**Lab Report # 04**



**CSE301 - L Signals & Systems Lab**

Submitted by: AWAIS SADDIQUI

Registration No: 21PWCSE1993

Class Section: “A”

Submitted to:

**Dr. Durr-e-Nayab**

**Department of Computer System Engineering**

**UET Peshawar**

**301L: Signals & Systems Lab**

**LAB ASSESSMENT RUBRICS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Marking Criteria** | **Exceeds expectation (5-4)** | **Meets expectation** **(3-2)** | **Does not meet expectation (1)** | **Score** |
| **1. Realization of Experiment** | Program compiles (noerrors and no warnings).  Program always works correctly and meets the specification(s).  Completed between 71-100% of the requirements. | Program compiles (no errors and some warnings).  Some details of the program specification are violated, program functions incorrectly for some inputs.  Completed between 41-70% of the requirements. | Program fails to or compile with lots of warnings.  Program only functions correctly in very limited cases or not at all.  Completed less than  40% of the requirements. | 30% |
| **2. Ability to apply required code utility or data structure** | Able to apply required data type or data structure and produce correct results. Familiarize and selects proper functions for simulation of given problem using  software tools like  MATLAB. | Able to apply required data type or data structure but does not produce correct results. Need guidance to select proper functions for simulation of given problem using  software tools like  MATLAB. | Unable to identify required data type or data structure.  Incapable of selecting proper functions for simulation of given problem using software tools like MATLAB. | 20% |
| **3. Documentation** | Clearly and effectively documented including descriptions of all variables/functions. Specific purpose is noted for each function, control structure, input requirements and output results. | Basic documentation including descriptions of all variables/functions. Specific purpose is noted for each function and control structure. | No documentation included. | 10% |
| **4. Ability to run/debug** | Executes Matlab codes without errors, excellent user | Executes Matlab codes without errors.  User prompts are | Does not execute Matlab codes due to errors. | 20% |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | prompts, good use of symbols, spacing in output.  Thorough and organized testing has been completed and output from test cases is included. | understandable, minimum use of symbols or spacing in output.  Some testing has been completed. | User prompts are misleading or nonexistent.  No testing has been completed. |  |
| **5. Results compilation** | Show processed results effectively by conducting simple computations and plotting using collected data | Show processed results effectively by conducting simple computations and plotting using collected data with minor error | Unable to show processed results effectively by conducting simple computations and plotting using collected data with minor error | 10% |
| **6. Efficiency** | Excellent use of CPU and Memory. | Good but not smart use of CPU and Memory. | Inefficient use of CPU and Memory. | 10% |
| **7. Lab**  **Performance (Team work and Lab**  **etiquettes)** | Actively engages and cooperates with other group members in an effective manner. Respectfully and carefully observes safety rules and procedures | Cooperates with other group members in a reasonable manner. Observes safety rules and procedures with minor deviation. | Distracts or discourages other group members from conducting the experiment. Disregards  safety rules and procedures. | 10% |

**Instructor:**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Signals & Systems Laboratory**

**DISCRETE-TIME SIGNAL REPRESENTATION IN MATLAB:**

In MATLAB, finite‐duration sequence (or discrete time signal) is represented by row matrix/vector of appropriate values. Such representation does not have any information about sample position therefore, for correct representation, two vectors are required, one for x and other for n.

**GRAPHICS**

Two‐ and three‐dimensional MATLAB graphs can be given titles, have their axes labeled, and have text placed within the graph.

**OBJECTIVES OF THE LAB**

* Discrete Signal representation in MATLAB
* MATLAB Graphics
* Two Dimensional Plots
* Plot and subplot
* Different Plotting Functions Used in MATLAB

**TASK 01**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CODE:**

%{

Task-01

x1[n] = [2 5 8 4 3]

x2[n] = [4 3 2]

%}

clear all

clc

%Part A

x1 = [2 5 8 4 3];

x2 = [4 3 2 0 0];

sum = x1+x2;

product = x1.\*x2;

%Part B

disp('Addition using loop is ')

for i=1:5

forSum(1,i)=x1(i)+x2(i);

end

disp(forSum)

for j=1:5

x4(1,j) = x1(j)\*x2(j);

end

disp('Multiplication of signals using loop is: ');

disp(x4);

%Part C

SigPlot(x1, x2, sum, product)

function SigPlot(sig1, sig2, sum, prod)

%seprate Figures

figure (1)

plot(sig1)

title('Signal 1 is ')

figure (2)

plot(sig2)

title('Signal 2 is ')

figure (3)

plot(sum)

title('Signal sum ')

figure (4)

plot(prod)

title('Signal product')

%Overlapping Signals

plot(sig1)

hold on

plot(sig2)

hold on

plot(sum)

hold on

plot(prod)

%using subplot

subplot(4,1,1)

plot(sig1)

subplot(4,1,2)

plot(sig2)

subplot(4,1,3)

plot(sum)

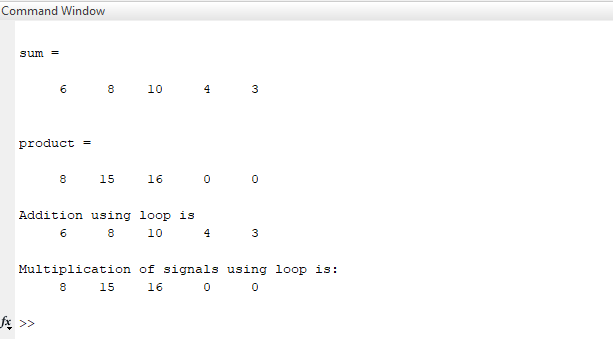
subplot(4,1,4)

plot(prod)

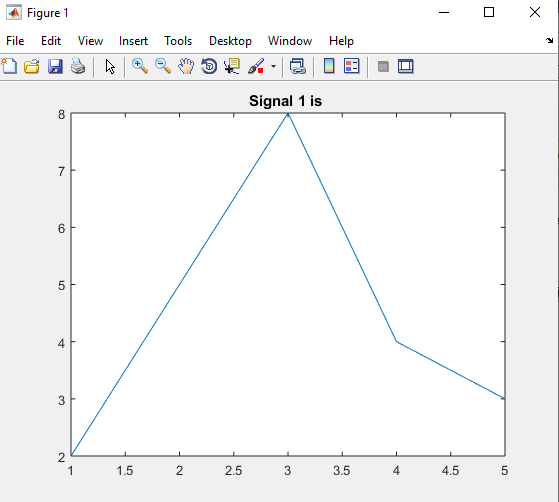
end

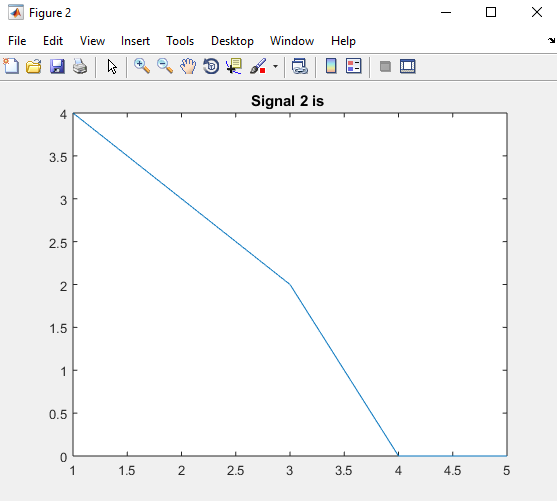
**OUPUT:**

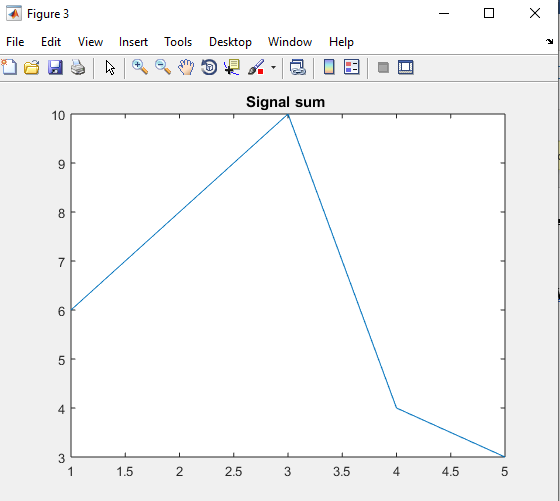
Part(a&b):

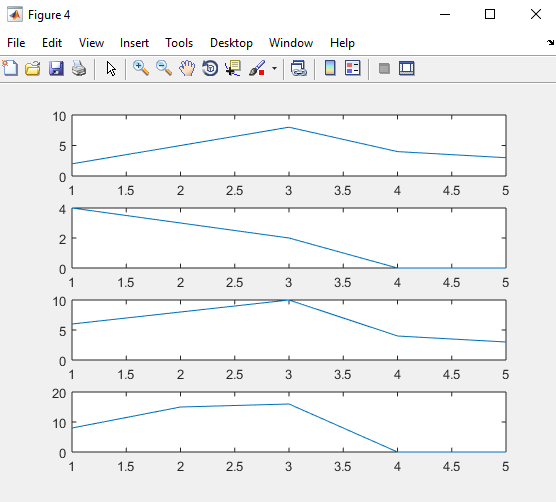


**Part B:**









**TASK 02**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CODE:**

%Task-02

clear all

clc

sig = [1 2 3 4 5];

scale=input('Enter Scaling factor value: ')

SigPlot2(sig,ScaleSig(sig,scale))

function[out]= ScaleSig(signal, fact)

out= signal\*fact;

end

function SigPlot2(s1, s2)

figure

plot(s1)

title('Signal 1');

figure

plot(s2)

title('Signal 2');

figure

%Now Overlapping Signals

hold on

plot(s1)

hold on

plot(s2)

figure

%Now using subplot

subplot(4,1,1)

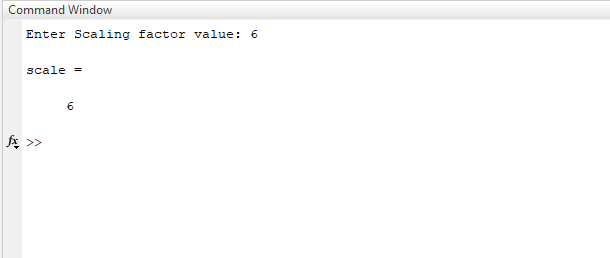
plot(s1)

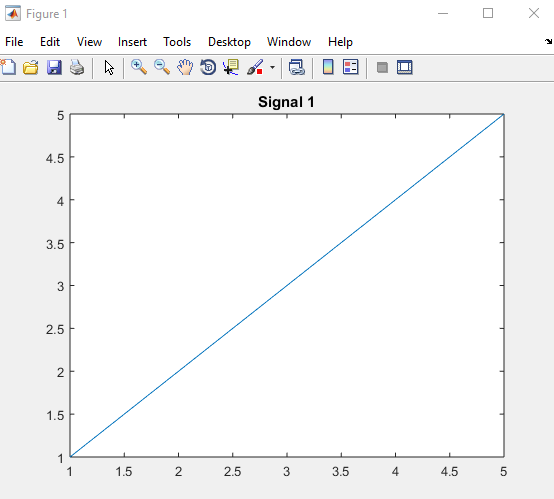
subplot(4,1,2)

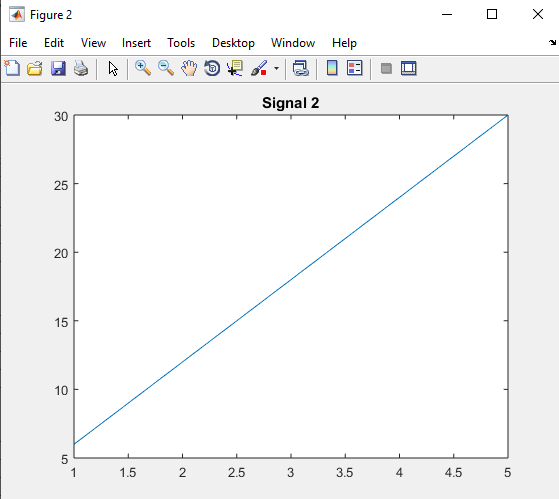
plot(s2)

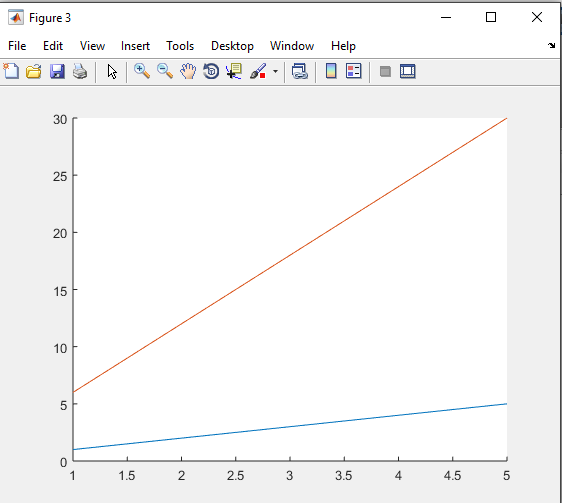
end

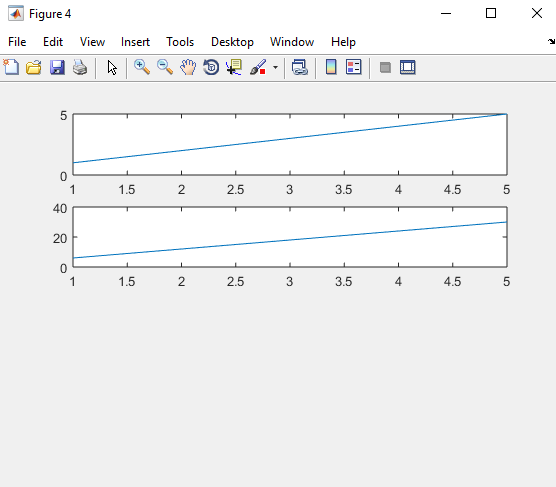
**OUPUT:**











**TASK 03**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CODE:**

clc;

clear all;

close all;

x1=[1 2 2 1 1]; %Given Signal 1

x2=[3 2 0 1 2]; %Given Signal 2

n=[0:4];

Index=[]; %Made to store indexs

y=1;

for i=1:5

if x1(i)<x2(i)

Index(y)=i-1;

y=y+1;

end

end

disp('x2 has smaller amplitude on following Indexs: ');

disp(Index);

Sigplot3(x1,x2,n)

function Sigplot3(sig1,sig2,n)

%For seprate widnow

figure

stem(n,sig1);

title('Signal 1')

figure

stem(n,sig2);

title('Signal 2')

figure

%On same Window

stem(n,sig1);

title('Signal 1 using hold on')

hold on

stem(n,sig2);

title('Signal 2 using hold on')

hold on

figure

%Different but same window

subplot(2,1,1);

stem(n,sig1);

title('Signal 1 using subplot')

hold on

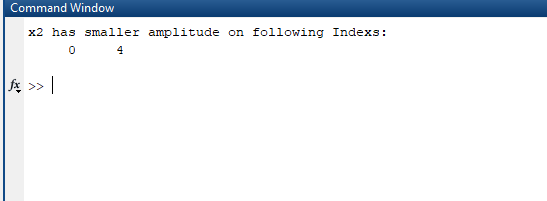
subplot(2,1,2);

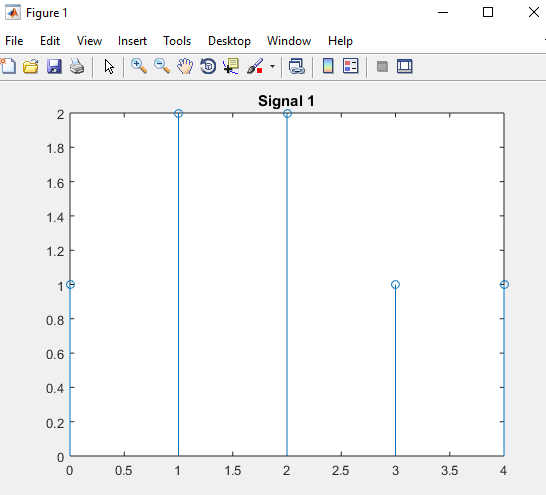
stem(n,sig2);

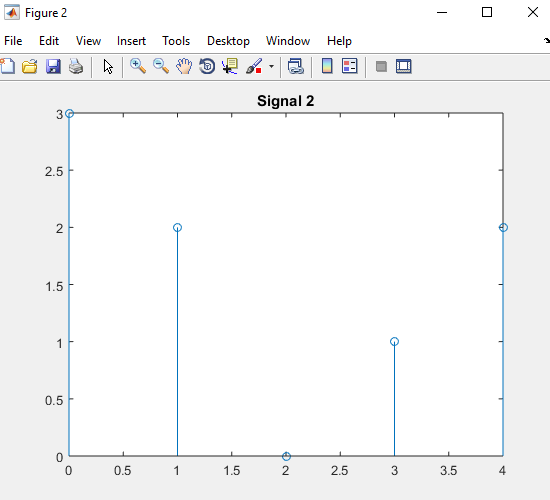
title('Signal 2 using subplot')

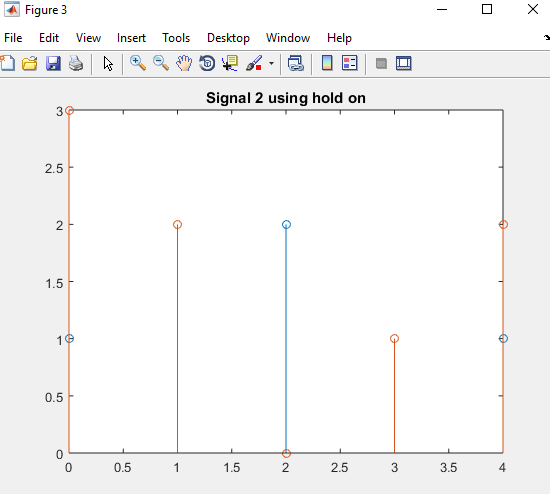
hold on

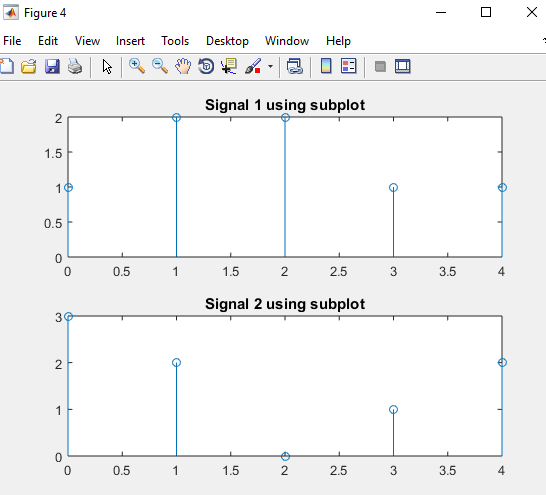
**OUPUT:**











**TASK 04**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CODE:**

clc;

clear all;

close all;

i=1;

T1=[-15:15];

T2=[-50:+50];

for x=-15:15

y1(1,i)=(2)\*(x.^2);

y2(1,i)=(4)\*(x.^3);

i=i+1;

end

figure

subplot(2,1,1)

plot(T1,y1,' -\*')

hold on

title('Signal y1 w.r.t T1');

subplot(2,1,2)

plot(T1,y2,'-.')

hold on

title('Signal y2 w.r.t T1');

i=1;

for x=-50:50

y1(1,i)=(2)\*(x.^2);

y2(1,i)=(4)\*(x.^3);

i=i+1;

end

figure

subplot(2,1,1)

plot(T2,y1,'-\*')

hold on

title('Signal y1 w.r.t T2');

subplot(2,1,2)

plot(T2,y2,'-.')

hold on

title('Signal y2 w.r.t T2');

**OUPUT:**

Graphical user interface

Description automatically generated

Graphical user interface, histogram

Description automatically generated

**TASK 05**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CODE:**

clc;

close all;

clear all;

c1=input('Enter x value of center :');

c2=input('Enter y value of center :');

rad=input('Enter radius of circle :');

PlotCircle(c1,c2,rad)

PlotCircle Function:

function PlotCircle(x,y,r)

theta=[0:1/100:2\*pi];

xe=(r\*cos(theta))+x;

ye=(r\*sin(theta))+y;

figure

plot(xe,ye)

**OUPUT:**

A screenshot of a computer

Description automatically generated with medium confidence

**Graphical user interface

Description automatically generated**

**TASK 06**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CODE:**

sig1=[2,5,8,4,3,0,0,0] % Signal 1 given

sig2=[0,0,0,0,1,4,3,2] % Signal 2 given

i=1:8; %There are 8 enteries

%sumsig is sum of two signals

sumsig=sig1+sig2;

disp('Resultant sum Signal is');

disp(sumsig);

%plotting sig1

subplot(3,1,1)

plot(sig1,i,'r -\*')

title('Signal 1')

%plotting sig2

subplot(3,1,2)

plot(sig2,i,'g -.')

title('Signal 2')

%plotting sumsig

subplot(3,1,3)

plot(sumsig,i,'b -\*')

title('Sum Signal')

**OUPUT:**

A screenshot of a computer

Description automatically generated with medium confidence

**Diagram

Description automatically generated**

**TASK 07**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CODE:**

clc;

clear all;

close all;

%num2str() function converts number to string that is how we will take

%signal from user below in this program

samp=input('How many sample values are You want to enter :');

%Above we are getting number about how many samples entered

for i=1:samp

y(i)=input(['Enter New Sample' num2str(i) ':']);

end

%Getting Threshold value

thresh=input('Enter Threshold: ');

AmpScale(y,thresh)

AmpScale Function:

function AmpScale(sig,t)

greater=0; %for samples greater than t

less=0; %for samples less than -t\

%numel(sig) is functon that will calculate number of element in signal sig

for i=1:1:numel(sig)

if sig(i)>t

sig(i)=sig(i)-t;

greater=greater+1;

end

if sig(i)<-t

sig(i)=sig(i)+t;

less=less+1;

end

end

disp('Samples Greater then Threshold are: ');

disp(greater);

disp('Samples Lesser then Threshold are: ');

disp(less);

figure

plot(sig,i)

**OUPUT:**

A screenshot of a computer

Description automatically generated with medium confidence

**TASK 08**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CODE:**

function DS=Downsample(S)

j=1;

for i=1:length(S)

if(mod(i,2)~=0)

DS(j)=S(i);

j=j+1;

end

end

end

S = input('Enter a Signal: ');

DS = Downsample(S);

disp(DS);

%figure

subplot(2,1,1)

stem(S);

title('Original Signal')

xlabel('Sample Number')

ylabel('Signal Amplitude')

%figure2

subplot(2,1,2)

stem(DS);

title('DownSampled Signal')

xlabel('Sample Number')

ylabel('Signal Amplitude')

**OUPUT:**

A screenshot of a computer

Description automatically generated with medium confidence

**TASK 09**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CODE:**

function US=Upsample(S)

j=1;

for i=1:2\*length(S)

if(mod(i,2)==0)

US(i)=0;

else

US(i)=S(j);

j=j+1;

end

end

S = input('Enter a Signal: ');

US = Upsample (S);

disp(US);

%figure

subplot(2,1,1)

stem(S, 'filled');

title('Original Signal')

xlabel('Sample Number')

ylabel('Signal Amplitude')

%figure2

subplot(2,1,2)

stem(US, 'filled');

title('UpSampled Signal')

xlabel('Sample Number')

ylabel('Signal Amplitude')

**OUPUT:**

A screenshot of a computer

Description automatically generated with medium confidence

**TASK 10**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CODE:**

clc;

clear all;

close all;

[x,y]=meshgrid([-2:.2:2]);

z= x.\*exp(-x.^2-y.^2);

figure

surf(x,y,z,gradient(z))

colorbar

**OUPUT:**

