```cpp
#include <bits/stdc++.h>
#define cin fin
using namespace std;
class Node
{
public:
    int number, gender, s1, s2, s3, score;
    double ave_score;
    string name;
    Node *next;
    Node(int _number = 0, string _name = "", int _gender = 0,
         int _s1 = 0, int _s2 = 0, int _s3 = 0)
        : number(_number), name(_name), gender(_gender), s1(_s1), s2(_s2), s3(_s3)
    {
        score = s1 + s2 + s3, ave_score = score / 3.0, next = NULL;
    }
    friend ostream &operator<<(ostream &out, const Node &t)
    {
        out << t.number << " " << t.name << " " << (t.gender ? "男" : "女")
            << " " << t.s1 << " " << t.s2 << " " << t.s3 << " 总分=" << t.score
            << " 平均成绩=" << t.ave_score << endl;
        return out;
    }
};
class List
{
private:
    Node *begin;
    int n;

public:
    List() { begin = new Node, n = 0; }
    // 1.添加学生信息，包括学号、姓名、性别、3门课成绩、总成绩、平均成绩
    void append(Node x)
    {
        Node *t = new Node{x};
        t->next = begin->next;
        begin->next = t;
        n++;
    }
    // 2.显示学生信息，将所有学生信息打印输出
    void printAll()
    {
        for (Node *p = begin->next; p; p = p->next)
            cout << *p;
    }
    // 3.查找学生信息，根据学生姓名，将其信息打印输出
    void print(string name)
    {
        for (Node *p = begin->next; p; p = p->next)
```

```
            if (p->name == name)
                cout << *p;
    }
    // 4.修改学生信息，可以根据学号查找到学生，然后可以修改学生成绩项
    void changeScore(int number, int s1, int s2, int s3)
    {
        for (Node *p = begin->next; p; p = p->next)
            if (p->number == number)
            {
                p->s1 = s1, p->s2 = s2, p->s3 = s3;
                p->score = s1 + s2 + s3;
                p->ave_score = (s1 + s2 + s3) / 3.0;
            }
    }
    // 5.删除学生信息，根据学号查找到学生，将其信息删除
    void erase(int number)
    {
        for (Node *p = begin, *t; p->next; p = p->next)
            if (p->next->number == number)
            {
                t = p->next;
                p->next = p->next->next;
                n--;
                delete t;
                break;
            }
    }
    // 6.按学生总成绩进行从高到低排序
    void Sort()
    {
        // 把链表各结点指针保存到指针数组a[]中
        Node *a[n];
        int i = 0;
        for (Node *p = begin->next; p; p = p->next, i++)
            a[i] = p;
        // 对a[]根据指针所指向结点的成绩大小进行排序
        sort(a, a + n, [](const Node *a, const Node *b)
             { return a->score > b->score; });
        Node *p = begin;
        // 重新按照指针数组各指针顺序调整链表顺序
        for (int i = 0; i < n; p = p->next, i++)
            p->next = a[i];
        p->next = NULL;
    }
    // 7.按科目输出不及格的学生信息
    void print()
    {
        for (Node *p = begin->next; p; p = p->next)
            if (p->s1 < 60 || p->s2 < 60 || p->s3 < 60)
                cout << *p;
    }
    int size() { return n; }
};
int main()
{
    ifstream fin("student.txt");
    int n;
    cin >> n; // #define cin fin 从文件中读入
    List A;
    Node x;
    while (n--)
    {
        cin >> x.number >> x.name >> x.gender >> x.s1 >> x.s2 >> x.s3;
        x.score = x.s1 + x.s2 + x.s3;
        x.ave_score = x.score / 3.0;
        A.append(x);
    }
    A.Sort();
    A.printAll();
}
```