3. Write a C++ program to read and write and student objects with variable-length records using any suitable record structure. Implement pack(),unpack(),modify() and search() methods

Variable length record

A record which can differ in length from the other records of the file.

- **delimited record**

  A variable length record which is terminated by a special character or sequence of characters.

- **delimiter**

  A special character or group of characters stored after a field or record, which indicates the end of the preceding unit.

  - The records within a file are followed by a delimiting byte or series of bytes.
  - The delimiter cannot occur within the records.
  - Records within a file can have different sizes.
  - Different files can have different length records.
  - Programs which access the file must know the delimiter.
  - Offset, or position, of the nth record of a file cannot be calculated.
  - There is external overhead for record separation equal to the size of the delimiter per record.
  - There should be no internal fragmentation (unused space within records.)
  - There may be no external fragmentation (unused space outside of records) after file updating.
  - Individual records cannot always be updated in place.

File_structure3.cpp

```
#include<iostream.h>
#include<fstream.h>
#include<process.h>
#include<string.h>
#include<conio.h>
class student
{
      private: char buf[45],name[10],sem[10],branch[10];
      int pos;
      public:
      void read()
      {
            cout<<"name:"<<endl;
            cin>>name;
            cout<<"semester:"<<endl;
            cin>>sem;
            cout<<"branch:"<<endl;
            cin>>branch;
      }

      void pack(fstream &ofile)
       {
            read();
            strcpy(buf,"");
            strcat(buf,name);
            strcat(buf,"|");
            strcat(buf,sem);
            strcat(buf,"|");
            strcat(buf,branch);
            strcat(buf,"|");
            strcat(buf,"\n");
            ofile.write(buf,strlen(buf));
      }

      void unpack(fstream &ifile)
      {
            char extra[45];
            while(!ifile.eof())
            {
                  ifile.getline(name,10,'|');
                  ifile.getline(sem,10,'|');
                  ifile.getline(branch,10,'|');
                  ifile.getline(extra,45,'\n');
                  cout<<name<<"\t";
                  cout<<sem<<"\t";
                  cout<<branch<<"\n";
            }
      }
```

```
int search(fstream &ifile,char key[])
{
      char extra[45];
      while(!ifile.eof())
      {
            ifile.getline(name,10,'|');
            ifile.getline(sem,10,'|');
            ifile.getline(branch,10,'|');
            ifile.getline(extra,45,'\n');

             if(strcmp(name,key)==0)
             {
                   cout<<" "<<"record found and details
                   are:"<<endl; cout<<" "<<"name"<<name<<endl;
                  cout<<" "<<"semester"<<sem<<endl;
                   cout<<" "<<"branch"<<branch<<endl;
                   return 1;
             }

      }
       return 0;
}

void modify(fstream &ifile,char key[])
{
      student s[10];
      char extra[50];
      int i=0;

       while(!ifile.eof())
       {
            ifile.getline(s[i].name,10,'|');
            ifile.getline(s[i].sem,10,'|');
            ifile.getline(s[i].branch,10,'|');
            ifile.getline(extra,45,'\n');
            i++;
       }
      ifile.close();
      int flag=0;
      for(int j=0;j<i;j++)
      {
            if(strcmp(key,s[j].name)==0)
            {
                  flag=1;
                  cout<<"record found details are:"<<endl;
                  cout<<s[j].name<<endl;
                  cout<<s[j].sem<<endl;
                  cout<<s[j].branch<<endl;
```

```
                        cout<<"enter the modification
                        details"<<endl; cout<<"enetr the
                        name"<<endl; cin>>s[j].name;
                        cout<<"enter the sem;"<<endl;
                        cin>>s[j].sem;
                        cout<<"enter the branch"<<endl;
                        cin>>s[j].branch;
                  }
            }
            if(flag==0)
            {
                  cout<<"Record not found\n";
                  return;
            }
            ifile.open("student.txt",ios::trunc|ios::app);
            for(int k=0;k<i;k++)
            {
                  strcpy(buf,"");
                  strcat(buf,s[k].name);
                  strcat(buf,"|");
                  strcat(buf,s[k].sem);
                  strcat(buf,"|");
                  strcat(buf,s[k].branch);
                  strcat(buf,"|");
                  strcat(buf,"\n");
                  ifile.write(buf,strlen(buf));
            }
      }
};
void main()
{
      int n,i,ch;
      char key[10];
      student stu;
      fstream ifile,ofile;
      ofile.open("student.txt",ios::trunc|ios::app);
      ofile.close();
      for(;;)
      {
            clrscr();
            cout<<"1.insert\n 2.display\n 3.search\n 4.modify\n
            5.exit\n"; cout<<"enter your choice"<<endl; cin>>ch;

            switch(ch)
            {
                  case 1: fstream ofile;
                        ofile.open("student.txt",ios::out|ios::app);
                        cout<<"enter the no of students";
                        cin>>n;
                        for(i=0;i<n;i++)
                        {
                              stu.pack(ofile);
                        }
                        ofile.close();
                        break;

                  case 2: fstream infile;
                         infile.open("student.txt",ios::in);
                         stu.unpack(infile);
                         getch();
                         infile.close();
                         break;

                  case 3:cout<<"enter the record name to be
                         searched"<<endl; cin>>key;
                         fstream ifile;
                         ifile.open("student.txt",ios::in);
                         if(stu.search(ifile,key)==0)
                         cout<<"record not found\n";
                         getch();
                         ifile.close();
                         break;

                  case 4: fstream iofile;
                         iofile.open("student.txt",ios::in|ios::out);
                         cout<<"enter the record name to be modified\n"<<endl;
                         cin>>key;
                         stu.modify(iofile,key);
                         getch();
                         iofile.close();
                         break;
                         default:exit(0);
                   }
             }
        }
```

Output:

```
1:write to file 2:display the file 3:modify the file 4:search
5.exit Enter the choice:1
Enter the number of students:2
Enter the student name = ajay
Enter the sem = 6
Enter the branch = ise

Enter the student name = rahul
Enter the sem = 6
Enter the branch = cse

1:write to file 2:display the file 3:modify the file 4:search 5.exit
```

```
Enter the choice:2
Name              Sem         Branch
ajay              6           ise
rahul             6           cse

1:write to file 2:display the file 3:modify the file 4:search 5.exit
Enter the choice:4
Enter the record name you want to search = rahul
Record found
rahul        6          cse

1:write to file 2:display the file 3:modify the file 4:search 5.exit
Enter the choice:3
Enter the record name you want to modify:rahul
record found and details are:
rahul        6          cse

enter modification details
Enter the student name =navya
Enter the sem = 6
Enter the branch = ise
1:write to file 2:display the file 3:modify the file 4:search 5.exit
Enter the choice:2
Name              Sem         Branch
ajay              6           ise
Navya             6           ise

1:write to file 2:display the file 3:modify the file 4:search 5.exit

Enter the choice:4
Enter the record name you want to search:keerthi
Record not found
```