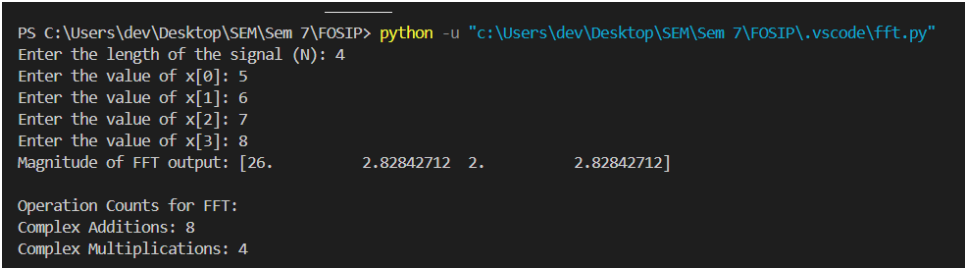
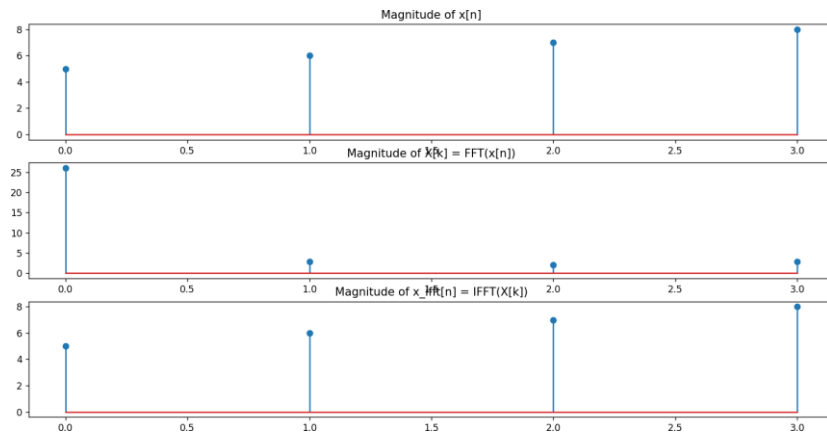


Name:	Hatim Sawai
UID:	2021300108
Experiment 4	
Aim:	The aim of this experiment is to implement computationally Fast Algorithms.
Objective:	<ol style="list-style-type: none"> 1. Develop a program to perform FFT of N point Signal. 2. Calculate FFT of a given DT signal and verify the results using mathematical formula. 3. Computational efficiency of FFT.
Input Specifications:	<ol style="list-style-type: none"> 1. Length of first Signal N 2. DT Signal values
Problem Definition:	<p>(1) Take any four-point sequence $x[n]$. Find FFT of $x[n]$ and IFFT of $\{X[k]\}$.</p> <p>(2) Calculate Real and Complex Additions & Multiplications involved to find $X[k]$.</p>
Experiment and Analysis:	 <pre> PS C:\Users\dev\Desktop\SEM\Sem 7\FOSIP> python -u "c:\Users\dev\Desktop\SEM\Sem 7\FOSIP\.vscode\fft.py" Enter the length of the signal (N): 4 Enter the value of x[0]: 5 Enter the value of x[1]: 6 Enter the value of x[2]: 7 Enter the value of x[3]: 8 Magnitude of FFT output: [26.0, 2.82842712, 2.0, 2.82842712] Operation Counts for FFT: Complex Additions: 8 Complex Multiplications: 4 </pre> <p>To find DFT of a 4 point sequence</p> <p>Input $x[n]$: {5, 6, 7, 8}</p> <p>Magnitude $X[k]$: {26, 2.82, 2, 2.82 }</p>



Conclusion:

1. Computational Efficiency in DFT:
 - a) Total Real Multiplications = $4N^2$
 - b) Total Real Additions = $4N^2 - 2N$
2. Computational Efficiency in FFT:
 - a) Total Real Multiplications = $2N \cdot \log_2 N$
 - b) Total Real Additions = $3N \cdot \log_2 N$
3. FFT produces fast results due to:
 - a) Less Computations
 - b) Parallel implementations