**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **26/05/2020** | **Name:** | **Namratha S Hipparagi** |
| **Course:** | **Digital signal processing** | **USN:** | **4AL16EC040** |
| **Topic:** | **Find the Z-Transform of sequence using Matlab**  **Transform and Laplace Transform** | **Semester & Section:** | **8 A** |
| **Github Repository:** | **namrathahipparagi\_1** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report**  **Convolution:**  Convolution is a mathematical way of combining two signals to form a third signal. It is the single most important technique in Digital Signal Processing. Using the strategy of impulse decomposition, systems are described by a signal called the impulse response.  Define the convolution of two functions f(x) and g(x) as f \_ g:    Calculating the Laplace F(s) transform of a function f(t) is quite simple in Matlab. First you need to specify that the variable t and s are symbolic ones. This is done with the command  >> syms t s  Next you define the function f(t). The actual command to calculate the transform is  >> F=laplace(f,t,s)  To make the expression more readable one can use the commands, simplify and pretty.  >> syms t s  >> f=-1.25+3.5\*t\*exp(-2\*t)+1.25\*exp(-2\*t);  >> F=laplace(f,t,s) F = -5/4/s+7/2/(s+2)^2+5/4/(s+2)  >> simplify(F) ans = (s-5)/s/(s+2)^2  >> pretty(ans)  The Z-transform converts a discrete-time signal, which is a sequence of real or complex numbers, into a complex frequency-domain representation. It can be considered as a discrete-time equivalent of the Laplace transform.  clc;  close all;  clear all;  syms 'z';  disp('If you input a finite duration sequence x(n), we will give you its z-transform');  nf=input('Please input the initial value of n = ');  nl=input('Please input the final value of n = ');  x= input('Please input the sequence x(n)= ');  syms 'm';  syms 'y';  f(y,m)=(y\*(z^(-m)));  disp('Z-transform of the input sequence is displayed below');  k=1;  for n=nf:1:nl      answer(k)=(f((x(k)),n));     k=k+1;  end  disp(sum(answer)); |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **26/5/2020** | **Name:** | **Namratha S Hipparagi** | |
| **Course:** | **Python** | **USN:** | **4al16ec040** | |
| **Topic:** | **Application 4: Build a**  **Personal Website with**  **Python and Flask** | **Semester & Section:** | **8 A** | |
| **AFTERNOON SESSION DETAILS** | | | |
| It is important to separate the content from the rules for how it should be rendered primarily because it is easier to reuse those rules across many pages. CSS separates the content contained in HTML files from how the content should be displayed. CSS files are also much easier to maintain on large projects than styles embedded within the HTML files.  View source screenshot for the fsp.css file in index.html.  **Navigation menu:**  /\* Add a black background color to the top navigation \*/ .topnav {   background-color: #333;   overflow: hidden; }  /\* Style the links inside the navigation bar \*/ .topnav a {   float: left;   color: #f2f2f2;   text-align: center;   padding: 14px 16px;   text-decoration: none;   font-size: 17px; }  /\* Change the color of links on hover \*/ .topnav a:hover {   background-color: #ddd;   color: black; }  /\* Add a color to the active/current link \*/ .topnav a.active {   background-color: #4CAF50;   color: white; } | | | |
|  | | | |