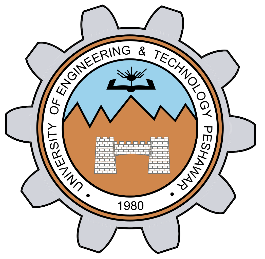
**TITLE OF LAB: (INTRODUCTION TO MATLAB GRAPHICS)**

**LAB # 04**



**Spring 2022**

**CSE301L Signals & Systems Lab**

Submitted by : **Safi Ullah Khan**

Registration No.: **20PWCSE1943**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

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

Submitted to:

**Engr. Durr-e-Nayab**

Day, Date (e.g Sunday, Jun 05th, 2022)

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**OBJECTIVES OF THE LAB**

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

In this lab, we will cover the following topics:

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

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

**-------------------------TASK 01--------------------------**

Given the signals:

x1[n] = [2 5 8 4 3] x2[n] = [4 3 2]

a) Write a MatLab program that adds these two signals. Use vector addition and multiplication.

b) Instead of using vector addition and multiplication, use for loop to add and multiply the signals.

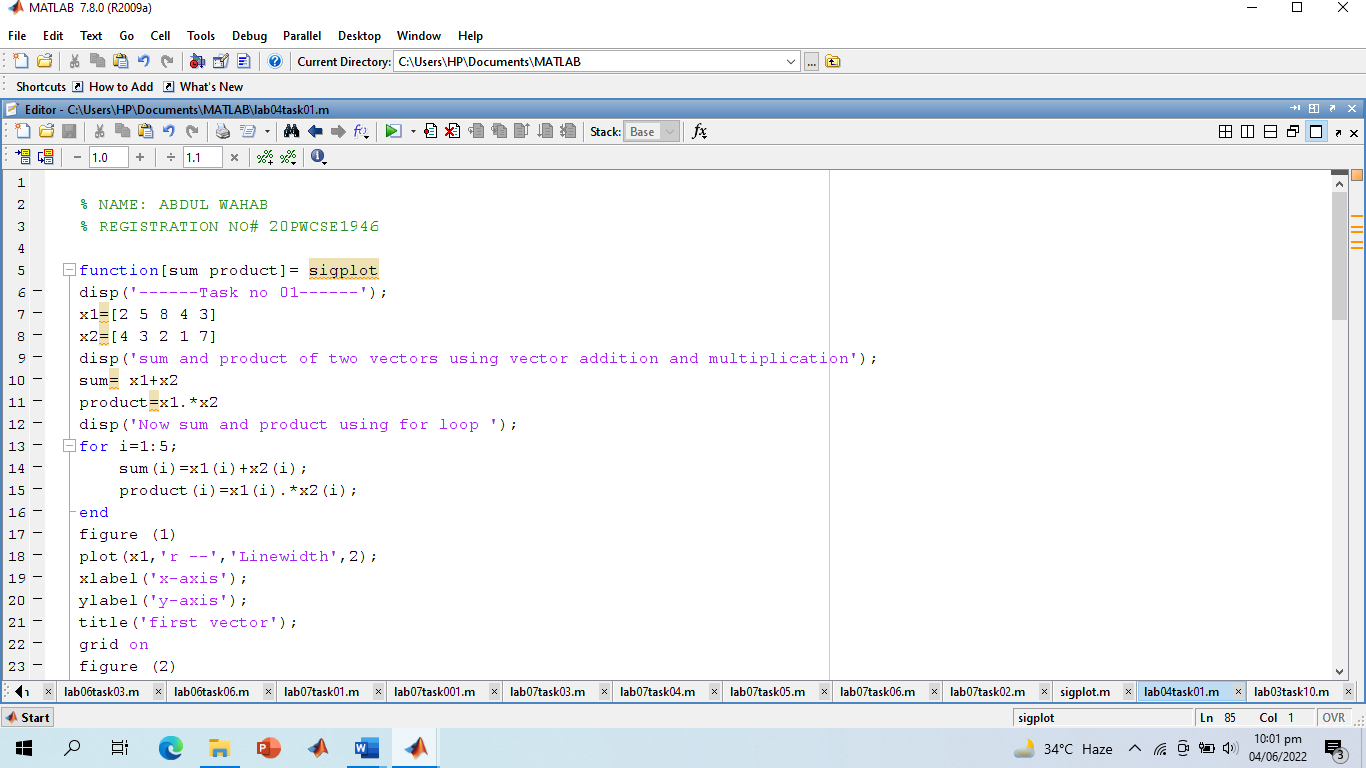
c) Design a function **SigPlot** that takes the original signals and their sum and product as input and plots them as:

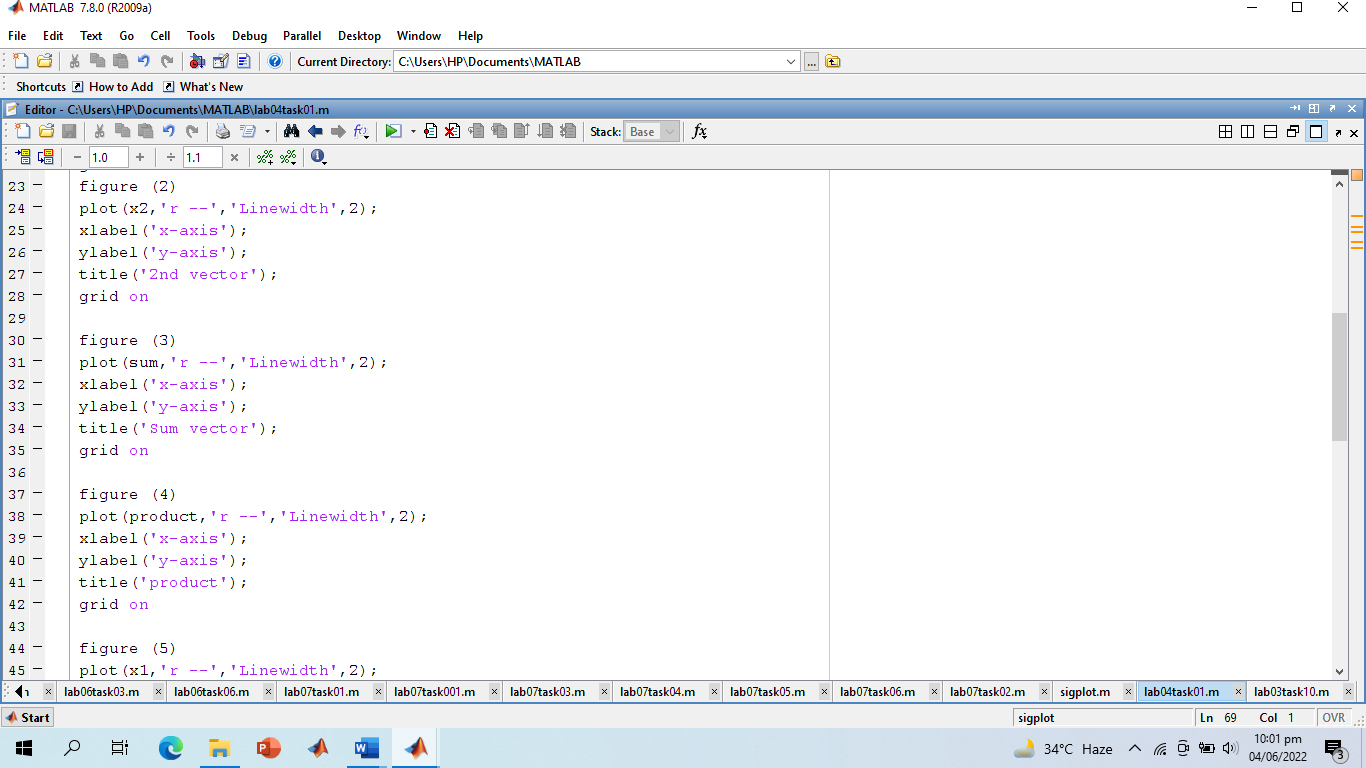
i) Separate Figures,

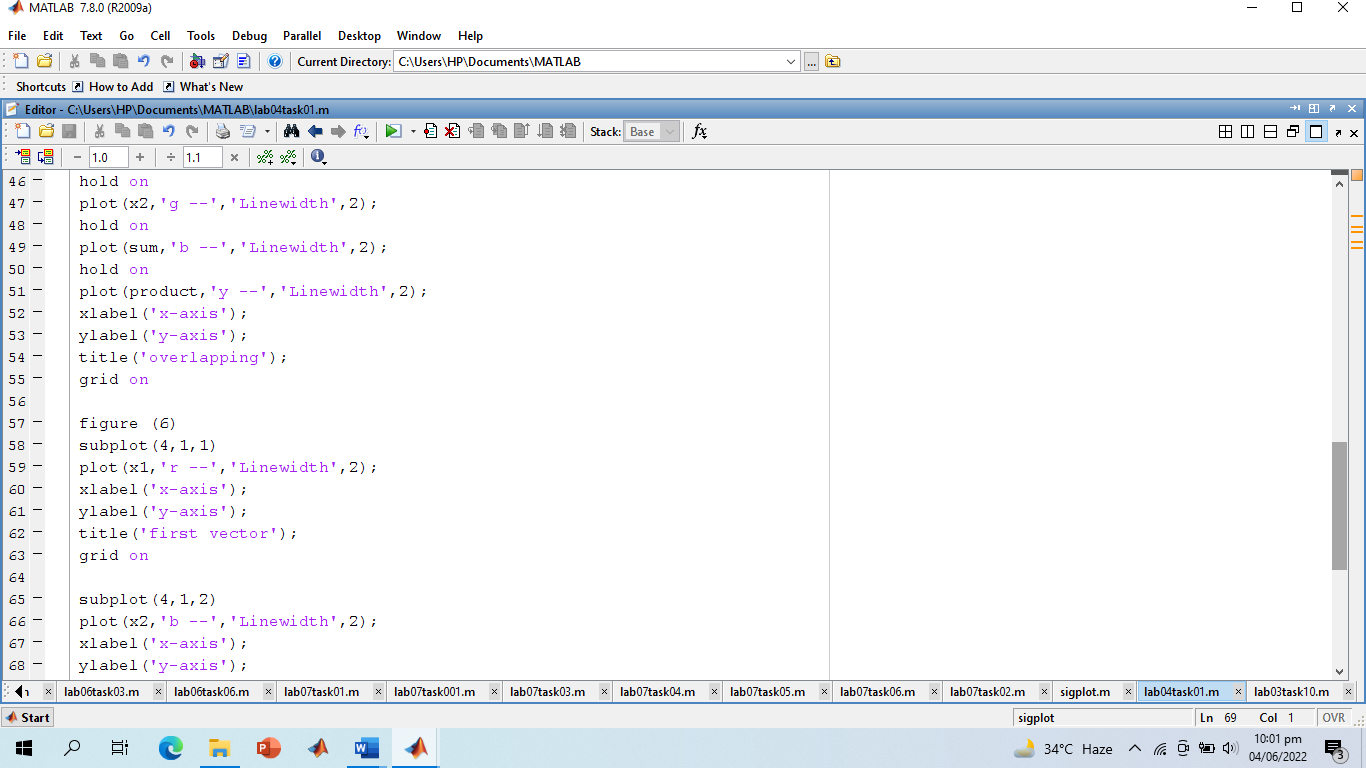
ii) Single Figure overlapping all the signals, and

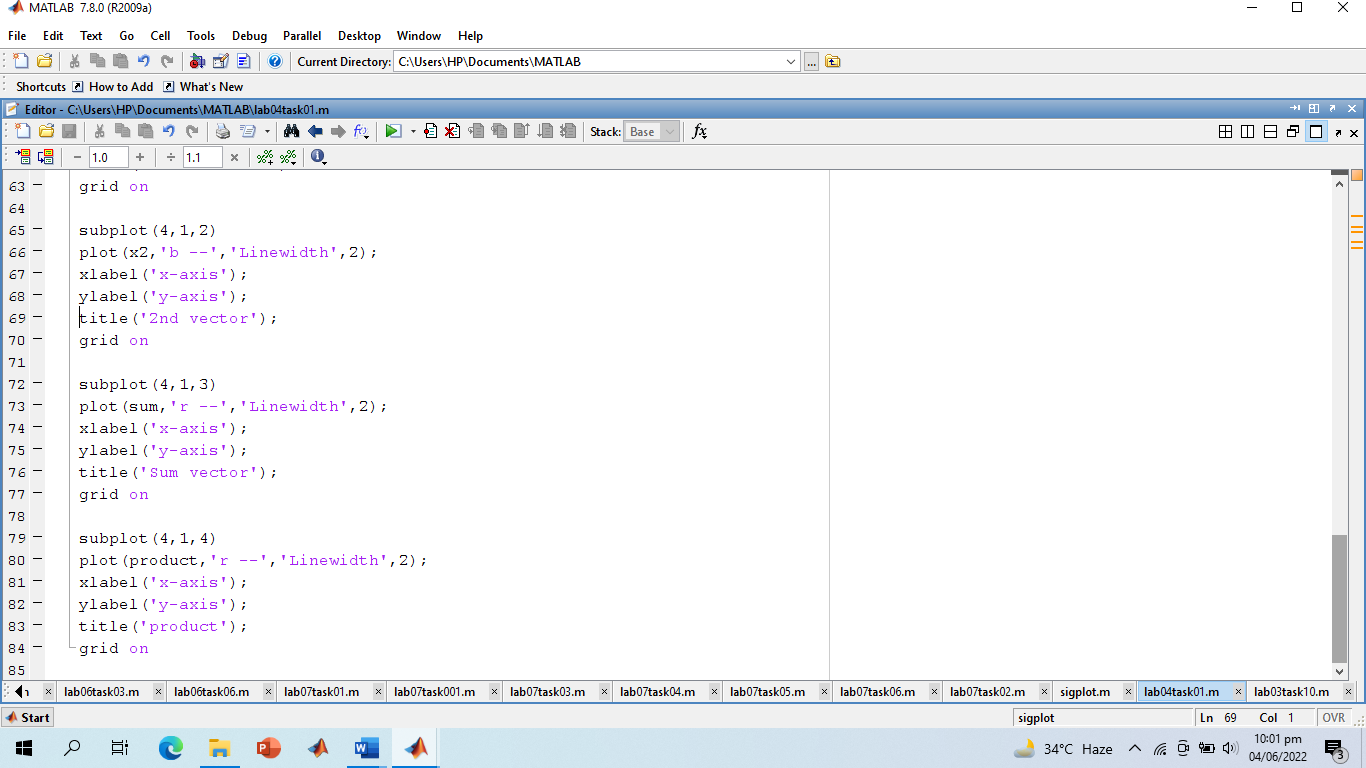
iii) Single Figure with separate signal plots using subplots.

**Screenshot of Input:**



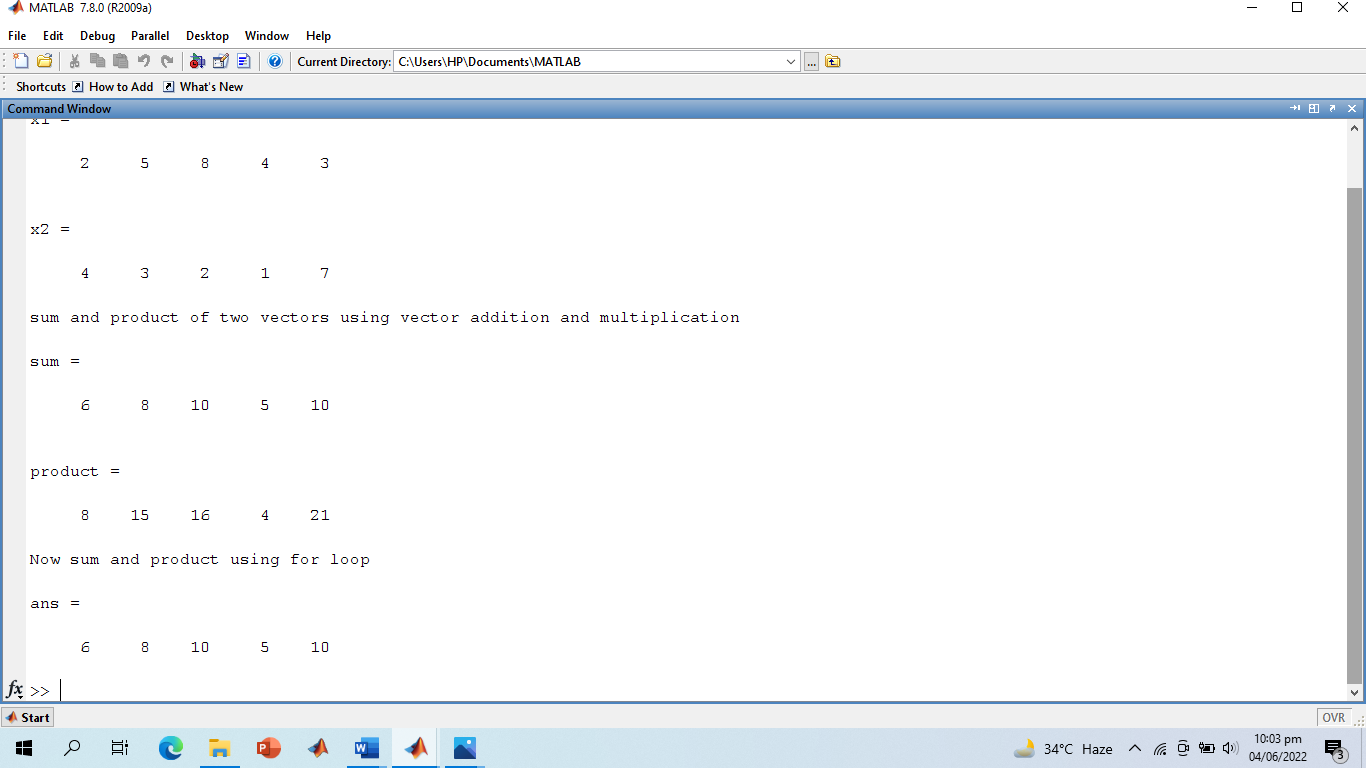






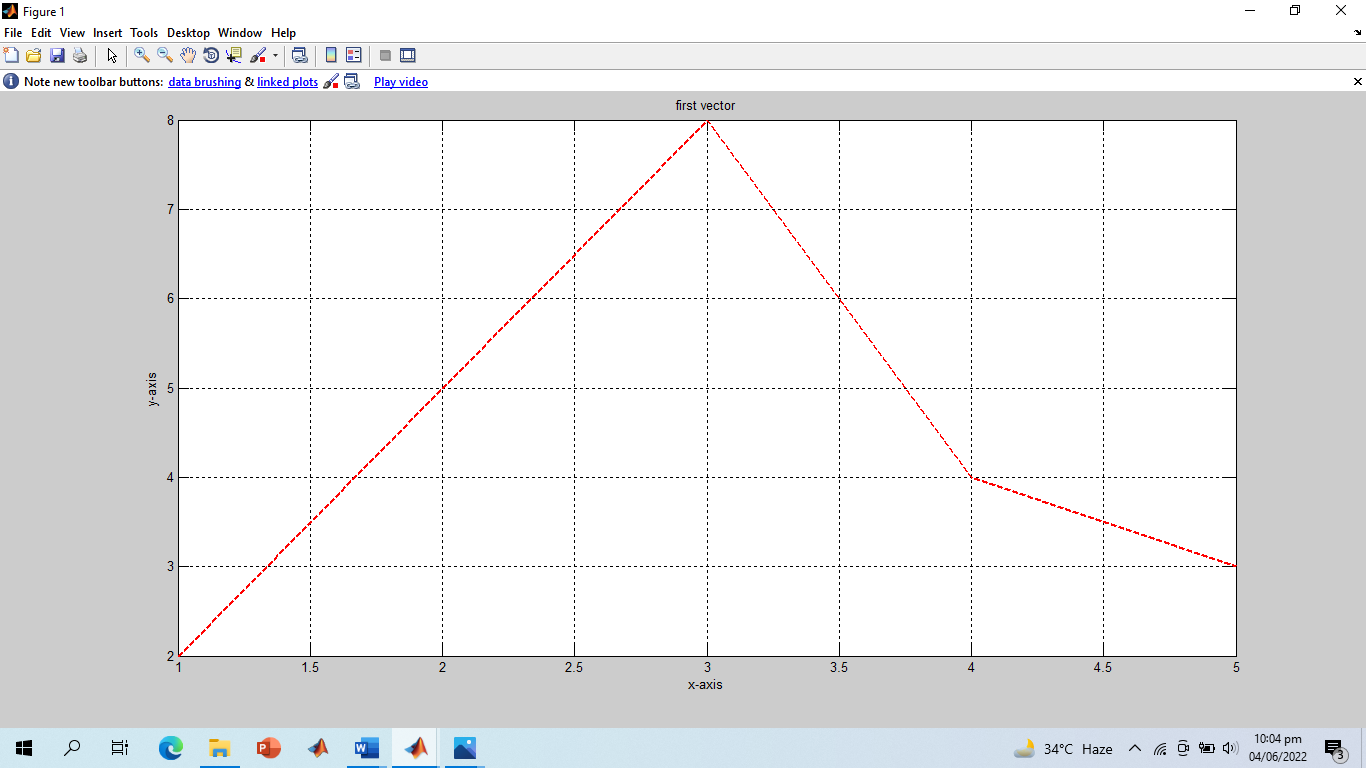
**Screenshot of output:**

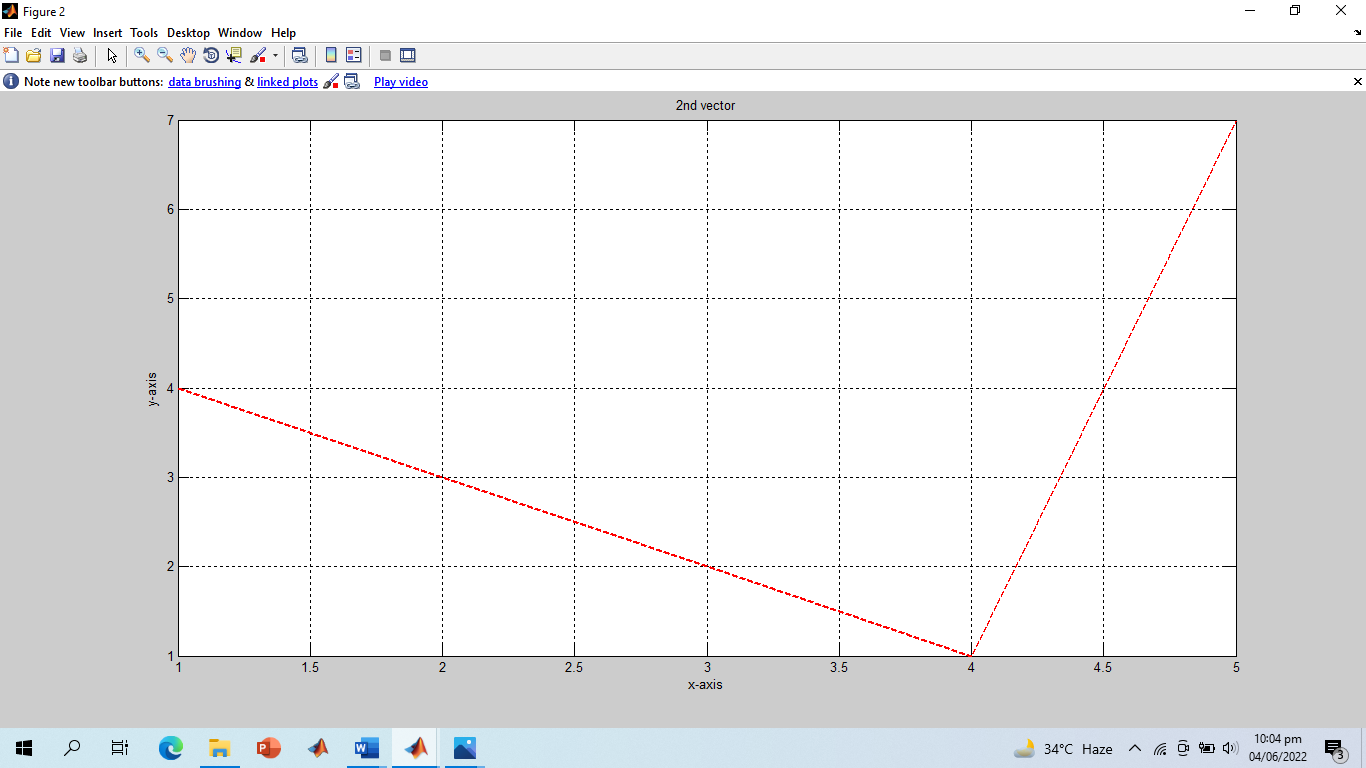


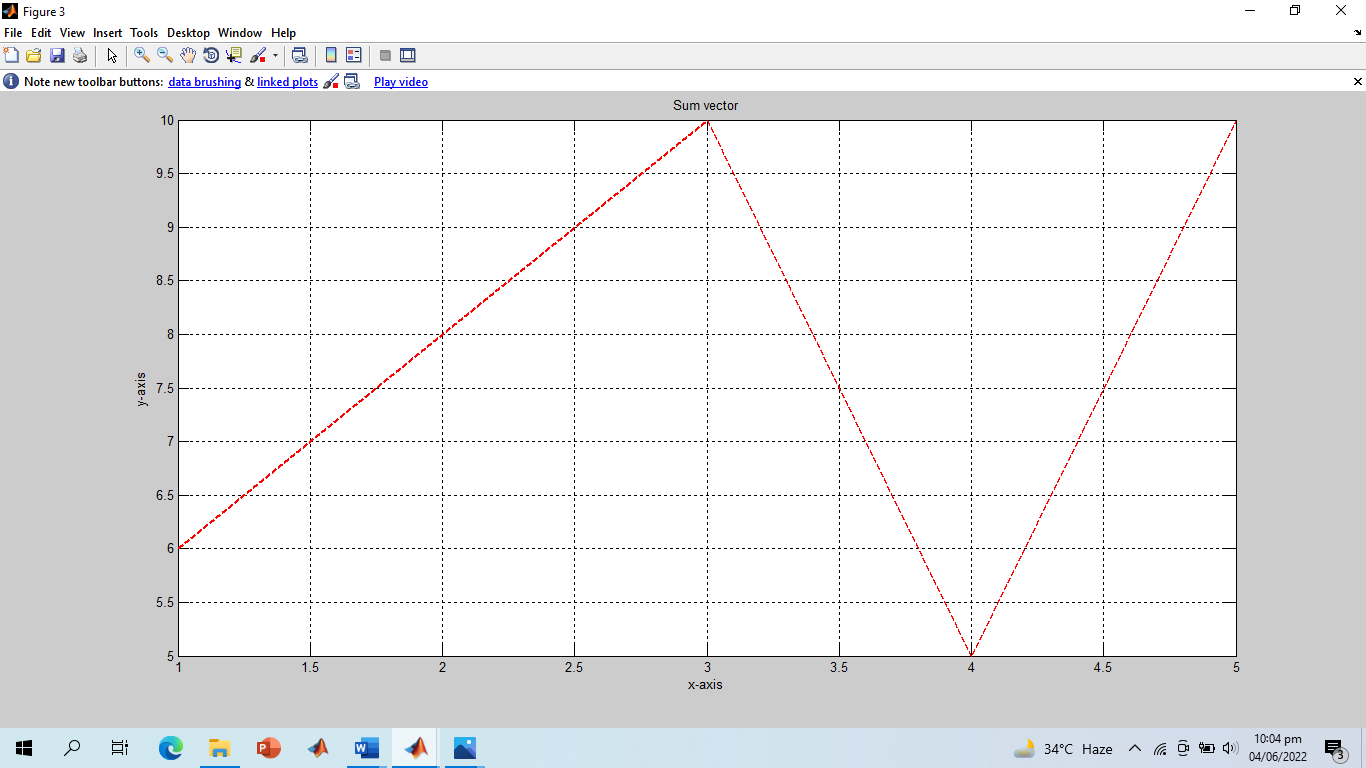


**Screenshot of Separate Figures:**

**1)**

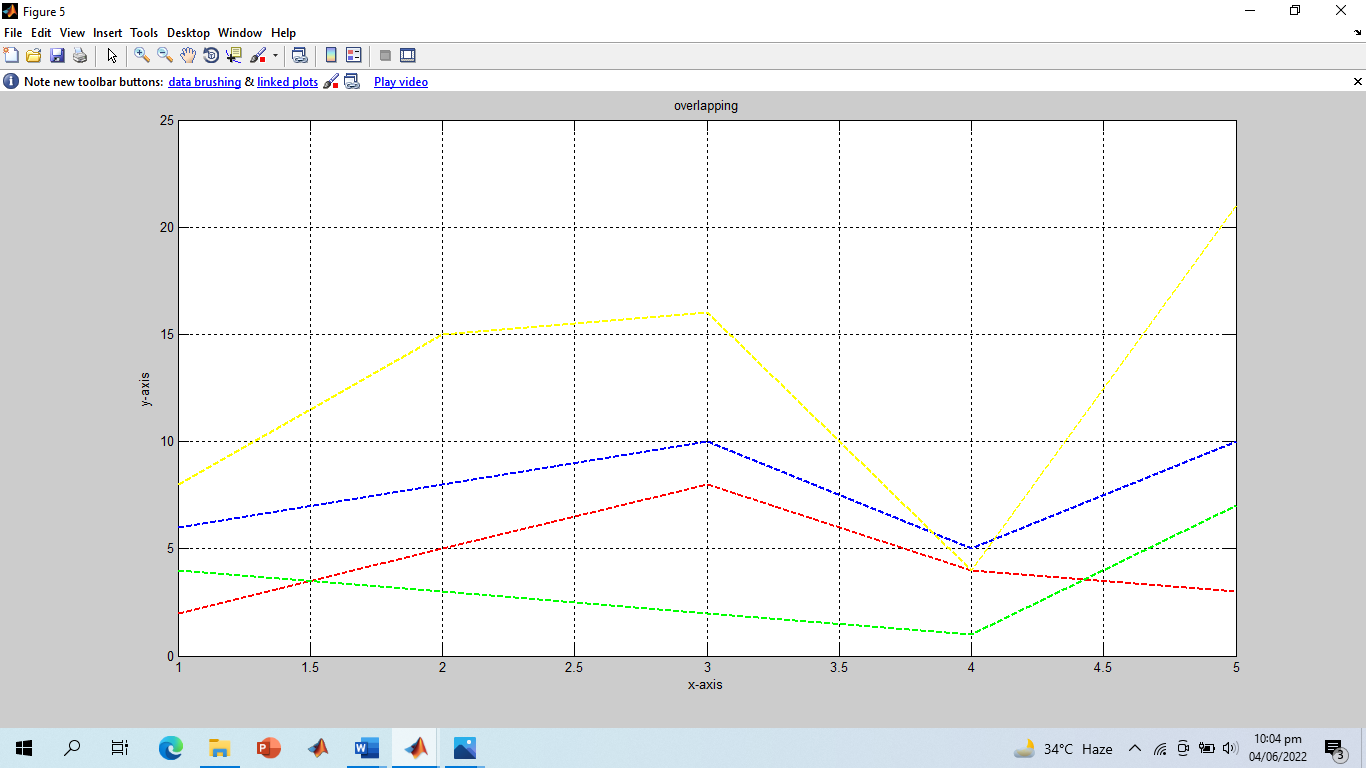




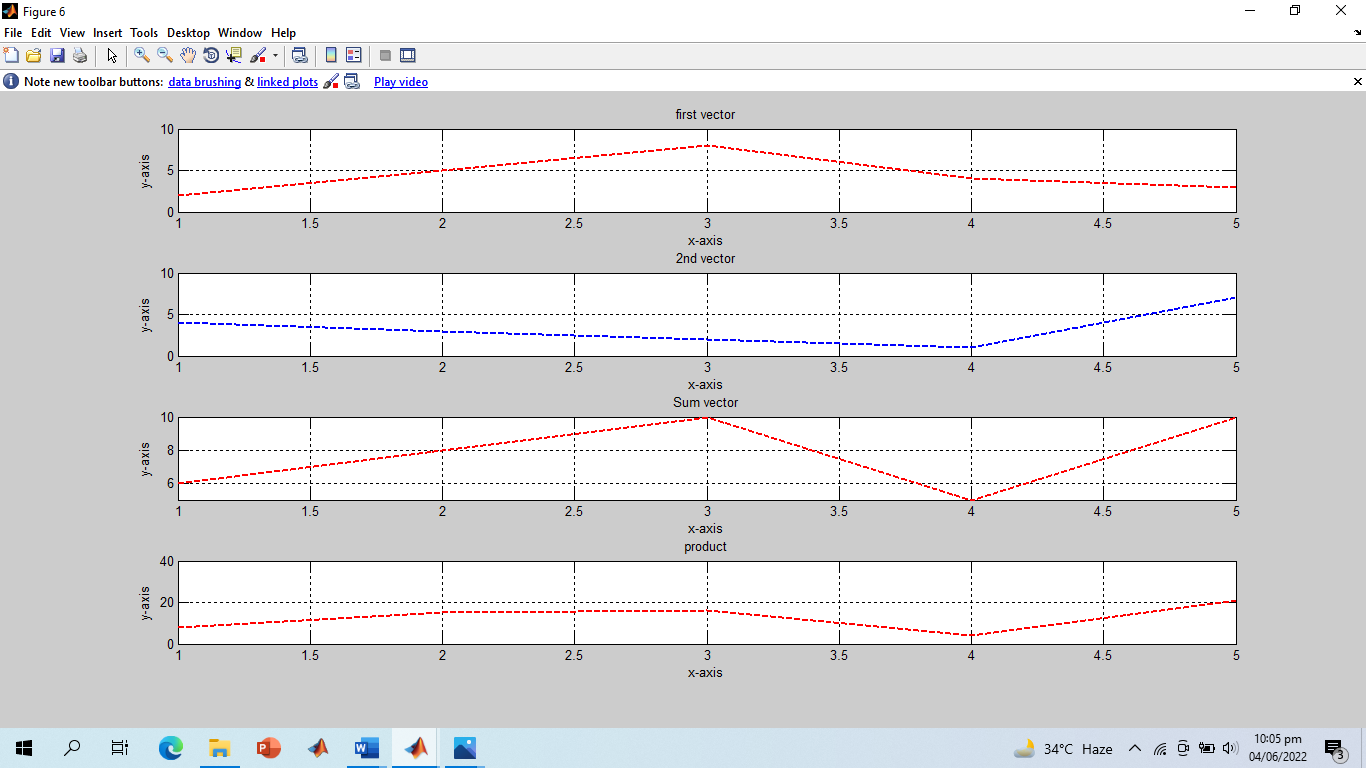




**2) Overlapping:**



**3) All Graphs in a Single Command:**



**---------------TASK 02--------------------------**

Amplitude scaling by a factor β causes each sample to get multiplied by β. Write a user‐defined function **ScaleSig** that has two input arguments: (i) a signal to be scaled and (ii) scaling factor β. The function should return the scaled output to the calling program. In the calling program, get the discrete time signal as well as the scaling factor from user and then call the above‐mentioned function.

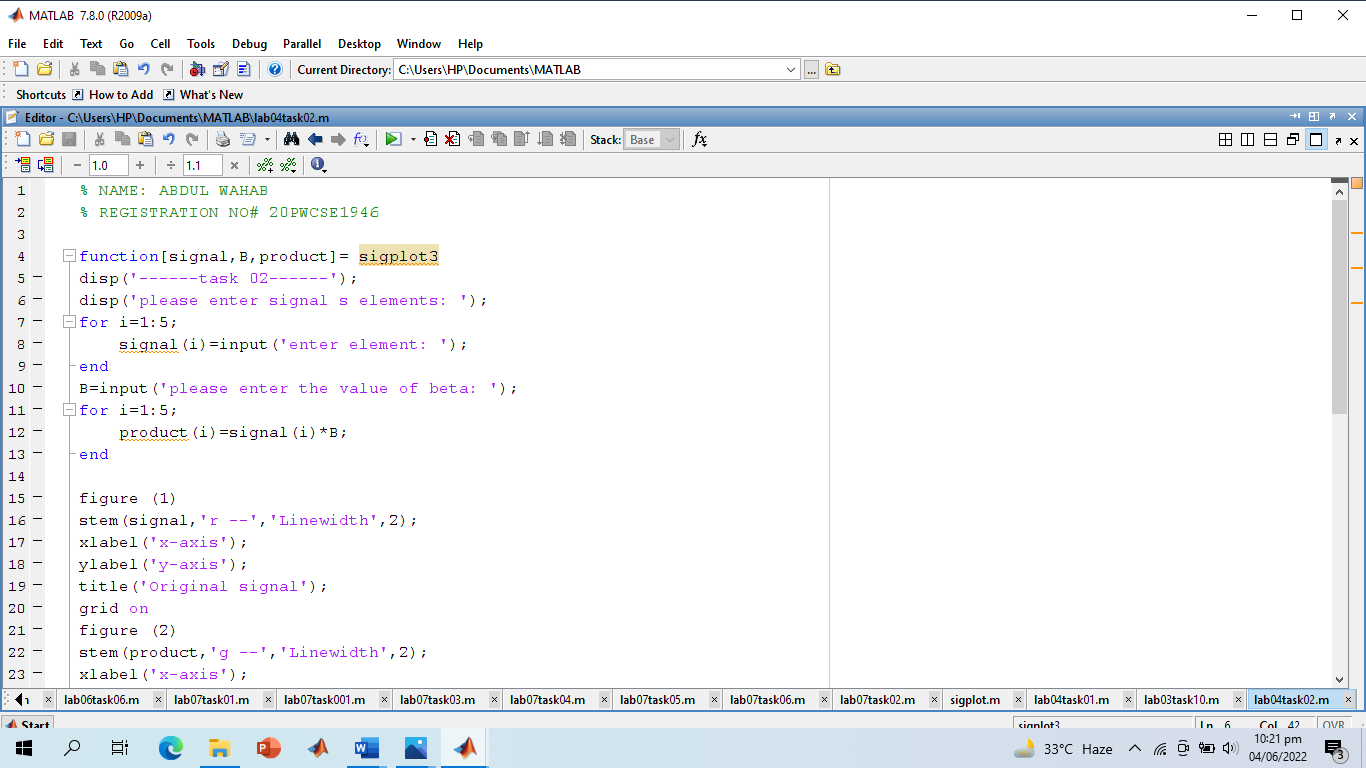
Design a function **SigPlot** that takes the original signals and their scaled versions as input from the main calling program and plots them as:

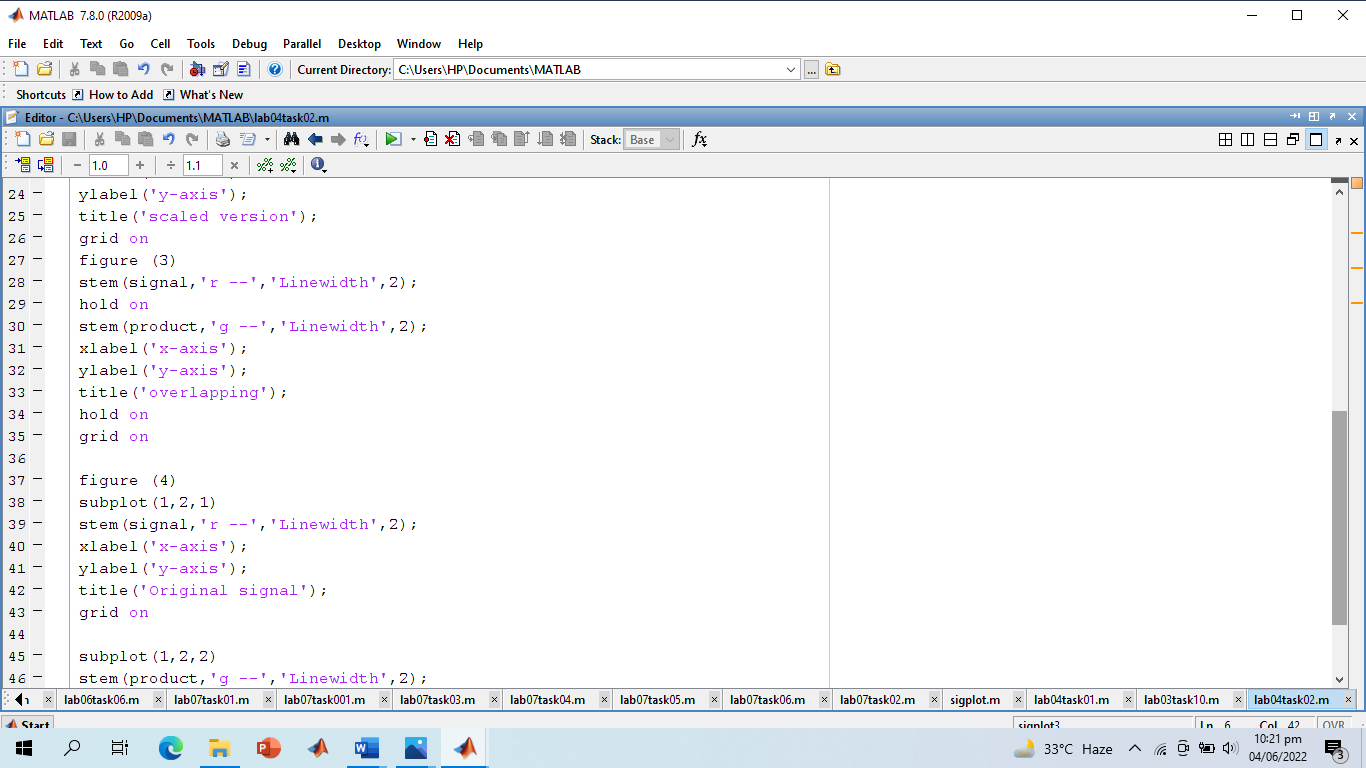
i) Separate Figures,

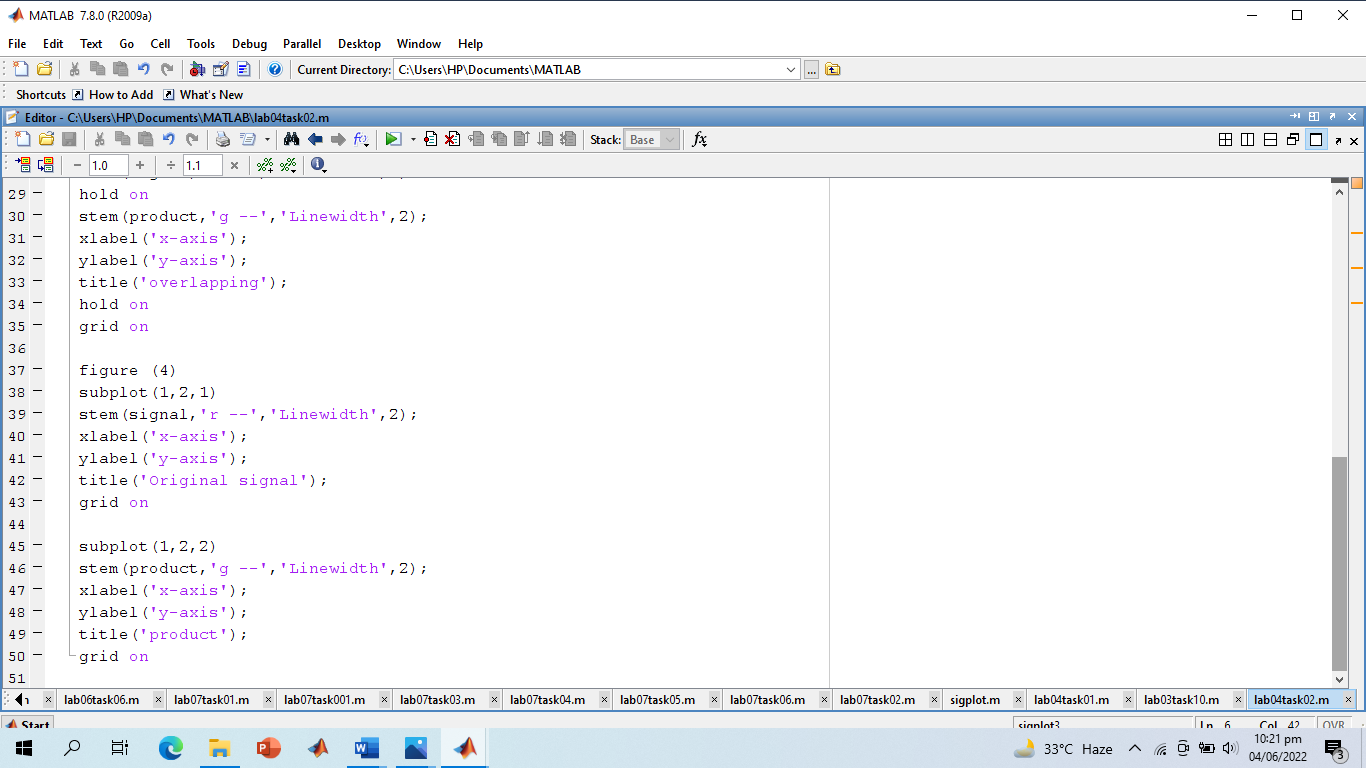
ii) Single Figure overlapping all the signals, and

iii) Single Figure with separate signal plots using subplot.

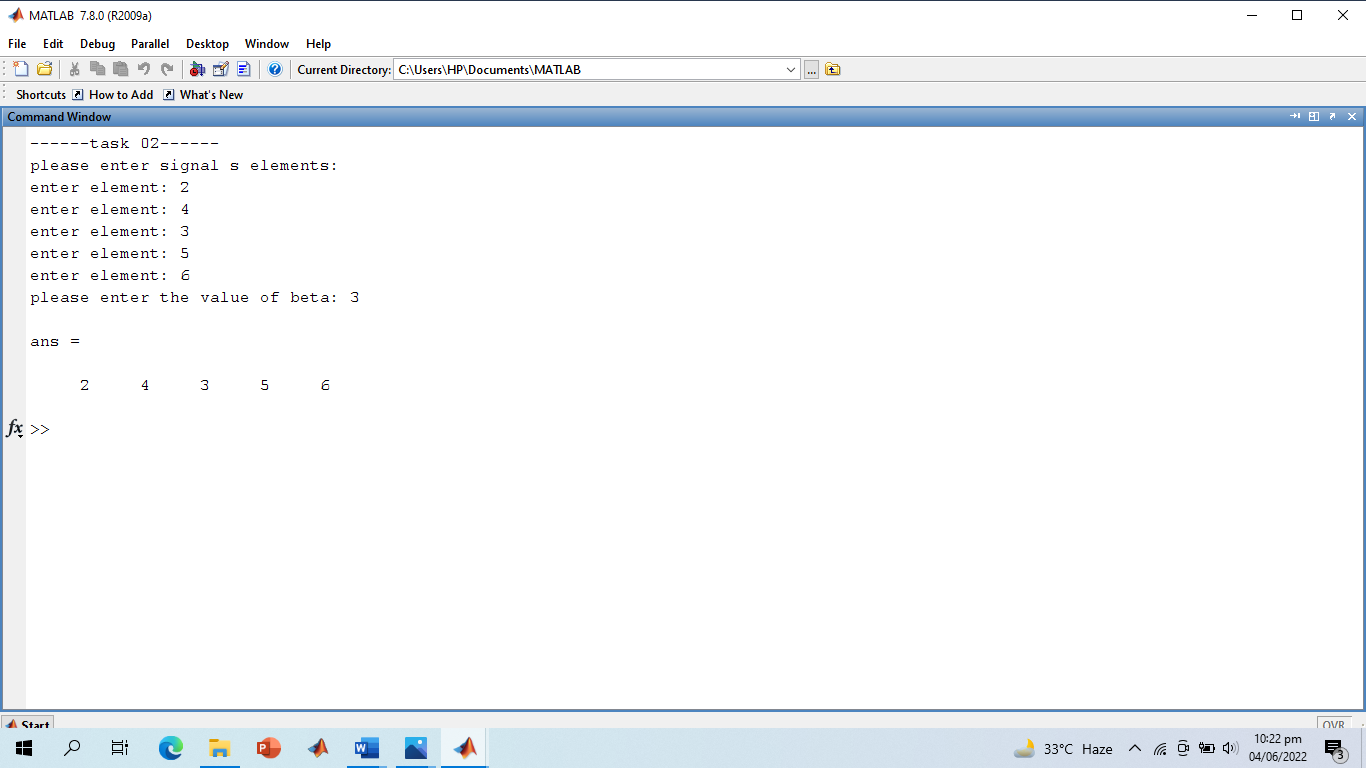
**Screenshot of Input:**





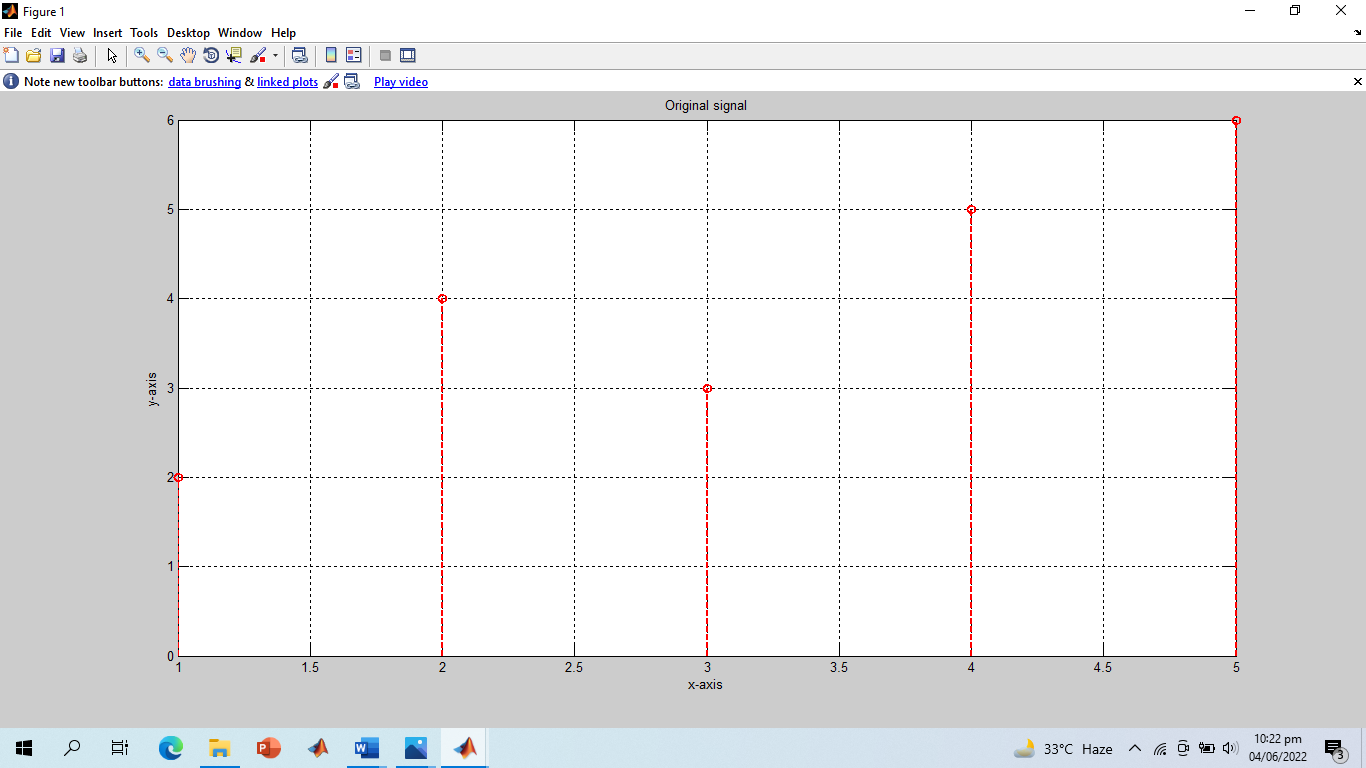


**Screenshot of Output:**



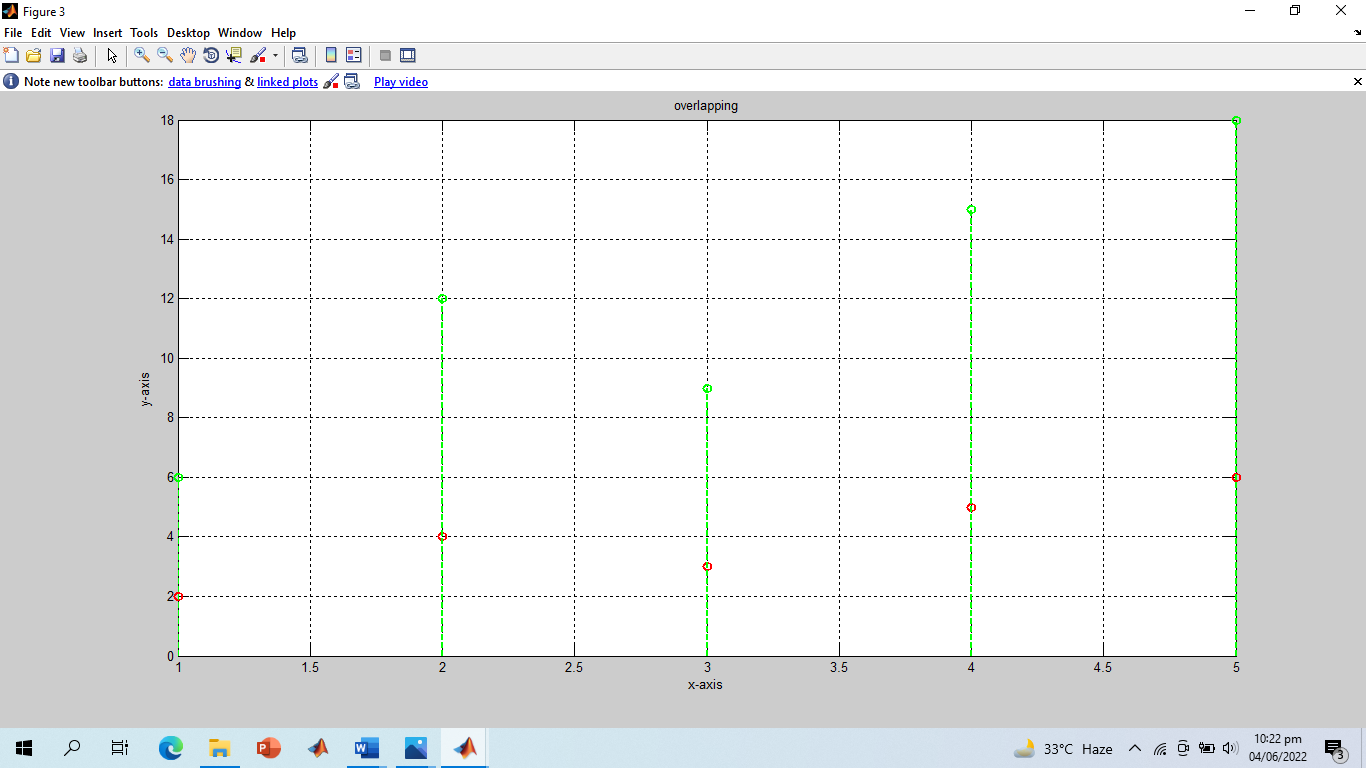
**Screenshot of Output Graphs:**

1. **Separate Graphs:**

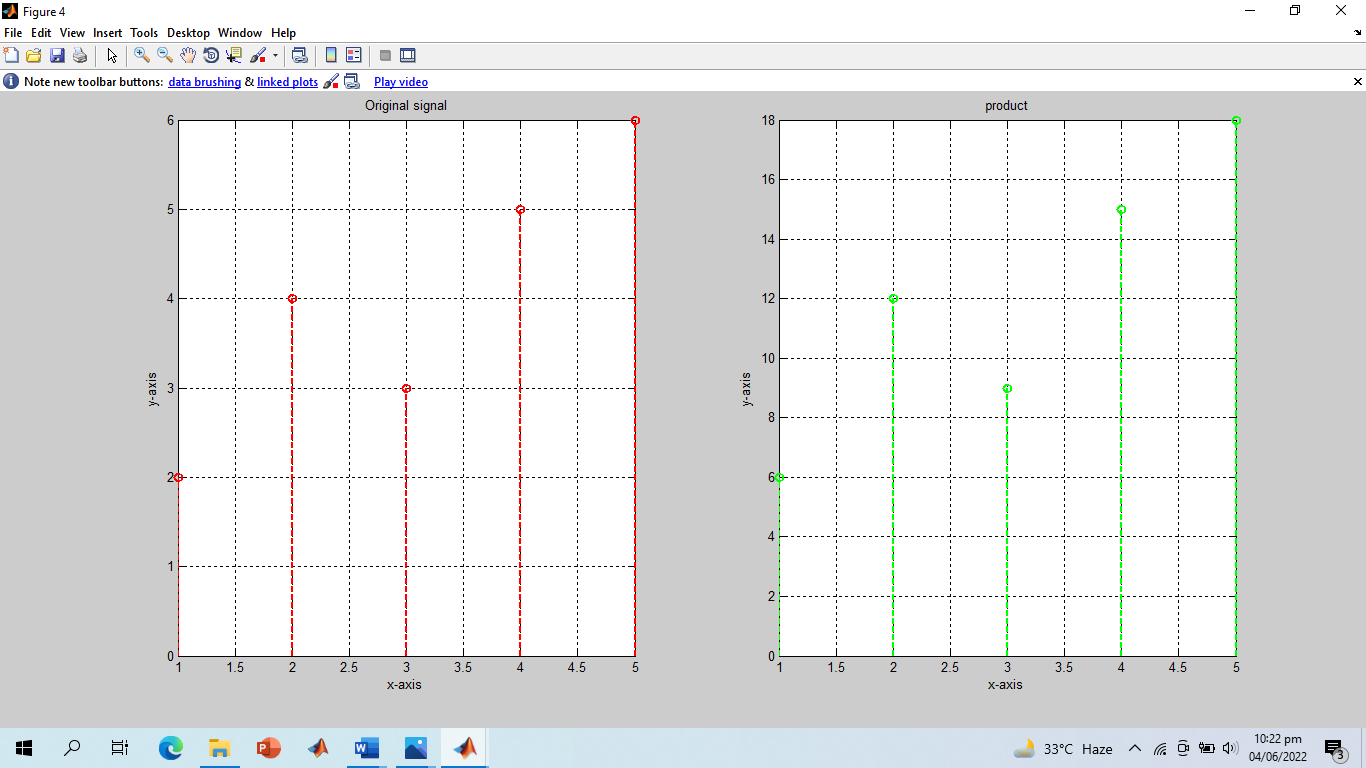




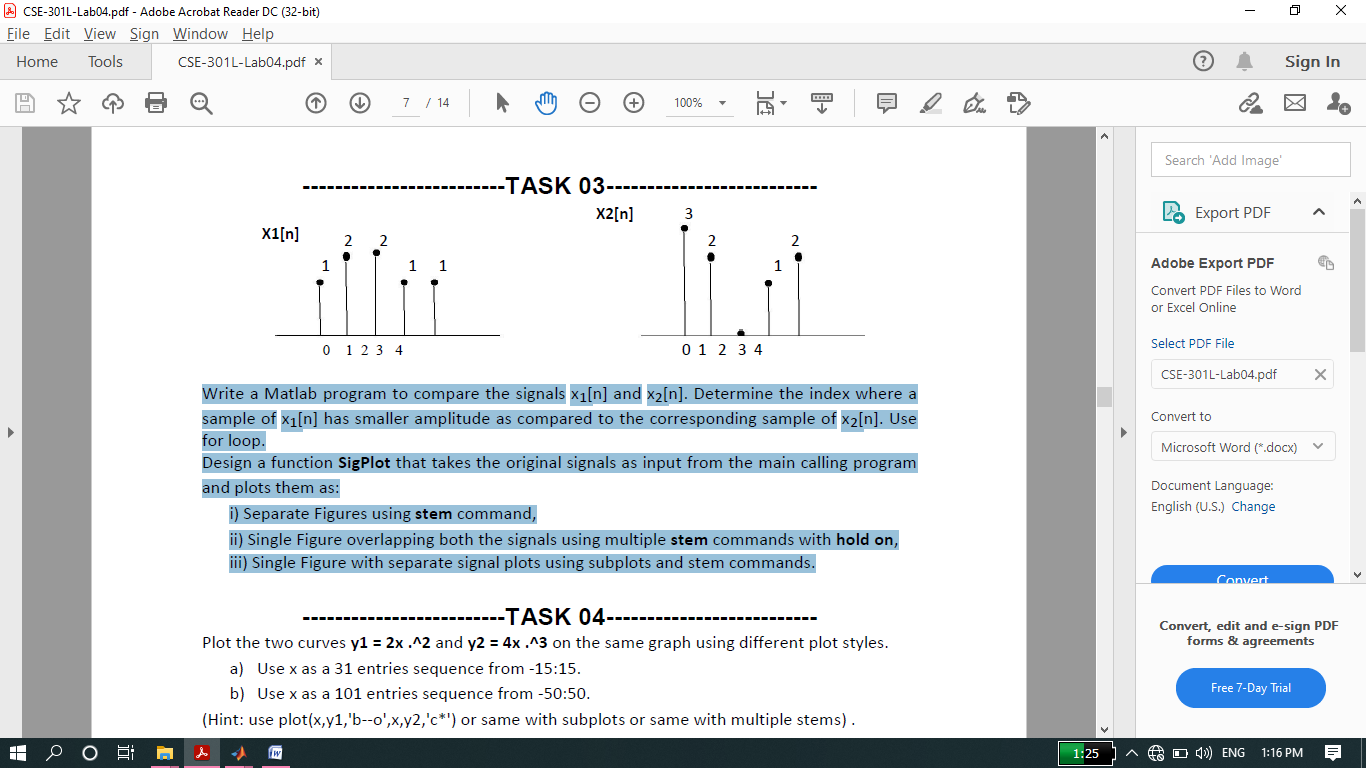
1. **Overlapping:**



1. **All in a Single Window:**



**-------------------------TASK 03--------------------------**



Write a Matlab program to compare the signals x1[n] and x2[n]. Determine the index where a sample of x1[n] has smaller amplitude as compared to the corresponding sample of x2[n]. Use for loop.

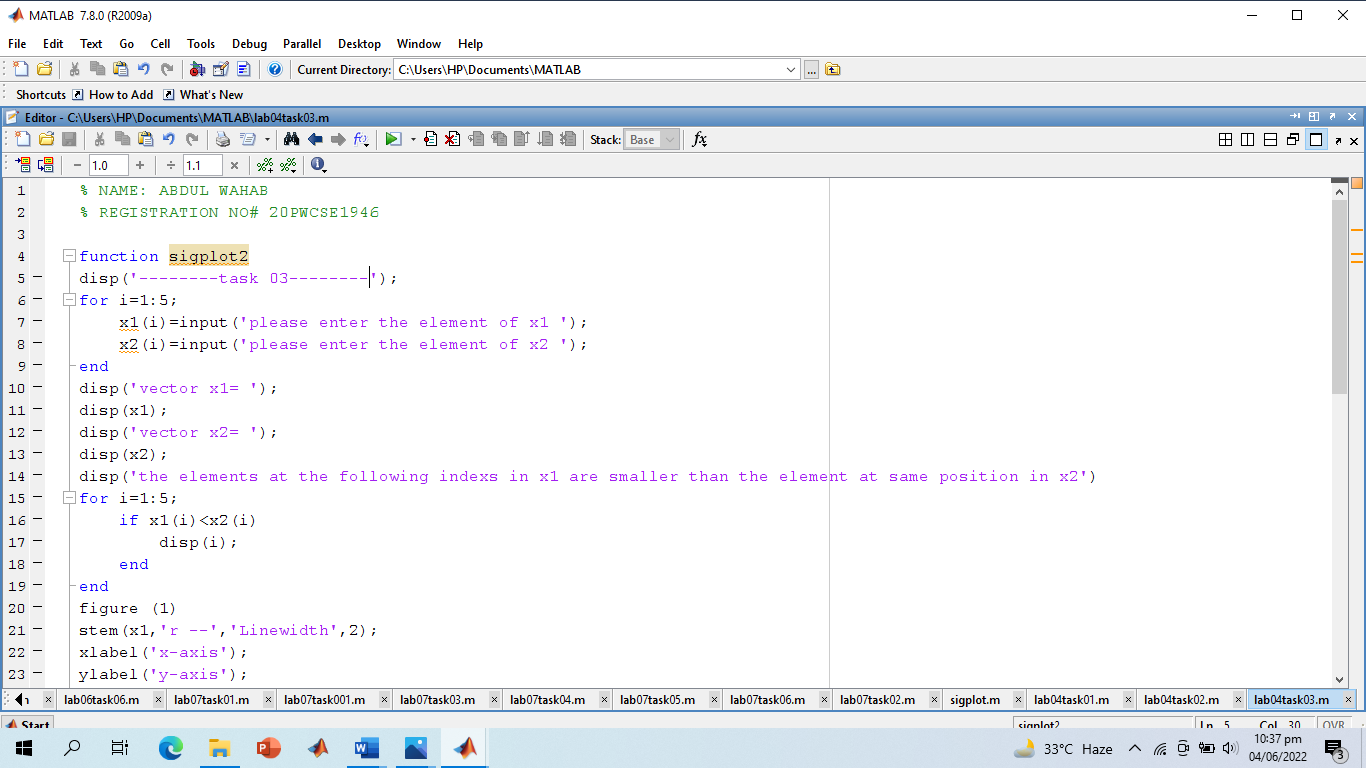
Design a function **SigPlot** that takes the original signals as input from the main calling program and plots them as:

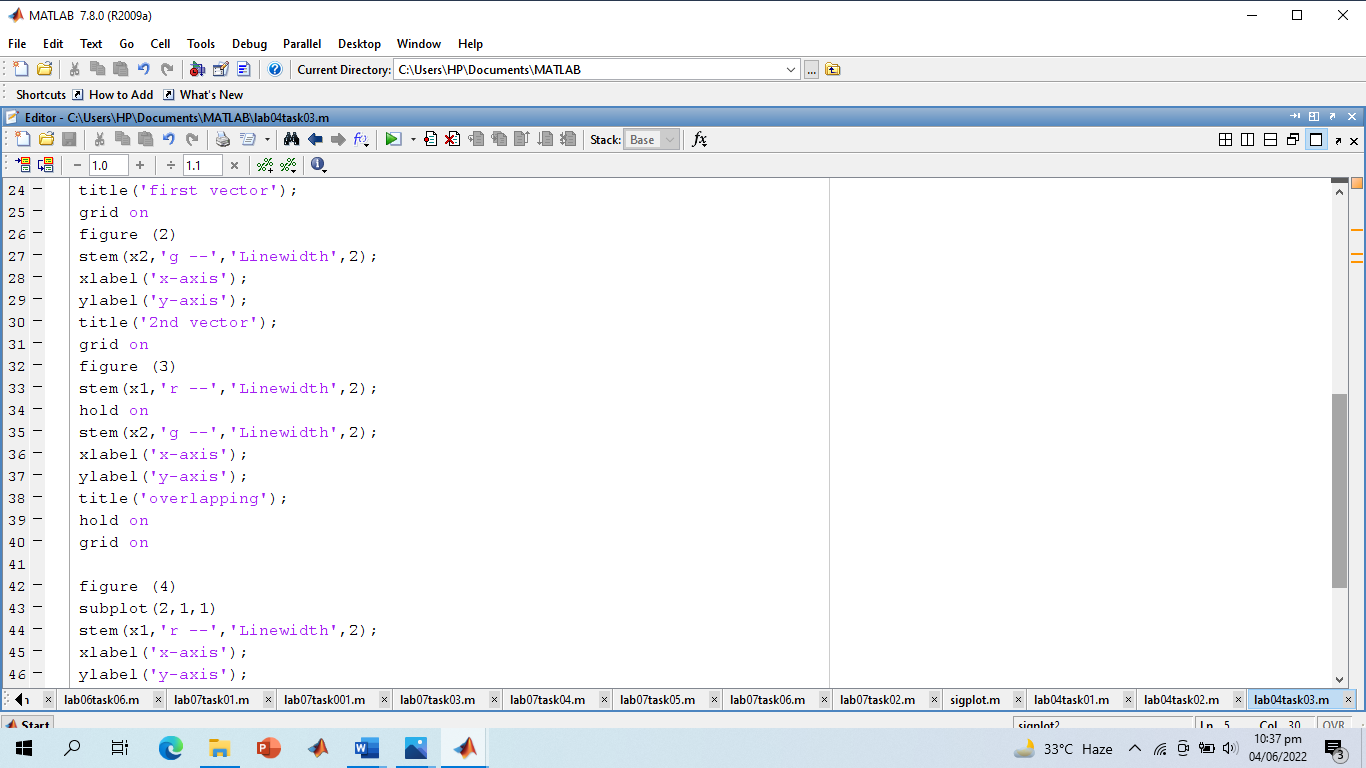
i) Separate Figures using **stem** command,

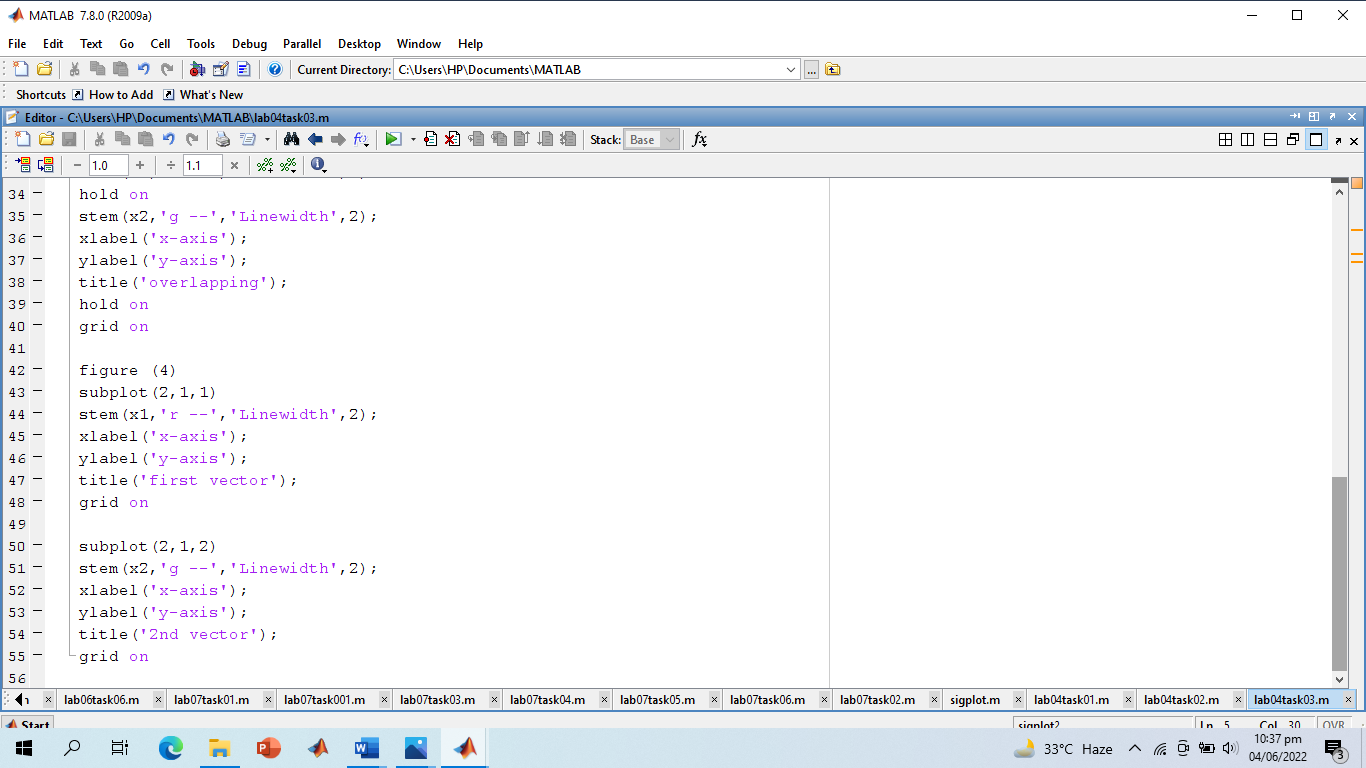
ii) Single Figure overlapping both the signals using multiple **stem** commands with **hold on**,

iii) Single Figure with separate signal plots using subplots and stem commands.

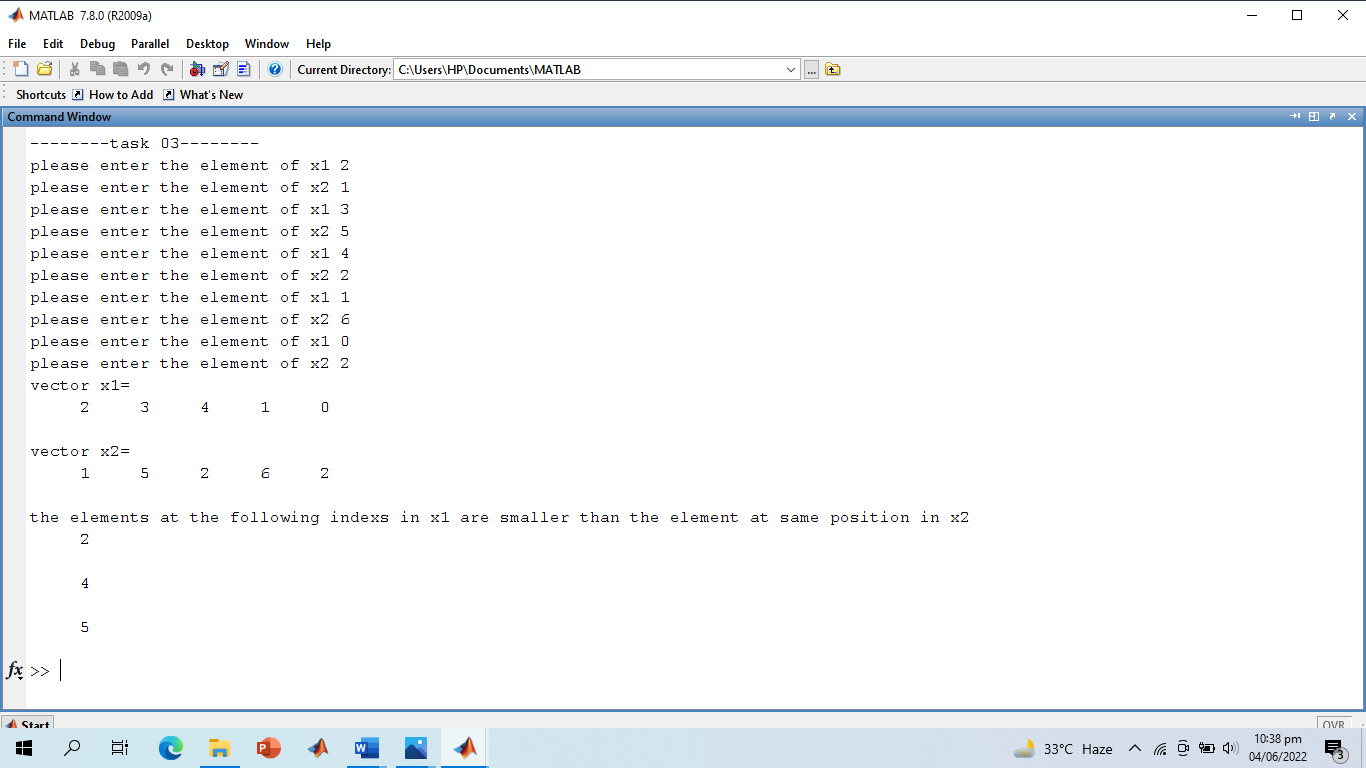
**Screenshot of Input:**





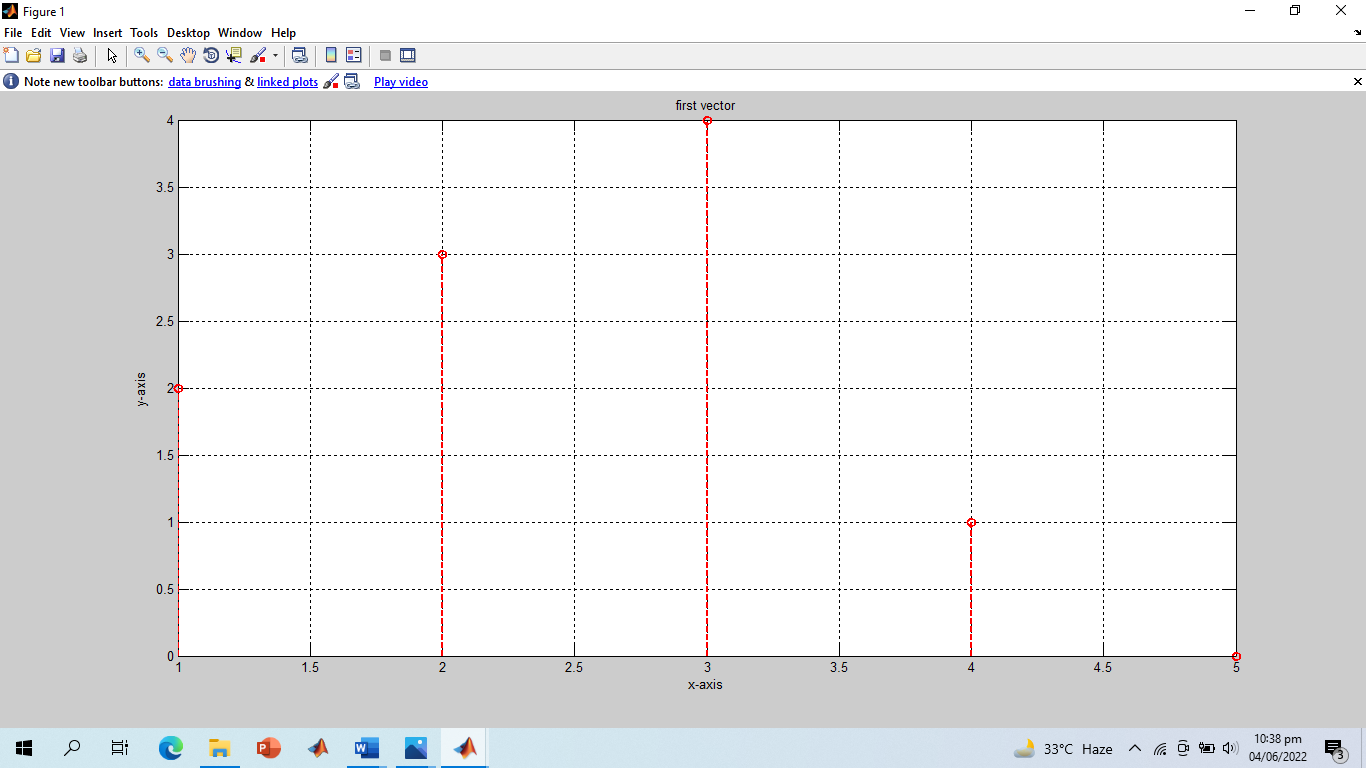


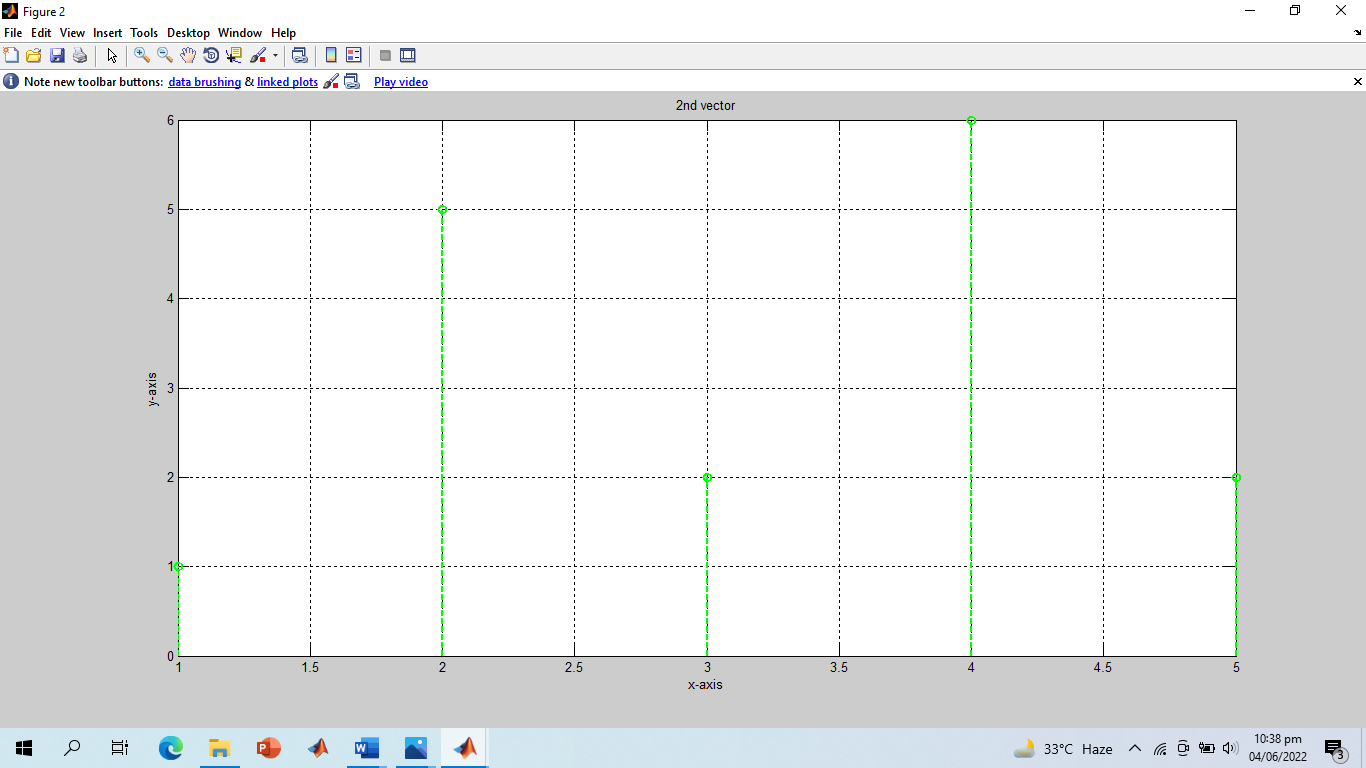
**Screenshot of Output:**



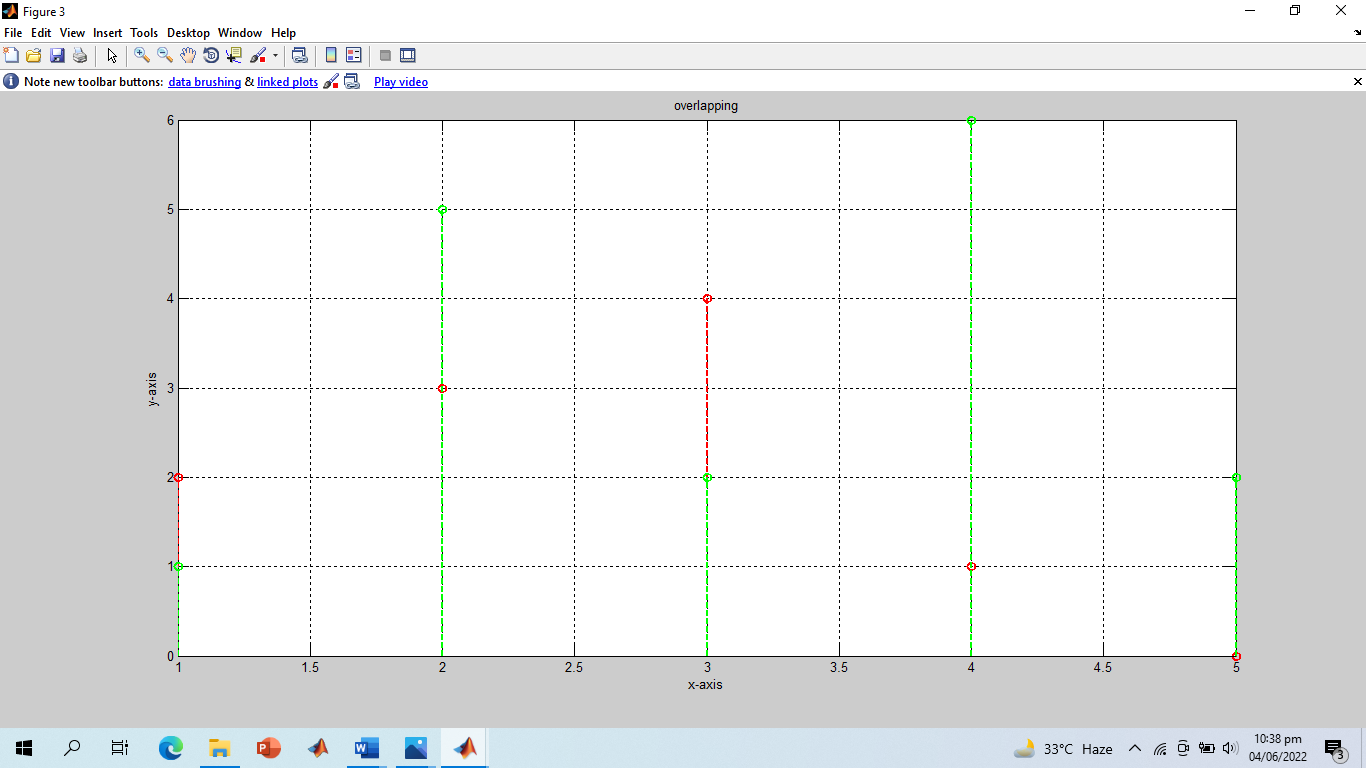
**Screenshot of Output Graphs**

1. **Separate Graph:**

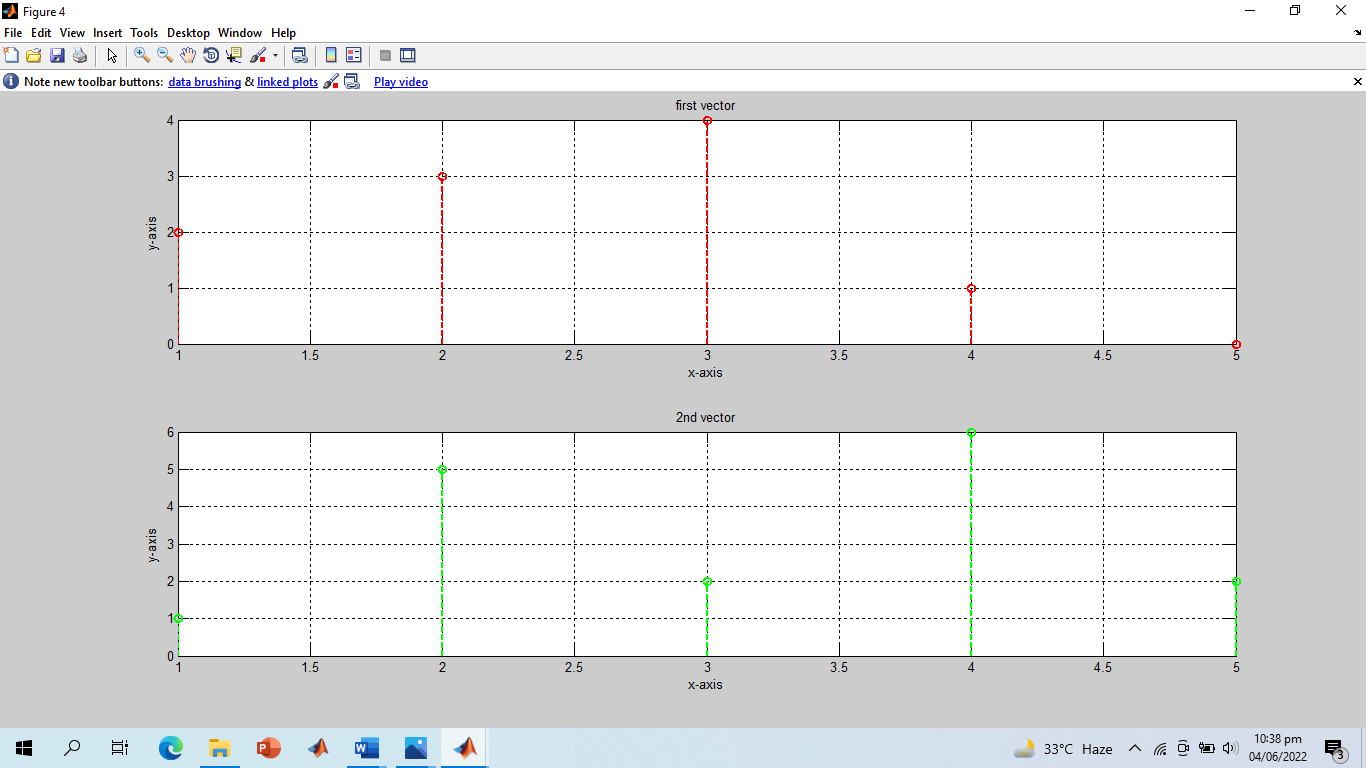




1. **Overlapping:**



1. **All Graphs in a Single Window:**



**-------------------------TASK 04--------------------------**

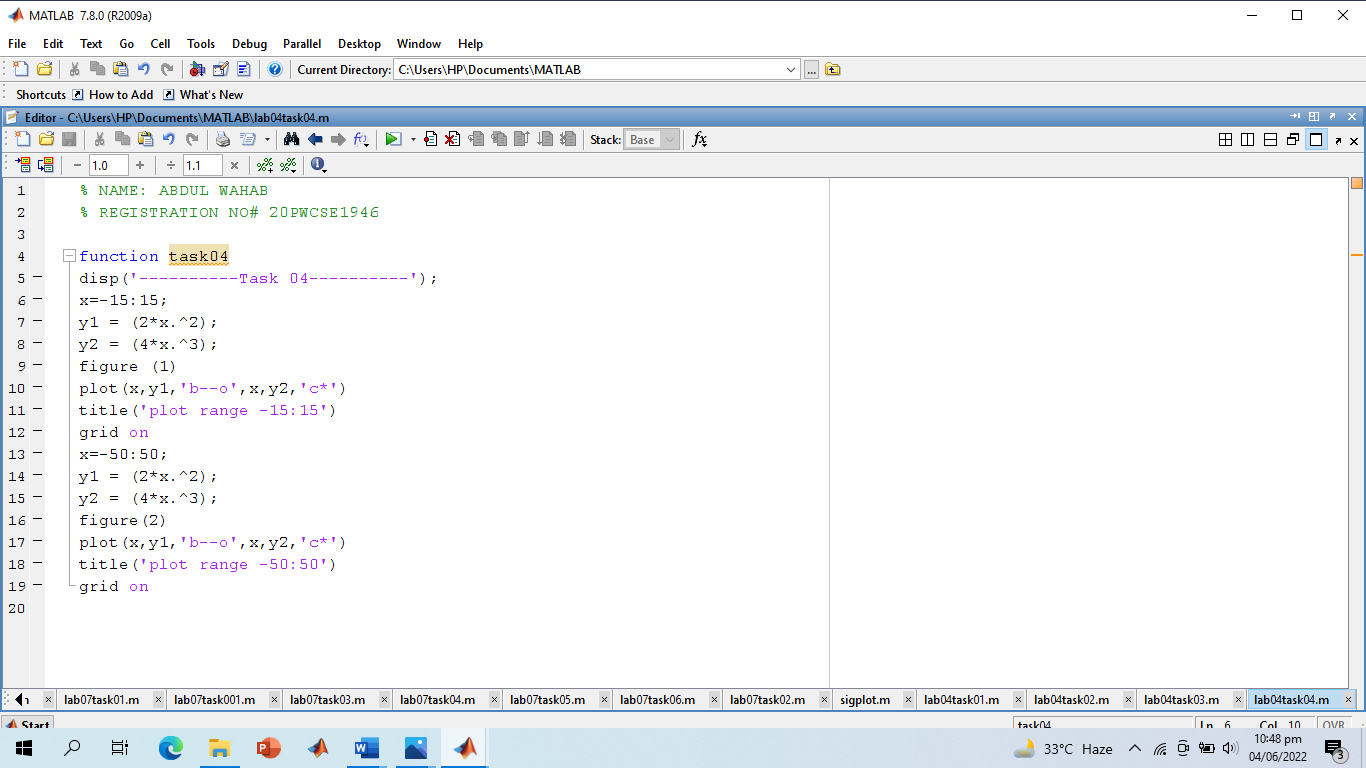
Plot the two curves **y1 = 2x.^2** and **y2 = 4x.^3** on the same graph using different plot styles.

a) Use x as a 31 entries sequence from -15:15.

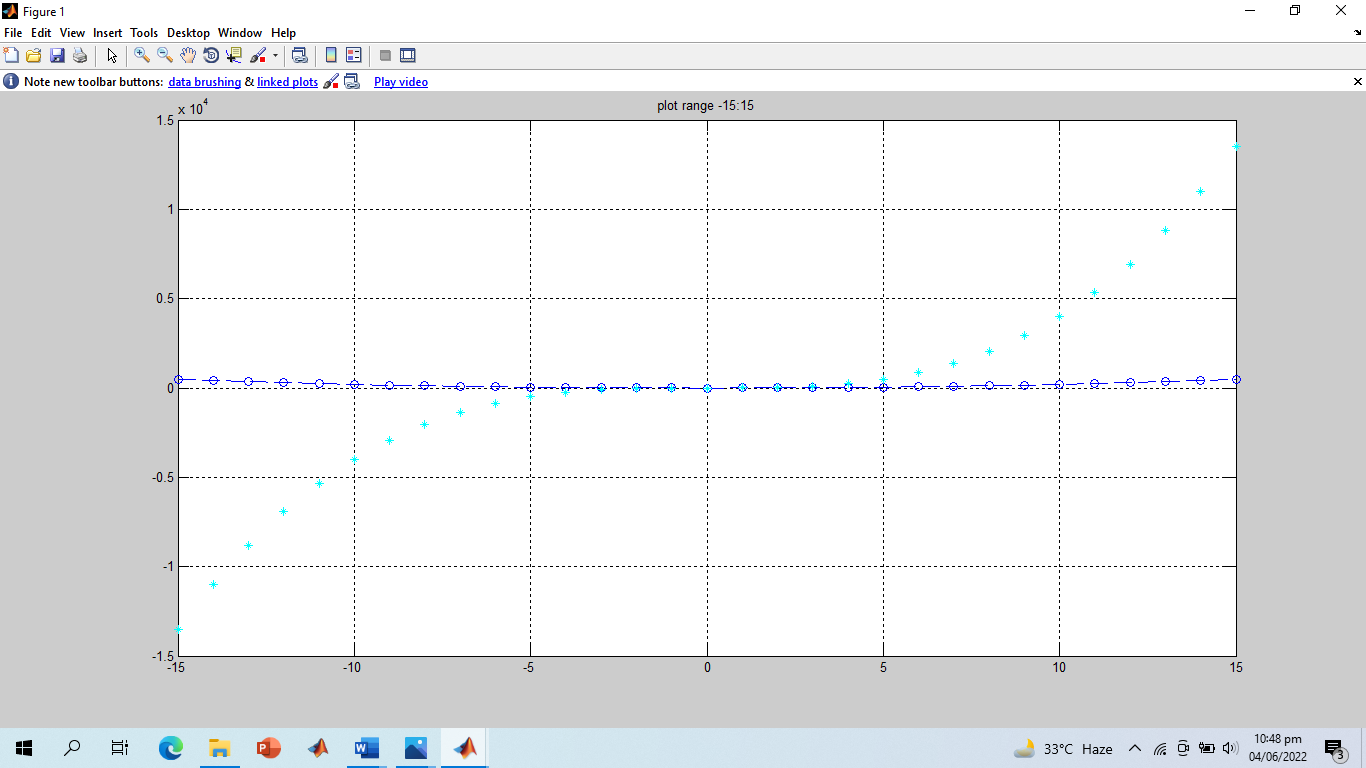
b) Use x as a 101 entries sequence from -50:50.

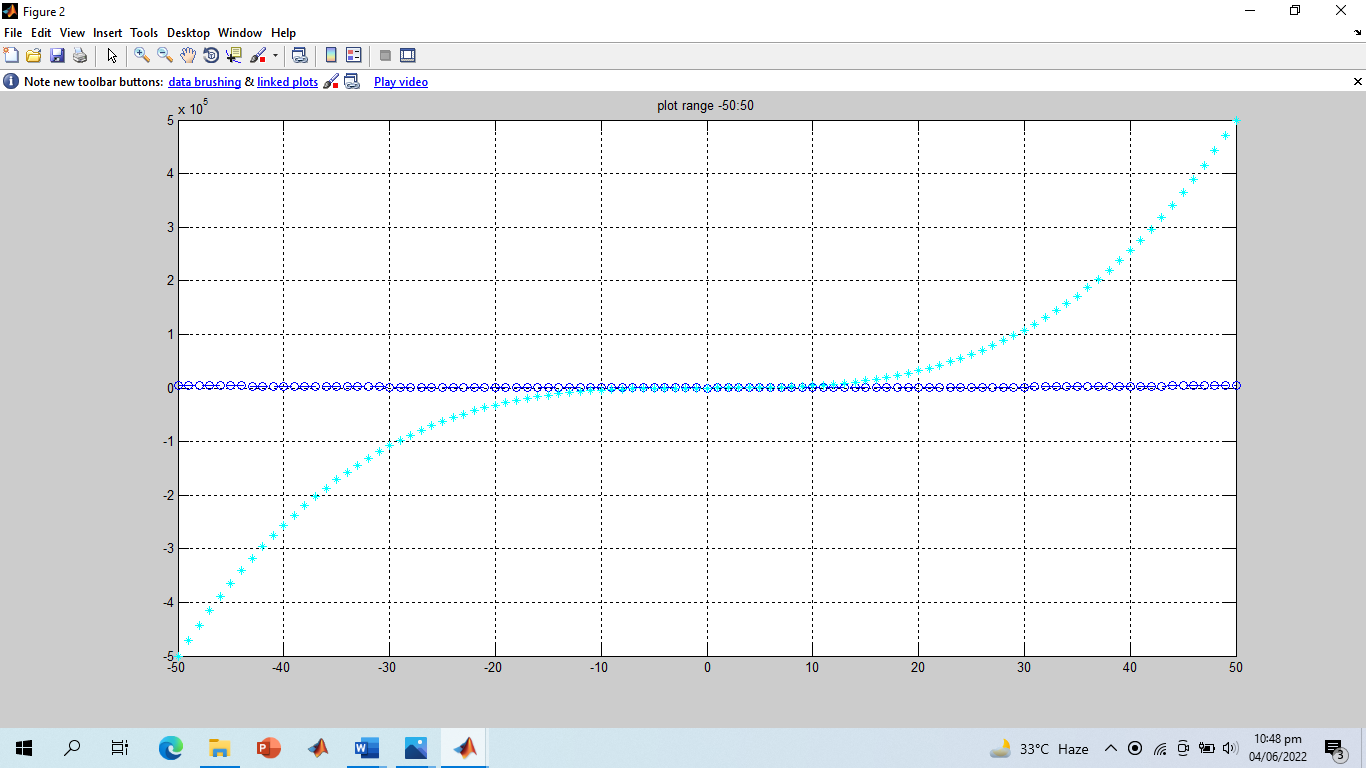
(Hint: use plot (x, y1,'b--o’, x, y2,'c\*') or same with subplots or same with multiple stems).

**Screenshot of Input:**



**Screenshot of Output:**



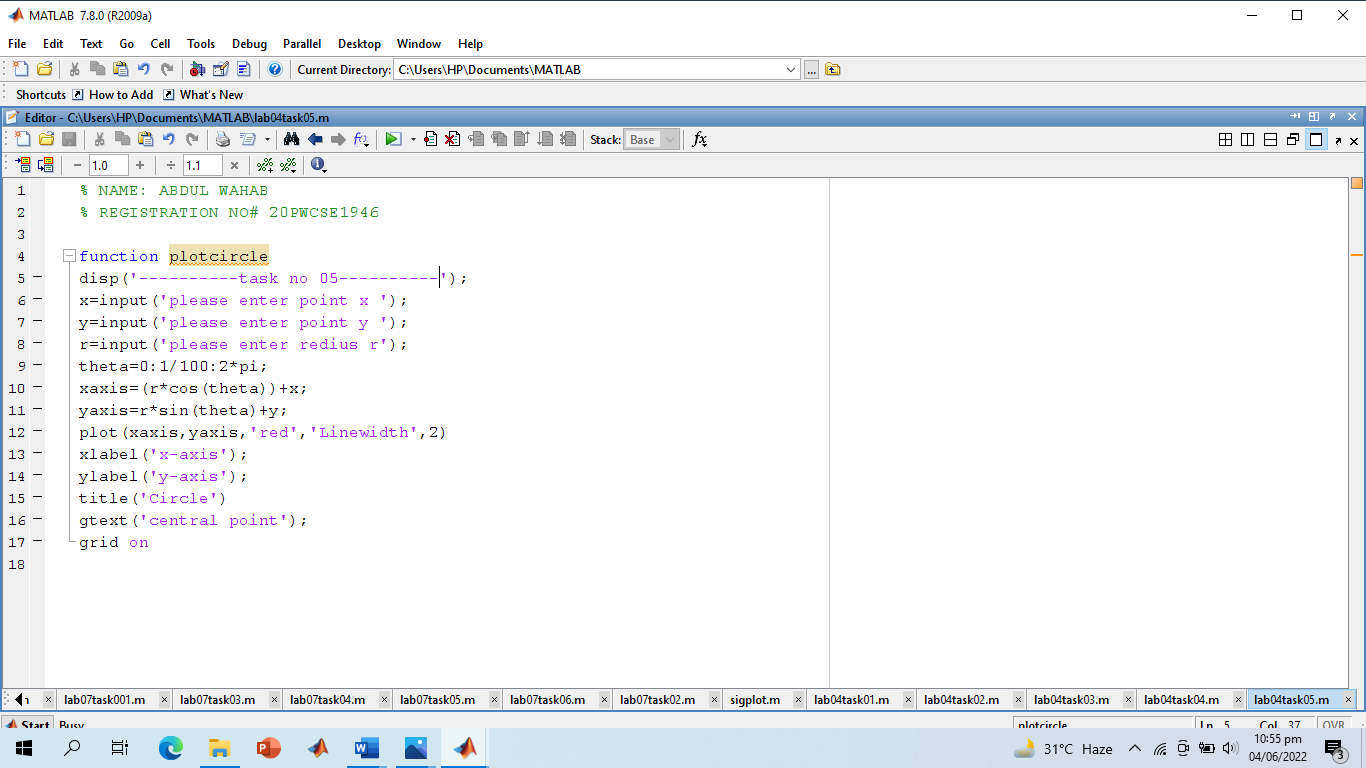


**-------------------------TASK 05--------------------------**

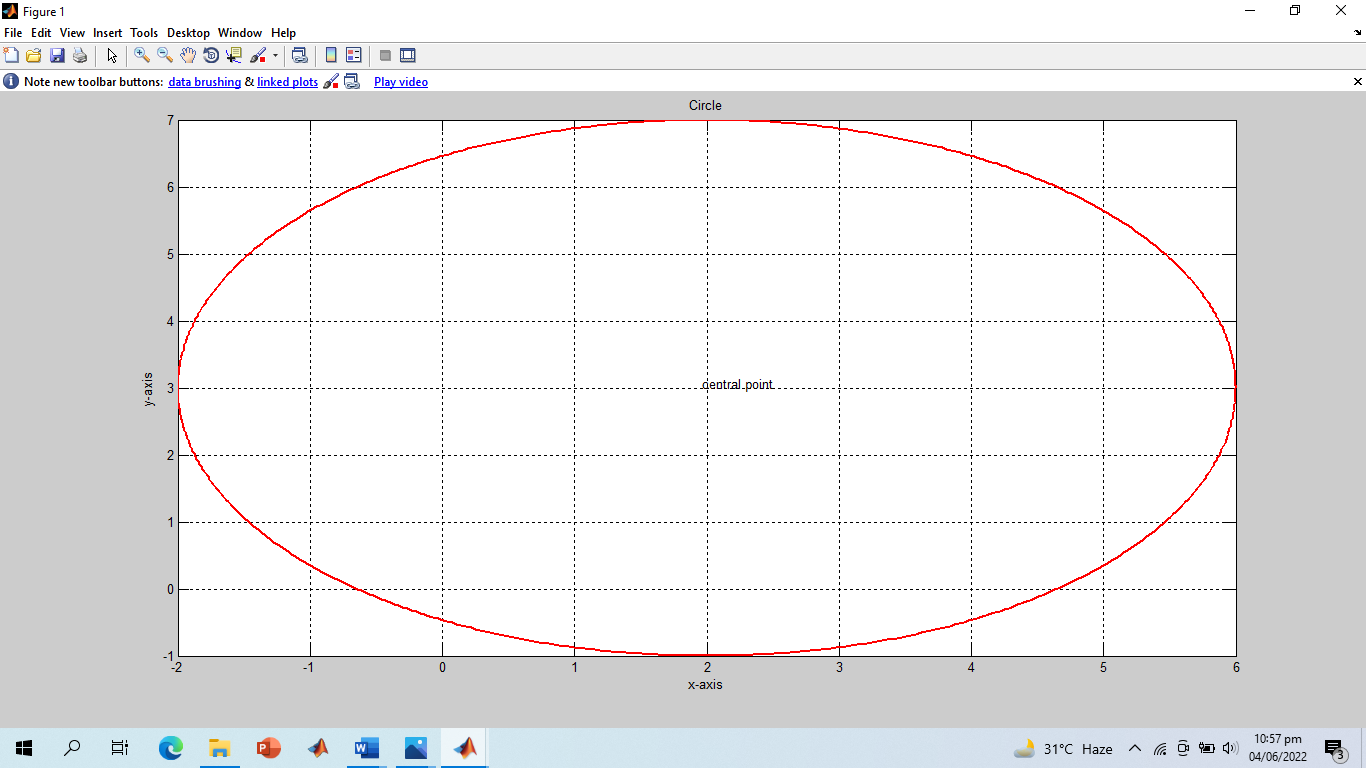
Create a function **PlotCircle** that takes points x, y and radius r from user as inputs and generates a graph of circle centered at point (x, y) with a radius equal to r. Use **axis equal** to use equal data units along each coordinate direction and use **axis square** to view square axis.

(Hint: use circle equation: x-axis = r\*cos(theta)+x; y-axis=r\*sin(theta)+y; where theta=0:1/100:2\*pi and plot x-axis versus y-axis)

**Screenshot of Input:**



**Screenshot of Output:**



**-------------------------TASK 06--------------------------**

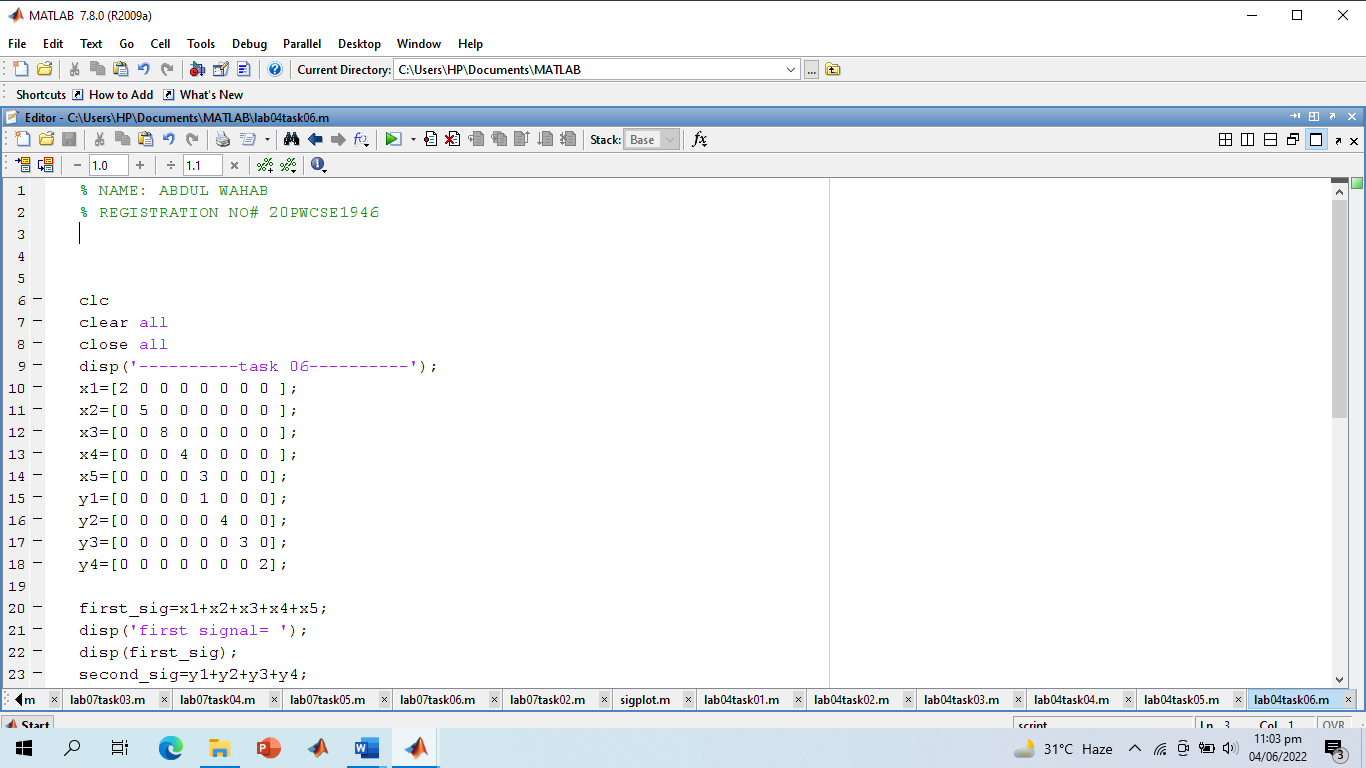
Given the signals:

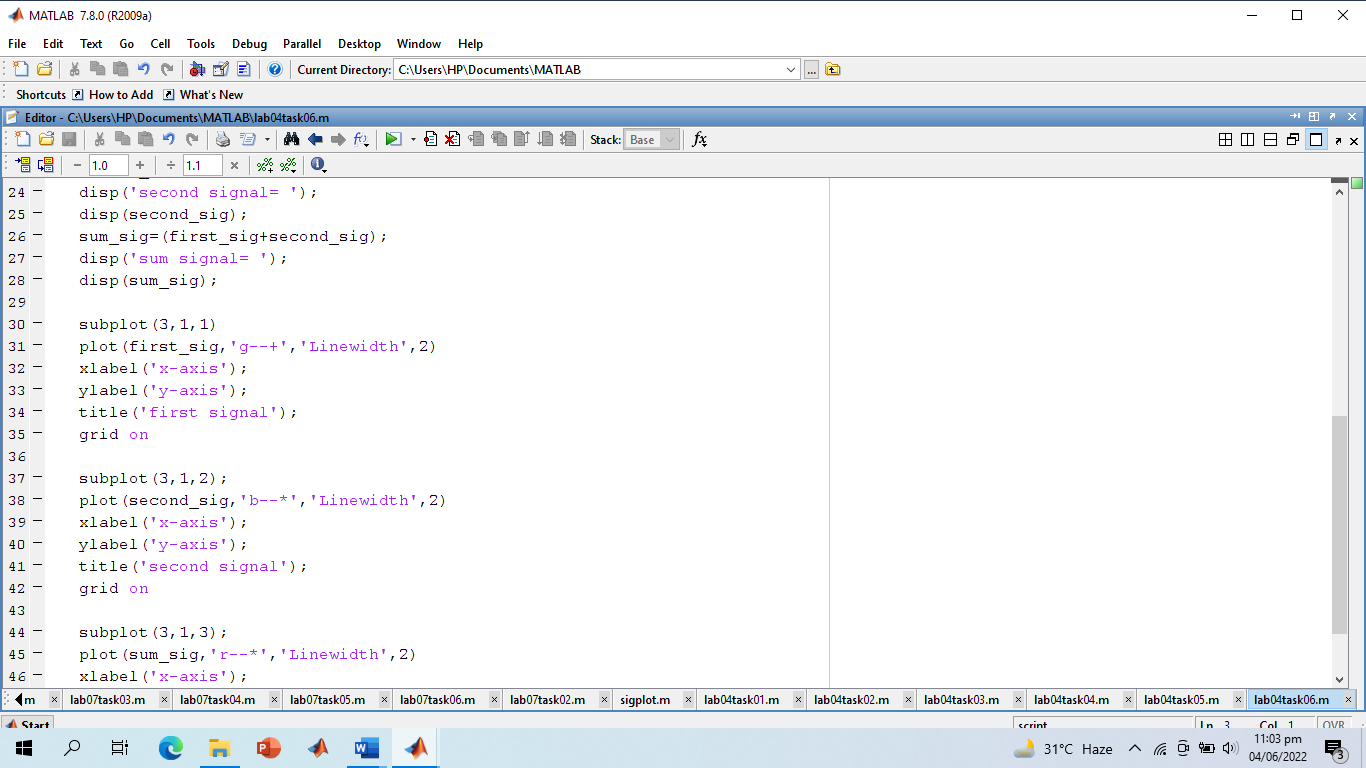
X1[n] = 2δ[n] + 5δ[n‐1] + 8δ[n‐2] + 4δ[n‐3] + 3δ[n‐4]

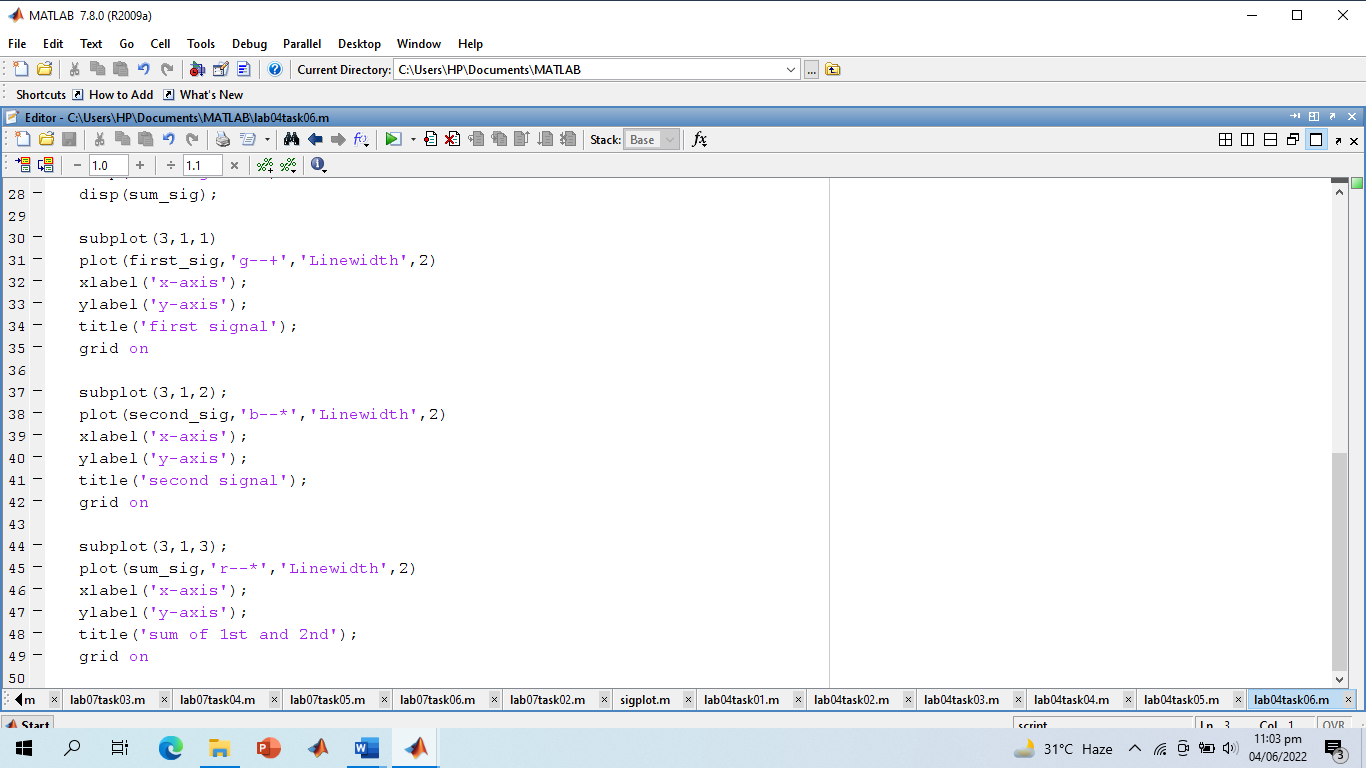
X2[n] = δ[n‐4] + 4δ[n‐5] +3δ[n‐6] + 2δ[n‐7]

Write a Matlab program that adds these two signals. Plot the original signals as well as the final results using different plotting designs.

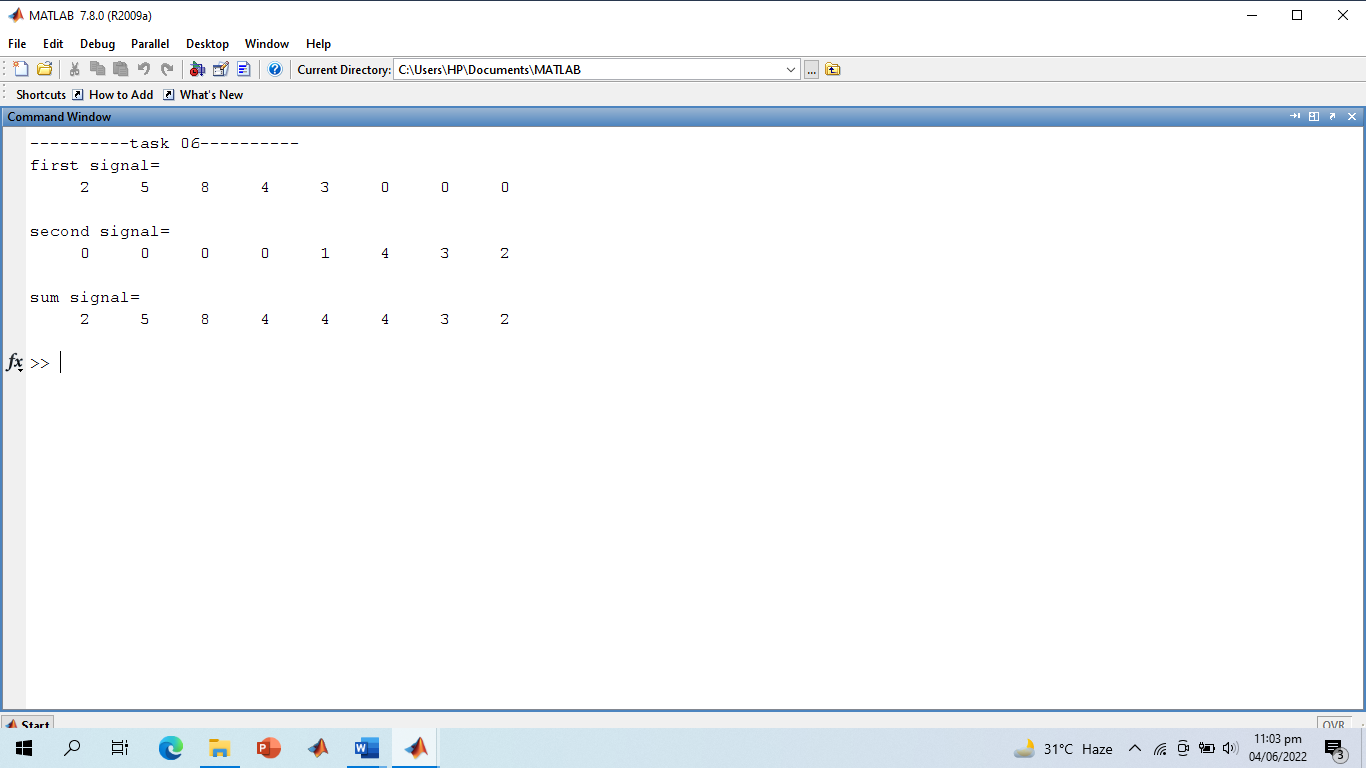
**Screenshot of Input:**

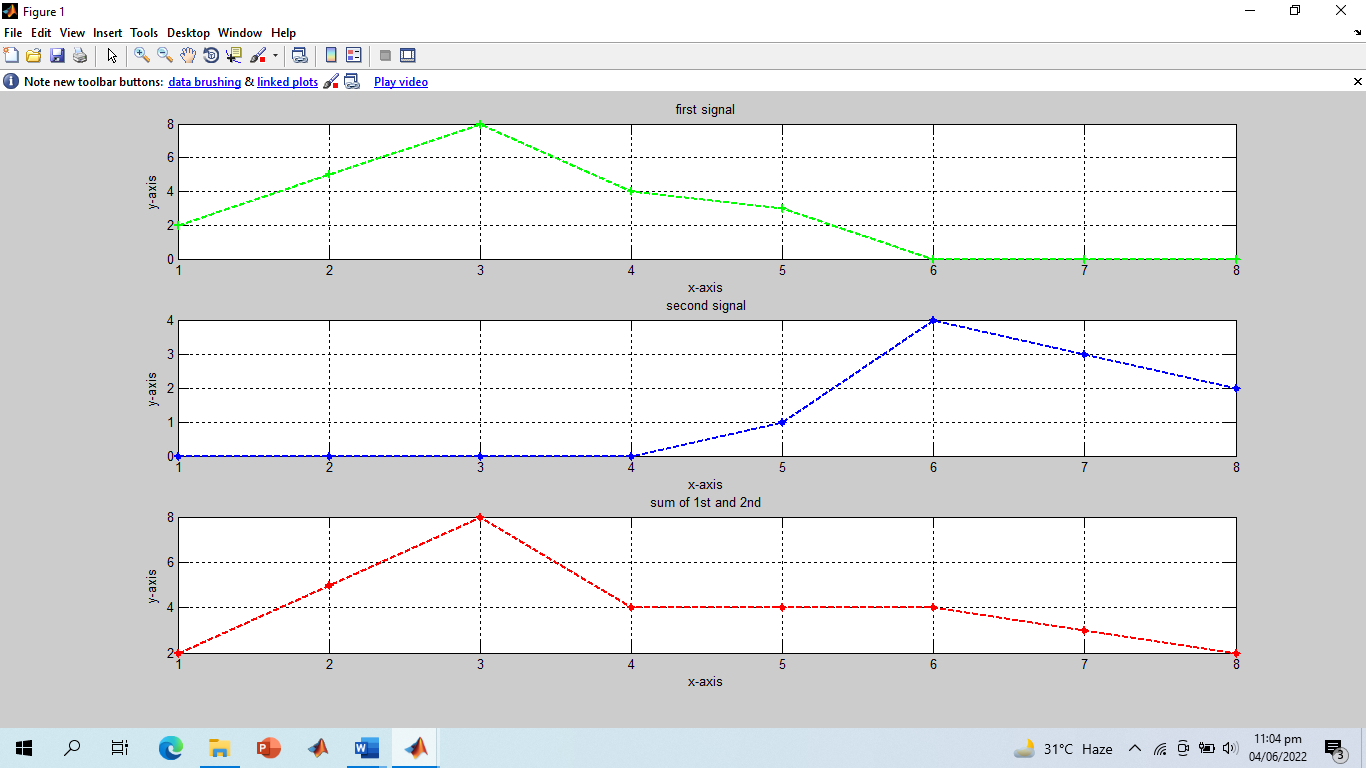






**Screenshot of Output:**

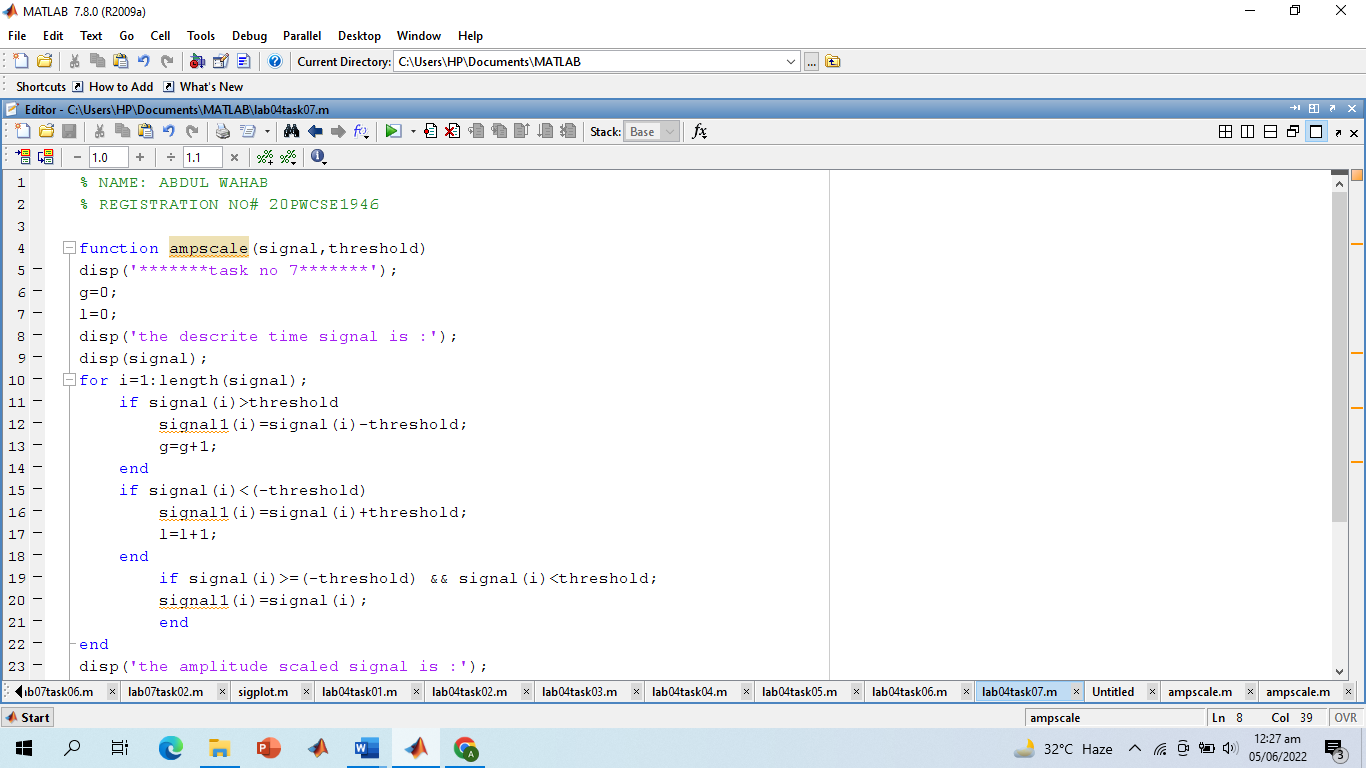


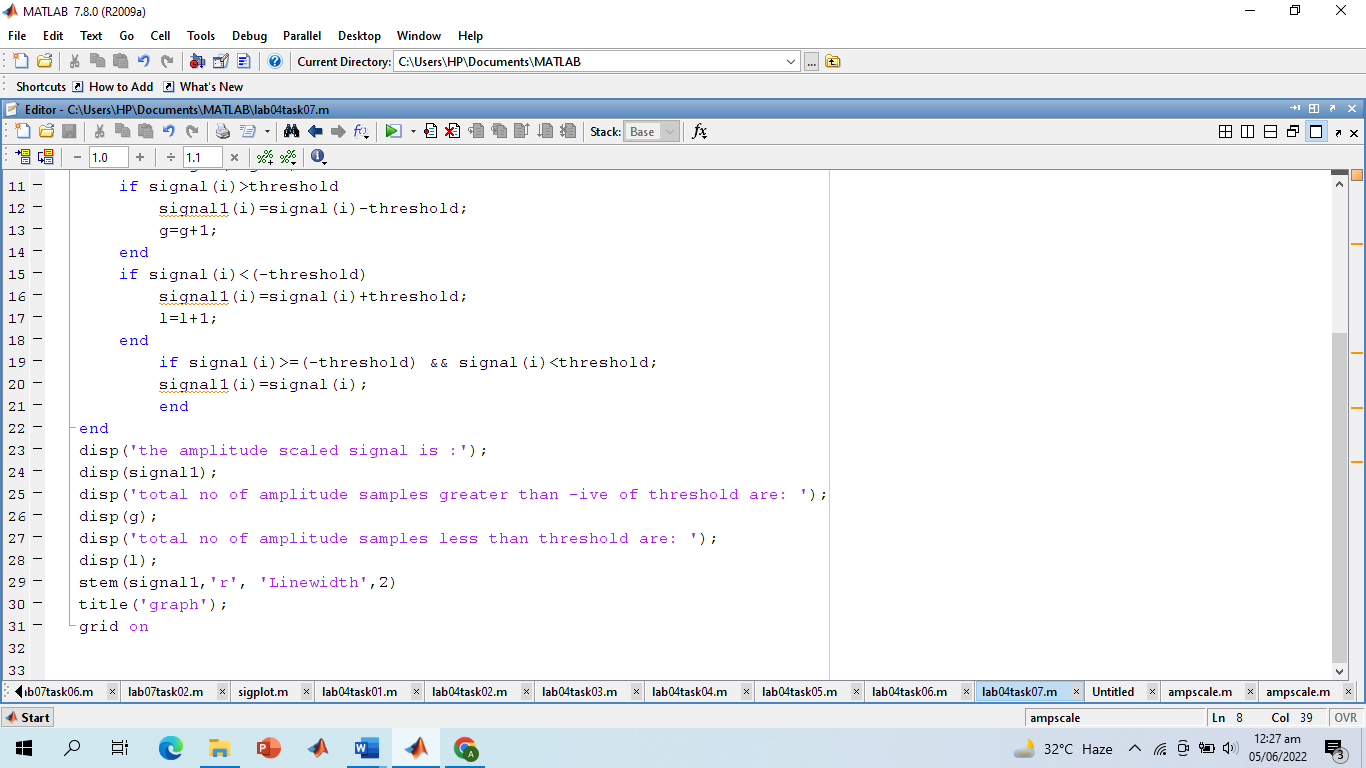


**-------------------------TASK 07--------------------------**

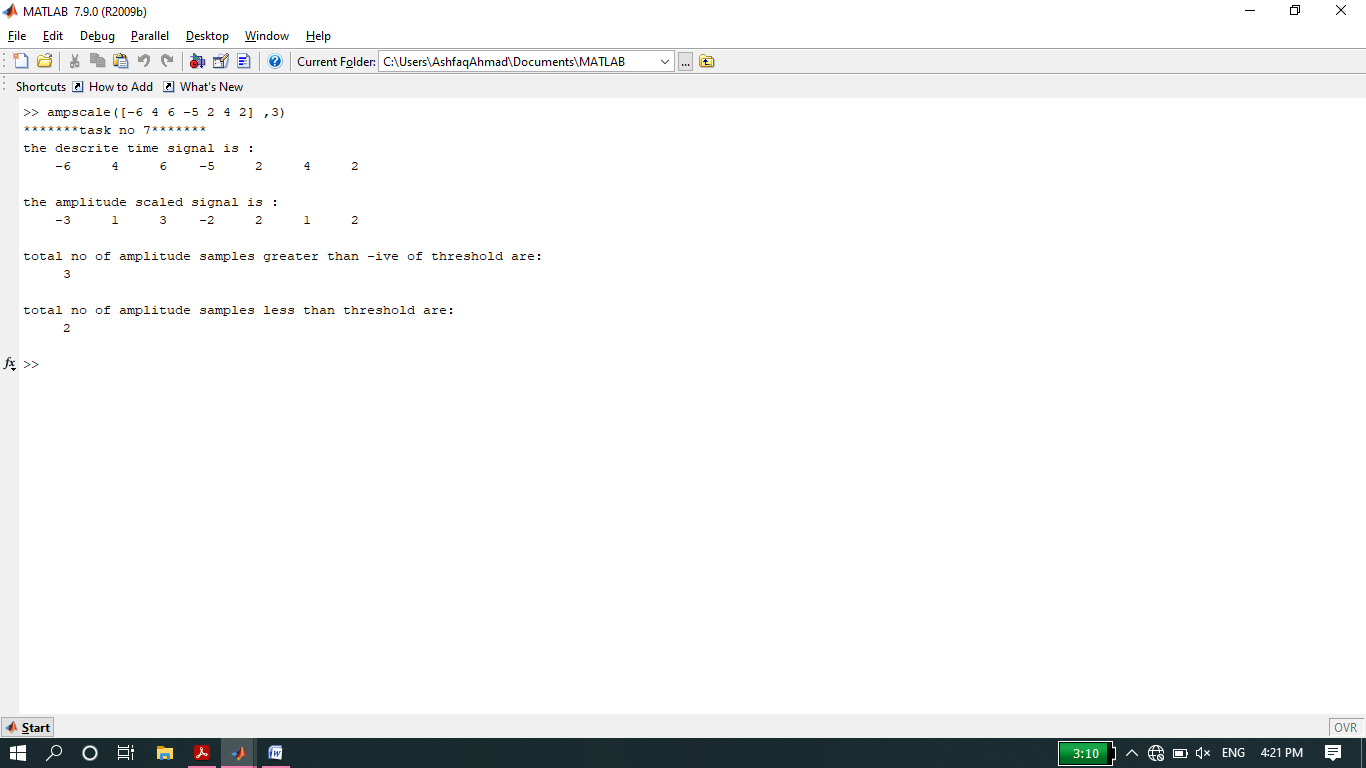
Create a function **AmpScale** that takes a discrete‐time signal **S** and a threshold **T** from user and scales the amplitude of the input signal. The function saves and counts the number of samples with amplitude greater than **T** and less than **-T** and plots the amplitude scaled signal and gives the number of samples within the thresholds as output.

**Screenshot of Input:**

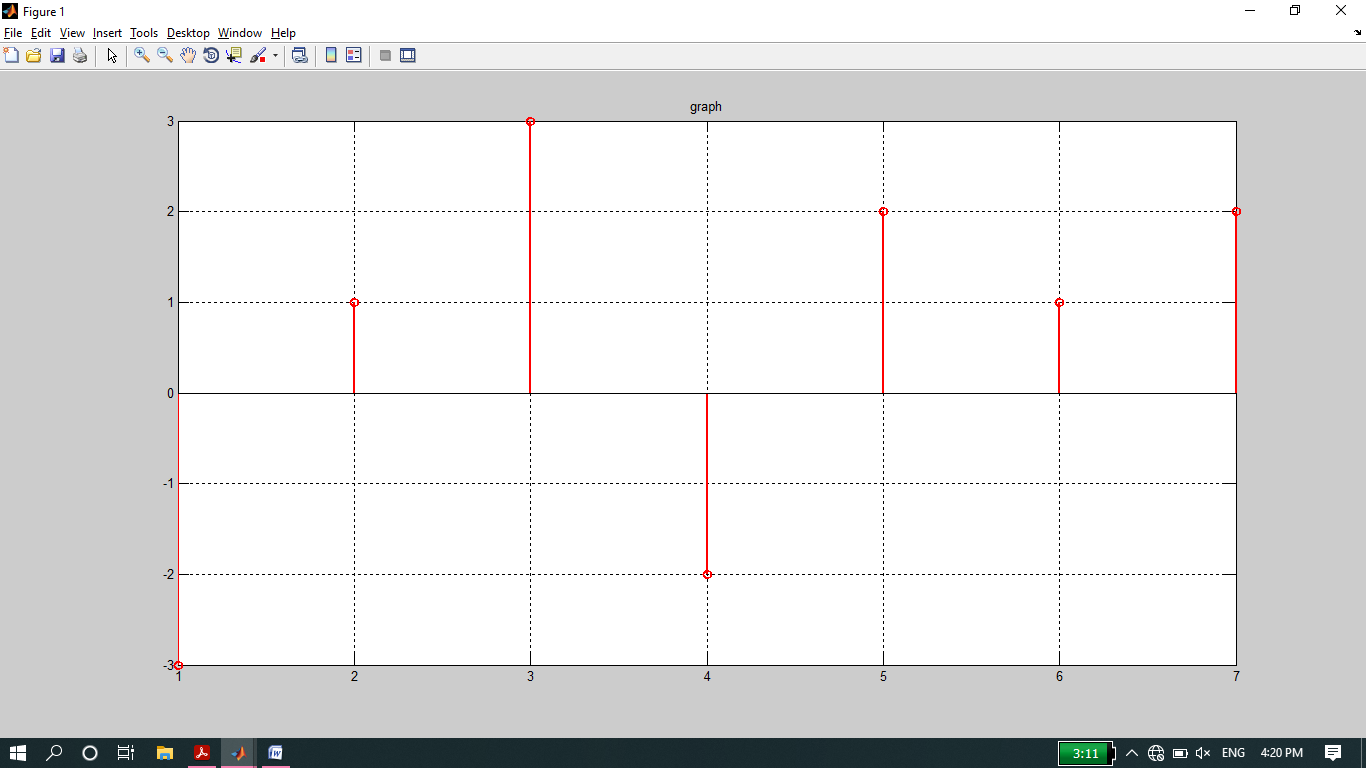




**Screenshot of Output:**

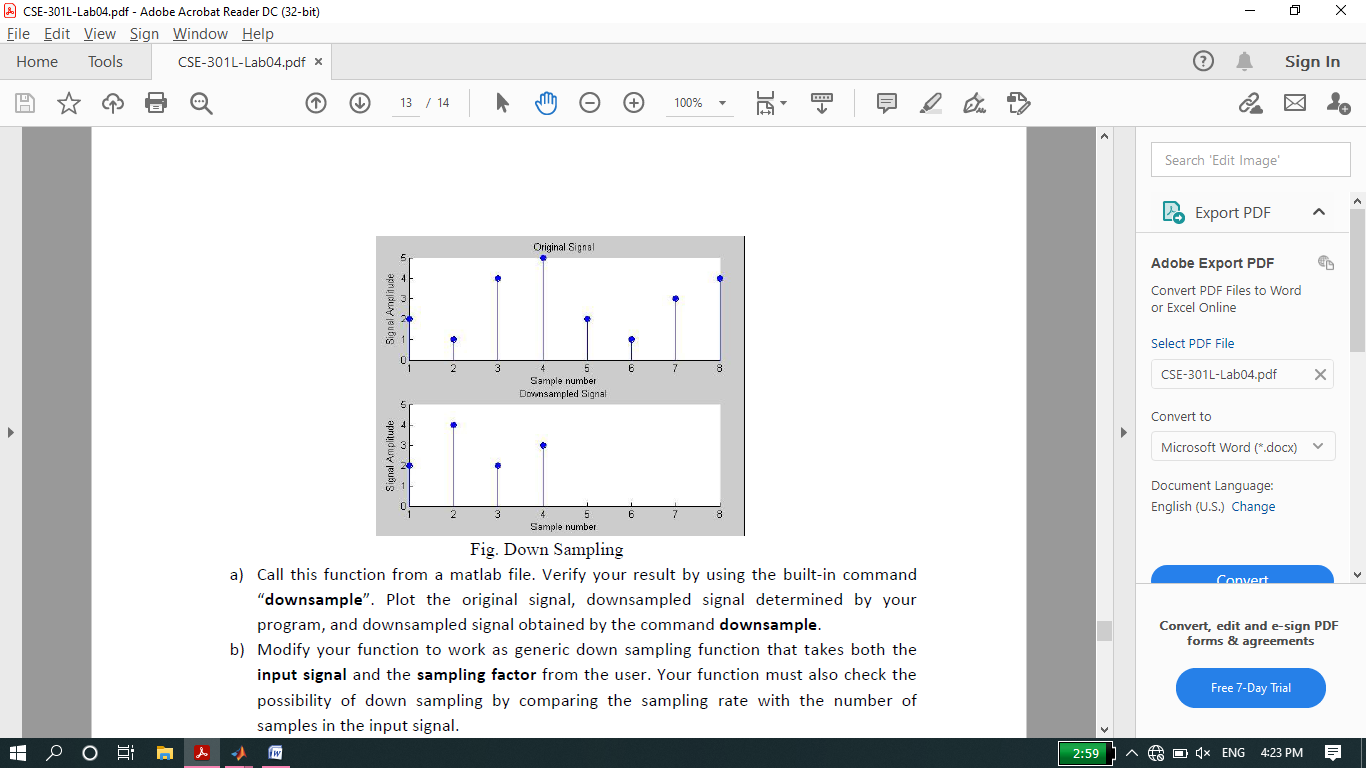


**Screenshot of Output Graph:**

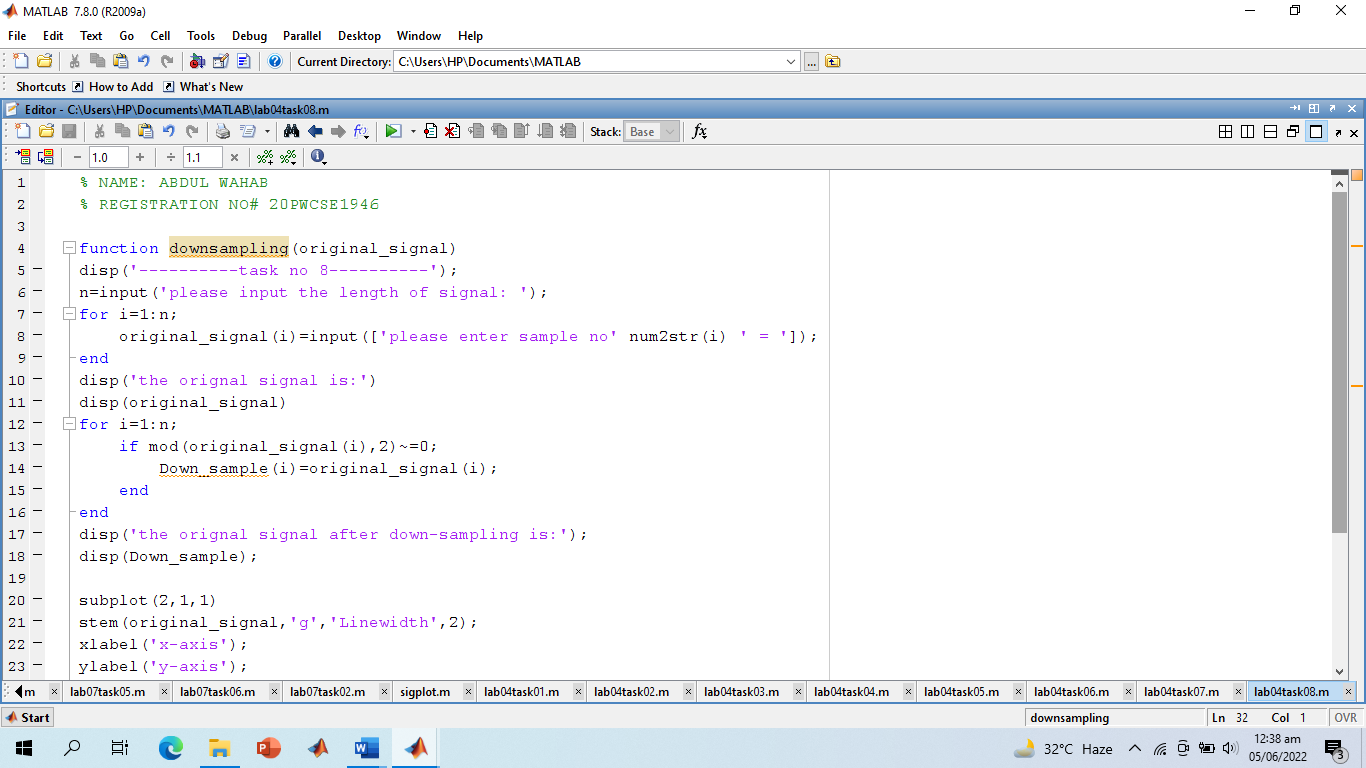


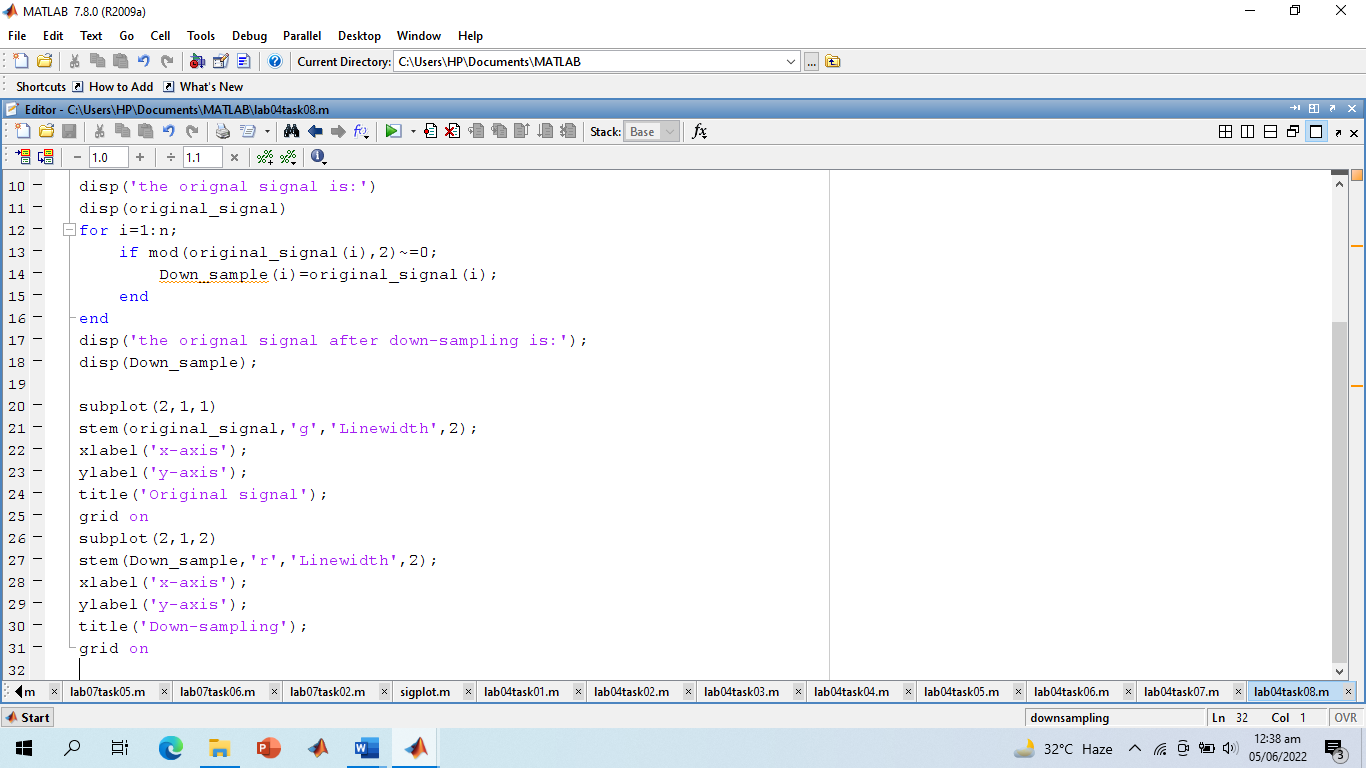
**-------------------------TASK 08--------------------------**

Write your own function **downsamp** that takes a signal as input, retain odd numbered samples of the original signal and discard the even‐numbered (down sampling by 2). The function must return the down sampled version of that signal as output. See Fig for example.

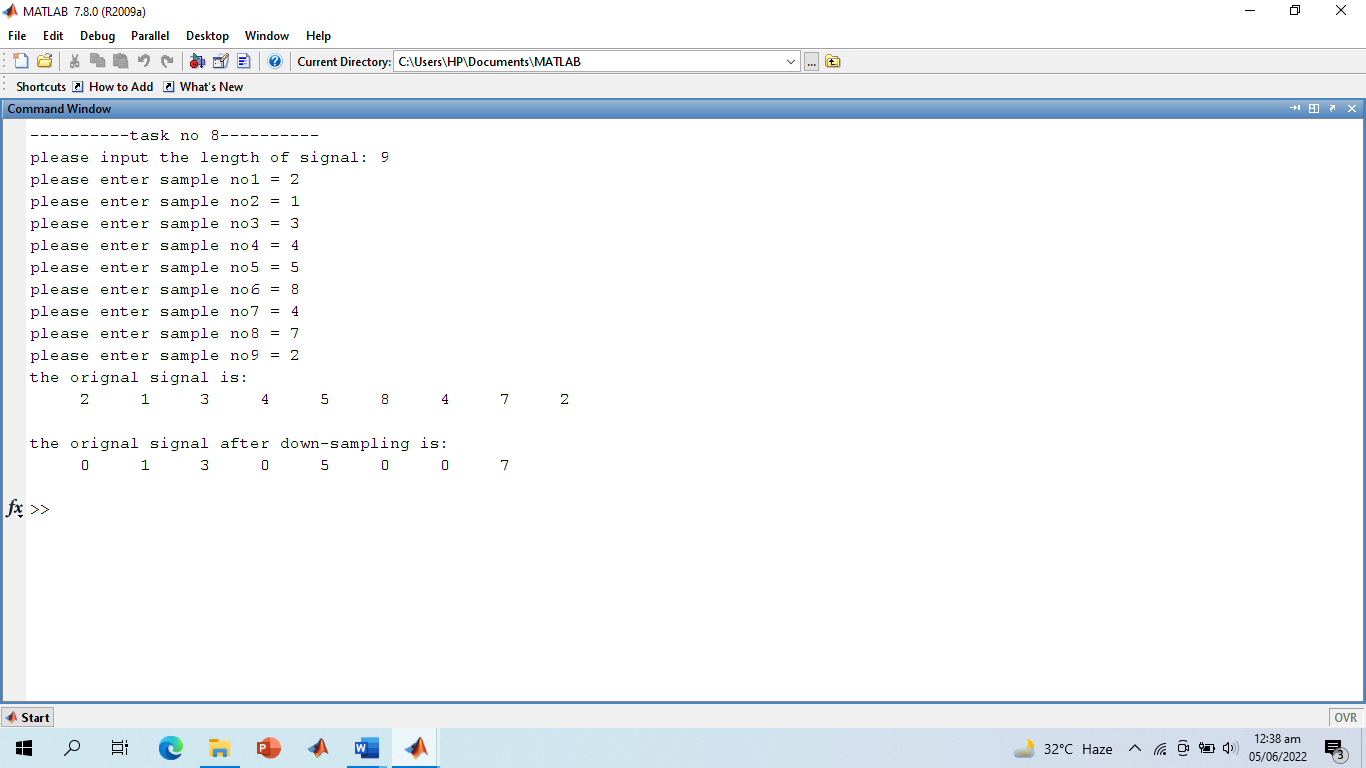
****

**Screenshot of Input:**

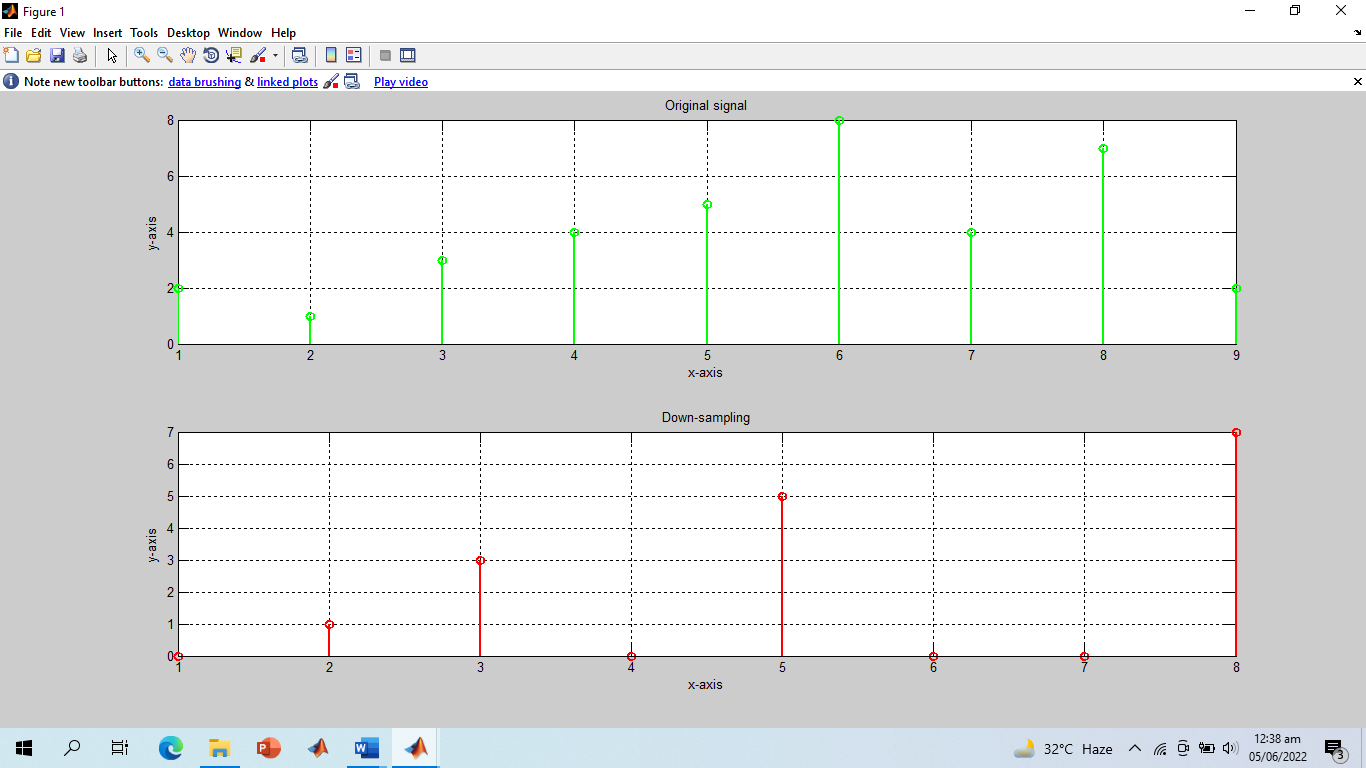




**Screenshot of Output:**

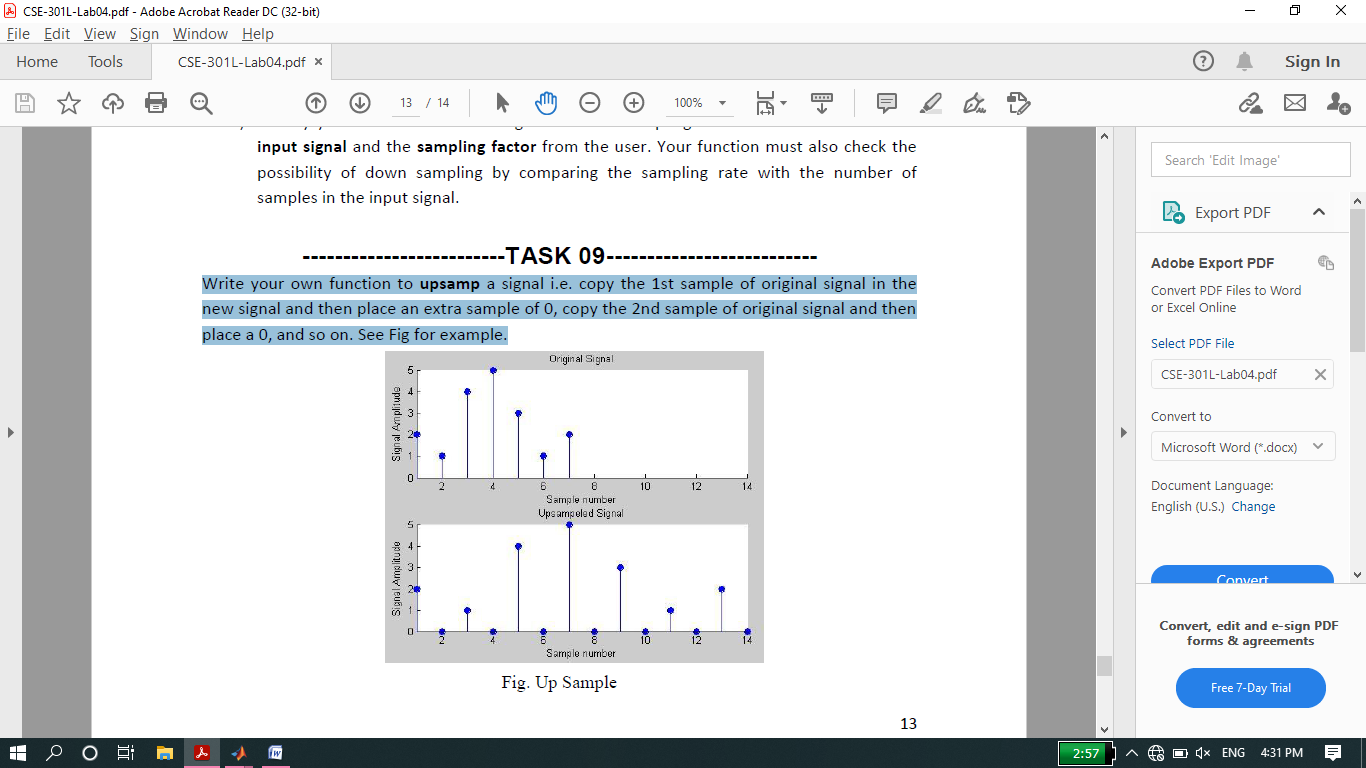


**Screenshot of Output Graph:**

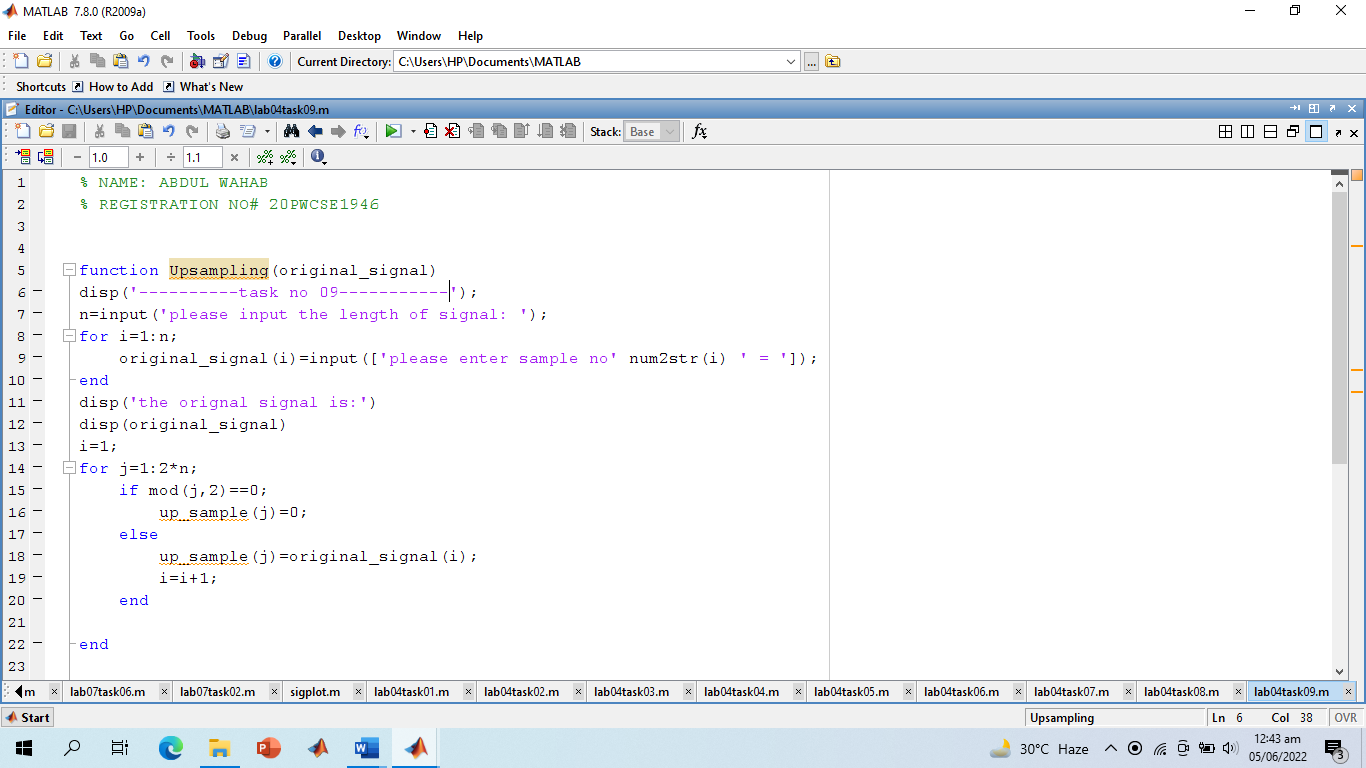


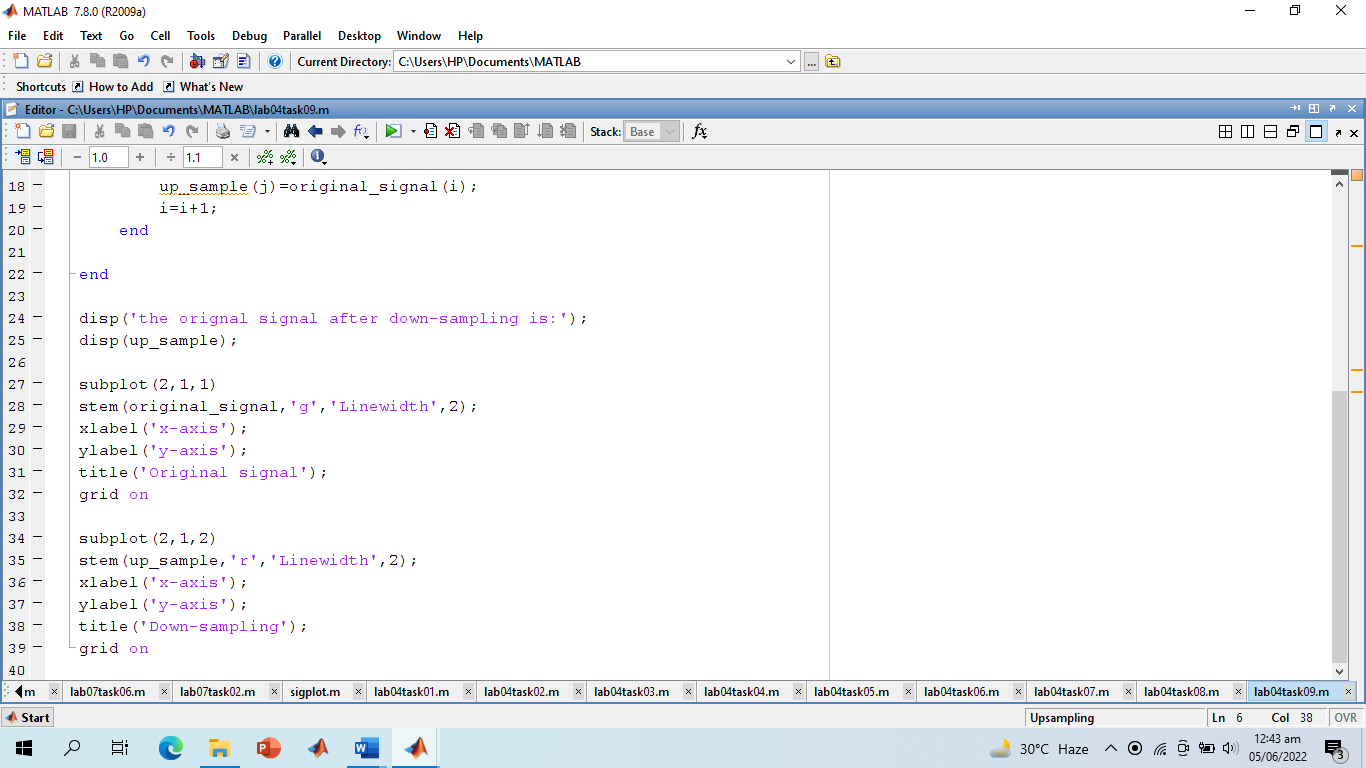
**-------------------------TASK 09--------------------------**

Write your own function to **upsamp** a signal i.e., copy the 1st sample of original signal in the new signal and then place an extra sample of 0, copy the 2nd sample of original signal and then place a 0, and soon. See Fig for example.

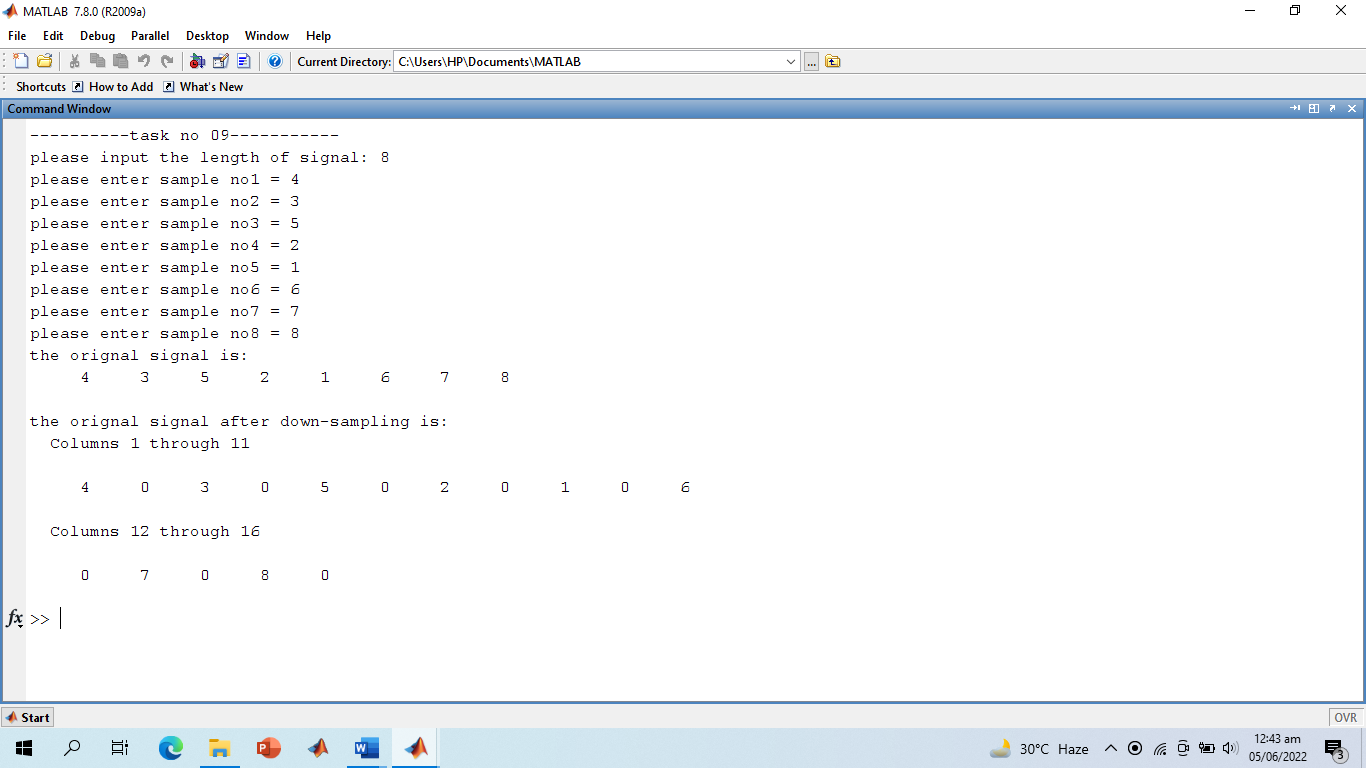


**Screenshot of Input:**

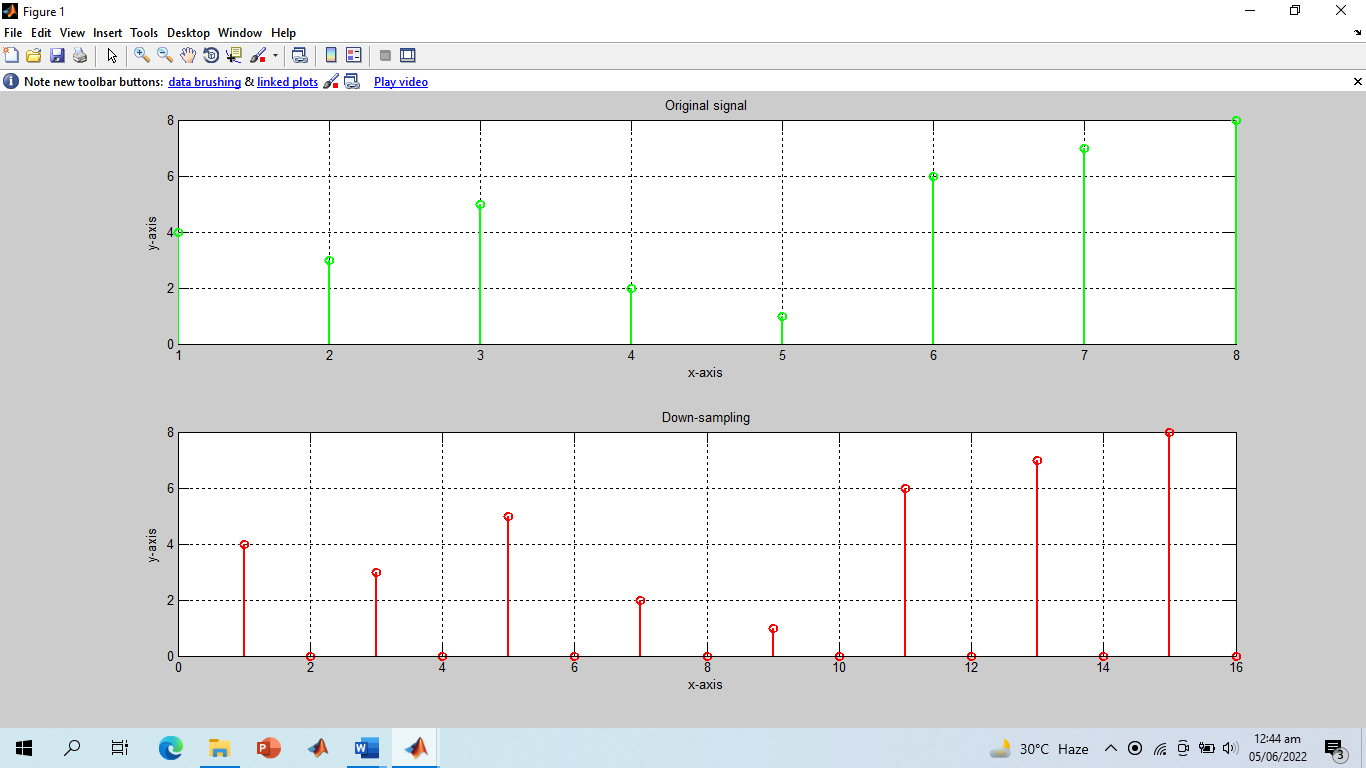




**Screenshot of Output:**



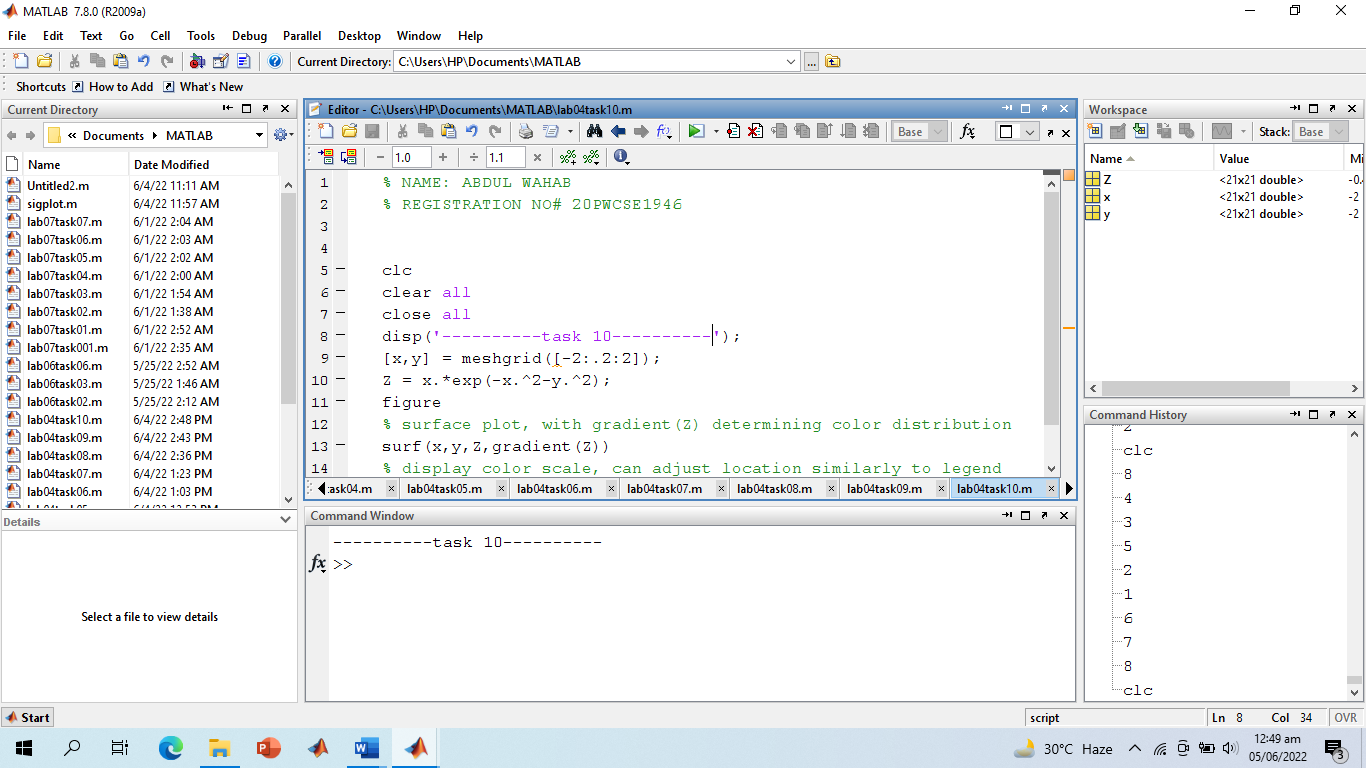
**Screenshot of Output Graphs:**



**-------------------------TASK 10--------------------------**

Plotting **3-D graphics** with MatLab. This is a complementary task for practicing 3d graphs in MatLab. **Surf** command is used in Matlab for plotting 3D graphs, the **meshgrid** command is used for setting up 2D plane.

**Screenshot of Input:**



**Screenshot of Output Graph:**



**-------------------------THE END--------------------------**