File Handing

Binary File Handling

Question 1

WAP to write a tuple of 5 elements into a binary file named tupbin.dat. Also read and display the contents of tupbin.dat

Source Code:

```
import pickle
tup = tuple(map(int, input("Enter 5 integers: ").split()))
with open('tupbin.dat', 'wb') as f:
    pickle.dump(tup, f)
with open('tupbin.dat', 'rb') as f:
    tup = pickle.load(f)
print(tup)
```

Output :

```
(1, 2, 3, 4, 5)
```

Question 2

WAP to store n random numbers of 4 digits each in a list named pwdlist. Write the contents of the list pwdlist into a binary file. Also read and display the binary file contents

Source code:

```
import random
import pickle

n = int(input("Enter the number of random numbers: "))

pwdlist = []

for i in range(n):
    pwd = random.randint(1000, 9999)
    pwdlist.append(pwd)

print(pwdlist)

with open('pwdbin.dat', 'wb') as f:
    pickle.dump(pwdlist, f)

with open('pwdbin.dat', 'rb') as f:
    pwdlist = pickle.load(f)

print(pwdlist)
```

Output:

```
Enter the number of random numbers: 4
[4762, 2910, 1406, 6100]
[4762, 2910, 1406, 6100]
```

Question 3

Write a program to create a binary file called emp.dat and write into it the employee details of n employees available in the form of dictionaries.

Source code:

```
import pickle

n = int(input("Enter the number of
employees: "))

emp_details = []

for i in range(n):
    emp_details.append(dict(
        name=input("Enter the employee's
name: "),
        age=int(input("Enter the employee's
age: ")),
        salary=float(input("Enter the
employee's salary: "))
    ))

with open('emp.dat', 'wb') as f:
    pickle.dump(emp_details, f)

print("Employee details written to emp.dat")
3
with open('emp.dat', 'rb') as f:
    emp_details = pickle.load(f)

print(emp_details)
```

Output:

```
Enter the number of employees: 3
Enter the employee's name: sai
Enter the employee's age: 16
Enter the employee's salary: 300000000
Enter the employee's name: karthik
Enter the employee's age: 16
Enter the employee's salary: 4899999999
Enter the employee's name: ketha
Enter the employee's age: 16
Enter the employee's salary: 57899999
Employee details written to emp.dat
```

Question 4

Write a program to open the file “Emp.dat” (created in program 3), read the objects written in it and display them

Source code:

```
import pickle

with open('emp.dat', 'rb') as f:
    emp_details = pickle.load(f)

print(emp_details)
```

Output :

```
[{'name': 'sai ', 'age': 16, 'salary':
300000000.0}, {'name': 'karthik ', 'age':
16, 'salary': 4899999999.0}, {'name':
'ketha', 'age': 16, 'salary': 57899999.0}]
```

Question 5

Source Code:

```
import pickle as p
import os

# (a)
def insert(filename: str):
    with open(filename, 'ab') as bin_f:
        print(f"In the file '{filename}' has
been opened.")
        tnum = int(input("Enter the travel
number: "))
        source = input("Enter the source: ")
        destination = input("Enter the
destination: ")
        cost = float(input("Enter the cost
of the trip: "))
        traTicket = {'tnum': tnum, 'source':
source, 'destination': destination, 'cost':
cost}
        p.dump(traTicket, bin_f)
        print(f"The dictionary '{traTicket}'
has been added successfully to the file
'{filename}'.")

# (b)
def displayall(filename):
    try:
        with open(filename, 'rb') as bf:
            try:
                while True:
                    content = p.load(bf)
                    print(content)
            except EOFError:
                print("End of file
reached.\n")
    except FileNotFoundError:
        print("This file doesn't exist!
Please try again.")

# (c)
def search(filename, kTnum):
    try:
        with open(filename, 'rb') as bf:
            try:
                a = True
                while a:
                    content = p.load(bf)
                    for key in
content.keys():
                        if content[key] ==
kTnum:
                            print(content)
                            a = False
                            break
            except EOFError:
                print("No matches found.")
    except FileNotFoundError or TypeError:
        print("This file name is wrong or
doesn't exist! Please try again.")

# (d)
def modify_withtemp(filename, kTnum):
    with open(filename, 'rb') as f1,
open('temp.dat', 'wb') as f2:
        try:
            while True:
                ticket = p.load(f1)
                if ticket['tnum'] == kTnum:
                    new_source =
input("Enter New Source: ")
                    new_destination =
input("Enter New Destination: ")
                    new_cost =
float(input("Enter New Cost: "))
                    modified_ticket =
{'tnum': kTnum, 'source': new_source,
'destination': new_destination, 'cost':
new_cost}
                    p.dump(modified_ticket,
f2)
                    print("Record modified
successfully.")
```

```
                else:
                    p.dump(ticket, f2)
        except EOFError:
            print("Modifications complete.")
            f1.close()
            f2.close()
            os.remove(filename)
            os.rename('temp.dat', filename)

# (e)
def modify_withouttemp(filename, kTnum):
    with open(filename, 'rb+') as bf:
        try:
            while True:
                position = bf.tell()
                ticket = p.load(bf)
                if ticket['tnum'] == kTnum:
                    new_source =
input("Enter New Source: ")
                    new_destination =
input("Enter New Destination: ")
                    new_cost =
float(input("Enter New Cost: "))
                    modified_ticket =
{'tnum': kTnum, 'source': new_source,
'destination': new_destination, 'cost':
new_cost}
                    bf.seek(position)
                    p.dump(modified_ticket,
bf)
                    print("Record modified
successfully.")
        except EOFError:
            print("Modifications complete.")
```

```
# (f)
def delete_withtemp(filename, kTnum):
    temp_filename = "temp.bin"
    try:
        with open(filename, 'rb') as rb,
open(temp_filename, 'wb') as temp_file:
            while True:
                try:
                    ticket = p.load(rb)
                    if ticket["tnum"] !=
kTnum:
                        p.dump(ticket,
temp_file)
                except EOFError:
                    break
        rb.close()
        temp_file.close()
        os.remove(filename)
        os.rename(temp_filename, filename)
        print("Record deleted
successfully.")
    except FileNotFoundError:
        print("File doesn't exist or no
records found.")

while True:
    print("""
    MENU:
    1 = Insert a New Ticket Record (Retain
previous data)
    2 = View all Records
    3 = Search a particular Ticket
    4 = Modify Ticket Details (Using
Temporary File)
```

```
    5 = Modify Ticket Details (Without Using
Temporary File)
    6 = Delete a record
    7 = Exit
    """)

    choice = input("Enter your choice here:
")

    if choice == '1':
        f_name = input("Enter the name of
the file: ")
        insert(f_name)
    elif choice == '2':
        f_name = input("Enter the name of
the file: ")
        displayall(f_name)
    elif choice == '3':
        f_name = input("Enter the name of
the file: ")
        k_tnum = int(input("Enter the Ticket
number: "))
        search(f_name, k_tnum)
    elif choice == '4':
        f_name = input("Enter the name of
the file: ")
        k_tnum = int(input("Enter the Ticket
number: "))
        modify_withtemp(f_name, k_tnum)
    elif choice == '5':
        f_name = input("Enter the name of
the file: ")
        k_tnum = int(input("Enter the Ticket
number: "))
        modify_withouttemp(f_name, k_tnum)
    elif choice == '6':
        f_name = input("Enter the name of
the file: ")
        k_tnum = int(input("Enter the Ticket
number: "))
        delete_withtemp(f_name, k_tnum)
    elif choice == '7':
        break
    else:
        print("**INVALID INPUT**")
```

Output:

```
MENU:
1 = Insert a New Ticket Record (Retain
previous data)
2 = View all Records
3 = Search a particular Ticket
4 = Modify Ticket Details (Using Temporary
File)
5 = Modify Ticket Details (Without Using
Temporary File)
6 = Delete a record
7 = Exit
Enter your choice here: 2
Enter the name of the file: ticket.dat
('tnum': 1212,
'source': 'Delhi',
'destination': 'Mumbai',
'cost': 75000.0)
('tnum': 1234,
'source': 'Mumbai', 'destination':
'Hydrabad',
'cost': 125000.0]
End of file reached.
```

```
MENU:
1 = Insert a New Ticket Record (Retain
previous data)
2 = View all Records
3 = Search a particular Ticket
```

Question 6

Source code:

```
import pickle as p
import os

# (a)
def delete_DM(filename):
    temp_filename = "temp.dat"
    try:
        with open(filename, 'rb') as rb,
open(temp_filename, 'wb') as temp_file:
            while True:
                try:
                    ticket = p.load(rb)
                    if ticket['source'] ==
"Delhi" and ticket['destination'] ==
"Mumbai":
                        pass
                    else:
                        p.dump(ticket,
temp_file)
                except EOFError:
                    break
        rb.close()
        temp_file.close()
        os.remove(filename)
        os.rename(temp_filename, filename)
        print("Record deleted
successfully.")
    except FileNotFoundError:
        print("File doesn't exist or no
records found.")

# (b)
def rec_count(filename):
    try:
        with open(filename, 'rb') as bf:
            count = 0
            while True:
                try:
                    p.load(bf)
                    count += 1
                except EOFError:
                    print(f"The number of
records in this file is {count}.")
                    break
    except FileNotFoundError:
        print("File doesn't exist or no
records found.")

# (c)
def sale(filename):
    try:
        with open(filename, 'ab+') as bf:
            while True:
                try:
                    ticket = p.load(bf)
                    ticket['cost'] =
ticket['cost'] * 1.20
```

```
                    p.dump(ticket, bf)
                except EOFError:
                    break
    except FileNotFoundError:
        print("File doesn't exist or no
records found.")

delete_DM("ticket.dat")
rec_count('ticket.dat')
sale('ticket.dat')

with open("ticket.dat", 'rb') as file:
    print(p.load(file))
```

Output:

```
Record deleted successfully.
The number of records in this file is 1.
('tnum': 1234,
'source': 'Hydrabad', 'destination':
'Chennai',
'cost': 150000.0}
```