

```

#include <iostream>
#include <string>
#include <fstream>
using namespace std;
class record{
public:
    int rollno, marks;
    char name[20];
    friend class student;};
class hashing{
public:
    int rollno, pos;
    hashing(){
        rollno = -1;
        pos = -1;}};
class student{
public:
    record rec;
    hashing h[10];
    int s, relt;
    student(){relt=0;}
    void create_wr();
    void create_wor();
    void modify();
    void display();
    void search();
    void display_all();};
void student::create_wr(){
    char ans;
    int nloc;
    hashing temp;
    fstream f;
    f.open("student.txt", ios::binary | ios::app);
    int loc;
    do{
        cout << "Enter details: \n";
        cout << "Enter roll number: ";
        cin >> rec.rollno;
        cout << "Enter name: ";
        cin >> rec.name;
        cout << "Enter marks: ";
        cin >> rec.marks;
        loc = rec.rollno % 10;
        if (h[loc].rollno == -1){
            h[loc].rollno = rec.rollno;
            h[loc].pos = relt ;
            f.write((char *)&rec, sizeof(rec));
            relt++;}
        else{
            temp.rollno = rec.rollno;
            temp.pos = relt;
            f.write((char *)&rec, sizeof(rec));
            relt++;
            if (loc != h[loc].rollno % 10){
                temp.rollno = h[loc].rollno;
                temp.pos = h[loc].pos;
                h[loc].rollno = rec.rollno;
                h[loc].pos = relt - 1;}
            nloc = (loc + 1) % 10;
            while (nloc != loc) {
                if (h[nloc].rollno == -1) {
                    h[nloc].rollno = temp.rollno;
                    h[nloc].pos = temp.pos;
                    break; }
                nloc = (nloc + 1) % 10;}}
        display();
        cout << "Do you want to add more records? (y/n): ";
        cin >> ans;
    }
}

```

```

    } while (ans == 'y' || ans == 'Y');
    f.close();}

void student::create_wor(){
    char ans;
    fstream f;
    f.open("student.txt", ios::binary | ios::app);
    int loc;do{
        cout << "\nEnter the details: \n";
        cout << "Roll no.: ";
        cin >> rec.rollno;
        cout << "Name: ";
        cin >> rec.name;
        cout << "Marks: ";
        cin >> rec.marks;
        loc = rec.rollno % 10;
    if (h[loc].rollno == -1){
        h[loc].rollno = rec.rollno;
        h[loc].pos = relt;
        f.write((char *)&rec, s);
        relt++;}
    else{
        int nloc = (loc + 1) % 10;
        while (nloc != loc){
            if (h[nloc].rollno == -1){
                h[nloc].rollno = rec.rollno;
                h[nloc].pos = relt;
                f.write((char *)&rec, s);
                relt++;
                break;}
            nloc = (nloc + 1) % 10;}}
        display();
        cout << "\nDo you want to add next student record? (y/n): ";
        cin >> ans;
    } while (ans == 'y' || ans == 'Y');
    f.close();}

void student::search(){
    int roll_no;
    cout << "Enter roll number to search: ";
    cin >> roll_no;
    int loc = roll_no % 10;
    bool found = false;
    if (h[loc].rollno == roll_no){
        ifstream f("student.txt", ios::binary);
        f.seekg(h[loc].pos * sizeof(record));
        f.read((char *)&rec, sizeof(rec));
        cout << "Record found:" << endl;
        cout << "Roll No: " << rec.rollno << endl;
        cout << "Name: " << rec.name << endl;
        cout << "Marks: " << rec.marks << endl;
        f.close();
        found = true;}
    else{
        int nloc = (loc + 1) % 10;
        while (nloc != loc){
            if (h[nloc].rollno == roll_no){
                ifstream f("student.txt", ios::binary);
                f.seekg(h[nloc].pos * sizeof(record));
                f.read((char *)&rec, sizeof(rec));
                cout << "Record found:" << endl;
                cout << "Roll No: " << rec.rollno << endl;
                cout << "Name: " << rec.name << endl;
                cout << "Marks: " << rec.marks << endl;
                f.close();
                found = true;
                break;}
            nloc = (nloc + 1) % 10;}}
    if (!found)
    {

```

```

        cout << "Record not found for roll number " << roll_no << endl;}}
void student::modify(){
    int roll_no;
    cout << "Enter roll number to modify: ";
    cin >> roll_no;
    int loc = roll_no % 10;
    bool found = false;
    if (h[loc].rollno == roll_no){
        fstream f("student.txt", ios::binary | ios::in | ios::out);
        f.seekg(h[loc].pos * sizeof(record));
        f.read((char *)&rec, sizeof(rec));
        cout << "Existing Record:" << endl;
        cout << "Roll No: " << rec.rollno << endl;
        cout << "Name: " << rec.name << endl;
        cout << "Marks: " << rec.marks << endl;
        cout << "\nEnter new details: \n";
        cout << "Enter name: ";
        cin >> rec.name;
        cout << "Enter marks: ";
        cin >> rec.marks;
        found = true;
        f.seekp(h[loc].pos * sizeof(record));
        f.write((char *)&rec, sizeof(rec));
        f.close();}
    else{
        int nloc = (loc + 1) % 10;
        while (nloc != loc){
            if (h[nloc].rollno == roll_no){
                fstream f("student.txt", ios::binary | ios::in | ios::out);
                f.seekg(h[nloc].pos * sizeof(record));
                f.read((char *)&rec, sizeof(rec));
                cout << "Existing Record:" << endl;
                cout << "Roll No: " << rec.rollno << endl;
                cout << "Name: " << rec.name << endl;
                cout << "Marks: " << rec.marks << endl;
                cout << "\nEnter new details: \n";
                cout << "Enter name: ";
                cin >> rec.name;
                cout << "Enter marks: ";
                cin >> rec.marks;
                found = true;
                f.seekp(h[nloc].pos * sizeof(record));
                f.write((char *)&rec, sizeof(rec));
                f.close();
                break;}
            nloc = (nloc + 1) % 10; } }
    if (!found){
        cout << "Record not found for roll number " << roll_no << endl; }
void student::display(){
    cout << "Hash Table:" << endl;
    cout << "-----" << endl;
    cout << "Index\tRoll No\tPosition" << endl;
    for (int i = 0; i < 10; i++){
        cout << i << "\t" << h[i].rollno << "\t" << h[i].pos << "\t" << endl;
        cout << "-----" << endl; }
void student :: display_all(){
    ifstream file("student.txt", ios::binary);
    record r;
    while (file.read((char *)&r, sizeof(r))){
        cout<<r.rollno<<"\t"<<r.name<<"\t"<<r.marks<<endl;}
    file.close();
    cout << "Hash Table:" << endl;
    cout << "-----" << endl;
    cout << "Index\tRoll No\tPosition" << endl;
    for (int i = 0; i < 10; i++){
        cout << i << "\t" << h[i].rollno << "\t" << h[i].pos << "\t" << endl;
        cout << "-----" << endl; }

```

```
int main(){
    student s;
    int choice;
    while (1){
        cout << "\n1. Create record (with replacement)\n2. Create record (without
replacement)\n3. Display record\n4. Search\n5. Modify\n6. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;
        switch (choice){
            case 1:
                s.create_wr();
                break;
            case 2:
                s.create_wor();
                break;
            case 3:
                s.display_all();
                break;
            case 4:
                s.search();
                break;
            case 5:
                s.modify();
                break;
            case 6:
                exit(0);
            default:
                cout << "Invalid choice! Please try again." << endl;}}}
```