**Assignment 3:**

Question 1:

AIM:

To write a code that reads movie and displays the required data.

# Code:

"""

Assignment 3 Question 1 done by Vansh Aggarwal

Date: 18/6/24

Time of start: 11:30

"""

import pickle as p

#create file

def create\_file():

    f = open("movie.dat", "wb")

    f.close()

    print("File Created")

while True:

    print ("Do you want to erase the file or continue with the previous file: ")

    opt2 = int(input ("Enter 1 for yes and 2 for no: "))

    if opt2 == 1:

        create\_file()

        break

    elif opt2 == 2:

        print ("File Retained.")

        break

    else:

        print ("Try again.")

# Main Code:

def addmovie():

    rating = 1000

    while rating<0 or rating > 10:

        rating = int(input("Enter the movie rating: "))

    movie = input ("Enter movie name: ")

    language = input ("Enter movie language: ")

    fout = open("movie.dat", "ab")

    temp\_input = [rating, movie, language]

    p.dump(temp\_input, fout)

    fout.close()

def readmovie():

  data = []

  try:

    with open("movie.dat", "rb") as fout:

      while True:

        data\_1 = p.load(fout)

        data.append(data\_1)

  except EOFError:

    print("End of file reached in 'movie.dat'.")

  finally:

    fout.close()

    return(data)

def bestrated():

    movie = ""

    rating = 0

    data = readmovie()

    for i in data:

        if i[0]>rating:

            movie = i

            rating = i[0]

    return(movie)

#Main Loop:

while True:

    print ('''

Options:

1) Write the Movie to a binary file.

2) Read the Movie from a binary file and display the records.

3) Search the record of the Movie which has the highest rating. ''' )

    opt1 = int(input("Enter the option to execute: "))

    if opt1==1:

        addmovie()

    if opt1 ==2:

        data = readmovie()

        for i in data:

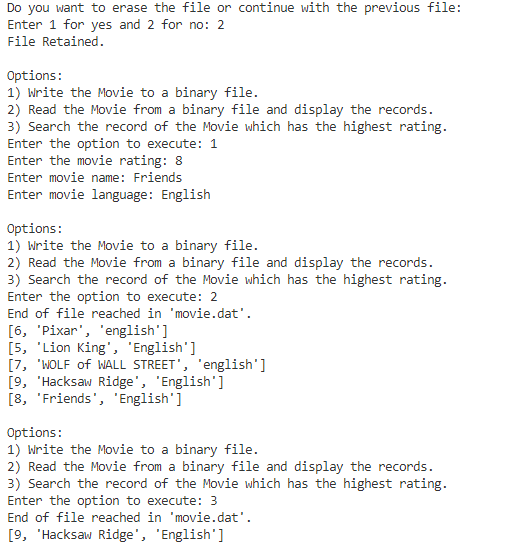
            print (i)

    if opt1 ==3:

        mov = bestrated()

        print (mov)

# Sample Output:



Question 2:

AIM:

Write menu driven python program to create a binary file (Product.dat) storing the   
Dictionary Product and perform given operations.

# Code:

"""

Assignment 3 Question 2 done by Vansh Aggarwal

Date: 21/6/24

Time of start: 11:29

"""

import pickle as p

import os

#create file

def create\_file():

    f = open("product.dat", "wb")

    f.close()

    print("File Created")

while True:

    print ("Do you want to erase the file or continue with the previous file: ")

    opt2 = int(input ("Enter 1 for yes and 2 for no: "))

    if opt2 == 1:

        create\_file()

        break

    elif opt2 == 2:

        print ("File Retained.")

        break

    else:

        print ("Try again.")

# Main Code:

def addproduct():

    fout = open("product.dat", "ab")

    pid = int(input("Enter product ID:"))

    pname = input("Enter Product Name: ")

    qty = int(input("Enter Quantity of the product: "))

    price = float(input("Enter Price of product: "))

    product = {"pid":pid,"pname":pname, "qty":qty, "price":price}

    p.dump(product, fout)

    fout.close()

def readproduct():

  data = []

  try:

    with open("product.dat", "rb") as fout:

      while True:

        data\_1 = p.load(fout)

        data.append(data\_1)

  except EOFError:

    print()

  finally:

    fout.close()

    return(data)

def searchproduct(search\_string):

    data = readproduct()

    for i in data:

        if i["pname"] == search\_string:

            return(i)

    else:

        return("No such product found. ")

def update\_price(search\_string):

    fout = open("product1.dat", "wb")

    fout.close()

    ################################

    fout = open("product1.dat", "ab")

    data = readproduct()

    for i in data:

        if i['pname'] != search\_string:

            p.dump(i,fout)

        elif i['pname'] == search\_string:

            new\_price = float(input("Enter the new price: "))

            i["price"] = new\_price

            p.dump(i,fout)

    fout.close()

    os.remove("product.dat")

    os.rename("product1.dat", "product.dat")

    print ("changes made")

def del\_product(search\_string):

    fout = open("product1.dat", "wb")

    fout.close()

    ################################

    fout = open("product1.dat", "ab")

    product = searchproduct(search\_string)

    data = readproduct()

    for i in data:

        if i != product:

            p.dump(i,fout)

    fout.close()

    os.remove("product.dat")

    os.rename("product1.dat", "product.dat")

    print ("Lines deleted")

def move\_product(search\_string):

    fout = open("product2.dat", "wb")

    fout.close()

    ################################

    fout = open("product2.dat", "ab")

    #################################

    fin = open("product1.dat", "wb")

    fin.close()

    ################################

    fin = open("product1.dat", "ab")

    product = searchproduct(search\_string)

    data = readproduct()

    for i in data:

        if i == product:

            p.dump(i,fout)

        if i != product:

            p.dump(i, fin)

    fin.close()

    fout.close()

    os.remove("product.dat")

    os.rename("product1.dat", "product.dat")

    print ("Lines moved. ")

#Main Loop:

while True:

    print ('''

Options:

1) Add more data

2) Display All

3) Search for a product by name

4) Update the product price

5) Delete the product

6) Move certain products (by name) to a separate file.''' )

    opt1 = int(input("Enter the option to execute: "))

    if opt1==1:

        addproduct()

    elif opt1 ==2:

        data = readproduct()

        for i in data:

            print (i)

    elif opt1 ==3:

        search\_string= input("Enter the product to search: ")

        required\_p = searchproduct(search\_string)

        print (required\_p)

    elif opt1 == 4:

        search\_string= input("Enter the product: ")

        update\_price(search\_string)

    elif opt1 == 5:

        search\_string= input("Enter the product: ")

        del\_product(search\_string)

    elif opt1 == 6:

        n = int(input("Enter the number of items to move: "))

        for i in range (n):

            search\_string= input("Enter the product: ")

            move\_product(search\_string)

    elif opt1 ==7:

        break

# Sample Output:

PS D:\Vansh Aggarwal-11b\Vansh\_12B> & C:/Python310/python.exe "d:/Vansh Aggarwal-11b/Vansh\_12B/Assignment 3/products.py"

Do you want to erase the file or continue with the previous file:

Enter 1 for yes and 2 for no: 2

File Retained.

Options:

1) Add more data

2) Display All

3) Search for a product by name

4) Update the product price

5) Delete the product

6) Move certain products (by name) to a separate file.

Enter the option to execute: 1

Enter product ID:2323

Enter Product Name: testpaper

Enter Quantity of the product: 123

Enter Price of product: 7878

Options:

1) Add more data

2) Display All

3) Search for a product by name

4) Update the product price

5) Delete the product

6) Move certain products (by name) to a separate file.

Enter the option to execute: 2

{'pid': 1829382, 'pname': 'books', 'qty': 128, 'price': 90.0}

{'pid': 83948, 'pname': 'chauhan', 'qty': 1902, 'price': 29.0}

{'pid': 2, 'pname': 'no', 'qty': 192, 'price': 48903.0}

{'pid': 2323, 'pname': 'testpaper', 'qty': 123, 'price': 7878.0}

Options:

1) Add more data

2) Display All

3) Search for a product by name

4) Update the product price

5) Delete the product

6) Move certain products (by name) to a separate file.

Enter the option to execute: 3

Enter the product to search: no

{'pid': 2, 'pname': 'no', 'qty': 192, 'price': 48903.0}

Options:

1) Add more data

2) Display All

3) Search for a product by name

4) Update the product price

5) Delete the product

6) Move certain products (by name) to a separate file.

Enter the option to execute: 4

Enter the product: books

Enter the new price: 100

changes made

Options:

1) Add more data

2) Display All

3) Search for a product by name

4) Update the product price

5) Delete the product

6) Move certain products (by name) to a separate file.

Enter the option to execute: 2

{'pid': 1829382, 'pname': 'books', 'qty': 128, 'price': 100.0}

{'pid': 83948, 'pname': 'chauhan', 'qty': 1902, 'price': 29.0}

{'pid': 2, 'pname': 'no', 'qty': 192, 'price': 48903.0}

{'pid': 2323, 'pname': 'testpaper', 'qty': 123, 'price': 7878.0}

Options:

1) Add more data

2) Display All

3) Search for a product by name

4) Update the product price

5) Delete the product

6) Move certain products (by name) to a separate file.

Enter the option to execute: 5

Enter the product: no

Lines deleted

Options:

1) Add more data

2) Display All

3) Search for a product by name

4) Update the product price

5) Delete the product

6) Move certain products (by name) to a separate file.

Enter the option to execute: 2

{'pid': 1829382, 'pname': 'books', 'qty': 128, 'price': 100.0}

{'pid': 83948, 'pname': 'chauhan', 'qty': 1902, 'price': 29.0}

{'pid': 2323, 'pname': 'testpaper', 'qty': 123, 'price': 7878.0}

Options:

1) Add more data

2) Display All

3) Search for a product by name

4) Update the product price

5) Delete the product

6) Move certain products (by name) to a separate file.

Enter the option to execute: 6

Enter the number of items to move: 1

Enter the product: chauhan

Lines moved.

Options:

1) Add more data

2) Display All

3) Search for a product by name

4) Update the product price

5) Delete the product

6) Move certain products (by name) to a separate file.

Enter the option to execute: 2

{'pid': 1829382, 'pname': 'books', 'qty': 128, 'price': 100.0}

{'pid': 2323, 'pname': 'testpaper', 'qty': 123, 'price': 7878.0}