3. /* Write a program to read and write student objects with variable -Length records using any suitable record structures. Implemet pack (), unpack (), modify () and search () methods. */

```
#include<iostream>
#include<fstream>
#include<string.h>
using namespace std;

#define filename "std3.txt"
fstream ifile;

class student {
    char usn[15], name[20], age[5], branch[6], sem[5];

public:
    void opener(fstream& ifile, char* fn, ios_base::openmode mode);
    void read();
    void pack();
    void display();
    void unpack();
    int search();
    void modify(int r);
};

void student::opener(fstream& sfile, char* fn, ios_base::openmode mode) {
    sfile.open(fn, mode);
    if (!sfile) {
        cout << "Unable to open the file" << endl;
        exit(1);
    }
}

void student::read() {
    cout << "Enter the USN number: ";
    cin >> usn;
    cout << "Enter the name: ";
    cin >> name;
    cout << "Enter the age: ";
    cin >> age;
    cout << "Enter the branch: ";
    cin >> branch;
    cout << "Enter the semester: ";
    cin >> sem;
    pack();
}

void student::pack() {
    char buffer[75];
    strcpy(buffer, usn);
    strcat(buffer, "|");
    strcat(buffer, name);
    strcat(buffer, "|");
    strcat(buffer, age);
    strcat(buffer, "|");
    strcat(buffer, branch);
    strcat(buffer, "|");
    strcat(buffer, sem);
    strcat(buffer, "|");
    ifile << buffer << "#";
}

void student::display() {
    char buffer[75];
    cout << left;
    cout  << "usn"  << "name"  << "age";
    cout  << "branch"  << "sem" << endl;
    while (1) {
        unpack();
        if (ifile.eof())
            break;
        if (usn[0] != '$') {
            cout  << usn  << name  << age;
            cout  << branch  << sem << endl;
        }
    }
}

void student::unpack() {
    char dummy[75];
    ifile.getline(usn, 15, '|');
    ifile.getline(name, 20, '|');
    ifile.getline(age, 5, '|');
    ifile.getline(branch, 6, '|');
    ifile.getline(sem, 5, '|');
    ifile.getline(dummy, 10, '#');
}

int student::search() {
    int flag;
    char k[15];
    cout << "Enter the USN to be searched: ";
    cin >> k;
    while (!ifile.eof()) {
        flag = ifile.tellg();
        unpack();
        if (usn[0] != '$' && strcmp(usn, k) == 0) {
            cout << "USN: " << usn << "\nName: " << name << "\nAge: " << age;
            cout << "\nBranch: " << branch << "\nSemester: " << sem << "\n";
            return flag;
        }
    }
    return -1;
```

```
}

void student::modify(int r) {
    ifile.seekp(r, ios::beg);
    ifile.put('$');
    ifile.seekp(0, ios::end);
    read();
}

int main() {
    int ch, flag;
    student s;
    for (;;) {
        cout << endl << "1. Read\t2. Display\t3. Search\t4. Modify\t5. Exit" << endl;
        cout << "Enter your choice: ";
        cin >> ch;
        switch (ch) {
            case 1:
                s.opener(ifile, filename, ios::app);
                cout << "Enter the student details:" << endl;
                s.read();
                break;
            case 2:
                s.opener(ifile, filename, ios::in);
                cout << "The student details are:" << endl;
                s.display();
                break;
            case 3:
                s.opener(ifile, filename, ios::in);
                cout << "Searching based on USN number" << endl;
                flag = s.search();
                if (flag == -1)
                    cout << "Record not found" << endl;
                break;
            case 4:
                s.opener(ifile, filename, ios::in | ios::out);
                cout << "To modify the record based on USN" << endl;
                flag = s.search();
                if (flag == -1)
                    cout << "Record not found" << endl;
                else
                    s.modify(flag);
                break;
            default:
                exit(0);
        }
        ifile.close();
    }
    return 0;
}
```