

电子科技大学信息与软件工程学院

实 验 报 告

学 号 2018091602012

姓 名 杨贺然

(实验) 课程名称 程序设计与算法基础 II

理论教师 郝宗波

实验教师 郝宗波

电子科技大学

实验报告

学生姓名：杨贺然 学号：2018091602012 指导教师：郝宗波

实验地点：基础实验大楼 A508 实验时间：2019.04.19

一、实验室名称：学校实验中心软件实验室

二、实验名称：学生课程成绩查询程序

三、实验学时：8 课时

四、实验原理：

链表是一种物理存储单元上非连续、非顺序的存储结构，数据元素的逻辑顺序是通过链表中的指针链接次序实现的。链表由一系列结点（链表中每一个元素称为结点）组成，结点可以在运行时动态生成。每个结点包括两个部分：一个是存储数据元素的数据域，另一个是存储下一个结点地址的指针域。由于其可扩展性高，适用于这样需要动态添加元素的操作，因此采用链表作为数据存储方式。

栈是一种先进先出的数据结构，队列是一种先进后出的数据结构，利用栈可以实现链表反向，利用队列可以动态管理链表中元素。

五、实验目的

通过本实验练习，掌握需要包括：磁盘文件的读写方法；掌握链表创建方法，掌握链表节点的数据结构定义、链表的排序方法、两个链表内容链接查询的方法；掌握创建栈、进栈和出栈的实现方法。

六、实验内容：

设有学生信息文件 `student.dat`，每个学生记录包括：学号 `sno`、姓名 `sname`、性别 `sex`、专业 `major`；课程信息文件 `course.dat`，课程记录包括：课程号 `cno`、课程名称 `cname`、课时数 `classHours`；课程成绩信息文件 `courseGrade.dat`，成绩记录包括学号 `sno`、课程号 `cno`、考试成绩 `score`；自学教材 61-64 页线性表的应用，设计应用程序完成如下功能，同时需要设计程序运行的功能菜单：

- (1) 输入 10 条学生记录，其中软件技术专业 5 人，人工智能专业 5 人，并存入文件 `student.dat` 中；
- (2) 输入 3 门课程（数据库、数据结构、程序设计）信息记录，并存入文件 `course.dat` 中；
- (3) 输入上述 10 位同学分别选修上述三门课程的考试成绩到文件 `courseGrade.dat` 中；
- (4) 从文件 `student.dat` 中读出学生信息，生成按照学号升序排列的单向链表，并在屏幕上显示输出；
- (5) 从文件 `course.dat` 中读出课程信息，生成按照课程号升序排列的单向链表，并在屏幕上显示输出；
- (6) 从文件 `courseGrade.dat` 中读出成绩信息，生成按照学号和课程号升序排列的单链表，并在屏幕上显示输出；
- (7) 查询所有学生所有课程的考试成绩，生成该课程的成绩单链表，要求包括学号、学生姓名、专业、课程名、考试成绩等信息，按照考试成绩降序排列，并将学生的该成绩信息输出到文件 `studentGrade.dat` 中，同时在屏幕上显示输出；
- (8) 在（7）的链表中，查询指定课程号的所有学生的考试成绩，生成该课程的成绩单链表，要求包括学号、学生姓名、专业、课程名、考试成绩等信息，按照考试成绩降序排列输出到屏幕上显示；
- (9) 在（7）的链表中，查询指定课程号的考试成绩小于 60 分的学生成绩信息，生成该课程的成绩链表，要求包括学号、学生姓名、专业、课程名、考试成绩等信息，并按照考试成绩降序排列在屏幕上显示输出；
- (10) 使用栈实现将（4）的单链表中的学生信息逆序生成新的链表；
- (11) 设计使用链式队列完成问题（7）的要求。

七、实验器材（设备、元器件）：

个人电脑一台。

八、数据结构与程序：

1. 明确学生与课程属性，定义结构与链表结构

学生信息包括学生姓名，学号，性别，专业；课程信息包括课程号，课程名

称和学时；成绩信息包括学号，课程号，成绩。其中用字符数组保存学生姓名，学号，课程号，课程名称，用布尔型保存学生性别和专业，用整数型保存学时数与成绩。链表结构包括学生信息，课程信息或成绩信息与下一节点指针。

学生信息，课程信息与成绩信息结构如下：

```
typedef struct {
    char sNo[MAX_ID_LEN];
    char sName[MAX_NAME_LEN];
    bool sex; /* 0 - 男    1 - 女 */
    bool major; /* 0 - 软件技术    1 - 人工智能 */
} Student;

typedef struct {
    char cNo[MAX_ID_LEN];
    char cName[MAX_NAME_LEN];
    int classHours;
} Course;

typedef struct {
    char sNo[MAX_ID_LEN];
    char cNo[MAX_NAME_LEN];
    int score;
} Grade;
```

链表结构如下：

```
typedef struct snode {
    Student info;
    struct snode *nxt;
} studentList;

typedef struct cnode {
    Course info;
    struct cnode *nxt;
} courseList;

typedef struct gnode {
    Grade info;
    struct gnode *nxt;
} gradeList;
```

2. 实现主程序部分

主程序部分需要输出菜单，接收用户选择。这里用标准输入输出与用户交互。实现如下：

```

void outputMenu()
{
    puts("*****");
    puts("1. 输入学生信息并保存；");
    puts("2. 输入课程信息并保存；");
    puts("3. 输入成绩信息并保存；");
    puts("4. 读取学生信息，创建按学号升序的单向链表；");
    puts("5. 读取课程信息，创建按课程号升序的单向链表；");
    puts("6. 读取成绩信息，创建按学号和课程号升序的单向链表；");
    puts("7. 读取成绩信息，查询所有学生所有课程的考试成绩（需先运行 1, 2, 3）；");
    puts("8. 读取成绩信息，查询所有学生指定课程的考试成绩（需先运行 7）；");
    puts("9. 读取成绩信息，查询指定课程成绩小于 60 分的学生（需先运行 7）；");
    puts("a. 将 4 中生成的链表反转（需先运行 4）；");
    puts("b. 用链式队列解决 7 的问题；");
    puts("其他. 退出系统；");
    puts("*****");
    return ;
}

```

3. 实现创建学生信息链表功能

该函数需要从标准输入中读入学生的所有信息，然后创建链表节点存储，并插入学生信息的链表中。最后需要把学生的信息全部输出到文件中。

详细实现如下：

```

void inputStudentInfo()
{
    FILE *fp = fopen("student.dat", "w");
    char t[200]; int i;
    for (i = 1; i <= 10; i++)
    {
        printf("请输入学号：");
        scanf("%s", t); fprintf(fp, "%s", t);
        printf("请输入姓名：");
        scanf("%s", t); fprintf(fp, " %s", t);
        printf("请输入性别 (F - 女, M - 男)：");
        scanf("%s", t); fprintf(fp, " %s", t);
        printf("请输入专业 (S - 软件技术, A - 人工智能)：");
        scanf("%s", t); fprintf(fp, " %s\n", t);
        puts("录入成功");
    }
    fclose(fp);
    puts("操作成功");
    return ;
}

```

4. 实现创建课程信息链表功能

该函数需要从标准输入中读入课程的所有信息，然后创建链表节点存储，并插入课程信息的链表中。最后需要把课程的信息全部输出到文件中。

详细实现如下：

```
void inputCourseInfo()
{
    FILE *fp = fopen("course.dat", "w");
    char t[200]; int h, i;
    for (i = 1; i <= 3; i++)
    {
        printf("请输入课程编号：");
        scanf("%s", t); fprintf(fp, "%s", t);
        printf("请输入课程名称：");
        scanf("%s", t); fprintf(fp, " %s", t);
        printf("请输入课时数：");
        scanf("%d", &h); fprintf(fp, " %d\n", h);
        puts("录入成功");
    }
    fclose(fp);
    puts("操作成功");
    return ;
}
```

5. 实现创建成绩信息链表功能

该函数需要从标准输入中读入成绩的所有信息，然后创建链表节点存储，并插入成绩信息的链表中。最后需要把成绩的信息全部输出到文件中。

详细实现如下：

```
void inputGradeInfo()
{
    FILE *fp = fopen("courseGrade.dat", "w");
    char t[200]; int h, i;
    for (i = 1; i <= 30; i++)
    {
        printf("请输入学号：");
        scanf("%s", t); fprintf(fp, "%s", t);
        printf("请输入课程号：");
        scanf("%s", t); fprintf(fp, " %s", t);
        printf("请输入考试成绩：");
        scanf("%d", &h); fprintf(fp, " %d\n", h);
        puts("录入成功");
    }
    fclose(fp);
    puts("操作成功");
    return ;
}
```


6. 实现读入学生信息并按学号排序的功能

该函数需要从文件读入学生信息，并将学生信息按学号排序后输出到屏幕。每次读取一条学生信息记录，都将它保存到一个链表节点中。从头遍历链表，直到找到第一个学号比这条记录大位置，并将其插入到这个位置前面。最后从头开始输出链表中元素。

详细实现如下：

```
void rwGenerateStudentLink()
{
    FILE *fp = fopen("student.dat", "r");

    if (fp == NULL)
    {
        puts("文件打开失败");
        return ;
    }

    char t[3]; int i;

    slist = NULL; slist_last = NULL;
    for (i = 1; i <= 10; i++)
    {
        studentList *a = (studentList *)malloc(sizeof(studentList));
        a -> nxt = NULL;
        fscanf(fp, "%s", (a -> info).sNo);
        fscanf(fp, "%s", (a -> info).sName);
        fscanf(fp, "%s", t);
        if (t[0] == 'F') (a -> info).sex = true;
        else (a -> info).sex = false;
        fscanf(fp, "%s", t);
```

```
        if (t[0] == 'A') (a -> info).major = true;
        else (a -> info).major = false;

        if (slist == NULL)
        {
            slist = a;
            slist_last = a;
        }
        else
        {
            studentList *p = slist, *pre = NULL;
            while (p)
            {
                if (strcmp(p -> info.sNo, a -> info.sNo) > 0)
                    break;
                pre = p;
                p = p -> nxt;
            }
            if (!pre)
            {
                a -> nxt = slist;
                slist = a;
            }
            else if (pre == slist_last)
```

```

        {
            slist_last -> nxt = a;
            slist_last = a;
        }
        else
        {
            pre -> nxt = a;
            a -> nxt = p;
        }
    }
}
puts("操作成功");
studentList *p = slist;
while (p)
{
    #define aa p -> info
    printf("%s %s %s %s\n", aa.sNo, aa.sName, aa.sex ? "F" : "M", aa.major ? "A" : "S");
    #undef aa
    p = p -> nxt;
}

fclose(fp);
return ;
}

```

7. 实现读入课程信息并按课程号排序功能

该函数需要从文件读入课程信息，并将课程信息按课程号排序后输出到屏幕。每次读取一条课程信息记录，都将它保存到一个链表节点中。从头遍历链表，直到找到第一个课程号比这条记录大位置，并将其插入到这个位置前面。最后从头开始输出链表中元素。

详细实现详见附件 studentsInfo.c 中 void rwGenerateCourseLink()。

```

void rwGenerateCourseLink()
{
    FILE *fp = fopen("course.dat", "r");

    if (fp == NULL)
    {
        puts("文件打开失败");
        return ;
    }

    int t, i;

    clist = NULL; clist_last = NULL;
    for (i = 1; i <= 3; i++)
    {
        courseList *a = (courseList *)malloc(sizeof(courseList));
        a -> nxt = NULL;
        fscanf(fp, "%s", (a -> info).cNo);
        fscanf(fp, "%s", (a -> info).cName);
        fscanf(fp, "%d", &(a -> info).classHours);

        if (clist == NULL)
        {

```



```

        clist = a;
        clist_last = a;
    }
    else
    {
        courseList *p = clist, *pre = NULL;
        while (p)
        {
            if (strcmp(p -> info.cNo, a -> info.cNo) > 0)
                break;
            pre = p;
            p = p -> nxt;
        }
        if (!pre)
        {
            a -> nxt = clist;
            clist = a;
        }
        else if (pre == clist_last)
        {
            clist_last -> nxt = a;
            clist_last = a;
        }
        else

```

```

    {
        pre -> nxt = a;
        a -> nxt = p;
    }
}

puts("操作成功");
courseList *p = clist;
while (p)
{
    #define aa p -> info
    printf("%s %s %d\n", aa.cNo, aa.cName, aa.classHours);
    #undef aa
    p = p -> nxt;
}

fclose(fp);
return ;
}

```

8. 实现读入成绩信息并按学号与课程号排序功能

该函数需要从文件读入成绩信息，并将成绩信息按学号与课程号排序后输出到屏幕。每次读取一条成绩信息记录，都将它保存到一个链表节点中。以课程号为第一关键字，学号为第二关键字，从头遍历链表，直到找到第一个课程号比这

条记录大，或课程号相同且第一个学号比这条记录大的位置，将这条记录插入至该位置前面。最后从头开始输出链表中元素。

详细实现详见附件 studentsInfo.c 中 void rwGenerateGradeLink()。

```
void rwGenerateGradeLink()
{
    FILE *fp = fopen("courseGrade.dat", "r");

    if (fp == NULL)
    {
        puts("文件打开失败");
        return ;
    }

    int t, i;

    glist = NULL; glist_last = NULL;
    for (i = 1; i <= 30; i++)
    {
        gradeList *a = (gradeList *)malloc(sizeof(gradeList));
        a -> nxt = NULL;
        fscanf(fp, "%s", (a -> info).sNo);
        fscanf(fp, "%s", (a -> info).cNo);
        fscanf(fp, "%d", &(a -> info).score);
```

```
        if (glist == NULL)
        {
            glist = a;
            glist_last = a;
        }
        else
        {
            gradeList *p = glist, *pre = NULL;
            while (p)
            {
                int tmp = strcmp(p -> info.cNo, a -> info.cNo);
                if (tmp > 0 || (!tmp && p -> info.score < a -> info.score))
                    break;
                pre = p;
                p = p -> nxt;
            }
            if (!pre)
            {
                a -> nxt = glist;
                glist = a;
            }
        }
    }
}
```

```

        else if (pre == glist_last)
        {
            glist_last -> nxt = a;
            glist_last = a;
        }
        else
        {
            pre -> nxt = a;
            a -> nxt = p;
        }
    }
}
puts("操作成功");
gradeList *p = glist;
while (p)
{
    #define aa p -> info
    printf("%s %s %d\n", aa.cNo, aa.sNo, aa.score);
    #undef aa
    p = p -> nxt;
}

fclose(fp);
return ;

```

9. 实现生成成绩单链表功能

该函数利用（4），（5），（6）中生成的链表生成成绩单链表。先从（6）中生成的链表获取学号，课程号和成绩信息，之后从（4），（5）中链表把信息补全。最后以课程号为第一关键字，成绩为第二关键字，从头遍历链表，直到找到第一个课程号比这条记录大，或课程号相同且第一个成绩比这条记录小的位置，将这条记录插入至该位置前面。最后从头开始输出链表中元素，注意要输出到屏幕和文件。

详细实现详见附件 studentsInfo.c 中 void generateReportLink()。

10. 实现查询指定课程成绩成绩单功能

该函数利用（7）中生成的链表生成成绩单链表。遍历链表中的每一个元素，将课程号与查询的课程号相同的记录输出。

详细实现如下：

```

void generatePartReportLink()
{
    printf("请输入查询的课程号：");
    char t[10];
    scanf("%s", t);
    reportList *p = rlist;
    bool haveCourse = false;
    while (p)
    {
        if (strcmp(p -> info.b.cNo, t) == 0)
        {
            haveCourse = true;
            #define aa p -> info.a
            printf("%s %s %s %s ", aa.sNo, aa.sName, aa.sex ? "F" : "M", aa.major ? "A" : "S");
            #undef aa
            #define aa p -> info.b
            printf("%s %s %d %d\n", aa.cNo, aa.cName, aa.classHours, p -> info.score);
            #undef aa
        }
        p = p -> next;
    }

    if (!haveCourse) puts("Didn't find the course...");
    return ;
}

```

11. 实现查询不及格学生记录功能

该函数利用（7）中生成的链表生成成绩单链表。遍历链表中的每一个元素，如果这条记录的成绩小于 60，则输出这条记录。

详细实现如下：

```

void getFailedStudent()
{
    printf("请输入查询的课程号：");
    char t[10];
    scanf("%s", t);
    reportList *p = rlist;

    bool haveCourse = false, haveFailed = false;
    while (p)
    {
        if (strcmp(p -> info.b.cNo, t) == 0)
        {
            haveCourse = true;
            if (p -> info.score < 60)
            {
                haveFailed = true;
                #define aa (p -> info.a)
                printf("%s %s %s %s ", aa.sNo, aa.sName, aa.sex ? "F" : "M", aa.major ? "A" : "S");
                #undef aa
                #define aa (p -> info.b)
                printf("%s %s %d %d\n", aa.cNo, aa.cName, aa.classHours, p -> info.score);
                #undef aa
            }
        }
        p = p -> next;
    }

    if (!haveCourse) puts("Didn't find the course...");
    else if (haveFailed) puts("No failed students!");
    return ;
}

```

12. 实现学生信息逆序存储功能

该函数利用学生信息链表，将链表中元素依次全部压入栈中，然后依次弹出

元素，将元素按弹出先后次序插入链表尾部。

详细实现详见附件 studentsInfo.c 中 void reverseStudentLink()。

13. 实现用队列管理成绩信息功能

该函数利用成绩信息链表，将链表中元素依次插入队列。再从队列中弹出。

详细实现详见附件 studentsInfo.c 中 void generateReportLink_queue()。

14. 反复调试直到程序可以正常实现所有功能

九、程序运行结果

1. 主界面实现

```
*****
1. 输入学生信息并保存;
2. 输入课程信息并保存;
3. 输入成绩信息并保存;
4. 读取学生信息, 创建按学号升序的单向链表;
5. 读取课程信息, 创建按课程号升序的单向链表;
6. 读取成绩信息, 创建按学号和课程号升序的单向链表;
7. 读取成绩信息, 查询所有学生所有课程的考试成绩 (需先运行 1, 2, 3);
8. 读取成绩信息, 查询所有学生指定课程的考试成绩 (需先运行 7);
9. 读取成绩信息, 查询指定课程成绩小于 60 分的学生 (需先运行 7);
a. 将 4 中生成的链表反转 (需先运行 4);
b. 用链式队列解决 7 的问题;
其他. 退出系统;
*****
输入您的选择:
```

2. 输入学生信息并保存

```
*****
输入您的选择: 1
请输入学号: 0 yhr1 M S
请输入姓名: 请输入性别 (F - 女, M - 男): 请输入专业 (S - 软件技术, A - 人工智能): 录入成功
请输入学号: 1 yhr2 F A
请输入姓名: 请输入性别 (F - 女, M - 男): 请输入专业 (S - 软件技术, A - 人工智能): 录入成功
请输入学号: 2 HeRaNO M A
请输入姓名: 请输入性别 (F - 女, M - 男): 请输入专业 (S - 软件技术, A - 人工智能): 录入成功
请输入学号: 3 jiasdfhdshf F S
请输入姓名: 请输入性别 (F - 女, M - 男): 请输入专业 (S - 软件技术, A - 人工智能): 录入成功
请输入学号: 4 sdfsadfdsc M A
请输入姓名: 请输入性别 (F - 女, M - 男): 请输入专业 (S - 软件技术, A - 人工智能): 录入成功
请输入学号: 5 werqwerqwe M S
请输入姓名: 请输入性别 (F - 女, M - 男): 请输入专业 (S - 软件技术, A - 人工智能): 录入成功
请输入学号: 6 Kanade F A
请输入姓名: 请输入性别 (F - 女, M - 男): 请输入专业 (S - 软件技术, A - 人工智能): 录入成功
请输入学号: 7 Yuzuru M A
请输入姓名: 请输入性别 (F - 女, M - 男): 请输入专业 (S - 软件技术, A - 人工智能): 录入成功
请输入学号: 8 Taiko F S
请输入姓名: 请输入性别 (F - 女, M - 男): 请输入专业 (S - 软件技术, A - 人工智能): 录入成功
请输入学号: 9 osu M A
请输入姓名: 请输入性别 (F - 女, M - 男): 请输入专业 (S - 软件技术, A - 人工智能): 录入成功
操作成功
```

3. 输入课程信息并保存

```
输入您的选择: 2
请输入课程编号: C01
请输入课程名称: Database
请输入课时数: 48
录入成功
请输入课程编号: C02
请输入课程名称: DataStructure
请输入课时数: 64
录入成功
请输入课程编号: C03
请输入课程名称: ProgramDesign
请输入课时数: 64
录入成功
操作成功
```

4. 输入成绩信息并保存


```

*****
您的选择: 3
请输入学号: 1 C01 100
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 2 C01 75
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 3 C01 66
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 4 C02 53
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 5 C02 60
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 6 C02 61
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 7 C03 98
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 8 C03 88
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 9 C03 13
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 0 C03 59
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 1 C02 70
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 2 C02 71
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 3 C02 45
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 4 C03 80
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 5 C03 63
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 6 C03 67
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 7 C01 98
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 8 C01 66
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 9 C01 40
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 0 C01 50
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 1 C03 95
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 2 C03 97
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 3 C03 88
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 4 C01 87
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 5 C01 83
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 6 C01 53
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 7 C02 42
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 8 C02 46
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 9 C02 100
请输入课程号: 请输入考试成绩: 录入成功
请输入学号: 0 C02 22
请输入课程号: 请输入考试成绩: 录入成功
操作成功

```

5. 读取学生信息，创建按学号升序的单向链表

```
输入您的选择: 4
操作成功
0 yhr1 M S
1 yhr2 F A
2 HeRaNO M A
3 jiasdfhdshf F S
4 sdfsadfdsc M A
5 werqwerqwe M S
6 Kanade F A
7 Yuzuru M A
8 Taiko F S
9 osu M A
```

6. 读取课程信息，创建按课程号升序的单向链表

```
输入您的选择: 5
操作成功
C01 Database 48
C02 DataStructure 64
C03 ProgramDesign 64
```

7. 读取成绩信息，创建按学号和课程号升序的单向链表

```
输入您的选择: 6
操作成功
C01 1 100
C01 7 98
C01 4 87
C01 5 83
C01 2 75
C01 3 66
C01 8 66
C01 6 53
C01 0 50
C01 9 40
C02 9 100
C02 2 71
C02 1 70
C02 6 61
C02 5 60
C02 4 53
C02 8 46
C02 3 45
C02 7 42
C02 0 22
C03 7 98
C03 2 97
C03 1 95
C03 8 88
C03 3 88
C03 4 80
C03 6 67
C03 5 63
C03 0 59
C03 9 13
```

8. 读取成绩信息，查询所有学生所有课程的考试成绩

```
输入您的选择：7
操作成功
1 yhr2 F A C01 Database 48 100
7 Yuzuru M A C01 Database 48 98
4 sdfsdsc M A C01 Database 48 87
5 werqwerqwe M S C01 Database 48 83
2 HeRaNO M A C01 Database 48 75
3 jiasdfhdshf F S C01 Database 48 66
8 Taiko F S C01 Database 48 66
6 Kanade F A C01 Database 48 53
0 yhr1 M S C01 Database 48 50
9 osu M A C01 Database 48 40
9 osu M A C02 DataStructure 64 100
2 HeRaNO M A C02 DataStructure 64 71
1 yhr2 F A C02 DataStructure 64 70
6 Kanade F A C02 DataStructure 64 61
5 werqwerqwe M S C02 DataStructure 64 60
4 sdfsdsc M A C02 DataStructure 64 53
8 Taiko F S C02 DataStructure 64 46
3 jiasdfhdshf F S C02 DataStructure 64 45
7 Yuzuru M A C02 DataStructure 64 42
0 yhr1 M S C02 DataStructure 64 22
7 Yuzuru M A C03 ProgramDesign 64 98
2 HeRaNO M A C03 ProgramDesign 64 97
1 yhr2 F A C03 ProgramDesign 64 95
8 Taiko F S C03 ProgramDesign 64 88
3 jiasdfhdshf F S C03 ProgramDesign 64 88
4 sdfsdsc M A C03 ProgramDesign 64 80
6 Kanade F A C03 ProgramDesign 64 67
5 werqwerqwe M S C03 ProgramDesign 64 63
0 yhr1 M S C03 ProgramDesign 64 59
9 osu M A C03 ProgramDesign 64 13
```

9. 读取成绩信息，查询所有学生指定课程的考试成绩
当课程存在时：

```
输入您的选择：8
请输入查询的课程号：C02
9 osu M A C02 DataStructure 64 100
2 HeRaNO M A C02 DataStructure 64 71
1 yhr2 F A C02 DataStructure 64 70
6 Kanade F A C02 DataStructure 64 61
5 werqwerqwe M S C02 DataStructure 64 60
4 sdfsdsc M A C02 DataStructure 64 53
8 Taiko F S C02 DataStructure 64 46
3 jiasdfhdshf F S C02 DataStructure 64 45
7 Yuzuru M A C02 DataStructure 64 42
0 yhr1 M S C02 DataStructure 64 22
```

当课程不存在时:

```
输入您的选择: 8
请输入查询的课程号: C04
Didn't find the course...
```

10. 读取成绩信息, 查询指定课程成绩小于 60 分的学生
当课程存在时:

```
输入您的选择: 9
请输入查询的课程号: C01
6 Kanade F A C01 Database 48 53
0 yhr1 M S C01 Database 48 50
9 osu M A C01 Database 48 40
```

当课程不存在时:

```
输入您的选择: 9
请输入查询的课程号: C04
Didn't find the course...
```

11. 将 4 中生成的链表反转

```
输入您的选择: a
9 osu M A
8 Taiko F S
7 Yuzuru M A
6 Kanade F A
5 werqwerqwe M S
4 sdfsadfdsc M A
3 jiasdfhdshf F S
2 HeRaNO M A
1 yhr2 F A
0 yhr1 M S
```

12. 用链式队列解决 7 的问题


```
输入您的选择: b
操作成功
1 yhr2 F A C01 Database 48 100
7 Yuzuru M A C01 Database 48 98
4 sdfsadfdsc M A C01 Database 48 87
5 werqwerqwe M S C01 Database 48 83
2 HeRaNO M A C01 Database 48 75
3 jiasdfhdshf F S C01 Database 48 66
8 Taiko F S C01 Database 48 66
6 Kanade F A C01 Database 48 53
0 yhr1 M S C01 Database 48 50
9 osu M A C01 Database 48 40
9 osu M A C02 DataStructure 64 100
2 HeRaNO M A C02 DataStructure 64 71
1 yhr2 F A C02 DataStructure 64 70
6 Kanade F A C02 DataStructure 64 61
5 werqwerqwe M S C02 DataStructure 64 60
4 sdfsadfdsc M A C02 DataStructure 64 53
8 Taiko F S C02 DataStructure 64 46
3 jiasdfhdshf F S C02 DataStructure 64 45
7 Yuzuru M A C02 DataStructure 64 42
0 yhr1 M S C02 DataStructure 64 22
7 Yuzuru M A C03 ProgramDesign 64 98
2 HeRaNO M A C03 ProgramDesign 64 97
1 yhr2 F A C03 ProgramDesign 64 95
8 Taiko F S C03 ProgramDesign 64 88
3 jiasdfhdshf F S C03 ProgramDesign 64 88
4 sdfsadfdsc M A C03 ProgramDesign 64 80
6 Kanade F A C03 ProgramDesign 64 67
5 werqwerqwe M S C03 ProgramDesign 64 63
0 yhr1 M S C03 ProgramDesign 64 59
9 osu M A C03 ProgramDesign 64 13
```

十、实验结论

对于这种学生信息管理需求，可以用链表实现，对于将链表反向，可以利用栈先进后出的性质实现。

十一、总结及心得体会

经过本实验，对学生成绩信息管理系统，链表，栈和队列有了更深刻的了解，锻炼了代码编写与调试能力。将功能写成函数既能简化程序，又能增加程序可读性，便于调试与日后维护。