```
#include <iomanip.h>
#include <iostream.h>
#include <fstream.h>
#include <conio.h>
#include <string.h>
struct student
{
int rollno;
char name[20];
float marks;
int status;
};
class sequential
{ char master1[30];
fstream mas;
public:
sequential(char *a)
{
strcpy(master1,a);
mas.open(master1,ios::binary|ios::in);
if(mas.fail())
mas.open(master1,ios::binary | ios::out);
mas.close();
}
void read(); //display master file void insert(student rec1);
int Delete(int rollno);
int search(int rollno);
void pack();
void update();
void display(int recno) // display a particular record
{ student rec1;
mas.open(master1,ios::binary | ios::in | ios::nocreate);
mas.seekg(recno*sizeof(student),ios::beg);
mas.read((char*) &rec1,sizeof(student));
cout<<“\n”<<rec1.rollno<<“”<<rec1.name<<“”<<setprecision(2)<<rec1.mark
s;
mas.close();
}
```

```
};
void main()
{ sequential object("master.txt");
int rollno,op,recno;
student rec1;
do
{ cout<<"\n\n1)Read(Print)\n2)Insert\n3)Delete\n4)Update";
cout<<"\n5)Search\n6)Pack\n7)Quit";
cout<<"\nEnter Your Choice:";
cin>>op;
switch(op)
{ case 1: object.read();break; case 2: cout<<"\nEnter a record to be
inserted(roll no,name,marks : ";
cin>>rec1.rollno>>rec1.name>>rec1.marks;
object.insert(rec1);
break;
case 3: cout<<"\nEnter the roll no.:";
cin>>rollno;
object.Delete(rollno);
break;
case 4: object.update(); break;
case 5: cout<<"\nEnter a roll no. : ";
cin>>rollno;
recno=object.search(rollno);
if(recno>=0)
{ cout<<"\n Record No.: "<< recno;
object.display(recno);
}
else
cout<<"\nRecord Not Found ";
break;
case 6: object.pack();break;
}
}while(op!=7);
}
void sequential::read()
{ student crec; int i=1,n;
cout<<"\n*********Data File*********\n";
mas.open(master1,ios::binary | ios::in | ios::nocreate);
```

```
mas.seekg(0,ios::end);/*go to the end of file */
n=mas.tellg()/sizeof(student);
mas.seekg(0,ios::beg);
for(i=1;i<=n;i++)
{ mas.read((char*)&crec,sizeof(student));
if(crec.status==0)
cout<<“\n”<<i<<“)
“<<crec.rollno<<“”<<crec.name<<“”<<setprecision(2)<<crec.marks;
else
cout<<“\n”<<i<<“) “<<” ****** deleted *********”;
}
mas.close();
}
void sequential::insert(student rec1)
{ student crec;
int n,i,k;
mas.open(master1,ios::in | ios::out | ios::nocreate);
rec1.status=0;
mas.seekg(0,ios::end);/*go to the end of file */
n=mas.tellg()/sizeof(student);
if(n==0)
{
mas.write((char*)&rec1,sizeof(student)); mas.close();
return;
}
/* Shift records until the point of insertion */
i=nÂ1;
while(i>=0)
{ mas.seekg(i*sizeof(student),ios::beg);
mas.read((char*)&crec,sizeof(student));
if(crec.rollno>rec1.rollno)
{ mas.seekp((i+1)*sizeof(student),ios::beg);
mas.write((char*)&crec,sizeof(student));
}
else
break;
iÂÂ;
}
/*insert the record at (i+1)th position */
```

```
i++;
mas.seekp(i*sizeof(student),ios::beg);
mas.write((char*)&rec1,sizeof(student));
mas.close();
}
int sequential::Delete(int rollno)
{ student crec;
int i,n; mas.open(master1,ios::in | ios::out | ios::nocreate);
mas.seekg(0,ios::end);/*go to the end of file */
n=mas.tellg()/sizeof(student);
mas.seekg(0,ios::beg);
for(i=0;i<n;i++)
{ mas.read((char*)&crec,sizeof(student));
if(crec.status==0)
{
if(crec.rollno>rollno)
{cout<<"\nRecord does not exist ...";
mas.close();
return(0);
}
if(crec.rollno==rollno)
{crec.status=1;
mas.seekp(i*sizeof(student),ios::beg);
mas.write((char*)&crec,sizeof(student));
mas.close();
return(1);
}
}
}
return(0);
}
int sequential::search(int rollno){ student crec;
int i,n;
mas.open(master1,ios::in | ios::out | ios::nocreate);
mas.seekg(0,ios::end);/*go to the end of file */
n=mas.tellg()/sizeof(student);
mas.seekg(0,ios::beg);
for(i=0;i<n;i++)
{ mas.read((char*)&crec,sizeof(student));
```

```
if(crec.status==0)
{
if(crec.rollno>rollno)
{mas.close();
return(Â1);
}
if(crec.rollno==rollno)
{ mas.close();
return(i);
}
}
}
return(Â1);
}
void sequential::pack()
{ fstream temp;
student crec;int i,n;
mas.open(master1,ios::binary | ios::in);
temp.open("temp.txt",ios::out | ios::trunc | ios::binary);
mas.seekg(0,ios::end);/*go to the end of file */
n=mas.tellg()/sizeof(student);
mas.seekg(0,ios::beg);
for(i=0;i<n;i++)
{ mas.read((char*)&crec,sizeof(student));
if(crec.status==0)
temp.write((char*)&crec,sizeof(student));
}
mas.close();
temp.close();
temp.open("temp.txt",ios::binary | ios::in);
mas.open(master1,ios::binary | ios::out | ios::trunc);
temp.seekg(0,ios::end);/*go to the end of file */
n=temp.tellg()/sizeof(student);
temp.seekg(0,ios::beg);
for(i=0;i<n;i++)
{ temp.read((char*)&crec,sizeof(student));
mas.write((char*)&crec,sizeof(student));
}
mas.close();
```

```
temp.close();
}void sequential::update()
{ int rollno;
student rec1;
cout<<“\n Enter the rollno of the record to be updated : “;
cin>>rollno;
cout<<“\nEnter a new record(roll no. name marks : “;
cin>>rec1.rollno>>rec1.name>>rec1.marks;
if(Delete(rollno))
insert(rec1);
else
cout<<“\n Record not found :”;
}
```