**Case Study #1: Network Traffic Analysis Tool using Tuples:** You are tasked with building a program to analyze network traffic logs using Python. Implement the following functionalities, focusing on networking concepts and utilizing tuples:

**1.**      **Add Log Entries:**

a.       Create an empty list to represent the network traffic log.

b.      Allow the user to add log entries by entering details such as source IP, destination IP, protocol, and bytes transferred.

c.       Store each log entry as a tuple within the network traffic log list.

**2.**      **Display Log Entries:**

a.       Provide an option for the user to display the list of all log entries in the network traffic log.

b.      Display each log entry's source IP, destination IP, protocol, and bytes transferred in a readable format.

**3.**      **Search for Log Entries by IP:**

a.       Implement a search functionality where the user can enter an IP address to find all log entries involving that IP.

b.      If log entries are found, display their details.

c.       If no log entries are found, inform the user.

**4.**      **Calculate Total Bytes Transferred:**

a.       Provide an option to calculate and display the total bytes transferred in the network traffic log.

**5.**      **Filter Log Entries by Protocol:**

a.       Allow the user to filter log entries based on the protocol used (e.g., TCP, UDP).

b.      Display the details of log entries that match the specified protocol.

**6.**      **Exit System:**

a.       Allow the user to exit the system.

b.      Ensure that your program is user-friendly, handles errors gracefully, and provides clear instructions to the user.

PROGRAM:

l = []

def add():

    s = input("Enter source ip: ")

    d = input("Enter destination ip: ")

    p = input("Enter protocol: ")

    b = int(input("Enter bytes transferred "))

    l.append((s,d,p,b))

def show():

    print("log entries :")

    print("source\tdestination\tprotocol\tbytes")

    for i in l: print(i[0],"\t",i[1],"\t",i[2],"\t",i[3])

def search():

    dn = input("Enter ip address: ")

    found=False

    for i in l:

        if (i[0]==dn or i[1]==dn):

            print(i[0],"\t",i[1],"\t",i[2],"\t",i[3])

        found=True

    if found==False:

        print("logs not found")

def filter():

    dn = input("Enter protocol: ")

    found=False

    for i in l:

        if i[2]==dn:

            print(i[0],"\t",i[1],"\t",i[2],"\t",i[3])

            found=True

    if found==False:

        print("logs not found")

def calc():

    total = 0

    for i in l: total+=i[3]

    print("total bytes transferred = ",total)

while(True):

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    print("welcome to network traffic analysis tool")

    print("1. Add log entries")

    print("2. display log entries")

    print("3. search for a log entries by ip")

    print("4. calculate total bytes transferred")

    print("5. filter log entries by protocol")

    print("6. exit")

    c = int(input("Enter ur choice: "))

    if(c==1):

        add()

    elif c==2:

        show()

    elif c==3:

        search()

    elif c==4:

        calc()

    elif c==5:

        filter()

    elif c==6:

        print("Exit")

        break

    else:

        print("invalid input")

OUTPUT:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

welcome to network traffic analysis tool

1. Add log entries

2. display log entries

3. search for a log entries by ip

4. calculate total bytes transferred

5. filter log entries by protocol

6. exit

Enter ur choice: 2

log entries :

source destination protocol bytes

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

welcome to network traffic analysis tool

1. Add log entries

2. display log entries

3. search for a log entries by ip

4. calculate total bytes transferred

5. filter log entries by protocol

6. exit

Enter ur choice: 1

Enter source ip: 123.34.3.2

Enter destination ip: 123.5.3.6

Enter protocol: tcp

Enter bytes transferred 2000

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

welcome to network traffic analysis tool

1. Add log entries

2. display log entries

3. search for a log entries by ip

4. calculate total bytes transferred

5. filter log entries by protocol

6. exit

Enter ur choice: 1

Enter source ip: 232.4.5.3

Enter destination ip: 232.4.5.4

Enter protocol: udp

Enter bytes transferred 1000

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

welcome to network traffic analysis tool

1. Add log entries

2. display log entries

3. search for a log entries by ip

4. calculate total bytes transferred

5. filter log entries by protocol

6. exit

Enter ur choice: 2

log entries :

source destination protocol bytes

123.34.3.2 123.5.3.6 tcp 2000

232.4.5.3 232.4.5.4 udp 1000

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

welcome to network traffic analysis tool

1. Add log entries

2. display log entries

3. search for a log entries by ip

4. calculate total bytes transferred

5. filter log entries by protocol

6. exit

Enter ur choice: 3

Enter ip address: 232.4.5.3

232.4.5.3 232.4.5.4 udp 1000

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

welcome to network traffic analysis tool

1. Add log entries

2. display log entries

3. search for a log entries by ip

4. calculate total bytes transferred

5. filter log entries by protocol

6. exit

Enter ur choice: 4

total bytes transferred = 3000

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

welcome to network traffic analysis tool

1. Add log entries

2. display log entries

3. search for a log entries by ip

4. calculate total bytes transferred

5. filter log entries by protocol

6. exit

Enter ur choice: 5

Enter protocol: tcp

123.34.3.2 123.5.3.6 tcp 2000

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

welcome to network traffic analysis tool

1. Add log entries

2. display log entries

3. search for a log entries by ip

4. calculate total bytes transferred

5. filter log entries by protocol

6. exit

Enter ur choice: 6

Exit

**Case Study #2: Event Attendance Tracking System with Sets:** You are tasked with building an Event Attendance Tracking System using Python, focusing on the utilization of sets. Implement the following functionalities:

**1.**      **Record Attendees:**

a.       Create an empty set to represent the attendees of an event.

b.      Allow the user to add attendees by entering their names.

c.       Use a set to ensure that each attendee is unique.

**2.**      **Display Attendees:**

a.       Provide an option for the user to display the list of all attendees at the event.

b.      Display each attendee's name in a readable format.

**3.**      **Check Attendance:**

a.       Implement a feature to check if a specific attendee is present.

b.      Allow the user to enter the name of an attendee and check if they are in the set.

c.       Display a message indicating whether the attendee is present or not.

**4.**      **Remove Attendee:**

a.       Allow the user to remove a specific attendee from the event.

b.      Allow the user to enter the name of the attendee to remove.

c.       If the attendee is found, remove them from the set.

d.      If the attendee is not found, inform the user.

**5.**      **Calculate Attendance Statistics:**

a.       Provide an option to calculate and display the total number of attendees and unique attendees.

b.      Use set operations to determine unique attendees.

**6.**      **Exit System:**

a.       Allow the user to exit the system.

PROGRAM:

l = set()

def add():

    s = input("Enter attendee name : ")

    l.add(s)

def show():

    print("attendee name ")

    for i in l: print(i)

def check():

    dn = input("Enter attendee name: ")

    if dn in l:

        print("present")

    else:

        print("absent")

def calc():

    print("total no. of attendees = ",len(l))

    show()

def remove():

    dn = input("Enter attendee name: ")

    if dn in l:

        l.remove(dn)

        print("removed")

    else:

        print("not present")

while(True):

    print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

    print("welcome to event attendance tracking system")

    print("1. Add attendees")

    print("2. display attendees")

    print("3. check attendance")

    print("4. remove attendance")

    print("5. calculate attendance statistics")

    print("6. exit")

    c = int(input("Enter ur choice: "))

    if(c==1):

        add()

    elif c==2:

        show()

    elif c==3:

        check()

    elif c==4:

        remove()

    elif c==5:

        calc()

    elif c==6:

        print("Exit")

        break

    else:

        print("invalid input")

OUTPUT:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

welcome to event attendance tracking system

1. Add attendees

2. display attendees

3. check attendance

4. remove attendance

5. calculate attendance statistics

6. exit

Enter ur choice: 1

Enter attendee name : charan

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

welcome to event attendance tracking system

1. Add attendees

2. display attendees

3. check attendance

4. remove attendance

5. calculate attendance statistics

6. exit

Enter ur choice: 1

Enter attendee name : aditya

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

welcome to event attendance tracking system

1. Add attendees

2. display attendees

3. check attendance

4. remove attendance

5. calculate attendance statistics

6. exit

Enter ur choice: 1

Enter attendee name : sandeep

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

welcome to event attendance tracking system

1. Add attendees

2. display attendees

3. check attendance

4. remove attendance

5. calculate attendance statistics

6. exit

Enter ur choice: 2

attendee name

charan

sandeep

aditya

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

welcome to event attendance tracking system

1. Add attendees

2. display attendees

3. check attendance

4. remove attendance

5. calculate attendance statistics

6. exit

Enter ur choice: 3

Enter attendee name: charan

present

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

welcome to event attendance tracking system

1. Add attendees

2. display attendees

3. check attendance

4. remove attendance

5. calculate attendance statistics

6. exit

Enter ur choice: 5

total no. of attendees = 3

attendee name

charan

sandeep

aditya

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

welcome to event attendance tracking system

1. Add attendees

2. display attendees

3. check attendance

4. remove attendance

5. calculate attendance statistics

6. exit

Enter ur choice: 4

Enter attendee name: charan

removed

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

welcome to event attendance tracking system

1. Add attendees

2. display attendees

3. check attendance

4. remove attendance

5. calculate attendance statistics

6. exit

Enter ur choice: 6

Exit