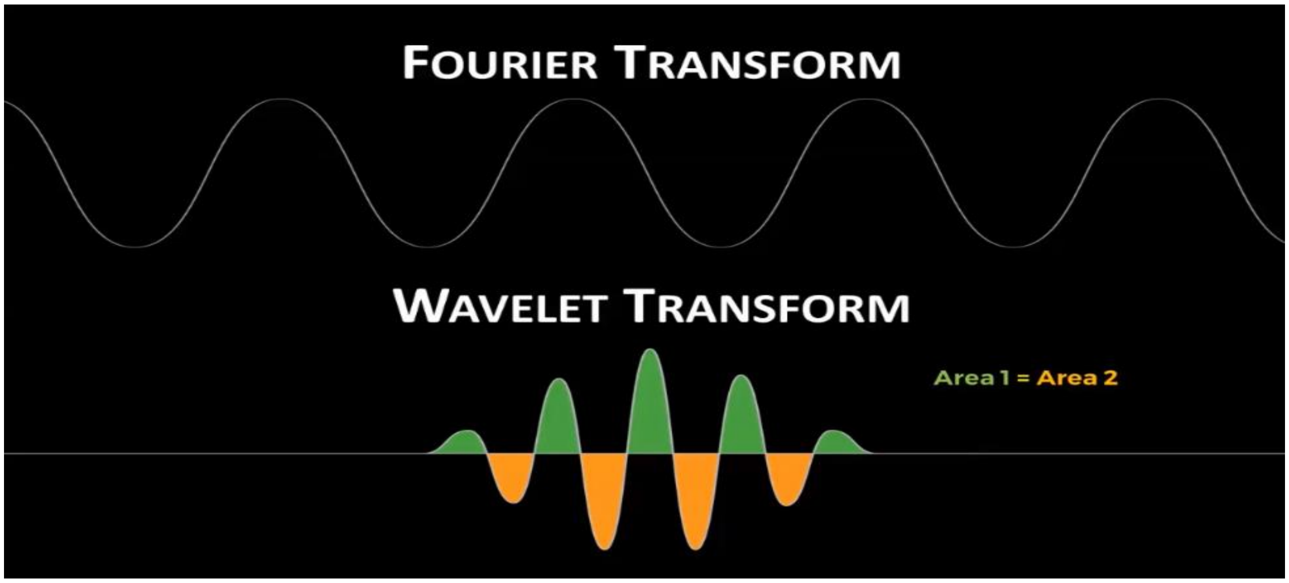
DAILY ASSESSMENT FORMAT

Date:	27/5/2020	Name:	SHILPA N
Course:	Digital signal processing	USN:	4AL16EC071
Topic:	DAY 3	Semester & Section:	8th sem B sec
Github Repository:	Shilpan-test		

FORENOON SESSION DETAILS

Image of session



Report –

FAST FOURIER TRANSFORM MATLAB:

A fast Fourier transform (FFT) is an algorithm that computes the discrete Fourier transform (DFT) of a sequence, or its inverse (IDFT). Fourier analysis converts a signal from its original domain (often time or space) to a representation in the frequency domain and vice versa. The DFT is obtained by decomposing a sequence of values into components of different frequencies. This operation is useful in many fields, but computing it directly from the definition is often too slow to be practical. An FFT rapidly computes such transformations by factorizing the DFT matrix into a product of sparse (mostly zero) factors. As a result, it manages to reduce the complexity computing the DFT from , which arises if one simply applies the definition of DFT, to , where is the data size. The difference in speed can be enormous, especially for long data sets where N may be in the thousands or millions. In the presence of round-off error, many FFT algorithms are much more accurate than evaluating the DFT definition directly or indirectly. There are many different FFT algorithms based on a wide range of published theories, from simple complex-number arithmetic to group theory and number theory.

SYNTAX:

Y = fft(X)

Y = fft(X,n)

Y = fft(X,n,dim)

DESCRIPTION:

Y = fft(X) computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm.

If X is a vector, then fft(X) returns the Fourier transform of the vector.

If X is a matrix, then fft(X) treats the columns of X as vectors and returns the Fourier transform of each column.

If X is a multidimensional array, then fft(X) treats the values along the first array dimension whose size does not equal 1 as vectors and returns the Fourier transform of each vector.

example

Y = fft(X,n) returns the n-point DFT. If no value is specified, Y is the same size as X.

If X is a vector and the length of X is less than n, then X is padded with trailing zeros to length n.

If X is a vector and the length of X is greater than n, then X is truncated to length n.

If X is a matrix, then each column is treated as in the vector case.

If X is a multidimensional array, then the first array dimension whose size does not equal 1 is treated as in the vector case.

example

Y = fft(X,n,dim) returns the Fourier transform along the dimension dim. For example, if X is a matrix, then fft(X,n,2) returns the n-point Fourier transform of each row.

FIR AND IIR FILTERS:

Both FIR and IIR are the two major classifications of digital filters used for signal filtration. The crucial difference between FIR and IIR filter is that the FIR filter provides an impulse response of finite period.

As against IIR is a type of filter that generates impulse response of infinite duration for a dynamic system.

Definition of FIR filter:

A type of digital filter that generates a finite impulse response of a dynamic system is known as FIR filters. More simply, we can say, here the impulse response provided by the filter is of finite duration.

In FIR filters the response gets fixed to zero in a finite period of time thus it is named so. In the case of FIR filters, the n_{th} order filter generates n+1 samples before getting fixed to 0. In FIR filters, feedback is not present thus these operate on only present and past input values. Hence the output is the summation of a finite quantity of finite samples of input values. So, these are highly stable. The generalized equation for FIR filter is given as:

$$y(n) = \sum_{k=0}^{N-1} h(k)\, x(n-k)$$

Where, N is the length of the filter.

FIR filters are used with systems having 0 response.

Definition of IIR filter:

The type of digital filter that is designed to generate infinite impulse response of a dynamic system is known as the IIR filter. As an internal feedback mechanism is present in these filters thus the filter operates for an indefinite period of time.

IIR filters are used by the systems that generate an infinite response. IIR filter operates in a way that not only the present and past inputs but the past output sample is also taken into consideration.

The equation in general form for IIR filter is given as:

$$y(n) = \sum_{k=0}^{\infty} h(k)\, x(n-k)$$

As these filters support the recursive operation. Thus it never allows its response to settle to 0 for the applied impulse.

The memory requirement and computational time enhance the efficiency of the IIR filters. Also, it is quite easy to convert an analog filter into a digital IIR filter.

WELCH'S METHOD AND WINDOWING:

As usual, the purpose of the window function w(n) is to reduce side-lobe level in the spectral density estimate, at the expense of frequency resolution, exactly as in the case of sinusoidal spectrum analysis.

When using a non-rectangular analysis window, the window hop-size R cannot exceed half the frame length M/2 without introducing a non-uniform sensitivity to the data x(n) over time. In the rectangular window case, we can set R=M and have non-overlapping windows. For Hamming, Hanning, and any other generalized Hamming window, one normally sees R=(M-1)/2 for odd-length windows. For the Blackman window, R~M/2 is typical. In general, the hop size R is chosen so that the analysis window w overlaps and adds to a constant at that hop size. This consideration is explored more fully in Chapter 8. An equivalent parameter used by most matlab implementations is the overlap parameter M-R .

Note that the number of blocks averaged in increases as R decreases. If Nx>=M denotes the total number of signal samples available, then the number of complete blocks available for averaging may be computed as

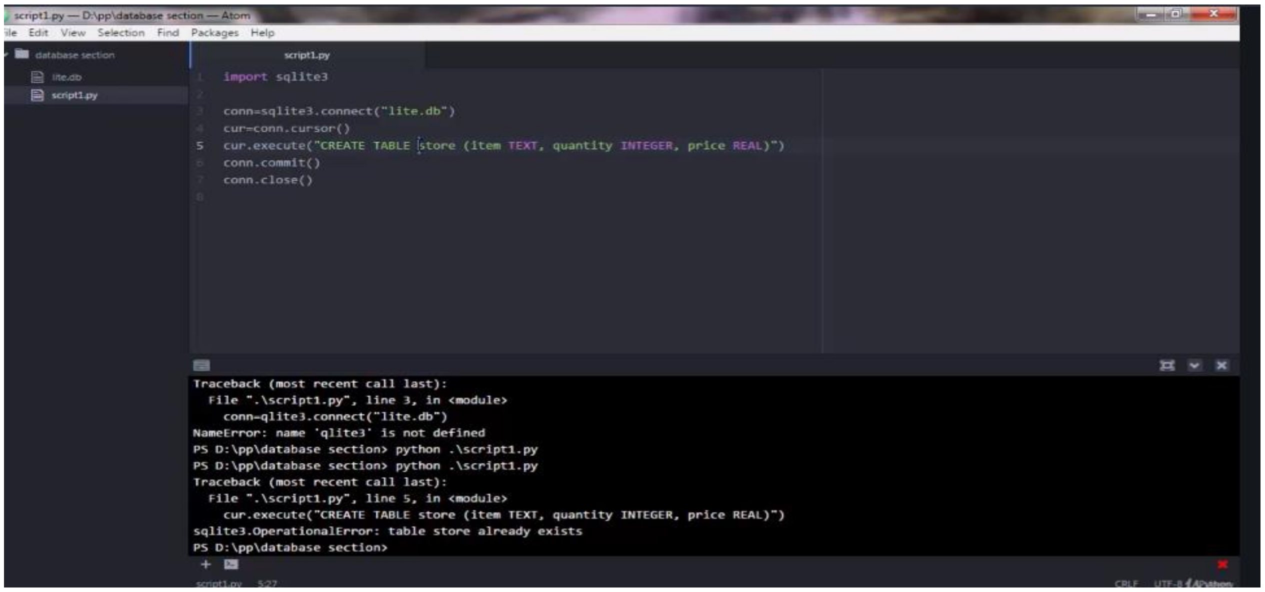
$$K = \left\lfloor \frac{N_x - M}{R} \right\rfloor + 1,$$

where the floor function [x] denotes the largest integer less than or equal to x .

Date:	27/5/2020	Name:	SHILPA N
Course:	Python	USN:	4AL16EC071
Topic:	Graphical user interface with Tkinter, interacting with databases	Sem & sec:	8th sem 'B' sec

AFTERNOON SESSION DETAILS

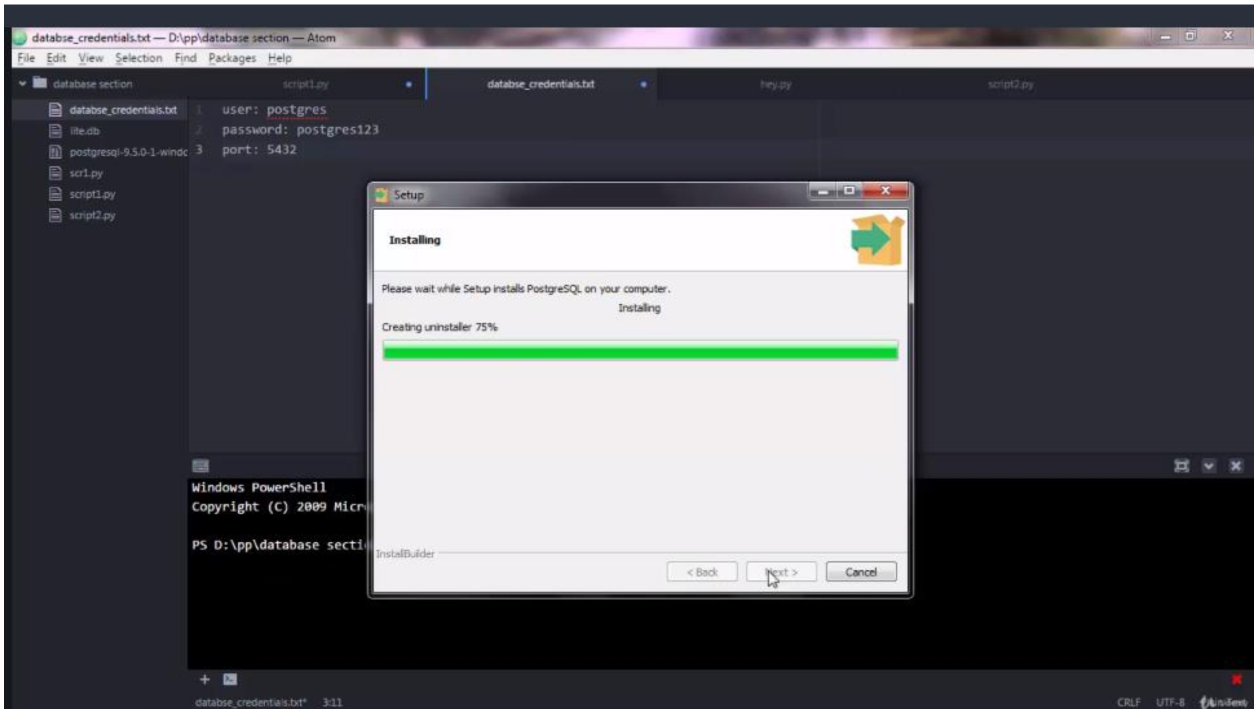
SESSION IMAGE :


script1.py — D:\pp\database section — Atom
File Edit View Selection Find Packages Help
database section
lite.db
script1.py

```
import sqlite3

conn=sqlite3.connect("lite.db")
cur=conn.cursor()
cur.execute("CREATE TABLE store (item TEXT, quantity INTEGER, price REAL)")
conn.commit()
conn.close()
```

```
Traceback (most recent call last):
  File ".\script1.py", line 3, in <module>
    conn=qlite3.connect("lite.db")
NameError: name 'qlite3' is not defined
PS D:\pp\database section> python .\script1.py
PS D:\pp\database section> python .\script1.py
Traceback (most recent call last):
  File ".\script1.py", line 5, in <module>
    cur.execute("CREATE TABLE store (item TEXT, quantity INTEGER, price REAL)")
sqlite3.OperationalError: table store already exists
PS D:\pp\database section>
```



```
user: postgres
password: postgres123
port: 5432
```

Installing

Please wait while Setup installs PostgreSQL on your computer.

Installing

Creating uninstaller 75%

```
Windows PowerShell
Copyright (C) 2009 Micr

PS D:\pp\database secti
```

REPORT:

MULTI WIDGET GUI:

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

To create a tkinter app:

1. Importing the module – tkinter
2. Create the main window (container)
3. Add any number of widgets to the main window
4. Apply the event Trigger on the widgets.

Importing tkinter is same as importing any other module in the Python code. Note that the name of the module in Python 2.x is ‘Tkinter’ and in Python 3.x it is ‘tkinter’.

```
import tkinter
```

There are two main methods used which the user needs to remember while creating the Python application with GUI.

1. **Tk(screenName=None, baseName=None, className=’Tk’, useTk=1):** To create a main window, tkinter offers a method ‘Tk(screenName=None, baseName=None, className=’Tk’, useTk=1)’. To change the name of the window, you can change the className to the desired one. The basic code used to create the main window of the application is:

   ```
   m=tkinter.Tk() where m is the name of the main window object
   ```

2. There is a method known by the name mainloop() is used when your application is ready to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed.

```
m.mainloop()
```

```
from tkinter import *
# Create an empty Tkinter window
window=Tk()
def from_kg():
    # Get user value from input box and multiply by 1000 to get kilograms
    gram=float(e2_value.get())*1000
    # Get user value from input box and multiply by 2.20462 to get pounds
).    pound=float(e2_value.get())*2.20462
.  # Get user value from input box and multiply by 35.274 to get ounces
ounce=float(e2_value.get())*35.274

Empty the Text boxes if they had text from the previous use and fill them again

t1.delete("1.0", END)  # Deletes the content of the Text box from start to END

t1.insert(END,gram)  # Fill in the text box with the value of gram variable

t2.delete("1.0", END)

t2.insert(END,pound)

t3.delete("1.0", END)

t3.insert(END,ounce)

reate a Label widget with "Kg" as label

=Label(window,text="Kg")

.grid(row=0,column=0) # The Label is placed in position 0, 0 in the window
```

```
_value=StringVar()  # Create a special StringVar object

=Entry(window,textvariable=e2_value)  # Create an Entry box for users to enter the value

.grid(row=0,column=1

Create a button widget

The from_kg() function is called when the button is pushed

=Button(window,text="Convert",command=from_kg)

.grid(row=0,column=2

Create three empty text boxes, t1, t2, and t3

=Text(window,height=1,width=20)

.grid(row=1,column=0)

=Text(window,height=1,width=20)

.grid(row=1,column=1)

=Text(window,height=1,width=20)

.grid(row=1,column=2)

This makes sure to keep the main window open

ndow.mainloop()
```

Querying data from a MySQL database:

If you prefer to work with MySQL instead of PostGreSQL, the below code is used.

I set up a remote MySQL database on a server with the IP address 108.167.140.122, so you don't have to install and set up a MySQL database yourself. To connect and query data from that remote database, you need a username, password, and the name of the database. These are written inside the Python script below.

You also need a Python library that interacts with MySQL databases. Many libraries are compatible, but I prefer mysql.connector. To install mysql.connector: simply execute `pip install mysql-connector` or `pip3 install mysql-connector` depending on whether you use pip or pip3.

```
import mysql.connector

word = input("Enter a word in English and press Enter: ")
con = mysql.connector.connect(
    user="ardit700_student",
    password = "ardit700_student",
    host="108.167.140.122",
    database = "ardit700_pm1database"
)
cursor = con.cursor()
query = cursor.execute("SELECT * FROM Dictionary WHERE Expression = '%s'" % word)
results = cursor.fetchall()
if results:
    for result in results:
        print(result[1])
else:
    print("We couldn't find any results about that.")
```