

中国地质大学课程设计报告



课程名称	C 语言课程设计	
姓 名	葛 雨 奇	
学 院	环境学院	
班 级	046181	
学 号		
联系电话		
邮 箱		
指导老师	陈 喆	
报告时间	2019 年 1 月 8 日	

问题及回答记录 (self-test Q&A related to your experience reports, and no less than 5 questions or contents of Q&A nearly full of the sheet as well as handwriting required)

[illegible]

课程设计评语

对课程论文的评语：

平时成绩：	课程论文成绩：
总 成 绩：	评阅人签名：

- 注：1、无评阅人签名成绩无效；
2、必须用钢笔或圆珠笔批阅，用铅笔阅卷无效；
3、如有平时成绩，必须在上面对评分表中标出，并计算入总成绩。

目 录

课程设计评语.....	3
目 录.....	4
1. 课程论文题目.....	2
2. 程序设计思路.....	2
3. 功能模块图.....	3
4. 数据结构设计.....	3
5. 算法设计.....	3
6. 程序代码.....	4
7. 程序运行结果.....	18
8. 编程中遇到的困难及解决方法.....	18
9. 总结心得及良好建议.....	22
10. 致谢.....	23

1. 课程论文题目

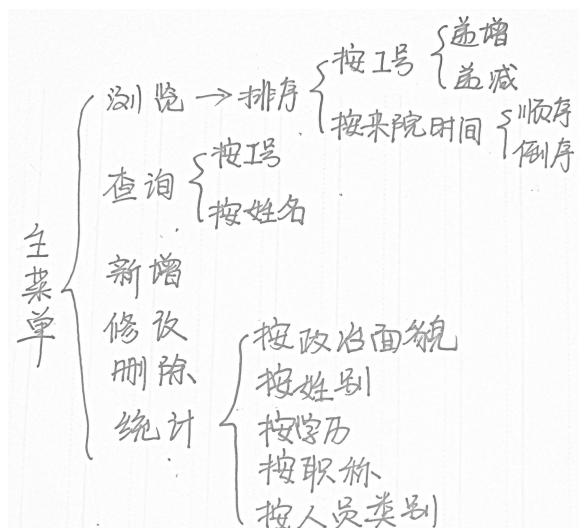
某高校主要人员有:在职人员(行政人员、教师、一般员工)退休人员 and 临时工。现在需要存储这些人员的人事档案信息,内容包括编号、姓名、性别、年龄、职务、职称、政治面貌、最高学历、任职时间、来院时间、人员类别。其中,人员编号唯一,不能重复

- (1) 添加删除功能: 能根据学院人事的变动情况, 添加删除记录。
- (2) 查询功能: 能根据编号和姓名进行查询。
- (3) 编辑功能(高级): 根据查询对相应的记录进行修改并存储。
- (4) 统计功能: 能根据多种参数进行人员的统计(在职人数、党员人数、女工人数、高学历高职称人数(硕士学位以上或者副教授以上)), 统计要求同时显示被统计者的信息。
- (5) 排序功能: 按照年龄、来院时间进行排序。
- (6) 保存功能: 能对输入的数据进行相应的存储

2. 程序设计思路

- 1. 功能模块的编写顺序: 新增→保存→浏览→修改→查询→统计→删除→排序
- 2. “查询”和“统计”的相似性很高, 部分代码可以通用
- 3. “保存”模块不单独列入菜单, 而在每一个其他的模块后都执行
- 4. “排序”模块设置在“浏览”模块内部, 可以更方便进行排序
- 5. 将“修改”模块嵌套在“查找”模块内部, 可以在查找后选择直接执行
- 6. 在所有要手动确认下一步执行的操作时, 提供“返回”功能

3. 功能模块图



4. 数据结构设计

将人员的每个信息都以 char 型保存

```
struct worker
```

```
{  
    char id[20];           //工号  
    char name[20];        //姓名  
    char sex[10];         //性别  
    char age[5];          //年龄  
    char edu[20];         //学历  
    char duty[10];        //职务  
    char political[20];    //政治面貌  
    char tittle[10];      //职称  
    char history[20];     //来院时间  
    char ago[5];          //任职时间(年)  
    char range[20];       //人员类别  
};
```

5. 算法设计

通过以下模块进行查找或统计：

```
k = -1;
for (i = 0; i < 文件中含有的项目数; i++)
    if (strcmp(要查找或统计的项, 原文件中的项) == 0)    //判断它们是否相同
    {
        k = i;          //k用于记录是否找到相同的相匹配的项。若无, k的值仍为-1
        printf_one(k); break;
        num++;           //num为统计后的总数
    }
```

6. 程序代码

```
#include<stdio.h>
#include<string.h>
#define N 50

struct worker
{
    char id[20];          //工号
    char name[20];        //姓名
    char sex[10];         //性别
    char age[5];          //年龄
    char edu[20];         //学历
    char duty[10];        //职务
    char political[20];    //政治面貌
    char tittle[10];      //职称
    char history[20];     //来院时间
    char ago[5];          //任职时间(年)
    char range[20];       //人员类别
} worker[N];

void save(int n)    //保存函数, 保存n个记录
{
    FILE *fp;
    int i;
    if ((fp = fopen("worker.txt", "wb")) == NULL)    //以只写方式为输出打开一个二进制文件
    {
        printf("\n 无法打开文件\n");
    }
    for (i = 0; i < n; i++)
        if (fwrite(&worker[i], sizeof(struct worker), 1, fp) != 1)
            printf("写入错误\n");
    fclose(fp);
}

add(int n)
{
```

```

FILE *fp;
int i;
if ((fp = fopen("worker.txt", "ab")) == NULL)           //以追加方式为输出打开一个二进制文件
{
    printf("\n 无法打开文件\n");
}
for (i = 0; i < n; i++)
    if (fwrite(&worker[i], sizeof(struct worker), 1, fp) != 1)
        printf("写入错误\n");
fclose(fp);
}

int load()        //加载函数
{
    FILE *fp;
    int i;
    if ((fp = fopen("worker.txt", "rb")) == NULL)
    {
        printf("\n 无法打开文件\n");
        return NULL;
    }
    for (i = 0; !feof(fp); i++)
        fread(&worker[i], sizeof(struct worker), 1, fp);
    fclose(fp);
    return(i - 1);
}

void no_input(int i, int n)    //工号输入函数, i 表示第 i 个职工的信息, n 表示比较到第 n 个职工
{
    int j, k, w1;
    do
    {
        w1 = 0;
        printf("工号: ");
        scanf("%s", worker[i].id);
        for (j = 0; worker[i].id[j] != '\0'; j++)
            if (worker[i].id[j] < '0' || worker[i].id[j] > '9')        //判断工号是否为数字
            {
                puts("请重新输入! \n");
                w1 = 1; break;
            }
        if (w1 != 1)
            for (k = 0; k < n; k++)                //比较到第 N 个职工
                if (k != i && strcmp(worker[k].id, worker[i].id) == 0)    //判断职工工号是否有雷同
                {
                    puts("请重新输入! \n");
                    w1 = 1; break;
                }
    } while (w1 == 1);
}

void input(int i)        //输入一个记录的函数
{
    no_input(i, i);
    printf("姓名:"); scanf("%s", worker[i].name);
    printf("性别:"); scanf("%s", worker[i].sex);
    printf("年龄:"); scanf("%s", worker[i].age);
    printf("职务:"); scanf("%s", worker[i].duty);
    printf("学历:"); scanf("%s", worker[i].edu);
    printf("政治面貌:"); scanf("%s", worker[i].political);
    printf("职称:"); scanf("%s", worker[i].tittle);
    printf("来院时间:"); scanf("%s", worker[i].history);
}

```



```

    printf("任职时间(年):");scanf("%s", worker[i].ago);
    printf("人员类别:");scanf("%s", worker[i].range);
}

void printf_one(int i)          //显示一个记录的函数
{
    printf("%10s %8s %5s %5s %8s %8s %10s %10s %15s %6s %15s\n",
        worker[i].id, worker[i].name, worker[i].sex, worker[i].age, worker[i].duty, worker[i].edu,
        worker[i].political, worker[i].tittle, worker[i].history, worker[i].ago, worker[i].range);
}

void printf_back()             //一个任务结束后选择浏览还是返回
{
    int k, w;
    printf("\n\n\t输入成功^~.\n\n");
    printf("请选择要执行的操作:\n\n\t1). 浏览全部\t2). 返回: [ ]\b\b");
    scanf("%d", &w);
    if (w == 1)browse();
    else menu();
}

modify_data(int i, int n)      //修改模块
{
    int c, w1;
    do
    {
        puts("\n 请选择=>\n\n 1). 工号 2). 姓名 3). 性别 4). 年龄 5). 职务 6). 学历 7). 政治面貌 8). 职称 9). 来
        院时间 10) 任职时间. 11). 人员类别 0). 取消");
        printf("选择:[ ]\b\b");
        scanf("%d", &c);
        if (c > 11 || c < 0)
        {
            puts("\n 选择有误, 请重选");
            getchar();
        }
    } while (c > 11 || c < 0);
    do
    {
        switch (c)
        {
            case 1: no_input(i, n); break;
            case 2: printf("姓名:"); scanf("%s", worker[i].name); break;
            case 3: printf("性别:"); scanf("%s", worker[i].sex); break;
            case 4: printf("年龄:"); scanf("%s", worker[i].age); break;
            case 5: printf("职务:"); scanf("%s", worker[i].duty); break;
            case 6: printf("学历:"); scanf("%s", worker[i].edu); break;
            case 7: printf("政治面貌:"); scanf("%s", worker[i].political); break;
            case 8: printf("职称:"); scanf("%s", worker[i].tittle); break;
            case 9: printf("来院时间:"); scanf("%s", worker[i].history); break;
            case 10: printf("任职时间"); scanf("%s", worker[i].ago); break;
            case 11: printf("人员类别"); scanf("%s", worker[i].range); break;
            case 0: menu(); break;
        }
        puts("\n 修改完成:\n");
        printf_face();
        printf_one(i);
        printf("\n 确定?\n\n\t1). 是 2). 否, 重新修改 3). 不保存退出 [ ]\b\b");
        scanf("%d", &w1);
        if (w1 == 1) save(n);
    } while (w1 == 2);
    return(w1);
}

```

```

printf_face()          //显示数据结构项目
{
    printf("\n      工号      姓名      性别      年龄      职务      学历      政治面貌      职称      来院时间      任
职时间(年)      人员类别\n\n");
}

void enter()           //输入模块
{
    int i, n;
    printf("新增多少个职工信息(0-%d):", N - 1);
    scanf("%d", &n);
    printf("\n 请输入职工信息\n\n");
    for (i = 0; i < n; i++)
        input(i);
    if (i != 0)
        add(n);
    printf_back();
}

browse()              //浏览模块
{
    int i, j, n;
    n = load();
    printf_face();
    for (i = 0; i < n; i++)
    {
        if ((i != 0) && (i % 10 == 0))
        {
            printf("\n\n 按任意键显示下一页...");
            getch();
            puts("\n\n");
        }
        printf_one(i);
    }
    printf("\n\t 这里有 %d 条记录.\n", n);
    printf("\n 是否进行排序? 1). 是 2). 否 [ ]\b\b");
    scanf("%d", &j);
    if (j == 1) rank();
    putchar(10);
    system("pause");
    menu();
}

modify()              //修改模块
{
    struct worker s;
    FILE *fp;
    int i, n, k, w0 = 1, w1, w2 = 0;
    n = load();
    do
    {
        k = -1;
        printf_face();
        for (i = 0; i < n; i++)
        {
            if ((i != 0) && (i % 10 == 0))          //目的是分屏显示
            {
                printf("\n\n 记住要修改的工号. 按任意键显示下一页");
                getch();
                puts("\n\n");
            }

```

```

        printf_one(i);          //调用显示一个记录的函数
    }
    do
    {
        printf("\n\n 输入要修改的职工工号:");
        scanf("%s", s.id);      //输入要修改的 ID
        for (i = 0; i < n; i++)  //查找要修改的数据
            if (strcmp(s.id, worker[i].id) == 0)
            {
                k = i;          //找到要修改的记录
                s = worker[i];   //把 worker[i] 备份, 以便恢复
            }
        if (k == -1)
            printf("\n\n 未找到 请重输");
    } while (k == -1);          //当 k=-1 表示没有找到
    printf_face();
    printf_one(k);
    w1 = modify_data(k, n);    //修改记录并返回保存控制值 w1, w1=1 表示用户已确认修改
    if (w1 == 1)
    {
        printf("\n 修改成功 ^_^.\n\n 修改另一个? \n\n\t1). 是 2). 否, 保存退出\t[ ]\b\b");
        scanf("%d", &w0);
        w2 = 1;                //用来控制保存, 使 w2=1 是标记已有过修改
    }
    else
    {
        w0 = 0;
        if (w2 == 1)
            worker[k] = s;
    }
    if (w0 != 1 && w2 == 1)
        save(n);
    } while (w0 == 1);
    menu();
}

search()          //查询模块
{
    int c, w1;
    do
    {
        printf("\n 请选择查找依据=>\n\n1). 工号 2). 姓名 3). 取消并返回: [ ]\b\b");
        scanf("%d", &c);
        if (c > 3 || c < 1)
        {
            puts("\n 选择有误! 请重选");
            getchar();
        }
    } while (c > 3 || c < 1);
    {switch (c)
    {
        case 1: search1(); break;
        case 2: search2(); break;
        case 3: menu(); break;
    }
    }
}

search1()         //按工号查询模块
{
    int i, n, k, w1 = 1, w2, w3;
    char s[20];

```

```

n = load();
do
{
    do
    {
        k = -1;
        printf("\n\n 输入要查询的工号:");
        scanf("%s", s);
        printf_face();
        for (i = 0; i < n; i++)
            if (strcmp(s, worker[i].id) == 0)
            {
                k = i;
                printf_one(k); break;
            }
        if (k == -1)
        {
            printf("\n\n 无符合项! ");
            printf("\n\n 继续查找?\n\t1). 是 2). 否, 返回 [ ]\b\b");
            scanf("%d", &w1);
            if (w1 == 2) menu();
        }
    } while (k == -1 && w1 == 1);
    w3 = 0;
    if (k != -1)
    {
        printf("\n\n 接下来干什么?\n\t1). 查找另一个 2). 修改 3). 返回菜单 [ ]\b\b");
        scanf("%d", &w2);
        switch (w2)
        {
            case 1: search1(); break;
            case 2: w3 = modify_data(k, n); break;
            case 3: {menu(); break;}
        }
    }
} while (w2 == 1);
menu();
}

search2()    //按姓名查询模块
{
    int i, n, k, w1 = 1, w2, w3, w4;
    char s[20];
    n = load();
    do
    {
        do
        {
            k = -1;
            printf("\n\n 请输入你要查询的姓名: \n");
            scanf("%s", s);
            printf_face();
            for (i = 0; i < n; i++)
                if (strcmp(s, worker[i].name) == 0)
                {
                    k = i;
                    printf_one(k);
                }
            if (k == -1)
            {
                printf("\n\n 无符合项! ");
                printf("\n\n 继续查找?\n\t1). 是 2). 否, 返回 [ ]\b\b");
            }
        } while (k == -1 && w1 == 1);
        w4 = 0;
        if (k != -1)
        {
            printf("\n\n 接下来干什么?\n\t1). 查找另一个 2). 修改 3). 返回菜单 [ ]\b\b");
            scanf("%d", &w2);
            switch (w2)
            {
                case 1: search1(); break;
                case 2: w4 = modify_data(k, n); break;
                case 3: {menu(); break;}
            }
        }
    } while (w2 == 1);
    menu();
}

```

```

        scanf("%d", &w1);
        if (w1 == 2) menu();
    }
} while (k == -1 && w1 == 1);
w4 = 0; w3 = 0;
if (k != -1)
{
    printf("\n\n 接下来干什么?\n\t1). 查找另一个 2). 修改 3). 返回菜单  [ ]\b\b");
    scanf("%d", &w2);
    switch (w2)
    {
        case 1: search2(); break;
        case 2: w3 = modify_data(k, n); break;
        case 3: {menu();
                break;
            }
    }
}
} while (w2 == 1);
menu();
}

statistics()          //统计模块
{
    int c, w1;
    do
    {
        printf("\n 请选择统计依据=>\n\n1). 政治面貌 2). 性别 3). 学历 4). 职称 5). 人员类别 6). 取消并返回:
[ ]\b\b");
        scanf("%d", &c);
        if (c > 6 || c < 1)
        {
            puts("\n 选择有误! 请重选");
            getchar();
        }
    } while (c > 6 || c < 1);
    {switch (c)
    {
        case 1: statistics1(); break;
        case 2: statistics2(); break;
        case 3: statistics3(); break;
        case 4: statistics4(); break;
        case 5: statistics5(); break;
        case 6: menu(); break;
    }
    }
}

statistics1()         //按政治面貌进行统计
{
    int i, n, k, w1 = 1, w2, w3, num = 0;
    char s[20];
    n = load();
    do
    {
        do
        {
            k = -1;
            printf("\n 输入要统计的政治面貌:");
            scanf("%s", s);
            printf_face();

```

```

        for (i = 0; i < n; i++)
            if (strcmp(s, worker[i].political) == 0)
            {
                k = i;
                printf_one(k);
                num++;
            }
        if (k == -1)
        {
            printf("\n\n 无符合项! ");
            printf("\n\n 重新统计?\n\t1). 是  2). 否, 返回  [ ]\b\b");
            scanf("%d", &w1);
            if (w1 == 2) menu();
        }
    } while (k == -1 && w1 == 1);
    w3 = 0;
    if (k != -1)
    {
        printf("\n 检索到%d 条信息", num);
        printf("\n\n 请选择下一步执行的操作:\n\t1). 统计另一项  2). 修改  3). 返回菜单  [ ]\b\b");
        scanf("%d", &w2);
        switch (w2)
        {
            case 1: statistics(); break;
            case 2: w3 = modify_data(k, n); break;
            case 3: {menu(); break;}
        }
    }
} while (w2 == 1);
menu();
}

statistics2()    //按性别进行统计
{
    int i, n, k, w1 = 1, w2, w3, num = 0;
    char s[20];
    n = load();
    do
    {
        do
        {
            k = -1;
            printf("\n 请输入选择的性别:");
            scanf("%s", s);
            printf_face();
            for (i = 0; i < n; i++)
                if (strcmp(s, worker[i].sex) == 0)
                {
                    k = i;
                    printf_one(k);
                    num++;
                }
            if (k == -1)
            {
                printf("\n\n 无符合项! ");
                printf("\n\n 重新统计?\n\t1). 是  2). 否, 返回  [ ]\b\b");
                scanf("%d", &w1);
                if (w1 == 2) menu();
            }
        } while (k == -1 && w1 == 1);
        w3 = 0;

```

```

        if (k != -1)
        {
            printf("\n 检索到%d 条信息", num);
            printf("\n\n 请选择下一步执行的操作:\n\t1). 统计另一项  2). 修改  3). 返回菜单  [ ]\b\b");
            scanf("%d", &w2);
            switch (w2)
            {
                case 1:statistics();break;
                case 2:w3 = modify_data(k, n);break;
                case 3: {menu();break;}
            }
        }
    } while (w2 == 1);
    menu();
}

statistics3()          //按学历进行统计
{
    int i, n, k, w1 = 1, w2, w3, num = 0;
    char s[20];
    n = load();
    do
    {
        do
        {
            k = -1;
            printf("\n 请输入学历:");
            scanf("%s", s);
            printf_face();
            for (i = 0; i < n; i++)
                if (strcmp(s, worker[i].edu) == 0)
                {
                    k = i;
                    printf_one(k);
                    num++;
                }
            if (k == -1)
            {
                printf("\n\n 无符合项! ");
                printf("\n\n 重新统计?\n\t1). 是  2). 否, 返回  [ ]\b\b");
                scanf("%d", &w1);
                if (w1 == 2) menu();
            }
        } while (k == -1 && w1 == 1);
        w3 = 0;
        if (k != -1)
        {
            printf("\n 检索到%d 条信息", num);
            printf("\n\n 请选择下一步执行的操作:\n\t1). 统计另一项  2). 修改  3). 返回菜单  [ ]\b\b");
            scanf("%d", &w2);
            switch (w2)
            {
                case 1:statistics();break;
                case 2:w3 = modify_data(k, n);break;
                case 3: {menu();break;}
            }
        }
    } while (w2 == 1);
    menu();
}

```

```

statistics4()          //按职称进行统计
{
    int i, n, k, w1 = 1, w2, w3, num = 0;
    char s[20];
    n = load();
    do
    {
        do
        {
            k = -1;
            printf("\n 请输入职称:");
            scanf("%s", s);
            printf_face();
            for (i = 0; i < n; i++)
                if (strcmp(s, worker[i].tittle) == 0)
                {
                    k = i;
                    printf_one(k);
                    num++;
                }
            if (k == -1)
            {
                printf("\n\n 无符合项! ");
                printf("\n\n 重新统计?\n\t1). 是 2). 否, 返回 [ ]\b\b");
                scanf("%d", &w1);
                if (w1 == 2) menu();
            }
        } while (k == -1 && w1 == 1);
        w3 = 0;
        if (k != -1)
        {
            printf("\n 检索到%d 条信息", num);
            printf("\n\n 请选择下一步执行的操作:\n\t1). 统计另一项 2). 修改 3). 返回菜单 [ ]\b\b");
            scanf("%d", &w2);
            switch (w2)
            {
                case 1: statistics(); break;
                case 2: w3 = modify_data(k, n); break;
                case 3: {menu(); break;}
            }
        }
    } while (w2 == 1);
    menu();
}

statistics5()          //按人员类别进行统计
{
    int i, n, k, w1 = 1, w2, w3, num = 0;
    char s[20];
    n = load();
    do
    {
        do
        {
            k = -1;
            printf("\n 请输入人员类别:");
            scanf("%s", s);
            printf_face();
            for (i = 0; i < n; i++)
                if (strcmp(s, worker[i].range) == 0)

```



```

        {
            k = i;
            printf_one(k);
            num++;
        }
    if (k == -1)
    {
        printf("\n\n 无符合项!");
        printf("\n\n 重新统计?\n\t1). 是 2). 否, 返回 [ ]\b\b");
        scanf("%d", &w1);
        if (w1 == 2) menu();
    }
} while (k == -1 && w1 == 1);
w3 = 0;
if (k != -1)
{
    printf("\n 检索到%d 条信息", num);
    printf("\n\n 请选择下一步执行的操作:\n\t1). 统计另一项 2). 修改 3). 返回菜单 [ ]\b\b");
    scanf("%d", &w2);
    switch (w2)
    {
        case 1: statistics(); break;
        case 2: w3 = modify_data(k, n); break;
        case 3: {menu(); break;}
    }
}
} while (w2 == 1);
menu();
}

rank() //排序模块
{
    int c;
    do
    {
        printf("\n 请输入排序依据: 1). 工号 2). 来院时间 3). 取消并返回 [ ]\b\b");
        scanf("%d", &c);
        if (c > 3 || c < 1)
        {
            puts("\n 选择有误! 请重选");
            getchar();
        }
    } while (c > 3 || c < 1);
    switch(c)
    {
        case 1: rank1(); break;
        case 2: rank2(); break;
        case 3: menu(); break;
    }
}

rank1() //按工号进行排序
{
    struct worker t;
    int n=1, c, m=0, v, k = 1, i;
    v = load();
    do
    {
        printf("\n1). 递增 2). 递减 [ ]\b\b");
        scanf("%d", &c);
        if (c > 2 || c < 1)

```

```

        {
            puts("\n 选择有误! 请重选");
            getchar();
        }
    } while (c > 2 || c < 1);
    if (c == 1)
    {
        for (m = 0; m < v - 1; m++, k++)
        {
            for (n = k; n < v; n++)
            {
                if (strcmp(worker[m].id, worker[n].id) > 0)
                {
                    t = worker[m];
                    worker[m] = worker[n];
                    worker[n] = t;
                }
            }
        }
    }
    else
    {
        for (m = 0; m < v - 1; m++, k++)
        {
            for (n = k; n < v; n++)
            {
                if (strcmp(worker[m].id, worker[n].id) < 0)
                {
                    t = worker[m];
                    worker[m] = worker[n];
                    worker[n] = t;
                }
            }
        }
    }
    printf_face();
    for (i = 0; i < v; i++)
        printf_one(i);
    save(v);
}

```

```

rank2()          //按来院时间进行排序
{
    struct worker t;
    int n = 1, c, m = 0, v, k = 1, i;
    v = load();
    do
    {
        printf("\n1). 顺序 2). 倒序 [ ]\b\b");
        scanf("%d", &c);

        if (c > 2 || c < 1)
        {
            puts("\n 选择有误! 请重选");
            getchar();
        }
    } while (c > 2 || c < 1);
    if (c == 1)
    {
        for (m = 0; m < v - 1; m++, k++)
        {
            for (n = k; n < v; n++)

```

```

        {
            if (strcmp(worker[m].history, worker[n].history) > 0)
            {
                t = worker[m];
                worker[m] = worker[n];
                worker[n] = t;
            }
        }
    }
}
else
{
    for (m = 0; m < v - 1; m++, k++)
    {
        for (n = k; n < v; n++)
        {
            if (strcmp(worker[m].history, worker[n].history) < 0)
            {
                t = worker[m];
                worker[m] = worker[n];
                worker[n] = t;
            }
        }
    }
    printf_face();
    for (i = 0; i < v; i++)
        printf_one(i);
    save(v);
}

```

```

delete_file()
{
    int i, n, k, w1 = 1, w2;
    char s[20];
    n = load();
    printf_face();
    for (i = 0; i < n; i++)
        printf_one(i);
    do
    {
        k = -1;
        printf("\n\n 输入要删除的工号:");
        scanf("%s", s);
        printf_face();
        for (i = 0; i < n; i++)
            if (strcmp(s, worker[i].id) == 0)
            {
                k = i;
                printf_one(k); break;
            }
        if (k == -1)
        {
            printf("\n\n 无符合项! ");
            printf("\n\n 重新查找?\n\t1). 是 2). 否, 返回 [ ]\b\b");
            scanf("%d", &w1);
            if (w1 == 2) menu();
        }
    } while (k == -1 && w1 == 1);
    if (k != -1)
    {
        printf("\n\n 确认删除?\n\t1). 是 2). 否 [ ]\b\b");
    }
}

```

```

        scanf("%d", &w2);
        switch (w2)
        {
            case 1:delete__(k, n);break;
            case 2:menu();break;
        }
    }
}

delete__(int k, int n)
{
    int i, w;
    for (i = k;i < n - 1;i++)
    {
        worker[i] = worker[i + 1];
    }
    save(n - 1);
    printf("删除成功! \n");
    system("pause");
    menu();
}

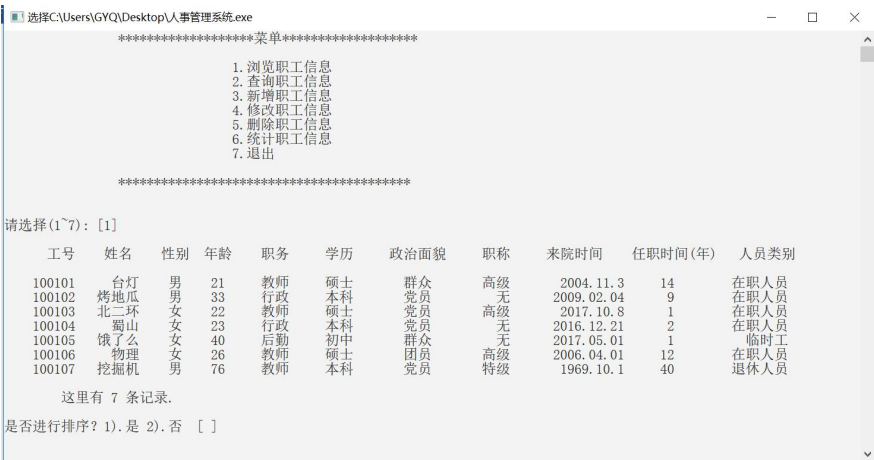
menu()
{
    int n, w1;
    do
    {
        system("cls");
        puts("\t\t*****菜单*****\n");
        puts("\t\t\t1. 浏览职工信息");
        puts("\t\t\t2. 查询职工信息");
        puts("\t\t\t3. 新增职工信息");
        puts("\t\t\t4. 修改职工信息");
        puts("\t\t\t5. 删除职工信息");
        puts("\t\t\t6. 统计职工信息");
        puts("\t\t\t7. 退出");
        puts("\n\t\t*****\n");
        printf("请选择(1~7): [ ]\b\b");
        scanf("%d", &n);
        if (n < 1 || n>7) //对选择的数字作判断
        {
            w1 = 1;getchar();
        }
        else w1 = 0;
    } while (w1 == 1);
    switch (n)
    {
        case 1:browse();break; //浏览
        case 2:search();break; //查询
        case 3:enter();break; //新增
        case 4:modify();break; //修改
        case 5:delete_file();break; //删除
        case 6:statistics();break; //统计
        case 7:exit(0); //退出
    }
}

main()
{
    menu();
}

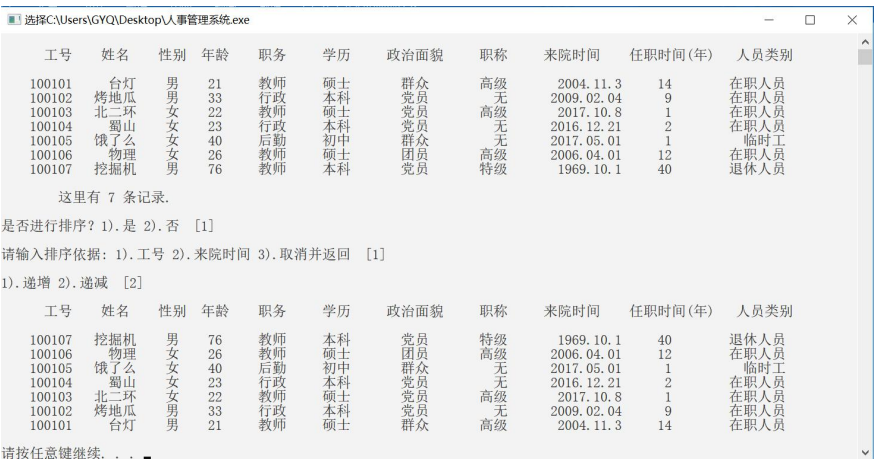
```

7. 程序运行结果

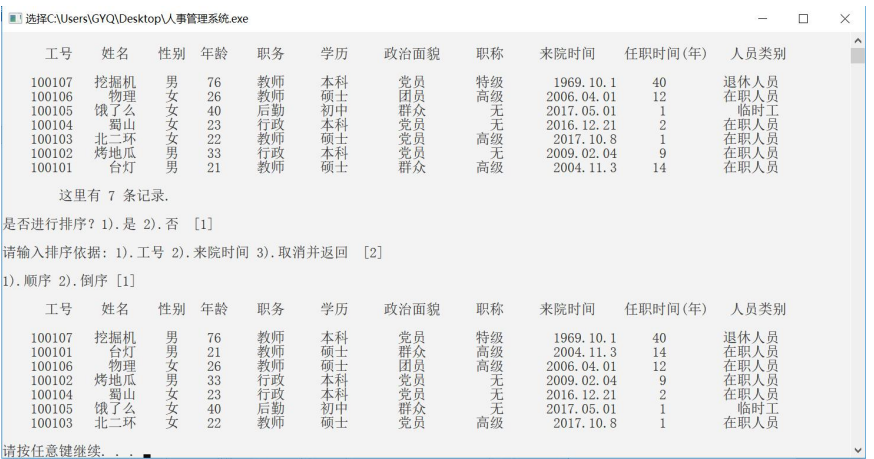
• 浏览模块:



• 按工号排序模块:



• 按来院时间排序模块:



• 按工号查找模块:

选择C:\Users\GYQ\Desktop\人事管理系统.exe

*****菜单*****

1. 浏览职工信息
2. 查询职工信息
3. 新增职工信息
4. 修改职工信息
5. 删除职工信息
6. 统计职工信息
7. 退出

请选择(1~7): [2]

请选择查找依据=>

1). 工号 2). 姓名 3). 取消并返回: [1]

输入要查询的工号:100103

工号	姓名	性别	年龄	职务	学历	政治面貌	职称	来院时间	任职时间(年)	人员类别
100103	北二环	女	22	教师	硕士	党员	高级	2017. 10. 8	1	在职人员

请选择下一步执行的操作:
1). 查找另一个 2). 修改 3). 返回菜单 []

• 按姓名查找模块:

选择C:\Users\GYQ\Desktop\人事管理系统.exe

*****菜单*****

1. 浏览职工信息
2. 查询职工信息
3. 新增职工信息
4. 修改职工信息
5. 删除职工信息
6. 统计职工信息
7. 退出

请选择(1~7): [2]

请选择查找依据=>

1). 工号 2). 姓名 3). 取消并返回: [2]

请输入你要查询的姓名:
挖掘机

工号	姓名	性别	年龄	职务	学历	政治面貌	职称	来院时间	任职时间(年)	人员类别
100107	挖掘机	男	76	教师	本科	党员	特级	1969. 10. 1	40	退休人员

请选择下一步执行的操作:
1). 查找另一个 2). 修改 3). 返回菜单 []

• 新增信息模块:

选择C:\Users\GYQ\Desktop\人事管理系统.exe

5. 删除职工信息
6. 统计职工信息
7. 退出

请选择(1~7): [3]

新增多少个职工信息(0-49):1

请输入职工信息

工号: 100108
姓名: 寿司
性别: 女
年龄: 21
职务: 教师
学历: 本科
政治面貌: 群众
职称: 高级
来院时间: 2000. 04. 06
任职时间(年): 18
人员类别: 在职人员

输入成功~`.

请选择要执行的操作:
1). 浏览全部 2). 返回: []

• 删除信息模块

选择C:\Users\GYQ\Desktop\人事管理系统.exe

5. 删除职工信息

6. 统计职工信息

7. 退出

请选择(1~7): [5]

工号	姓名	性别	年龄	职务	学历	政治面貌	职称	来院时间	任职时间(年)	人员类别
100101	台灯	男	21	教师	硕士	群众	高级	2004.11.3	14	在职人员
100102	烤地瓜	男	33	行政	本科	党员	无	2009.02.04	9	在职人员
100103	北二环	女	22	教师	硕士	党员	高级	2017.10.8	1	在职人员
100104	蜀山	女	23	行政	本科	党员	无	2016.12.21	2	在职人员
100105	饿了么	女	40	后勤	初中	群众	无	2017.05.01	1	临时工
100106	物理	女	26	教师	硕士	团员	高级	2006.04.01	12	在职人员
100107	挖掘机	男	76	教师	本科	党员	特级	1969.10.1	40	退休人员
100108	寿司	女	21	教师	本科	群众	高级	2000.04.06	18	在职人员

输入要删除的工号:100108

工号	姓名	性别	年龄	职务	学历	政治面貌	职称	来院时间	任职时间(年)	人员类别
100108	寿司	女	21	教师	本科	群众	高级	2000.04.06	18	在职人员

确认删除?

1). 是 2). 否 [1]

删除成功!

请按任意键继续. . .

• 修改信息模块:

选择C:\Users\GYQ\Desktop\人事管理系统.exe

请选择(1~7): [4]

工号	姓名	性别	年龄	职务	学历	政治面貌	职称	来院时间	任职时间(年)	人员类别
100101	台灯	男	21	教师	硕士	群众	高级	2004.11.3	14	在职人员
100102	烤地瓜	男	33	行政	本科	党员	无	2009.02.04	9	在职人员
100103	北二环	女	22	教师	硕士	党员	高级	2017.10.8	1	在职人员
100104	蜀山	女	23	行政	本科	党员	无	2016.12.21	2	在职人员
100105	饿了么	女	40	后勤	初中	群众	无	2017.05.01	1	临时工
100106	物理	女	26	教师	硕士	团员	高级	2006.04.01	12	在职人员
100107	挖掘机	男	76	教师	本科	党员	特级	1969.10.1	40	退休人员

输入要修改的职工工号:100102

工号	姓名	性别	年龄	职务	学历	政治面貌	职称	来院时间	任职时间(年)	人员类别
100102	烤地瓜	男	33	行政	本科	党员	无	2009.02.04	9	在职人员

请选择=>

1). 工号 2). 姓名 3). 性别 4). 年龄 5). 职务 6). 学历 7). 政治面貌 8). 职称 9). 来院时间 10) 任职时间. 11). 人员类别 0). 取消

选择:[4]

年龄:42

修改完成:

工号	姓名	性别	年龄	职务	学历	政治面貌	职称	来院时间	任职时间(年)	人员类别
100102	烤地瓜	男	42	行政	本科	党员	无	2009.02.04	9	在职人员

确定?

1). 是 2). 否, 重新修改 3). 不保存退出 [1]

修改成功_".

是否修改另一个?

1). 是 2). 否, 保存退出 []

• 统计党员人数:

选择C:\Users\GYQ\Desktop\人事管理系统.exe

*****菜单*****

1. 浏览职工信息

2. 查询职工信息

3. 新增职工信息

4. 修改职工信息

5. 删除职工信息

6. 统计职工信息

7. 退出

请选择(1~7): [6]

请选择统计依据=>

1). 政治面貌 2). 性别 3). 学历 4). 职称 5). 人员类别 6). 取消并返回: [1]

输入要统计的政治面貌:党员

工号	姓名	性别	年龄	职务	学历	政治面貌	职称	来院时间	任职时间(年)	人员类别
100102	烤地瓜	男	42	行政	本科	党员	无	2009.02.04	9	在职人员
100103	北二环	女	22	教师	硕士	党员	高级	2017.10.8	1	在职人员
100104	蜀山	女	23	行政	本科	党员	无	2016.12.21	2	在职人员
100107	挖掘机	男	76	教师	本科	党员	特级	1969.10.1	40	退休人员

检索到4条信息

请选择下一步执行的操作:

1). 统计另一项 2). 修改 3). 返回菜单 []

• 统计在职人数

选择C:\Users\GYQ\Desktop\人事管理系统.exe

*****菜单*****

1. 浏览职工信息
2. 查询职工信息
3. 新增职工信息
4. 修改职工信息
5. 删除职工信息
6. 统计职工信息
7. 退出

请选择(1~7): [6]

请选择统计依据=>

1). 政治面貌 2). 性别 3). 学历 4). 职称 5). 人员类别 6). 取消并返回: [5]

请输入人员类别: 在职人员

工号	姓名	性别	年龄	职务	学历	政治面貌	职称	来院时间	任职时间(年)	人员类别
100101	台灯	男	21	教师	硕士	群众	高级	2004. 11. 3	14	在职人员
100102	烤地瓜	男	42	行政	本科	党员	无	2009. 02. 04	9	在职人员
100103	北二环	女	22	教师	硕士	党员	高级	2017. 10. 8	1	在职人员
100104	蜀山	女	23	行政	本科	党员	无	2016. 12. 21	2	在职人员
100106	物理	女	26	教师	硕士	团员	高级	2006. 04. 01	12	在职人员

检索到5条信息

请选择下一步执行的操作:

1). 统计另一项 2). 修改 3). 返回菜单 []

• 统计女工人数

选择C:\Users\GYQ\Desktop\人事管理系统.exe

*****菜单*****

1. 浏览职工信息
2. 查询职工信息
3. 新增职工信息
4. 修改职工信息
5. 删除职工信息
6. 统计职工信息
7. 退出

请选择(1~7): [6]

请选择统计依据=>

1). 政治面貌 2). 性别 3). 学历 4). 职称 5). 人员类别 6). 取消并返回: [2]

请输入选择的性别: 女

工号	姓名	性别	年龄	职务	学历	政治面貌	职称	来院时间	任职时间(年)	人员类别
100103	北二环	女	22	教师	硕士	党员	高级	2017. 10. 8	1	在职人员
100104	蜀山	女	23	行政	本科	党员	无	2016. 12. 21	2	在职人员
100105	饿了么	女	40	后勤	初中	群众	无	2017. 05. 01	1	临时工
100106	物理	女	26	教师	硕士	团员	高级	2006. 04. 01	12	在职人员

检索到4条信息

请选择下一步执行的操作:

1). 统计另一项 2). 修改 3). 返回菜单 []

• 统计高学历人数

选择C:\Users\GYQ\Desktop\人事管理系统.exe

*****菜单*****

1. 浏览职工信息
2. 查询职工信息
3. 新增职工信息
4. 修改职工信息
5. 删除职工信息
6. 统计职工信息
7. 退出

请选择(1~7): [6]

请选择统计依据=>

1). 政治面貌 2). 性别 3). 学历 4). 职称 5). 人员类别 6). 取消并返回: [3]

请输入学历: 硕士

工号	姓名	性别	年龄	职务	学历	政治面貌	职称	来院时间	任职时间(年)	人员类别
100101	台灯	男	21	教师	硕士	群众	高级	2004. 11. 3	14	在职人员
100103	北二环	女	22	教师	硕士	党员	高级	2017. 10. 8	1	在职人员
100106	物理	女	26	教师	硕士	团员	高级	2006. 04. 01	12	在职人员

检索到3条信息

请选择下一步执行的操作:

1). 统计另一项 2). 修改 3). 返回菜单 []

8. 编程中遇到的困难及解决方法

已解决：

1. 定义结构体数组时，各种不同的数据类型导致后期输入和处理的混乱。

解决方法：统一定义为 `char` 型，包括纯数字的数据（例如工号，年龄）。输入时统一以字符串进行输入，比较时统一用 `strcmp` 函数进行对比。

2. 编写排序模块时虽然显示出了排序后的状态，但未保存到磁盘中。

解决方法：查阅资料后得知，打开文件是将磁盘中的文件输入到文件缓冲区，进行编辑后仍然在文件缓冲区中。编写 `save()` 函数，将编辑后的文件以“只写”的方式输入到原磁盘文件，覆盖旧文件的内容，保存成功。

3. 编写“新增”模块时，发现下一次输入会覆盖旧内容，而不是在后面添加。

解决方法：查阅课本，找到了“只写”和“追加”的区别，在 `save()` 函数的基础上新编写 `add()` 函数，以“追加”的方式输入到原文件。

未解决：

1. 统计模块繁复且占用篇幅过多。虽然每个统计模块之间的差异不大，由于关键变量（例如 `worker[i].political`）中含有循环变量 `i`，未能将 5 个统计模块做到进一步统一的模块化。
2. 在统计高学历人数时，只能分别查不同的学历，而不能一次性统计。

9. 总结心得及良好建议

越厉害的程序员写出来的代码越简洁……显然我还不是，这个程序还有些繁琐。在编写这个程序的过程中，我得以充分理解函数的嵌套调用、文件的读写、以及自定义结构体数组的使用。程序中声明了相当多的函数，它们之间层层调用，有着密不可分的联系。

设计编写一个程序，从头开始可能毫无头绪，但从简单到复杂，从一个函数写到多个函数，从杂乱的显示界面到排版后整洁的界面，这些都离不开之前的练习和对程序的理解。编写时我也发现了 `do-while` 结构更多的用法，我使用 `do-while` 取代了之前常用的 `if + goto` 进行的判断并循环。

出于程序的易读性，整个程序中几乎没有用到指针。全部编写完成后，考虑再把指针加进去，结果出现 bug，时间不允许再仔细修改，还是放弃了。说起易读性，在编写这样八百多行的程序时，还是少不了必要的注释的，没有注释很容易打乱思路。

虽然还有几个小问题没有解决，我对这个程序还比较满意。当然，如果能解决程序繁杂的问题就更好了。日后进行类似程序设计时务必要从简至繁，尽可能地多用模块化函数和指针以提高效率。

10. 致谢

在此致谢：

陈喆 老师

樊俊青 老师

以及每位同学和学长的支持和帮助