Assignment - 10.

Problem Statement -

Department maintains student information the file contains rollno, name, division and address. Allow user to add, delete, search student details. If record does not exist, print a particular message. Use sequential file for implementing file handling.

Objective -

To study and implement file handling operations.

Outcome -

Studied and implemented file handling operations using sequential file, which can be used for various applications.

S/W & H/W Requirement -

Dell Optiflex 3020 MT, Keyboard, Monitor, Fedora 20, Eclipse.

Theory -

File is a stream of bytes. Size of file is expressed in number of bytes. Sequential file is a type in which, records are added in order of arrival. Length of record is not fixed. Search is time consuming.
Insertion, deletion is time consuming.

Random File -
We can read/write a particular word without having read permission records.
We can position file to a particular location.

Modes -
ios::in - Read from file
ios::out - Write to file
ios::ate - Go to end
ios::afp - Append to end of file
ios::trunc - Truncate file
ios::binary - Treat file as binary

Class structure -

```
class phoneBook {
    char name[10], add[10];
    int div, roll;
    public:
        void getdata();
        void showdata();
        char *getname() { return name; }
        int getroll() { return roll; }
        void update();
};
```

Pseudocodes -
Add a student record -

```
{ phoneBook rec; fstream file;
    rec.getdata();
    cin.get(ch); file.open("data.dat", ios::ate|ios::in|ios::out);
    file.write((char*)& rec, sizeof(rec));
}
```

2. Display all records -

```
file.seekg(0, ios::beg);
cout << "Records -" << endl;
while (file){
    file.read((char *) &rec, sizeof(rec));
    if (!file.eof()){
        rec.showdata();
    }
}
file.clear();
```

3. Search record on basis of roll no. -

```
cout << "Enter roll no." << endl;
cin >> roll;
file.seekg(0, ios::beg);
found = 0;
while(file.read((char *) &rec, sizeof(rec)))
{
    if (roll == rec.getroll()){
        found = 1
        rec.showdata(); break;
    }
}
file.clear();
if (found == 0)
    cout << "Record not found!" << endl;
```

4. Delete a record -

```
fstream f;
fl.open(newname, ios::out);
cout << "Enter roll no. to be deleted" << endl;
cin >> roll;
file.seekg(0, ios::beg);
found = 0;
while (file.read((char*)&rec, sizeof(rec))){
 if (roll == rec.getroll()){
     found = 1;
     continue;
  }
  fl.write((char*)&rec, sizeof(rec));
}
cout << remove(oldname);
fl.close();
cout << rename(newname, oldname);
```

Test cases -

	Case	Expected O/P	Actual O/P
1.	Add student name = a, Roll no. = 1 Address = MH	Added successfully to file.	Success
2.	Search student Roll no = 4	Student not found!	Success
3.	Display student data	name = a roll no. = 1 address = MH	Success
4.	Delete (roll no. = 1)	Data deleted successfully	Success

Conclusion - We have successfully implemented file handling operation using sequential file.