

```
int main() {
    Welcome();
    return 0;
}
```

欢迎界面——void Welcome () 函数，无参数，功能为欢迎用户

```
int Start() {
    int i;
    for(i=1; i<=10; i++) {
        printf("请输入第%d个学生的学号\n", i);
        scanf("%d", &stu[i].id);
        stu[i].next = NULL;
    }
    return 0;
}
```

初始化——int Start () 函数，无参数，功能为选择对全部毕设信息或者抽查毕设信息建立原始链表

```
void Read() {
    FILE *p;
    Student *p;
    int i;
    for(i=1; i<=10; i++) {
        p = (Student *)malloc(sizeof(Student));
        p->id = i;
        p->name = " ";
        p->grade = " ";
        p->score = " ";
        p->teacher = " ";
        p->status = " ";
        p->next = NULL;
        stu[i] = *p;
    }
}
```

void Read()函数，参数为（FILE \*p一个指向打开文件的指针，STUDENT \*p一个待存入数据的STUDENT结构体指针，int number学生学号）函数功能为从文件中读取数据，存入原始链表

这个函数会被多次调用

```
void Save() {
    FILE *p;
    Student *p;
    int i;
    for(i=1; i<=10; i++) {
        p = (Student *)malloc(sizeof(Student));
        p->id = i;
        p->name = " ";
        p->grade = " ";
        p->score = " ";
        p->teacher = " ";
        p->status = " ";
        p->next = NULL;
        stu[i] = *p;
    }
}
```

void save()函数，无参数，功能：将stu[]存下的数据分类标题写入文件第一行，将原始链表的每一节按顺序存入文件的一行，最后无空行

```
void Operate() {
    int i;
    for(i=1; i<=10; i++) {
        printf("请输入第%d个学生的学号\n", i);
        scanf("%d", &stu[i].id);
        stu[i].next = NULL;
    }
}
```

进入操作界面——void Operate()无参数，功能为当用户输入数字1-9时跳转至对应界面

```
void View() {
    int i;
    for(i=1; i<=10; i++) {
        printf("请输入第%d个学生的学号\n", i);
        scanf("%d", &stu[i].id);
        stu[i].next = NULL;
    }
}
```

查看所有毕设信息——void View () 无参数，功能为查看当前链表的全部毕设信息

```
void Print() {
    Student *p;
    p = stu[1];
    while(p != NULL) {
        printf("%d\t", p->id);
        printf("%s\t", p->name);
        printf("%s\t", p->grade);
        printf("%s\t", p->score);
        printf("%s\t", p->teacher);
        printf("%s\t", p->status);
        printf("%s\t", p->next);
        p = p->next;
    }
}
```

void Print () 函数，参数为（STUDENT \* p），功能为打印一个学生的所有毕设信息

```
Student *traverse(int ID) {
    Student *p;
    p = stu[1];
    while(p != NULL) {
        if(p->id == ID) {
            return p;
        }
        p = p->next;
    }
    return NULL;
}
```

STUDENT \*traverse (i) 函数，参数为（int ID 学生学号），功能为输入ID，返回链表中对应毕设信息的指针，没有的返回NULL

增加——void Add () 无参数，功能：使用者逐项输入一个学生毕设的各项信息，最后将其添加至链表尾以及保存至文件末尾

void Append () 函数，无参数，功能为将添加的信息保存至文件中

这个函数会被多次调用

```
int c_or_r() {
    int i;
    for(i=1; i<=10; i++) {
        printf("请输入第%d个学生的学号\n", i);
        scanf("%d", &stu[i].id);
        stu[i].next = NULL;
    }
}
```

int c\_or\_r () 函数，无参数，功能为引导用户输入1重新进入当前界面，输入2得到operate界面，输入其他数字无效

```
Student *confirm_ID() {
    Student *p;
    p = stu[1];
    while(p != NULL) {
        if(p->id == ID) {
            return p;
        }
        p = p->next;
    }
    return NULL;
}
```

STUDENT \*confirm\_ID () 函数，无参数，用户输入ID并确认后若ID存在链表中则返回对应指针；若不存在则返回NULL

traverse () 函数

\_Print () 函数

修改——void Modify () 无参数，功能为修改链表和文件中用户输入ID对应的各类信息

save () 函数

c\_or\_r () 函数

```
void Strikeout() {
    int i;
    for(i=1; i<=10; i++) {
        printf("请输入第%d个学生的学号\n", i);
        scanf("%d", &stu[i].id);
        stu[i].next = NULL;
    }
}
```

删除——void Strikeout () 无参数，功能为删除链表和文件中用户输入ID对应的各类信息

Confirm\_ID () 函数

save () 函数

c\_or\_r () 函数

```
void Sort() {
    int i;
    for(i=1; i<=10; i++) {
        printf("请输入第%d个学生的学号\n", i);
        scanf("%d", &stu[i].id);
        stu[i].next = NULL;
    }
}
```

排序——void Sort () 函数，无参数，功能为按学号或者按成绩 或者指导教师工号排序毕设信息

void ID\_Sort () 函数，无参数，功能为按学号从小到大排序毕设信息

void grades\_Sort () 函数，无参数，功能为按成绩优、良、中、及格排序毕设信息

void instructornum\_Sort () 函数，无参数，功能为按指导教师工号排序毕设信息

save () 函数

c\_or\_r () 函数

void STUDENT\_copy () 函数，参数为（STUDENT \*p, STUDENT \*q）功能为把q指针指向的STUDENT类型数据拷贝到q指针指向的STUDENT

STUDENT\_copy () 函数

STUDENT\_copy () 函数

void show () 函数，无参数，功能为提供各种界面

```
void show() {
    int i;
    for(i=1; i<=10; i++) {
        printf("请输入第%d个学生的学号\n", i);
        scanf("%d", &stu[i].id);
        stu[i].next = NULL;
    }
}
```

部分函数过长无法截图敬请谅解

毕业设计信息管理系统  
main () 函数

全局变量

STUDENT \*Last,STUDENT\*head代表原始链表的头和尾

STUDENT \* h代表备用链表的头

char buf[1024]储存文件中头一行即数据分类的标题

char c[1024]用于全局要求输入数字后的gets () 函数，避免使用者输入字符后程序不能重新读入数字

STUDENT \*x代表抽查链表的头

int luck判断其他函数判断打开哪个文件

枚举类型变量InterFace,表示界面

\*j = NULL,链表的头

x \*k = NULL,链表的尾

int luck(1000) = 0;抽查的学号

int luck\_num =0;抽查的数量

int pc;要抽查的总数

这些全局变量功能仅为抽查函数的实现

```
struct Student {
    int id;
    char name[20];
    char grade[20];
    char score[20];
    char teacher[20];
    char status[20];
    struct Student *next;
};
```

创建结构体STUDENT包含学号、班级、专业、毕设标题、毕设成绩、指导教师工号、指导教师职称、毕设分类、是否在实践中完成、一个指向链表下一节的指针

查询——void Inquire () 函数，无参数，功能为用组合的方式查询毕设信息

void cut () 函数，参数为（STUDENT \* tcut）功能为将t为头的备用链表中删去一节

\_Print () 函数

c\_or\_r () 函数

```
void Count() {
    int i;
    for(i=1; i<=10; i++) {
        printf("请输入第%d个学生的学号\n", i);
        scanf("%d", &stu[i].id);
        stu[i].next = NULL;
    }
}
```

统计——void Count () 函数，无参数，功能为统计毕设信息中符合某型特点的人数

void grades\_Count () 函数，参数为（int a[4]），功能为统计毕设信息中各成绩的人数

void classification\_Count () 函数，参数为（int s[5]），功能为统计毕设信息中各题目分类的人数

void whe\_Count () 函数，参数为（int c[2]），功能为统计毕设是否在实践中完成的人数

c\_or\_r () 函数

```
void create_x(Student *p) {
    Student *q;
    q = (Student *)malloc(sizeof(Student));
    q->id = p->id;
    q->name = p->name;
    q->grade = p->grade;
    q->score = p->score;
    q->teacher = p->teacher;
    q->status = p->status;
    q->next = p->next;
    p->next = q;
}
```

void create\_x (STUDENT\*p) 函数，无参数，功能为将传入的指针指向的毕设信息创建为x链表的下一节

void Sportcheck\_1 () 函数，无参数，功能为删除已抽查过的毕设信息

void Sportcheck\_print () 函数，无参数，功能为将抽查的毕设信息全部打印出来

void Sportcheck\_save () 函数，无参数，功能为将抽查结果保存至函数

void Sportcheck\_init () 函数，无参数，功能为初始化随机种子和计算抽查点数

```
Student *h_traverse(int ID) {
    Student *p;
    p = stu[1];
    while(p != NULL) {
        if(p->id == ID) {
            return p;
        }
        p = p->next;
    }
    return NULL;
}
```

STUDENT \*h\_traverse (int ID) 函数 参数为（int ID学号），功能为返回学号对应的结构体的指针

int judge () 函数，无参数，功能为判断抽查结果是否符合要求，不符合则返回0，符合则返回1

```
void End() {
    Student *q;
    Student *p;
    for(p = head; p != NULL; p = p->next) {
        free(p);
    }
}
```

End () 函数，无参数，功能为将申请的原始链表内存空间全部释放