Name: Omkar Dhaigude
Roll no: 223015
Gr no:17U052

ASSIGNMENT NO.8.

Aim :- Department maintains a student information. The file contains roll number, name, division and address. Allow user to add, delete information of student. Display information of particular employee. If record of student does not exist an appropriate message is displayed. If it is, then the system displays the student

Objective:- To study the different data structure concepts to implement this program.

Theory:-

Input/output formatting

Writing to or reading from a file is similar to writing onto a terminal screen or reading from a keyboard. Differences are:

- File must be opened with an OPEN statement, in which the unit number and (optionally) the filename are given
- Subsequent writes (or reads) must refer to a known unit number (used for open)
- File should be closed at the end

File opening and closing

The syntax is:

```
OPEN([unit=]lunit,file='name' [,options])

CLOSE([unit=]lunit [,options])
```

For example:

```
OPEN(10, file='output.dat', status='new')

CLOSE(unit=10)
```

- The first parameter is the unit number and the keyword unit= can be omitted.
- The unit numbers 0,5 and 6 are predefined.
  - 0 is output for standard (system) error messages
  - 5 is for standard (user) input
  - 6 is for standard (user) output
  - These units are opened by default and should not be re-opened nor closed by users

Some options for opening a file:

- status: existence of the file ('old', 'new', 'replace', 'scratch', 'unknown')
- position: offset, where to start writing ('append')
- action: file operation mode ('write','read','readwrite')
- form: text or binary file ('formatted', 'unformatted')
- access: direct or sequential file access ('direct','sequential','stream')
- iostat: error indicator, (output) integer (non zero only upon an error)
- err: the label number to jump upon error
- recl: record length, (input) integer for direct access files only. Be careful, it can be in bytes or words...

Program Code:-

```
#include <iostream>

#include <fstream>

#include <cstring>

#include <iomanip>

#include<cstdlib>
```

```
#define max 50

using namespace std;

class Student

{

   char name[max];

   int rollNo;

   int year;

   int division;

   char address[50];

   friend class FileOperations;

   public:     Student()

               {

                           strcpy(name,"");

                           rollNo=year=division=0;

                           strcpy(address,"");

               }

               Student(char name[max],int rollNo,int year,int division,char
address[max])

               {

                     strcpy(this->address,address);
```

```
				strcpy(this->name,name);

				this->division=division;

				this->rollNo=rollNo;

				this->year=year;

			}

			int getRollNo()

			{

				return rollNo;

			}

			void displayStudentData()

			{


		cout<<endl<<setw(3)<<rollNo<<setw(10)<<name<<setw(3)<<year<<setw(2
)<<division<<setw(10)<<address;

			}

};

class FileOperations

{

		fstream file;

		public:FileOperations(char *name)

			 {
            //strcpy(this->name,name);

            this->file.open(name,ios::in|ios::out|ios::ate|ios::binary);

        }

        void insertRecord(int rollNo,char name[max],int year,int
division,char address[max])

        {

             Student s=Student(name,rollNo,year,division,address);

             file.seekp(0,ios::end);

             file.write((char*)&s,sizeof(Student));

             file.clear();

        }

        void displayAllRecords()

        {

             Student s;

             file.seekg(0,ios::beg);

             while(file.read((char *)&s,sizeof(Student)))

             {

                     s.displayStudentData();

             }

             file.clear();
```

```
}
void displayRecord(int rollNo)
{
        Student s;
        file.seekg(0,ios::beg);
        void *p;
        while(file.read((char *)&s,sizeof(Student)))
        {
                if(s.rollNo==rollNo)
                {
                        s.displayStudentData();
                        break;
                }
        }
        if(p==NULL)
                throw "Element not present";
        file.clear();
}
void deleteRecord(int rollNo)
{
```

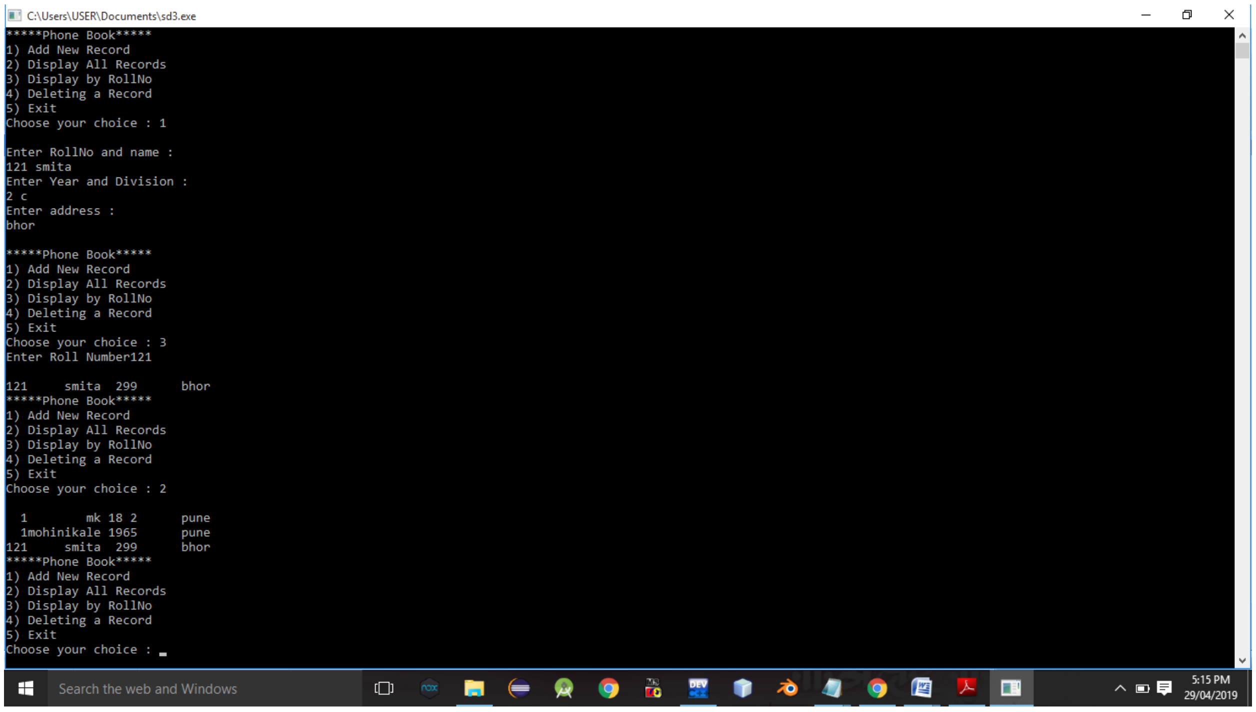
```
ofstream newFile("new.txt",ios::binary);

file.seekg(0,ios::beg);

bool flag=false;

Student s;

while(file.read((char *)&s,sizeof(s)))

{

        if(s.rollNo==rollNo)

        {

                flag=true;

                continue;

        }

        newFile.write((char *)&s,sizeof(s));

}

if(!flag)

{

        cout<<"Element Not Present";

}

file.close();

newFile.close();

remove("student.txt");
```

```
				rename("new.txt","student.txt");

				file.open("student.txt",ios::in|ios::ate|ios::out|ios::binary);

			}

			~FileOperations()

			{

				file.close();

				cout<<"Closing file..";

			}


};

int  main()

{

	ofstream newFile("student.txt",ios::app|ios::binary);

	newFile.close();

	FileOperations file((char *)"student.txt");

   int rollNo,year,choice=0;

   char division;

   char name[max],address[max];

   while(choice!=5)

   {
        //clrscr();

        cout<<"\n*****Phone Book*****\n";

        cout<<"1) Add New Record\n";

        cout<<"2) Display All Records\n";

        cout<<"3) Display by RollNo\n";

        cout<<"4) Deleting a Record\n";

        cout<<"5) Exit\n";

        cout<<"Choose your choice : ";

        cin>>choice;

        switch(choice)

        {

           case 1 : //New Record

                        cout<<endl<<"Enter RollNo and name : \n";

                        cin>>rollNo>>name;

                        cout<<"Enter Year and Division : \n";

                        cin>>year>>division;

                        cout<<"Enter address : \n";

                        cin>>address;

                        file.insertRecord(rollNo,name,year,division,address);

                        break;
```

```
case 2 :

                file.displayAllRecords();

                break;

case 3 :

                cout<<"Enter Roll Number";

                cin>>rollNo;

                try

                {

                        file.displayRecord(rollNo);

                }

                catch(const char *str)

                {

                        cout<<str;

                }

                break;

case 4:

                cout<<"Enter rollNo";

                cin>>rollNo;

                file.deleteRecord(rollNo);

                break;
```

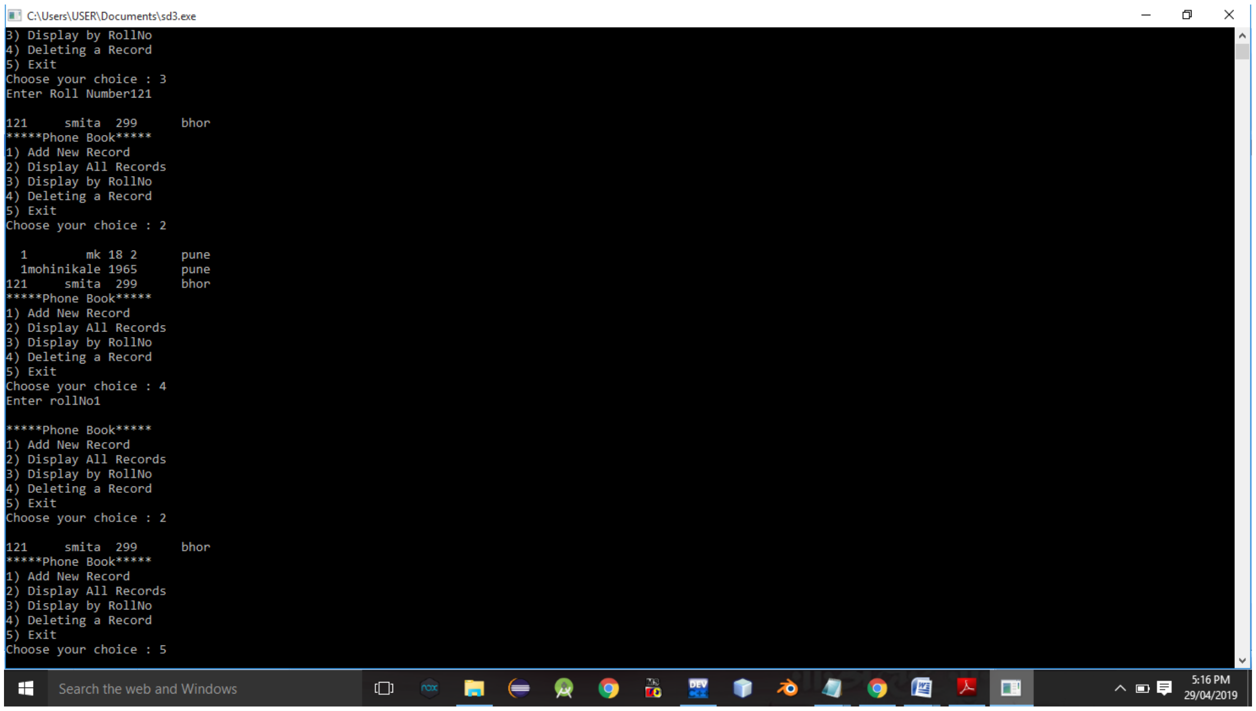
```
          case 5 :break;

        }



    }

}
```

Output Screenshots:-



C:\Users\USER\Documents\sd3.exe

```
*****Phone Book*****
1) Add New Record
2) Display All Records
3) Display by RollNo
4) Deleting a Record
5) Exit
Choose your choice : 1

Enter RollNo and name :
121 smita
Enter Year and Division :
2 c
Enter address :
bhor

*****Phone Book*****
1) Add New Record
2) Display All Records
3) Display by RollNo
4) Deleting a Record
5) Exit
Choose your choice : 3
Enter Roll Number121

121     smita  299      bhor
*****Phone Book*****
1) Add New Record
2) Display All Records
3) Display by RollNo
4) Deleting a Record
5) Exit
Choose your choice : 2

  1        mk 18 2      pune
  1mohinikale 1965      pune
121     smita  299      bhor
*****Phone Book*****
1) Add New Record
2) Display All Records
3) Display by RollNo
4) Deleting a Record
5) Exit
Choose your choice : _
```



```
3) Display by RollNo
4) Deleting a Record
5) Exit
Choose your choice : 3
Enter Roll Number121

121      smita  299      bhor
*****Phone Book*****
1) Add New Record
2) Display All Records
3) Display by RollNo
4) Deleting a Record
5) Exit
Choose your choice : 2

  1        mk 18 2      pune
  1mohinikale 1965      pune
121      smita  299      bhor
*****Phone Book*****
1) Add New Record
2) Display All Records
3) Display by RollNo
4) Deleting a Record
5) Exit
Choose your choice : 4
Enter rollNo1

*****Phone Book*****
1) Add New Record
2) Display All Records
3) Display by RollNo
4) Deleting a Record
5) Exit
Choose your choice : 2

121      smita  299      bhor
*****Phone Book*****
1) Add New Record
2) Display All Records
3) Display by RollNo
4) Deleting a Record
5) Exit
Choose your choice : 5
```

Conclusion:-_Thus,this assignment implemented successfully.