are square numbers.

Algorithm:

Step 1: Start the process.

Step 2: Open the new window.

Step 3: Specify the starting and ending values with an interval 1 for the variable x as [0:1:4]

Step 4: Calculate y by using the formula y = x.^2.

Step 5: Display the square number by using disp(y).

Step 6: Stop the process.

Program:

```
x = [0:1:4];
 y = x.^2;
disp(y);
```

Output:

```
0.  1.   4.      9.       16
```

Result:

The above program has been executed successfully and output is verified.

Ex.No: 08

TO DRAW A GRAPH

Date:

Aim:

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

Algorithm:

Step 1: Start the process.

Step 2: Open the new window.

Step 3: Assign the values [0 4 2 5] to the variable X.

Step 4: Assign the values [1 3 0 -2] to the variable Y.

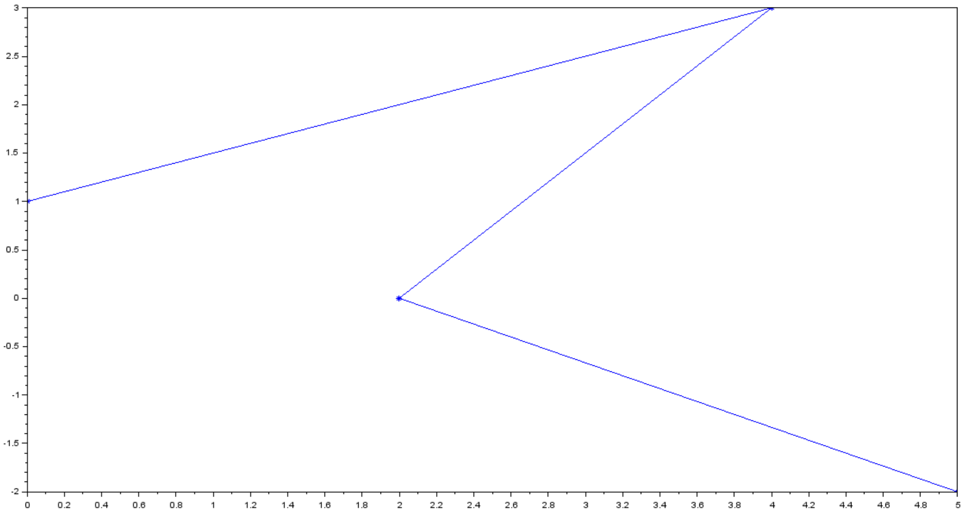
Step 5: Display the graph by using plot(X,Y,'*-').

Step 6: Stop the process.

Program:

```
X= [0 4        2        5];

Y= [1 3        0        -2];

plot(X,Y,’*-‘);
```

Output:



Result:

The above program has been executed successfully and output is verified.

Ex. No: 09

CREATING SUNFLOWER

Date:

Aim:

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

$$r_n = \sqrt{n}$$

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

Algorithm:

Step 1: Start the process

Step 2: Open the new window

Step 3: Assign the range from 1 to 1000 for the variable n

Step 4: Calculate r by using the formula r= sqrt (n)

Step 5: Calculate theta value by using 137.51*%pi*n/180

Step 6: Calculate X and Y values by using r*.cos(theta) and r*.sin(theta).

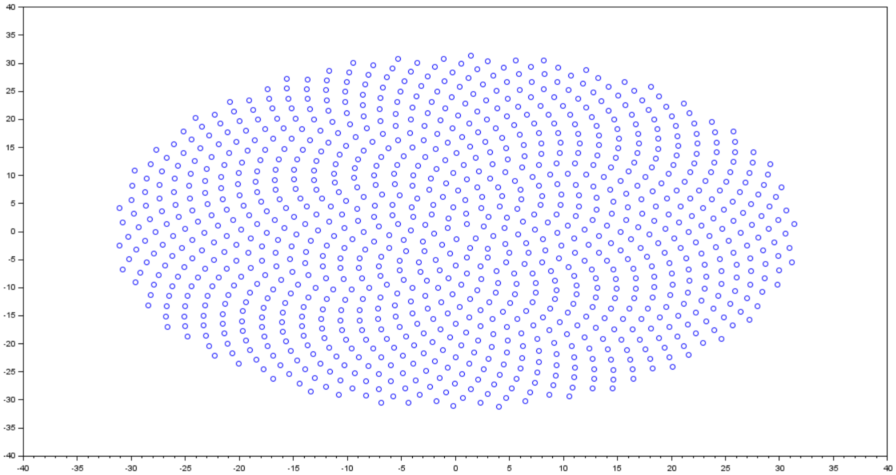
Step 7: Display the result by using plot(X,Y,’’).

Step 8: Stop the process.

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');
```

Output:



Result:

The above program has been executed successfully and output is verified.

Ex.No:10

CALCULATE APPROXIMATE POINTS FROM FUNCTION

Date:

Aim:

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

i. xn = n

ii. yn = 2n + 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.

Algorithm:

Step 1: Start the process

Step 2: Open the new window

Step 3: Declare the variables as x and y.

Step 4: The line is best fitted to the points using the function polyfit()

Step 5: Co-ordinates are generated from the line of best fit using y2=polyval(p,x).

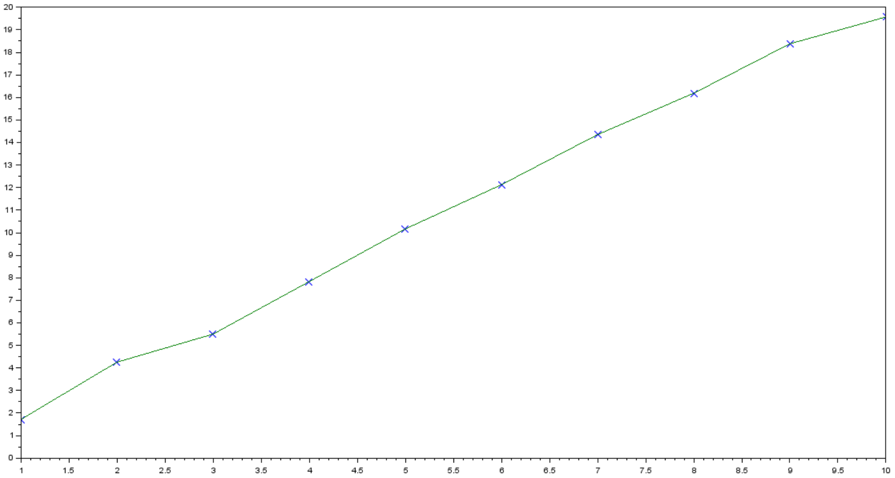
Step 6: Display the appropriate points using plot(x,y,’x’,x,y,’--‘).

Step 7: Stop the process.

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,y,'-');
```

Output:



Result:

The above program has been executed successfully and output is verified.

Ex.No:11

CALCULATE AND REPLAY 1 SECOND OF A SIN WAVE

Date:

Aim:

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 fi!

Algorithm:

Step 1: Start the process

Step 2: Open the new window.

Step 3: Declare the srate value as 11025.

Step 4: Calculate t value by using the formula 0:1/srate:1.

Step 5: Calculate s value by using the formula sin(2*pi*500*t)

Step 6: Save the sound file using the function wavwrite(s,srate,16,'ex35.wav')

Step 7: Display the appropriate points using plot(t(1:100),s(1:100),'-')

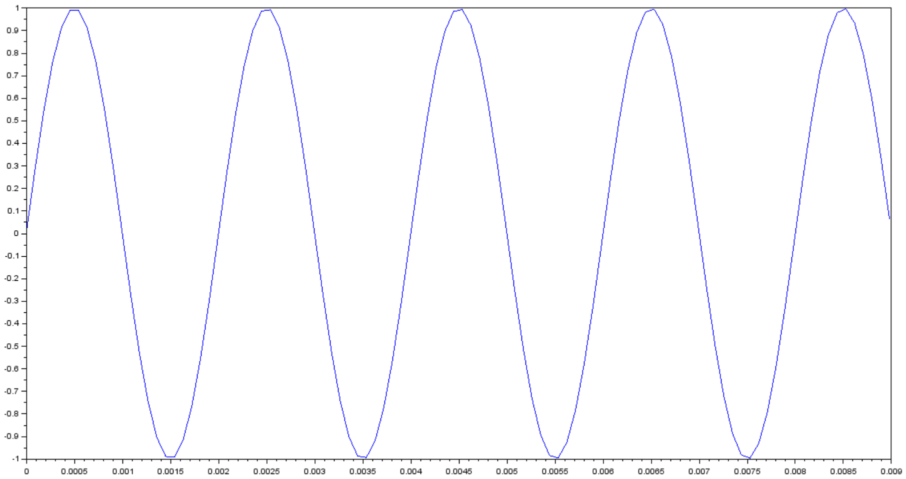
Step 8: Display the result.

Step 9: Stop the process

Program:

```
srate=11025;

t=0:1/srate:1;

s=sin(2*pi*500*t);

sound(s,srate);

wavwrite(s,srate,16,'ex35.wav');

plot(t(1:100),s(1:100),'-');
```

Output:



Result:

The above program has been executed successfully and the output is verified.

Ex.no:12

CALCULATE AND REPLAY 2SECOND CHIRP SOUND

Date:

Aim:

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.

Algorithm:

Step 1: Start the process

Step 2: Open the new window.

Step 3: Declare srate value as 11025.

Step 4: Calculate nsamp value by using the formula 2*srate.

Step 5: Calculate t and f value by using the formula 0:2/nsamp:2 and 250:750/nsamp:1000.

Step 6: Calculate s value by using the formula sin(2*pi*500*t)

Step 7: The values of the audio signal is specified in the parameters of the function sound(y,srate).

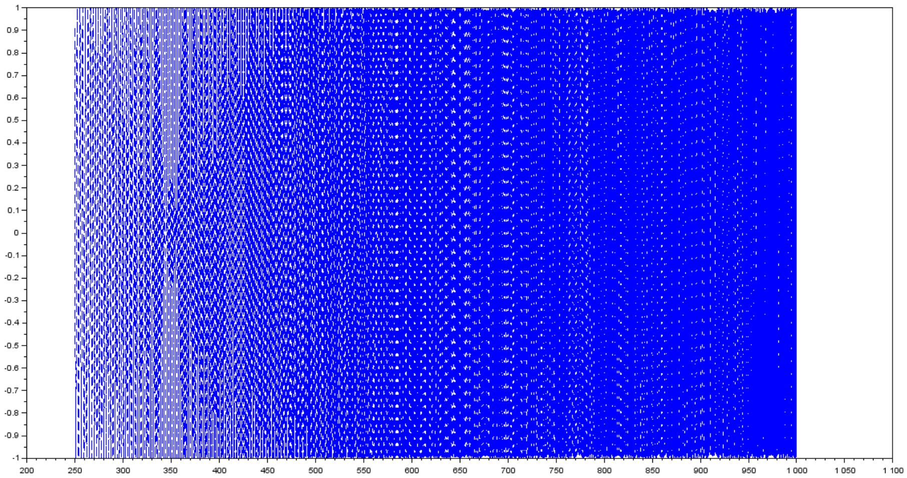
Step 8: Display the appropriate points using plot(t(1:100),s(1:100),'-')

Step 9: Stop the process.

Program:

```
srate=11025;

nsamp=2*srate;

t=0:2/nsamp:2;

f=250:750/nsamp:1000;

y=sin(2+3.14*f.*t);

sound(y,srate);

plot(f,y,'--');
```

Output:



Result:

The above program has been executed successfully and the output is verified.

Ex.no:13

BUILDING A SQUARE WAVE BY ADDING

10 ODD HARMONICS

Date:

Aim:

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

Algorithm:

Step 1: Start the process.

Step 2: Using MATLAB.

Step 3: Specify bthe starting and ending ramge from 0 to 10 for the variable t.

Step 4: Calculate y value by using the formula sin(t)+sin(3*t)/3+sin(5*t)/5+sin(7*t)/7+sin(9*t)/9.

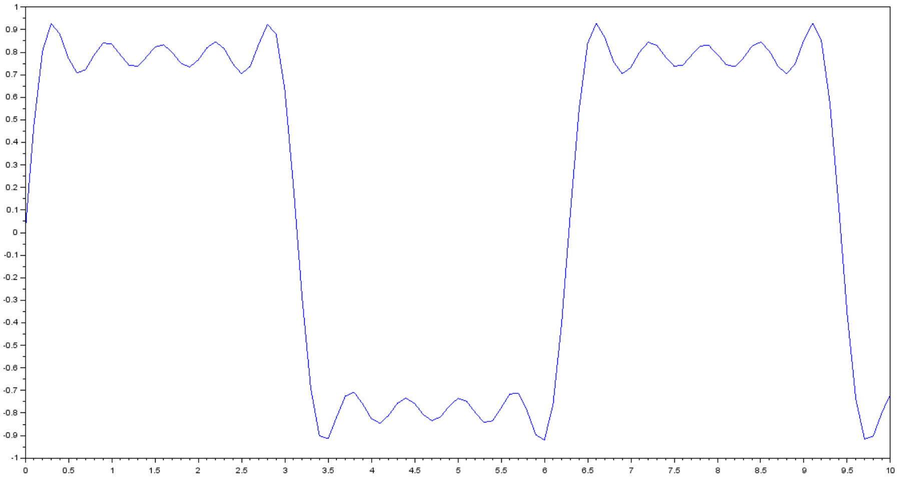
Step 5 :To draw the square wave using the plot(+y).

Step 6: Stop the process.

Program:

```
t=0:.1:10;

y=sin(t)+sin(3*t)/3+sin(5*t)/5+sin(7*t)/7+sin(9*t)/9;

plot(t,y);
```

Output:



Result:

The above program has been executed and the output is verified successfully.

Ex.no:14

CONVERT FAHRENHEIT TO CELCIUS

Date:

Aim:

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

i. FtoC(96)
ii. lookfor Fahrenheit
iii. help FtoC

Algorithm:

Step 1: Start the process.

Step 2: Using MATLAB.

Step 3: Define the function ftoc(f) . Calculate celcisus value

c= c=5*(f-32)/9 within the function

Step 4: Pass the parameter value 40 to the function ftoc.

Step 5: Display the value of celcius in disp(‘the celcius

value is’+string(c)) format.

Step 6: Stop the process.

Program:

```
function [c]=ftoc(f)
 c=5*(f-32)/9;
 endfunction
c=ftoc(40)
disp('the celcius value is '+string(c))
```

Output:

```
4.4444444

The celcius value is 4.4444444
```

Result:

The above program has been executed and the output is verified successfully.

Ex.no:15

STRING MANIPULATION

Date:

Aim:

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: i. Enter string 1: Mark ii. Enter string 2: Huckvale
iii. Mark Huckvale iv. ************* v. elavkcuH kraM

Algorithm:

Step 1: Start the process.

Step 2: Open the new window.

Step 3: Assign values for the string variable S1,S2 as s1=’good’,s2=’morning’.

Step 4: Concatenate the strings using s=strcat(‘good’+’morning’).Print the line of asterisks using the s1=sprintf('****************')

Step 5: Reverse the string concatenation using disp(strrev(s)).

Step 6: Stop the process.

Program:

```
s1='good';
s2='morning';
 s=strcat('good'+' '+'morning');
s1=sprintf('****************');
disp(s1,s);
disp(strrev(s));
```

Output:

```
good morning

**** *******

gninrom doog
```

Result:

The above program has been executed and the output is verified ssuccessfully.