Binary File –Exam based questions

A binary file "student.dat" has structure [rollno, name, marks].

i. Write a user defined function insertRec() to input data for a student and add to student.dat.

ii. Write a function searchRollNo(r) in Python which accepts the student's rollno as parameter and searches the record in the file "student.dat" and shows the details of student i.e. rollno, name and marks (if found) otherwise shows the message as 'No record found'.

(i)

```
import pickle
def insertRec():
        f=open("student.dat","ab")
        rollno = int (input("Enter Roll Number : "))
        name=input("Enter Name :")
        marks = int(input("Enter Marks : "))
        rec = [rollno, name, marks ]
        pickle.dump( rec, f )
        f.close()
```

(ii)

```
def searchRollNo( r ):
        f=open("student.dat","rb")
        flag = False
      while True:
                try:
                    rec=pickle.load(f)
                    if rec[0] == r :
                            print(rec['Rollno'])
                            print(rec['Name'])
                            print(rec['Marks])
                            flag == True
                  except EOFError:
                      break
          if flag == False:
                print("No record Found")
      f.close()
```

A binary file "emp.dat" has structure [EID, Ename, designation, salary].

i. Write a user defined function CreateEmp() to input data for a record and create a file emp.dat.

ii. Write a function display() in Python to display the detail of all employees whose salary is more than 50000.

(i)

```
import pickle
def CreateEmp():
        f1=open("emp.dat",'wb')
        eid=input("Enter E. Id")
        ename=input("Enter Name")
        designation=input("Enter Designation")
        salary=int(input("Enter Salary"))
        l=[eid,ename,designation,salary]
        pickle.dump(l,f1)
        f1.close()
```

(ii)

```
import pickle
def display():
      f2=open("emp.dat","rb")
      try:
         while True:
            rec=pickle.load(f2)
            if rec[3]>5000:
                print(rec[0],rec[1],rec[2],rec[3])
     except:
         f2.close()
```

Write a python program to append a new records in a binary file –"student.dat". The record can have Rollno, Name and Marks.

```
import pickle
while True:
        rollno = int(input("Enter your rollno: "))
        name = input("Enter your name: ")
```

Write a python program to search and display the record of the student from a binary file "Student.dat" containing students records (Rollno, Name and Marks). Roll number of the student to be searched will be entered by the user.

```
        marks = int(input("enter marks obtained: "))
        d = [rollno, name, marks]
        f1 = open("Student.dat", "wb")
        pickle.dump(d, f1)
        choice = input("enter more records: y/n")
         if choice== "N":
               break
         f1.close()
```

```
import pickle
f1 = open("Student.dat", "rb")
rno = int(input("Enter the roll no to search: "))
flag = 0
try:
    while True:
          r = pickle.load(f1)
          if rno == r[0]:
                print (rollno, name, marks)
                flag = 1
except:
      if flag == 0:
          print("Record not found…")
f1.close()
```

i. A binary file "emp.DAT" has structure (EID, Ename, designation,salary). Write a function to add more records of employes in existing file emp.dat.
ii. Write a function Show() in Python that would read detail of employee from file "emp.dat" and display the details of those employee whose designation is "Salesman".

(i)

```
import pickle
def createemp:
     f1=open("emp.dat",'ab')
     eid=input("Enter E. Id")
     ename=input("Enter Name")
     designation=input("Enter Designation")
     salary=int(input("Enter Salary"))
     l=[eid,ename,designation,salary]
     pickle.dump(l,f1)
     f1.close()
```

(ii)

```
def display():
        f2=open("emp.dat","rb")
        try:
            while True:
                rec=pickle.load(f2)
                if (rec[2]=='Manager'):
                      print(rec[0],rec[1], rec[2],rec[3])
         except:
               break
       f2.close()
```

A binary file named "EMP.dat" has some records of the structure [EmpNo, EName, Post, Salary]
(a) Create a binary file "EMP.dat" that stores the records of employees and display them one by one.
(b) Display the records of all those employees who are getting salaries between 25000 to 30000.

(a)

```
import pickle
f1 = open('emp.dat','rb')
try:
        while True:
                e = pickle.load(f1)
                print(e)
except:
          f1.close()
```

(b)

```
import pickle
f1 = open('emp.dat','rb')
try:
        while True:
                e = pickle.load(f1)
                if(e[3]>=25000 and e[3]<=30000):
                        print(e)
except:
          f1.close()
```

A binary file "Book.dat" has structure [BookNo, Book_Name, Author, Price].

i. Write a user defined function CreateFile() to input data for a record and add to “Book.dat” .
ii. Write a function CountRec(Author) in Python which accepts the Author name as parameter and count and return number of books by the given Author are stored in the binary file “Book.dat”

(i)

```
import pickle
def createFile():
        f=open("Book.dat","ab")
        BookNo=int(input("Book Number : "))
        Book_name=input("Name :")
        Author = input("Author:" )
        Price = int(input("Price : "))
        rec=[BookNo,Book_Name,Author,Price]
        pickle.dump(rec,f)
        f.close()
```

(ii)

```
def CountRec(Author):
        f=open("Book.dat","rb")
        num = 0
        try:
            while True:
                rec=pickle.load(f)
                if Author==rec[2]:
                        num = num + 1
        except:
            f.close()
        return num
```

A binary file student.dat has structure (rollno,name,class,percentage). Write a program to updating a record in the file requires roll number to be fetched from the user whose name is to be updated

```
import pickle
import os
f1 = open('student.dat','rb')
f2=open("temp.dat","wb")
r=int(input("enter rollno which you want to search"))
try:
        while True:
                e = pickle.load(f1)
                if e[0]==r:
                    e[1]=input("enter name")
                    pickle.dump(e,f2)
```

A binary file “STUDENT.DAT” has structure (admission_number, Name, Percentage). Write a function countrec() in Python that would read contents of the file “STUDENT.DAT” and display the details of those students whose percentage is above 75. Also display number of students scoring above 75%

```
import pickle
def CountRec():
        f=open("STUDENT.DAT","rb")
        num = 0
        try:
            while True:
                rec=pickle.load(f)
                if rec[2] > 75:
                        print(rec[0],rec[1],rec[2])
```

```
                else:
                    pickle.dump(e,f2)
except:
        f1.close()
        f2.close()
os.remove("student.dat")
os.rename("temp.dat","student,dat")
```

```
                        num = num + 1
    except:
        f.close()
    return num
```

A binary file named "EMP.dat" has some records of the structure [EmpNo, EName, Post, Salary]
(a) Write a user-defined function named NewEmp() to input the details of a new employee from the user and store it in EMP.dat.
(b) Write a user-defined function named SumSalary(Post) that will accept an argument the post of employees & read the contents of EMP.dat and calculate the SUM of salary of all employees of that Post.

(a)

```
import pickle
def NewEmp ( ):
        f = open("EMP.dat","wb")
        EmpNo = int(input("Enter employee number: "))
        EName = input("Enter name:")
        Post = input("Enter post:")
        Salary = int(input("Enter salary"))
        rec = [EmpNo, Ename, Post,Salary]
        pickle.dump(rec, f)
        f.close()
```

(b)

```
def SumSalary(Post):
    f = open("EMP.dat", "rb")
    c=0
    while True:
        try:
            g = p.load(f)
            if g[2]==Post:
                c=c+g[3]
        except:
            f.close()
      print("sum of salary", c)
```

A binary file "Items.dat" has structure as [Code, Description, Price].
i. Write a user defined function MakeFile( ) to input multiple items from the user and add to Items.dat
ii. Write a function SearchRec(Code) in Python which will accept the code as parameter and search and display the details of the corresponding code on screen from Items.dat.

(i)

```
import pickle
def MakeFile( ):
    while True:
        code = input("Enter Item Code :")
        desc = input("Enter description :")
        price = float(input("Enter price:"))
        d= [code,desc,price]
        f = open ("Items.dat", "ab")
        pickle.dump( d,f )
        ch = input("Add more record? (y/n) :")
        if ch=='n':
            break
    f.close( )
```

(ii)

```
def SearchRec(code):
    f = open("Items.dat", "rb")
    found = False
    while True:
        try:
            g = p.load(f)
            if g[0]==code:
                print(g[0],g[1],g[2])
                found=True
                break
        except:
            if found == False:
                print("No such record")
            f.close()
```

A binary file named “TEST.dat” has some records of the structure [TestId, Subject, MaxMarks, ScoredMarks] Write a function in Python named DisplayAvgMarks(Sub) that will accept a subject as an argument and read the contents of TEST.dat. The function will calculate & display the Average of the ScoredMarks of the passed Subject on screen.

```
def SumSalary(Sub):
    f = open("ABC.dat", "rb")
    c=0
    s=0
    while True:
        try:
            g = p.load(f)
            print(g)
            if g[1]==Sub:
                s=s+g[3]
                c=c+1
        except:
            f.close()
      print("sum of salary", s/c)
      f.close()
```

Consider a binary file emp.dat having records in the form of dictionary. E.g {eno:1, name:”Rahul”, sal: 5000} write a python function to display the records of above file for those employees who get salary between 25000 and 30000

```
import pickle
def search():
    f=open(“emp.dat”,”rb”)
    while True:
        try:
            d=pickle.load(f)
            if(d[‘sal’]>=25000 and
d[‘sal’]<=30000):
                print(d)
        except EOFError:
            break
    f.close()
```

A binary file “Bank.dat” has structure as [account_no, cust_name, balance].

i. Write a user-defined function addfile( ) and add a record to Bank.dat.

ii. Create a user-defined function CountRec( ) to count and return the number of customers whose balance amount is more than 100000.

(i)

```
import pickle
def addfile( ):
    f = open(“bank.dat”,”wb”)
    acc_no = int(input(“Enter account
number: “))
    cust_name = input(“Enter name:”)
    bal = int(input(“Enter balance”))
    rec = [acc_no, cust_name, bal]
    p.dump(rec, f)
    f.close()
```

(ii)

```
def CountRec( ):
    f = open(“bank.dat”,”rb”)
    c = 0
    try:
        while True:
            rec = p.load(f)
```

Consider an employee data, Empcode, empname and salary.

(i) Write python function to create binary file emp.dat and store their records.

(ii) write function to read and display all the records

Ans

```
import pickle
def add_record():
    f = open(“emp.dat”,”ab”)
    empcode =int(input(“employee code:”))
    empname = int(input(“empName:”))
    salary = int(input(“salary:”))
    d = [empcode, empname, salary]
    pickle.dump(d,f)
    f.close()
import pickle

def search():
    f=open(“emp.dat”,”rb”)
    while True:
        try:
            d=pickle.load(f)
            print(d)
```

```
            if rec[2] > 100000:
                c += 1
        except:
            f.close()
        return c
```

```
            except EOFError:
                break
        f.close()
```

Write a function SCOUNT( ) to read the content of binary file “NAMES.DAT‟ and display number of records (each name occupies 20 bytes in file) where name begins from “S‟ in it

```
def SCOUNT( ):
    s=' '
    count=0
    f=open('Names.dat', 'rb'):
    while True:
        s = f.read(20)
        if len(s)!=0:
            if s[0].lower()=='s':
                count+=1
    print('names beginning from "S" are ',count)
```

Given a binary file “emp.dat” has structure (Emp_id, Emp_name, Emp_Salary). Write a function in Python countsal() in Python that would read contents of the file “emp.dat” and display the details of those employee whose salary is greater than 20000

```
import pickle
def countsal():
        f = open (“emp.dat”, “rb”)
        n = 0
        try:
            while True:
                rec = pickle.load(f)
                if rec[2] > 20000:
                    print(rec[0], rec[1], rec[2])
                    n = n + 1
        except:
            print(n)
            f.close()
```

Write Python function DISPEMP( ) to read the content of file emp.csv and display only those records where salary is 4000 or above

```
import csv
def DISPEMP():
    csvfile=open('emp.csv'):
    myreader = csv.reader(csvfile,delimiter=',')
    print(EMPNO,EMP NAME,SALARY)
    for row in myreader:
        if int(row[2])>4000:
            print(row[0], row[1],row[2])
```

Consider the following CSV file (emp.csv):

```
Sl,name,salary
1,Peter,3500
2,Scott,4000
3,Harry,5000
4,Michael,2500
5,Sam,4200
```

Write Python function DISPEMP( ) to read the content of file emp.csv and display only those records where salary is 4000 or above

```
import csv
def DISPEMP():
    csvfile=open('emp.csv'):
    myreader = csv.reader(csvfile,delimiter=',')
    print(EMPNO,EMP NAME,SALARY)
    for row in myreader:
        if int(row[2])>4000:
            print(row[0], row[1],row[2])
```

A binary file “Stu.dat” has structure (rollno, name, marks).
Write a function in Python add_record() to input data for a record and add to Stu.dat.

A binary file “Stu.dat” has structure (rollno, name, marks).

```
import pickle
def add_record():
        fobj = open("Stu.dat","ab")
        rollno =int(input("Roll no:"))
        name = int(input("Name:"))
        marks = int(input("Marks:"))
        data = [rollno, name, marks]
        pickle.dump(data,fobj)
        fobj.close()
```

Write a function in python Search_record() to search a record from binary file "Stu.dat" on the basis of roll number.

```
def Search_record():
    f = open("Stu.dat", "rb")
    stu_rec = pickle.load(f)
    found = 0
    rno = int(input("roll number to search:"))
    try:
        for R in stu_rec:
            if R[0] == rno:
                print (R[1], "Found!")
                found = 1
                break
    except:
        if found == 0:
            print ("Sorry, record not found:")
        f.close()
```