

Write a program for transaction processing that write and read object randomly to and from a random access file so that user can add, update, delete and display the account information (accountnumber, lastname, firstname, totalbalance).

```
#include <iostream>
#include <fstream>
#include <cstring>
using namespace std;
class Transaction{
private:
    char firstName[20], lastName[20];
    int accountNumber, totalBalance;
public:
    Transaction(){};
    Transaction(char fn[], char ln[], int an, int tb):accountNumber(an),totalBalance(tb){
        strncpy(firstName,fn,20);
        strncpy(lastName,ln,20);
    }
    friend istream& operator>>(istream& in, Transaction &tr);
    friend ostream& operator<<(ostream& out, Transaction tr);
};

istream& operator>>(istream& in, Transaction &tr)
{
    cout << "-----" << endl;
    cout << "First Name: ";
    in >> tr.firstName;
    cout << "Last Name: ";
    in >> tr.lastName;
    cout << "Account Number: ";
    in >> tr.accountNumber;
    cout << "Total Balance: ";
    in >> tr.totalBalance;
    cout << "-----" << endl;
    return in;
}

ostream &operator<<(ostream &out, Transaction tr)
{
    cout << "-----" << endl;
```

```

    out << "First Name: " << tr.firstName << endl;
    out << "Last Name: " << tr.lastName << endl;
    out << "Account Number: " << tr.accountNumber << endl;
    out << "Total Balance: " << tr.totalBalance << endl;
    cout << "-----" << endl;
    return out;
}
int main()
{
    int ans;
    do
    {
        cout << "Menu" << endl;
        cout << "1.create record" << endl;
        cout << "2.add record" << endl;
        cout << "3.delete record" << endl;
        cout << "4.edit record" << endl;
        cout << "5.display record" << endl;
        cout << "Enter your choice";

        cin >> ans;
        fstream acc;
        if(ans== 1)
        {
            char ans;
            acc.open("account.dat",ios::out|ios::binary);
            do
            {
                Transaction tr;
                cin >> tr;
                acc.write(reinterpret_cast<char *>(&tr),sizeof(tr));
                if (!acc)
                {
                    cerr << "Couldnot write the data to the file";
                    return 1;
                }
                cout << "Do you want to continue y/n ";
                cin >> ans;
            }
            while(ans=='y');

```

```

    acc.close();
}
else if(ans == 2)
{
    char ans;
    acc.open("account.dat",ios::out|ios::app|ios::binary);
    do
    {
        Transaction tr;
        cin >> tr;
        acc.write(reinterpret_cast<char *>(&tr),sizeof(tr));
        if (!acc)
        {
            cerr << "Couldnot modify the data of the file";
            return 2;
        }
        cout << "Do you want to continue y/n ";
        cin >> ans;
    }
    while(ans=='y');
    acc.close();
}
else if(ans == 3)
{
    char ns;
    Transaction tr;
    acc.open("account.dat",ios::in|ios::binary);
    fstream newacc;
    newacc.open("tmp.dat",ios::out|ios::binary);
    if(!newacc)
    {
        cerr << "couldnot create tmp record file";
    }
    while(!acc.eof())
    {
        acc.read(reinterpret_cast<char *>(&tr),sizeof(tr));
        if (acc)
        {
            cout << tr;
            cout << "Do you want to delete this record y/n ";

```

```

cin >> ns;
if (ns != 'y')
{
    newacc.write(reinterpret_cast<char *>(&tr),sizeof(tr));
    if(!newacc)
    {
        cerr << "unable to write to a temp file";
        return 3;
    }
}
else
{
    cout << "record deleted" << endl;
}
}
}

```

```

acc.close();
newacc.close();
remove("account.dat");
rename("tmp.dat","account.dat");
}
else if(ans == 4)
{
    char ans;
    acc.open("account.dat",ios::in|ios::binary|ios::out);
    while(!acc.eof())
    {
        Transaction tr;
        int pos = acc.tellg();
        acc.read(reinterpret_cast<char *>(&tr),sizeof(tr));
        if (acc)
        {
            cout << tr;
            cout << "Do you want to edit this record y/n";
            cin >> ans;
            if (ans == 'y')
            {
                cin >> tr;
                acc.seekp(pos);
            }
        }
    }
}

```

```

    acc.write(reinterpret_cast<char *>(&tr),sizeof(tr));
    if(acc)
    {
        cout << "Record sucessfully edited"<< endl;
    }
    else
    {
        cerr<< "Unable to modify the record";
        return 4;
    }
}
}
}

acc.close();
}
else if (ans == 5)
{
    acc.open("account.dat",ios::in|ios::binary);
    while(!acc.eof())
    {
        Transaction tr;
        acc.read(reinterpret_cast<char *>(&tr),sizeof(tr));
        if(acc)
        {
            cout << tr;
        }
        else
        {
            cerr << "Couldnot read through the file" << endl;
        }
    }
    acc.close();
}
}
while(ans <= 5 && ans > 0 );
return 0;
}

```

```

#include<iostream>//or

```

```

#include<fstream>
#include<iomanip>
using namespace std;
class transaction
{
    long int account_number;
    char lastname[20],firstname[20];
    float total_balance;
public:
    void getdata()
    {
        cout<<endl<<"Enter the following information:"<<endl;
        cout<<"Account number:\t";
        cin>>account_number;
        cin.ignore();
        cout<<endl<<"First name:\t";
        cin.getline(firstname,20);
        cout<<endl<<"Last name:\t";
        cin.getline(lastname,20);
        cout<<endl<<"Total balance:\t";
        cin>>total_balance;
    }
    void display()
    {
        cout<<setiosflags(ios::left)<<setw(25)<<account_number<<setw(20)<<firstname<<setw(20)<<lastname<<setw(20)<<total_balance<<endl;
    }
};
int main()
{
    transaction tr;
    fstream file;
    int choice;
    while(1){
        cout<<"\n\nEnter your choice:"<<endl;
        cout<<"1) Create record\n";
        cout<<"2) Add record\n";
        cout<<"3) Update record\n";
        cout<<"4) Delete record\n";
        cout<<"5) Display record\n";
    }
}

```

```

cout<<"6) Exit\n";
cout<<"Enter your choice:";
cin>>choice;
int record_pos,location;
char ch;
switch(choice)
{
case 1:{
    file.open("transaction.txt",ios::out|ios::binary);
    do
    {
        tr.getdata();
        file.write(reinterpret_cast<char*>(&tr),sizeof(tr));
        cout<<"Do you want to continue (y/n)?";
        cin>>ch;
    }while(ch!='n');
    file.close();
    break;
}
case 2:{
    file.open("transaction.txt",ios::out|ios::app|ios::binary);
    do
    {
        tr.getdata();
        file.write(reinterpret_cast<char*>(&tr),sizeof(tr));
        cout<<"Do you want to continue (y/n)?";
        cin>>ch;
    }while(ch!='n');
    file.close();
    break;
}
case 3:{
    file.open("transaction.txt",ios::out|ios::in|ios::binary);
    cout<<"Enter record number to be updated:";
    cin>>record_pos;
    location=(record_pos-1)*sizeof(tr);
    tr.getdata();
    file.seekp(location);
    file.write(reinterpret_cast<char*>(&tr),sizeof(tr));
    file.close();
}
}

```

```

    break;
}
case 4:{
    file.open("transaction.txt",ios::in | ios::binary);
    ofstream file_new("temp.txt",ios::out | ios::binary);
    cout<<"Enter record number to be deleted:";
    cin>>record_pos;
    location=(record_pos)*sizeof(tr);
    while(file.read(reinterpret_cast<char*>(&tr),sizeof(tr)))
    {
        if(file.tellg()!=location)
        {
            file_new.write(reinterpret_cast<char*>(&tr),sizeof(tr));
        }
        else
        {
            continue;
        }
    }
    file.close();
    file_new.close();
    remove("transaction.txt");
    rename("temp.txt","transaction.txt");
    break;
}
case 5:{
file.open("transaction.txt",ios::in | ios::binary);
    cout<<endl<<setiosflags(ios::left)<<setfill('*')<<setw(78)<<'*<<endl;
    cout<<setfill(' ')<<setw(25)<<"Account number"<<setw(20)<<"First
name"<<setw(20)<<"Last name"<<setw(20)<<"Total balance"<<endl;
    cout<<setiosflags(ios::left)<<setfill('*')<<setw(78)<<'*<<setfill(' ')<<endl;
    while(file.read(reinterpret_cast<char*>(&tr),sizeof(tr)))
    {
        tr.display();
    }
    file.close();
    break;
default:
    exit(0);
}

```



}  
}  
}