Day 5

|  |
| --- |
| **Program 1:**  **Global Positioning system**  **Write a Python program to track a route on Google map of given latitude and longitude values in the CSV file.**  **Required Libraries:**   1. **CSV Library** 2. **Gmplot library** 3. **Csv files having latitude and longitude values** |

|  |
| --- |
| **Program 2:**  **Write a Python Program for Speech recognition (Convert Audio file into Text file)**  **Required Libraries and Files**   1. **Speech Recognition Library** 2. **Audio File should be in .wav extension** |

import speech\_recognition as sr

r = sr.Recognizer()

with sr.AudioFile('C:/Users/kelka/Documents/Internship\_Python/Vedant\_audio.wav.wav') as audio:

audio\_text = r.listen(audio)

try:

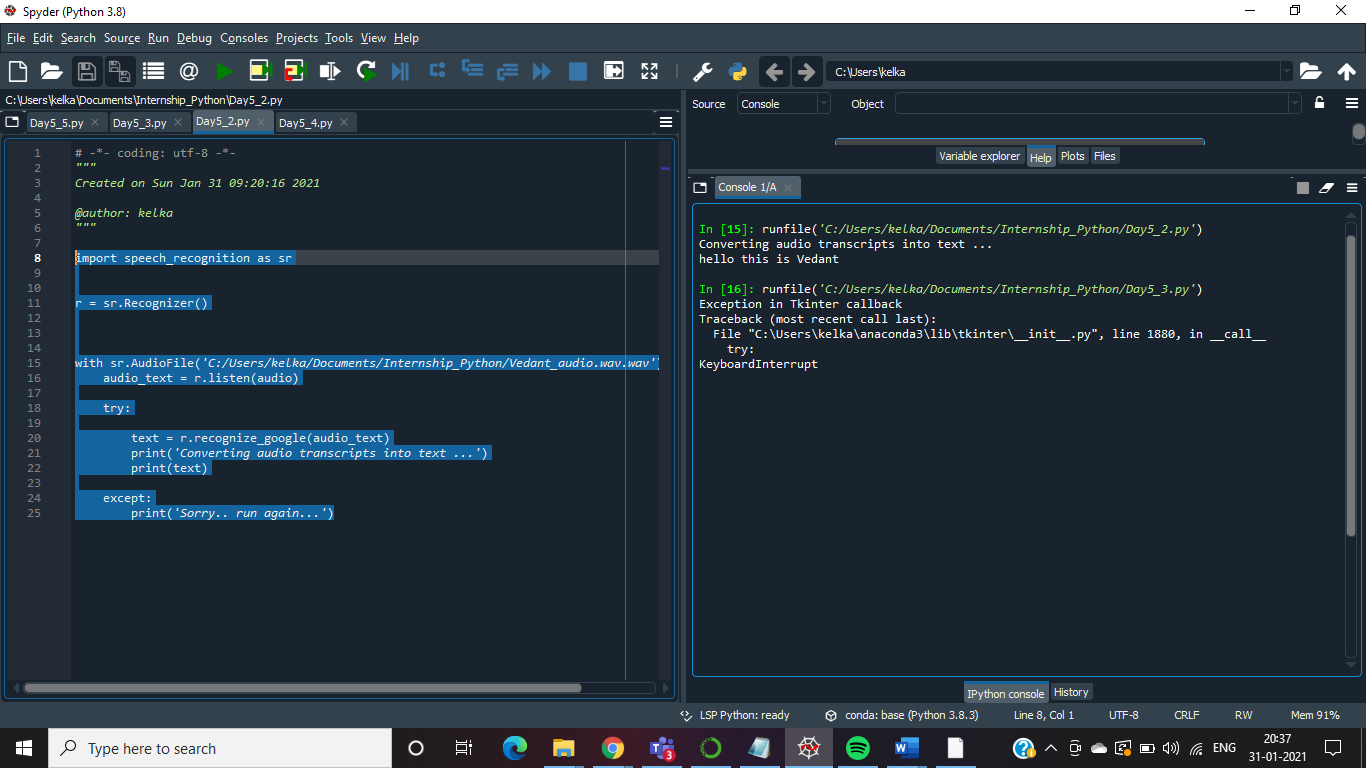
text = r.recognize\_google(audio\_text)

print('Converting audio transcripts into text ...')

print(text)

except:

print('Sorry.. run again...')



|  |
| --- |
| **Program 3:**  **Write a Program to design a login page using Tkinter library** |

from tkinter import \*

from tkinter import messagebox

def loginpage():

uname = e1.get()

password = e2.get()

if (uname == "" and password == ""):

messagebox.showinfo("warning","Blank Not allowed")

elif (uname == "vedant" and password == "12345"):

messagebox.showinfo("", "Login Success")

root.destroy()

else:

messagebox.showinfo("", "Incorrent Username and Password")

root = Tk()

root.title("Login")

root.geometry("350x350")

global e1

global e2

Label(root, text="UserName").place(x=10, y=10)

Label(root, text="Password").place(x=10, y=40)

e1 = Entry(root)

e1.place(x=140, y=10)

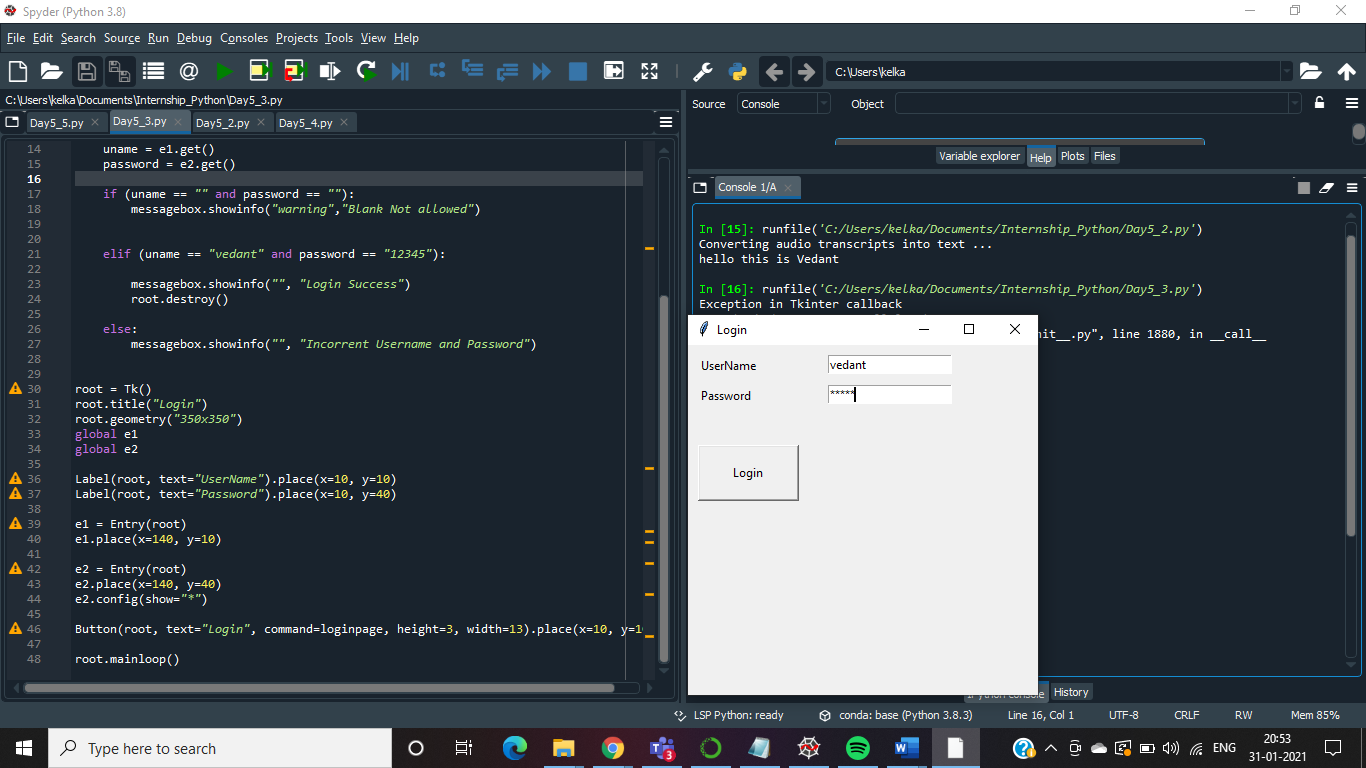
e2 = Entry(root)

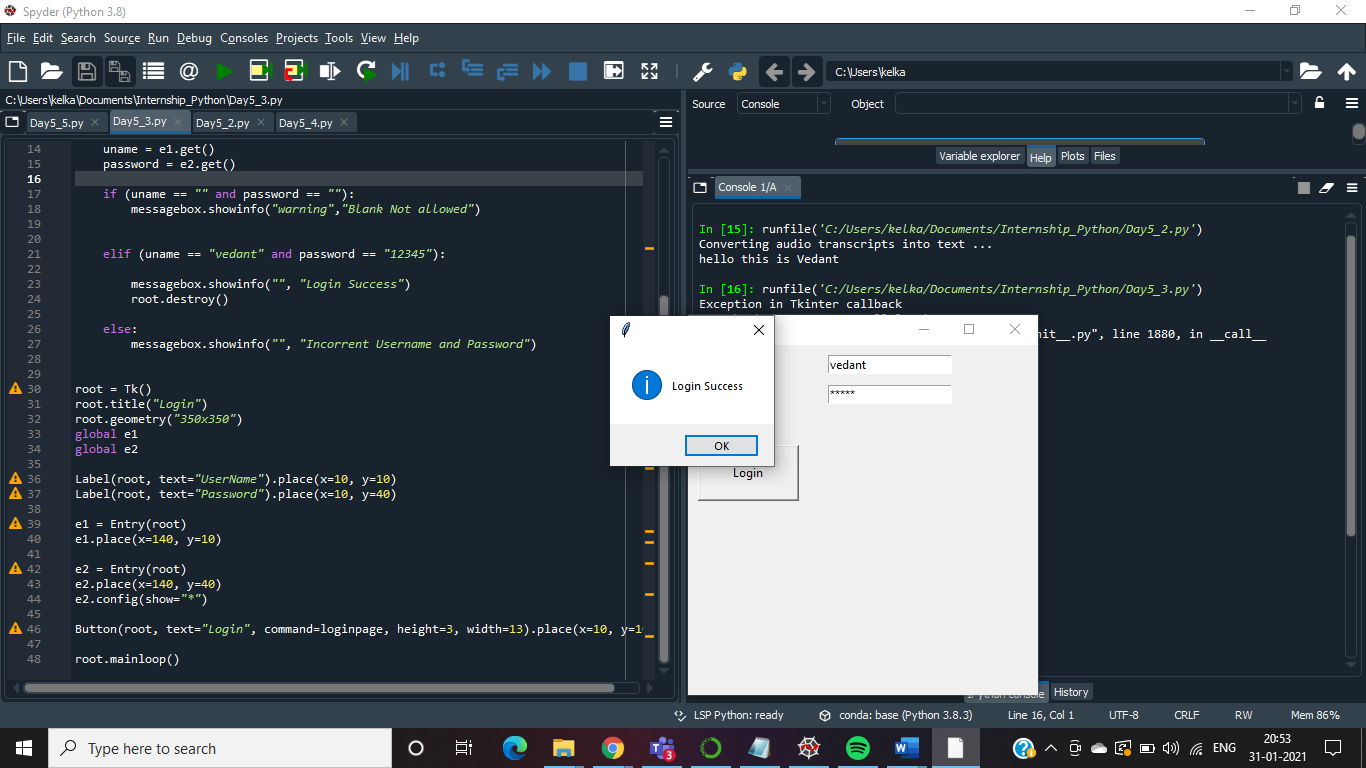
e2.place(x=140, y=40)

e2.config(show="\*")

Button(root, text="Login", command=loginpage, height=3, width=13).place(x=10, y=100)

root.mainloop()





**Program 4:**

Spiral Traversing Animation by using turtle library

**Required Libraries and Files**

1. **Turtle Library**
2. **Example: To animate Star,  run the following code**

**import turtle**

**tur = turtle.Turtle()**

**for i in range(50):**

**turtle.forward(50)**

**turtle.right(144)**

**turtle.done()**

import turtle

turtle.bgcolor("green")

sr=turtle.Turtle()

dot\_distance=25

sr.setposition(-200,200)

def spiral(m,n):

k=0

l=0

f=0

sr.color("black")

while(k<m and l<n):

if f==1:

sr.right(90)

for i in range(l,n):

sr.forward(dot\_distance)

k+=1

f=1

sr.right(90)

for i in range(k,m):

sr.forward(dot\_distance)

n-=1

sr.right(90)

if(k<m):

for i in range(n-1,l-1,-1):

sr.forward(dot\_distance)

m-=1

sr.right(90)

if(l<n):

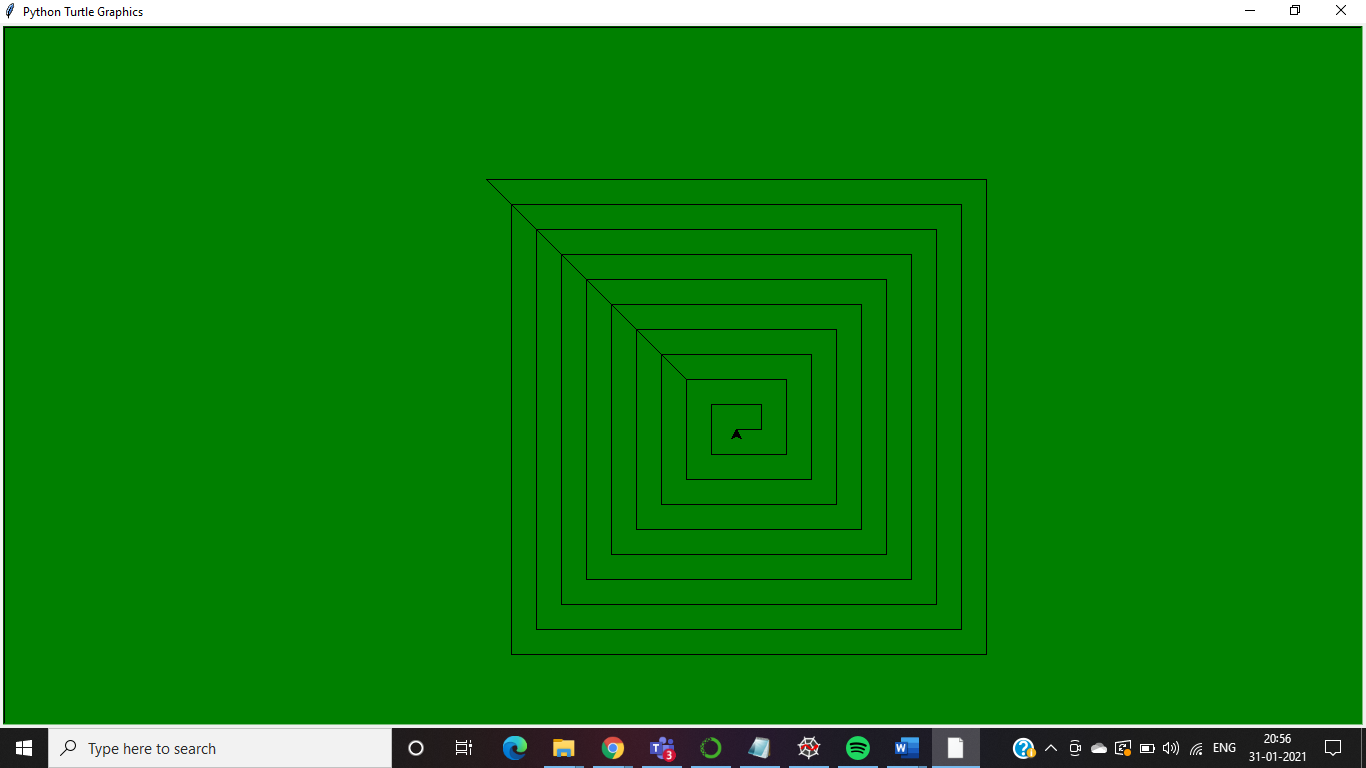
for i in range(m-1,k-1,-1):

sr.forward(dot\_distance)

l+=1

spiral(20,20)

turtle.done()



**Program 5: Tic Tac Toe game**

tic\_tac\_toe\_Board = {'1': ' ', '2': ' ', '3': ' ',

'4': ' ', '5': ' ', '6': ' ',

'7': ' ', '8': ' ', '9': ' '}

board\_keys = []

for key in tic\_tac\_toe\_Board:

board\_keys.append(key)

def printBoard(board):

print(board['1'] + '|' + board['2'] + '|' + board['3'])

print('-+-+-')

print(board['4'] + '|' + board['5'] + '|' + board['6'])

print('-+-+-')

print(board['7'] + '|' + board['8'] + '|' + board['9'])

def game():

turn = 'X'

count = 0

for i in range(10):

printBoard(tic\_tac\_toe\_Board)

print("It's your turn," + turn + ".Move to which place?")

move = input()

if tic\_tac\_toe\_Board[move] == ' ':

tic\_tac\_toe\_Board[move] = turn

count += 1

else:

print("That place is already filled.\nMove to which place?")

continue

if count >= 5:

if tic\_tac\_toe\_Board['7'] == tic\_tac\_toe\_Board['8'] == tic\_tac\_toe\_Board['9'] != ' ':

printBoard(tic\_tac\_toe\_Board)

print("\nGame Over.\n")

print(" \*\*\*\* " + turn + " won. \*\*\*\*")

break

elif tic\_tac\_toe\_Board['4'] == tic\_tac\_toe\_Board['5'] == tic\_tac\_toe\_Board['6'] != ' ':

printBoard(tic\_tac\_toe\_Board)

print("\nGame Over.\n")

print(" \*\*\*\* " + turn + " won. \*\*\*\*")

break

elif tic\_tac\_toe\_Board['1'] == tic\_tac\_toe\_Board['2'] == tic\_tac\_toe\_Board['3'] != ' ':

printBoard(tic\_tac\_toe\_Board)

print("\nGame Over.\n")

print(" \*\*\*\* " + turn + " won. \*\*\*\*")

break

elif tic\_tac\_toe\_Board['1'] == tic\_tac\_toe\_Board['4'] == tic\_tac\_toe\_Board['7'] != ' ':

printBoard(tic\_tac\_toe\_Board)

print("\nGame Over.\n")

print(" \*\*\*\* " + turn + " won. \*\*\*\*")

break

elif tic\_tac\_toe\_Board['2'] == tic\_tac\_toe\_Board['5'] == tic\_tac\_toe\_Board['8'] != ' ':

printBoard(tic\_tac\_toe\_Board)

print("\nGame Over.\n")

print(" \*\*\*\* " + turn + " won. \*\*\*\*")

break

elif tic\_tac\_toe\_Board['3'] == tic\_tac\_toe\_Board['6'] == tic\_tac\_toe\_Board['9'] != ' ':

printBoard(tic\_tac\_toe\_Board)

print("\nGame Over.\n")

print(" \*\*\*\* " + turn + " won. \*\*\*\*")

break

elif tic\_tac\_toe\_Board['7'] == tic\_tac\_toe\_Board['5'] == tic\_tac\_toe\_Board['3'] != ' ':

printBoard(tic\_tac\_toe\_Board)

print("\nGame Over.\n")

print(" \*\*\*\* " + turn + " won. \*\*\*\*")

break

elif tic\_tac\_toe\_Board['1'] == tic\_tac\_toe\_Board['5'] == tic\_tac\_toe\_Board['9'] != ' ':

printBoard(tic\_tac\_toe\_Board)

print("\nGame Over.\n")

print(" \*\*\*\* " + turn + " won. \*\*\*\*")

break

if count == 9:

print("\nGame Over.\n")

print("It's a Tie!!")

if turn == 'X':

turn = 'O'

else:

turn = 'X'

restart = input("Do want to play Again?(y/n)")

if restart == "y" or restart == "Y":

for key in board\_keys:

tic\_tac\_toe\_Board[key] = " "

game()

game()

