```
/*
-Company maintains employee information as employee ID, name, designation and salary. Allow user to add, delete information of employee. Display information of particular employee. If employee does not exist an appropriate message is displayed. If it is, then the system displays the employee details. Use index sequential file to maintain the data
*/

#include <iostream>
#include <fstream>
#include <cstring>

using namespace std;

const int MAX_NAME_LENGTH = 50;
const int MAX_DESIGNATION_LENGTH = 50;

struct Employee {
   int employeeID;
   char name[MAX_NAME_LENGTH];
   char designation[MAX_DESIGNATION_LENGTH];
   double salary;
};

void addEmployee(fstream& file) {
   Employee employee;

   cout << "Enter employee ID: ";
   cin >> employee.employeeID;

   cout << "Enter employee name: ";
   cin.ignore();
   cin.getline(employee.name, MAX_NAME_LENGTH);
```

```
    cout << "Enter employee designation: ";
    cin.getline(employee.designation,
MAX_DESIGNATION_LENGTH);

    cout << "Enter employee salary: ";
    cin >> employee.salary;

    file.write(reinterpret_cast<char*>(&employee),
sizeof(Employee));

    cout << "Employee added successfully!" << endl;
}

void deleteEmployee(fstream& file) {
    int employeeID;
    bool found = false;
    Employee employee;

    cout << "Enter employee ID to delete: ";
    cin >> employeeID;

    fstream tempfile("temp.dat", ios::out | ios::binary);

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

    while (file.read(reinterpret_cast<char*>(&employee),
sizeof(Employee))) {
        if (employee.employeeID != employeeID) {
            tempfile.write(reinterpret_cast<char*>(&employee),
sizeof(Employee));
        } else {
            found = true;
        }
    }
```

```
    file.close();
    tempfile.close();

    remove("employee_data.dat");
    rename("temp.dat", "employee_data.dat");

    if (found) {
        cout << "Employee deleted successfully!" << endl;
    } else {
        cout << "Employee not found!" << endl;
    }

    file.open("employee_data.dat", ios::in | ios::out |
ios::binary);
}

void displayEmployee(fstream& file) {
    int employeeID;
    bool found = false;
    Employee employee;

    cout << "Enter employee ID to display: ";
    cin >> employeeID;

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

    while (file.read(reinterpret_cast<char*>(&employee),
sizeof(Employee))) {
        if (employee.employeeID == employeeID) {
            found = true;
            break;
        }
    }

    if (found) {
```

```
        cout << "Employee ID: " << employee.employeeID << endl;
        cout << "Name: " << employee.name << endl;
        cout << "Designation: " << employee.designation << endl;
        cout << "Salary: " << employee.salary << endl;
    } else {
        cout << "Employee not found!" << endl;
    }
}

int main() {
    fstream file("employee_data.dat", ios::in | ios::out |
ios::binary);

    if (!file) {
        cout << "Error opening file!" << endl;
        return 1;
    }

    while (true) {
        cout << "\nEmployee Information System" << endl;
        cout << "1. Add Employee" << endl;
        cout << "2. Delete Employee" << endl;
        cout << "3. Display Employee" << endl;
        cout << "4. Quit" << endl;
        cout << "Enter your choice: ";

        int choice;
        cin >> choice;

        switch (choice) {
            case 1:
                addEmployee(file);
                break;
            case 2:
                deleteEmployee(file);
```

```
            break;
        case 3:
            displayEmployee(file);
            break;
        case 4:
            file.close();
            return 0;
        default:
            cout << "Invalid choice! Please try again." << endl;
    }
  }
}
```