

# 中南大學

## 高級程序設計實驗報告

學生姓名	孔德彬
學生學號	8208181404
指導教師	楊希
專業班級	計科 1801
完成日期	2019.12.16

計算機學院

## 目录

实验内容 .....	3
代码说明 .....	4
类与函数功能 .....	4
main .....	4
print.....	4
function .....	5
employee.....	8
运行截图 .....	11
心得体会 .....	17

## 1 实验内容

设计一个职工信息管理案例，实现不同类别职工的工资数据。职工的种类包括：正式职工和临时工。定义一个基本的职工类，基本信息包括：编号、姓名、性别、年龄、家庭住址、基本职务工资。派生出正式职工类，增加的属性包括：岗位津贴、养老金、住房公积金、所得税、医疗保险、实发工资；临时工增加的属性包括：奖金、所得税、实发工资。

1. 正式职工：实发工资=基本职务工资+岗位津贴-养老金-住房公积金-所得税-医疗保险-所得税；

2. 临时工：实发工资=基本职务工资+奖金-所得税；

### 基本功能与技术要求

1、数据文件管理功能：创建新文件、打开文件、保存文件。

2、计算与统计功能：完成记录中相关数据的统计；

1) 能够打开已存在的数据表文件，对数据表中与某关键字匹配的相关记录进行数据统计（包括：工资总额、平均值）

2) 计算每个职工的实发工资，排序输出。

3、记录管理功能：记录的管理采用链表或者指针数组实现。

1) 输入 / 添加/插入记录，并保存到数据文件中。

2) 查询、显示记录，根据用户要求按所给编号、姓名、工资等记录关键字排序，查询一个或多个相关联记录的各项信息。

3) 修改记录：能够打开已存在的数据表文件，并对确定的任意记录进行修改，在修改过程中，应显示记录信息，给出确认提示，并对更新的记录信息进行文件保存。

4) 删除记录：能够打开已存在的数据表文件，可以删除数据表中的任一记录，要求具备逻辑删除（具有恢复功能）和物理删除功能，并对新的数据表信息进行文件保存。

## 2 代码说明

### 2.1 类与函数功能

**main:**

主要实现功能选择, 运用 flag 变量, 用户输入, 进入 switch 选择对应功能, 在程序的开始和结束及时读取和写入文件, 及时更新用户信息。

**print:**

主要负责打印出不同功能下所要对应的界面, 有以下函数:

打印选择功能界面:

```
void printFunction();
```

打印功能一对应界面:

```
void printFunction1();
```

打印功能一用户选择的数据:

```
void printFunction1(listpoint1 *head, int i);
```

```
void printFunction1(listpoint2 *head, int i);
```

打印功能二对应界面:

```
void printFunction2(listpoint1 *head1, listpoint2 *head2);
```

打印功能三对应界面:

```
void printFunction3();
```

打印功能四选择界面, 以及需要操作的员工类型界面:

```
void printFunction4(string s);
```

打印功能四选择查询方式界面：

```
void printFunction4();
```

打印功能五选择更改信息界面：

```
void printFunction5(listpoint1 *head);
```

```
void printFunction5(listpoint2 *head);
```

打印功能六选择删除方式界面：

```
void printFunction6();
```

**function:**

主要负责对员工进行操作的一些函数，其中的函数有以下几个：

建立正式员工对应链表：

```
typedef struct listpoint1 {  
    fullTimeEmployee *e;  
    bool fireEmployee = false;  
    struct listpoint1 *next;  
    struct listpoint1 *last;  
}listpoint1;
```

建立临时员工对应链表：

```
typedef struct listpoint2 {  
    temporaryEmployee *e;  
    bool fireEmployee = false;  
    struct listpoint2 *next;  
    struct listpoint2 *last;  
}listpoint2;
```

增加正式员工和临时员工：

```
void addEmployee(listpoint1 *head);  
void addEmployee(listpoint2 *head);
```

写入文件：

```
void writeCsv(fullTimeEmployee e);  
void writeCsv(temporaryEmployee e);
```

读取csv文件：

```
void readCsv(listpoint1 *temp);  
void readCsv(listpoint2 *temp);
```

对员工进行排序：

```
void update(listpoint1 *temp);  
void update(listpoint2 *temp);
```

输出员工总工资：

```
void printWages(listpoint1 *temp, int i);  
void printWages(listpoint2 *temp, int i);
```

打印单个员工工资：

```
void printEmployee(listpoint1 *temp);  
void printEmployee(listpoint2 *temp);
```

打印即将被操作的员工全部信息：

```
void print(listpoint1 *flag);  
void print(listpoint2 *flag);
```

打印全部员工信息：

```
void printEmployees(listpoint1 *head);
```

```
void printEmployees(listpoint2 *head);
```

按不同方式查找员工：

```
listpoint1* findEmployee(listpoint1 *head, int i);
```

```
listpoint2* findEmployee(listpoint2 *head, int i);
```

打印全部查询到的正式员工或临时员工：

```
void printEmployee(int i, int j, listpoint1 *head1, listpoint2  
*head2);
```

更改员工信息：

```
void changeEmployee(int i, listpoint1 *head1, listpoint2 *head2);
```

删除员工：

```
void deleteEmployee(int i, int j, listpoint1 *head1, listpoint2  
*head2);
```

保存员工信息：

```
void saveEmployee(listpoint1 *head);
```

```
void saveEmployee(listpoint2 *head);
```

**employee:**

主要员工类，包含了员工的一些基本信息，：

普通员工类：

```
class employee {
private:
    int num = 0;           //职工编号
    string name = "";      //职工姓名
    string gender = "";    //性别
    int age = 0;           //年龄
    string address = "";   //家庭住址
    int wages = 0;         //基本职务工资
public:
    void setNum(int _num);
    void setName(string _name);
    void setGender(string _gender);
    void setAge(int _age);
    void setAddress(string _address);
    void setWages(int _wages);
    void set();
    int getNum();
    string getName();
    string getGender();
    int getAge();
    string getAddress();
    int getWages();
    employee();
};
```



派生出正式员工类:

```
class fullTimeEmployee : public employee {
private:
    int moreMoney1 = 0; //岗位津贴
    int moreMoney2 = 0; //养老金
    int moreMoney3 = 0; //住房公积金
    int moreMoney4 = 0; //所得税
    int moreMoney5 = 0; //医疗保险
    int sumMoney = 0; //实发工资
public:
    void setMoney1(int _moreMoney1);
    void setMoney2(int _moreMoney2);
    void setMoney3(int _moreMoney3);
    void setMoney4(int _moreMoney4);
    void setMoney5(int _moreMoney5);
    void setSumMoney();
    void set();
    int getMoney1();
    int getMoney2();
    int getMoney3();
    int getMoney4();
    int getMoney5();
    int getSumMoney();
    fullTimeEmployee();
    fullTimeEmployee(int i);
};
```

派生出临时员工类:

```
class temporaryEmployee : public employee {
```

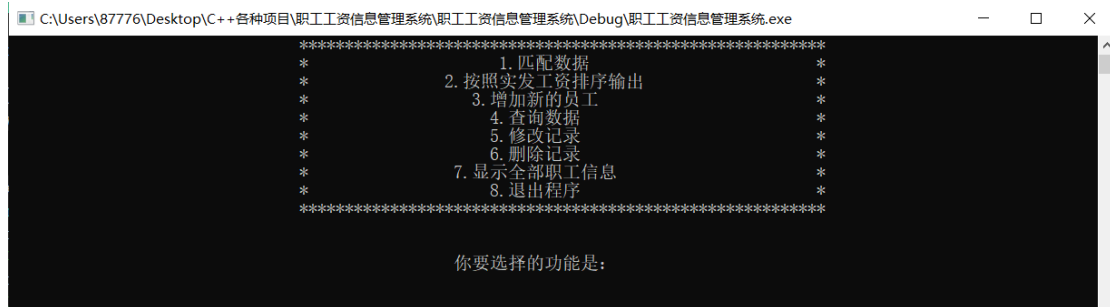
```

    int moreMoney1 = 0;    //奖金
    int moreMonry2 = 0;    //所得税
    int sumMoney = 0;      //实发工资
public:
    void setMoney1(int _moreMoney1);
    void setMoney2(int _moreMoney2);
    void setSumMoney();
    void set();
    int getMoney1();
    int getMoney2();
    int getSumMoney();
    temporaryEmployee();
    temporaryEmployee(int i);
};

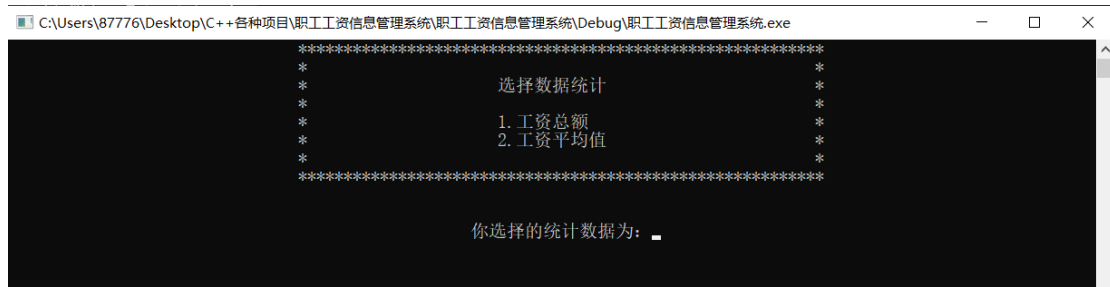
```

## 2.2 运行截图

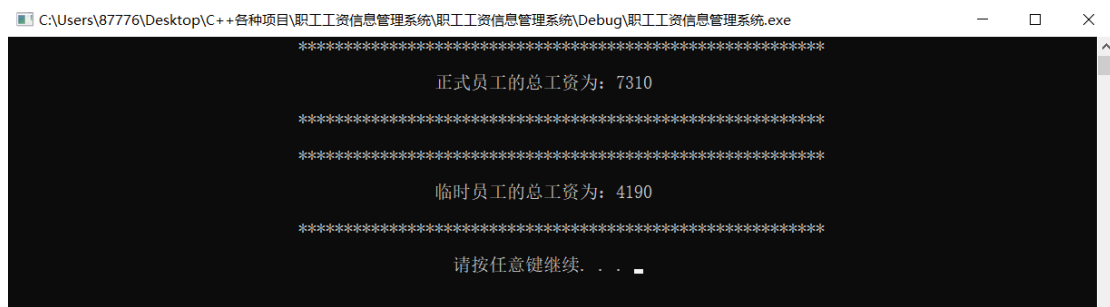
开始界面：



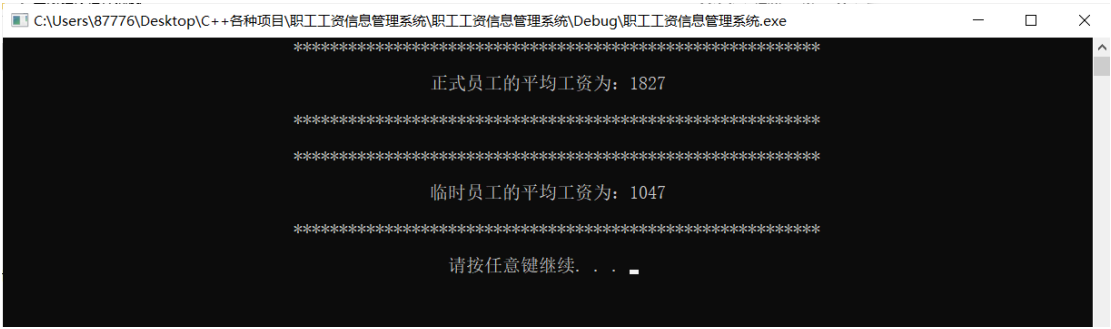
功能一功能选择界面：



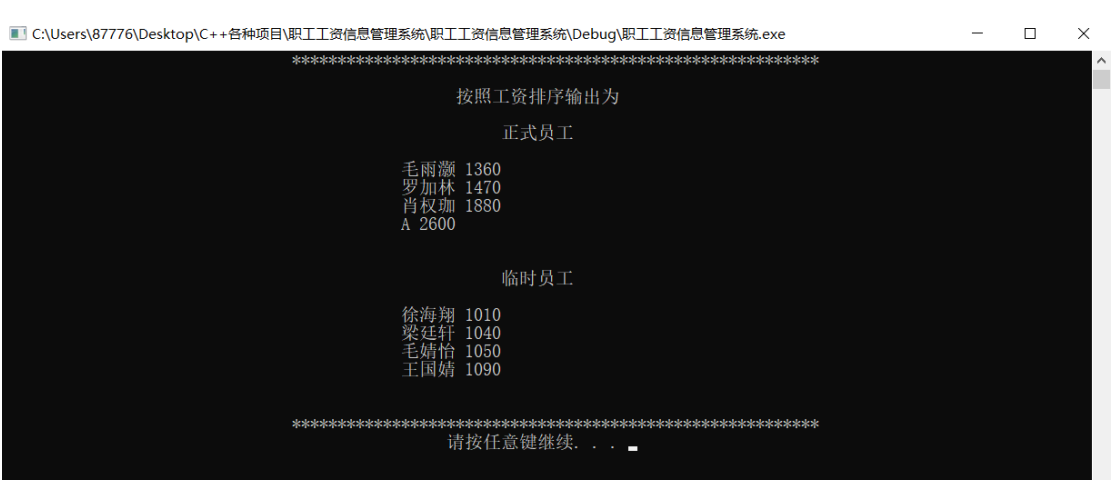
功能一工资总额输出界面：



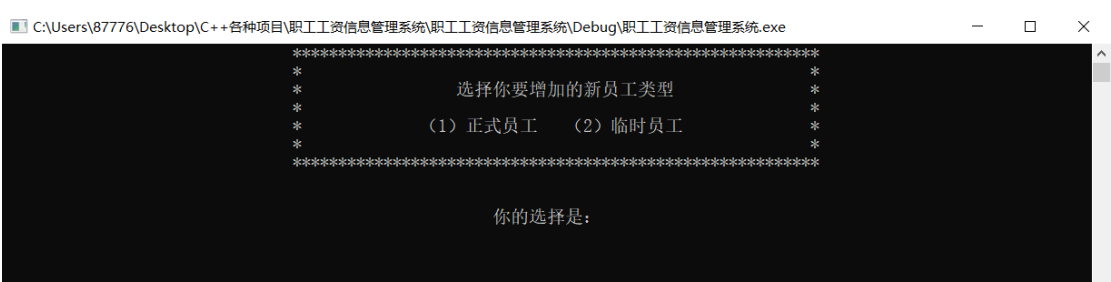
功能一平均工资输出界面：



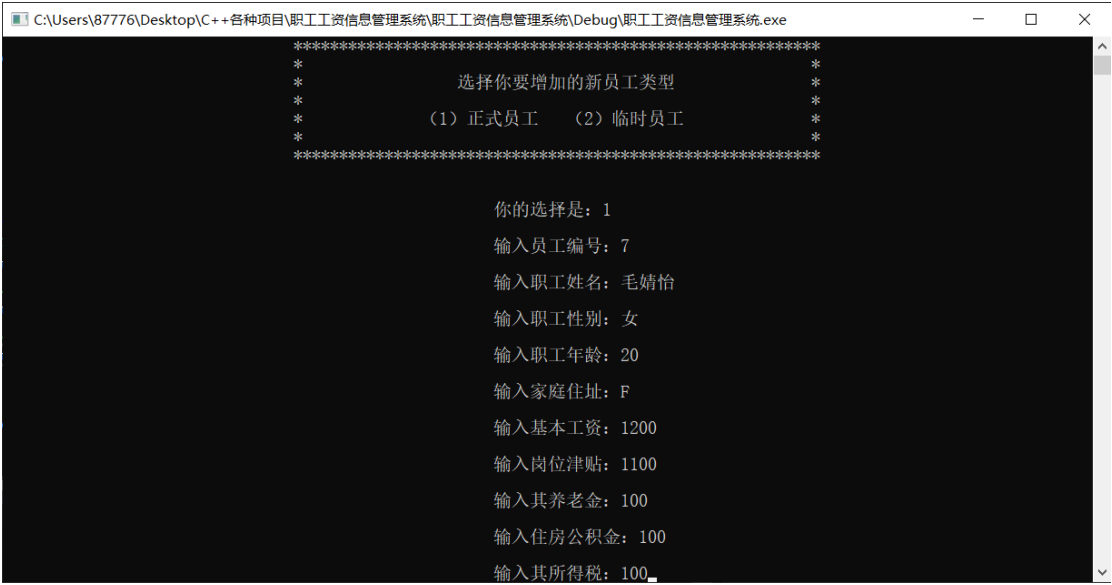
功能二输出界面：



功能三选择截图：



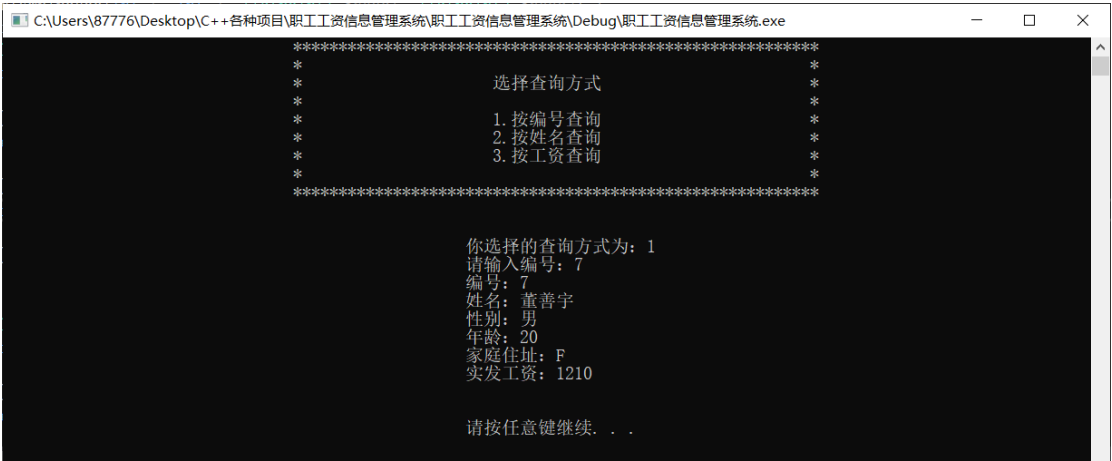
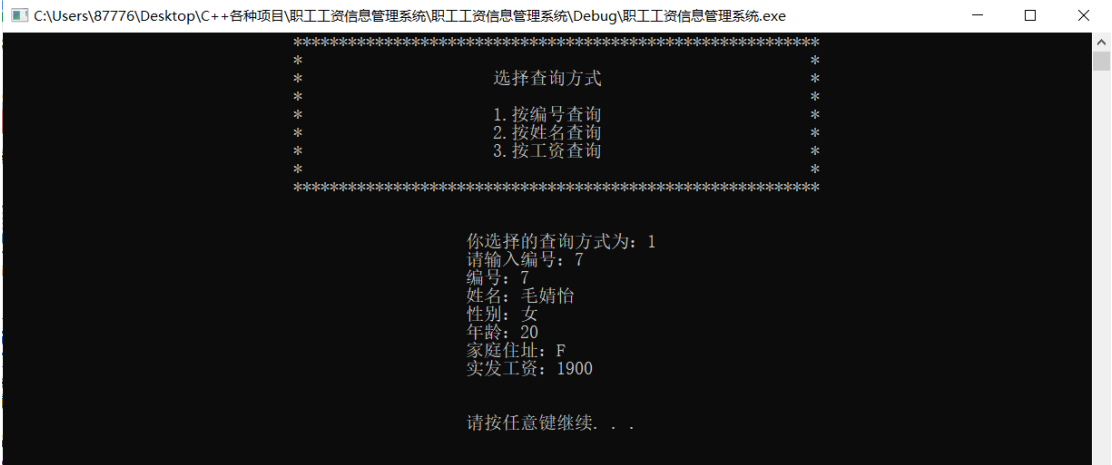
增加正式员工：



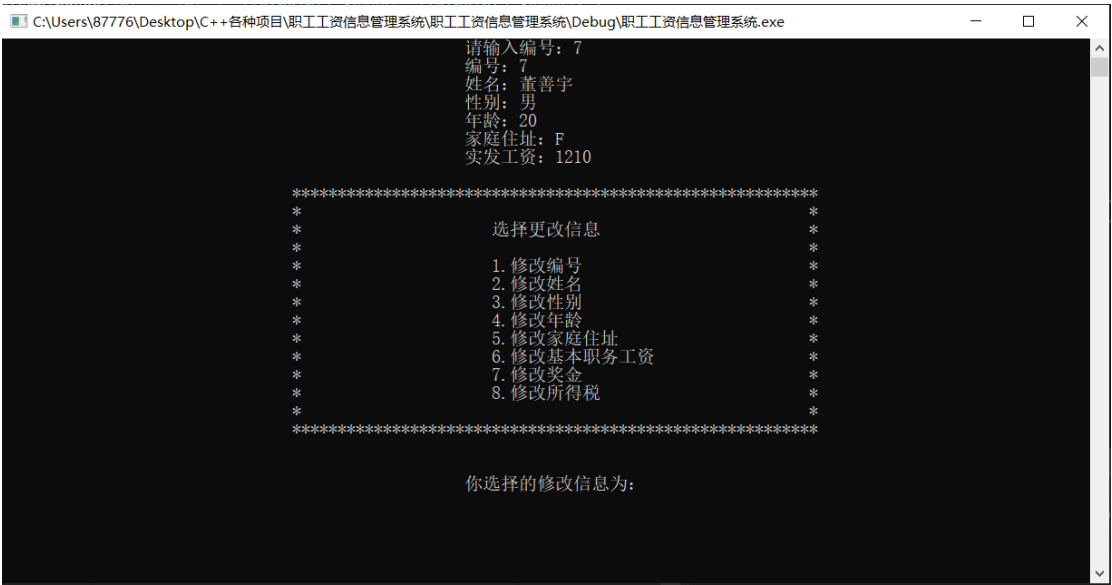
增加临时员工：



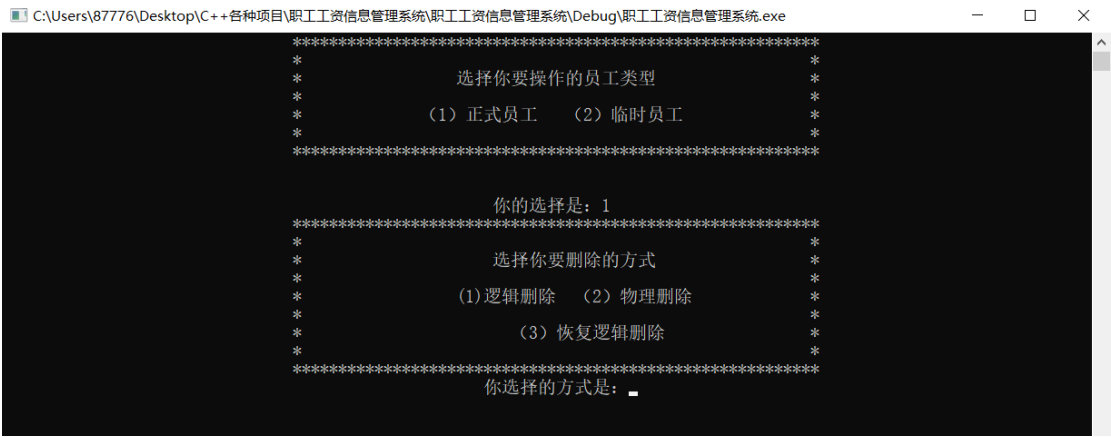
功能四查询数据（例 按编号查询）：



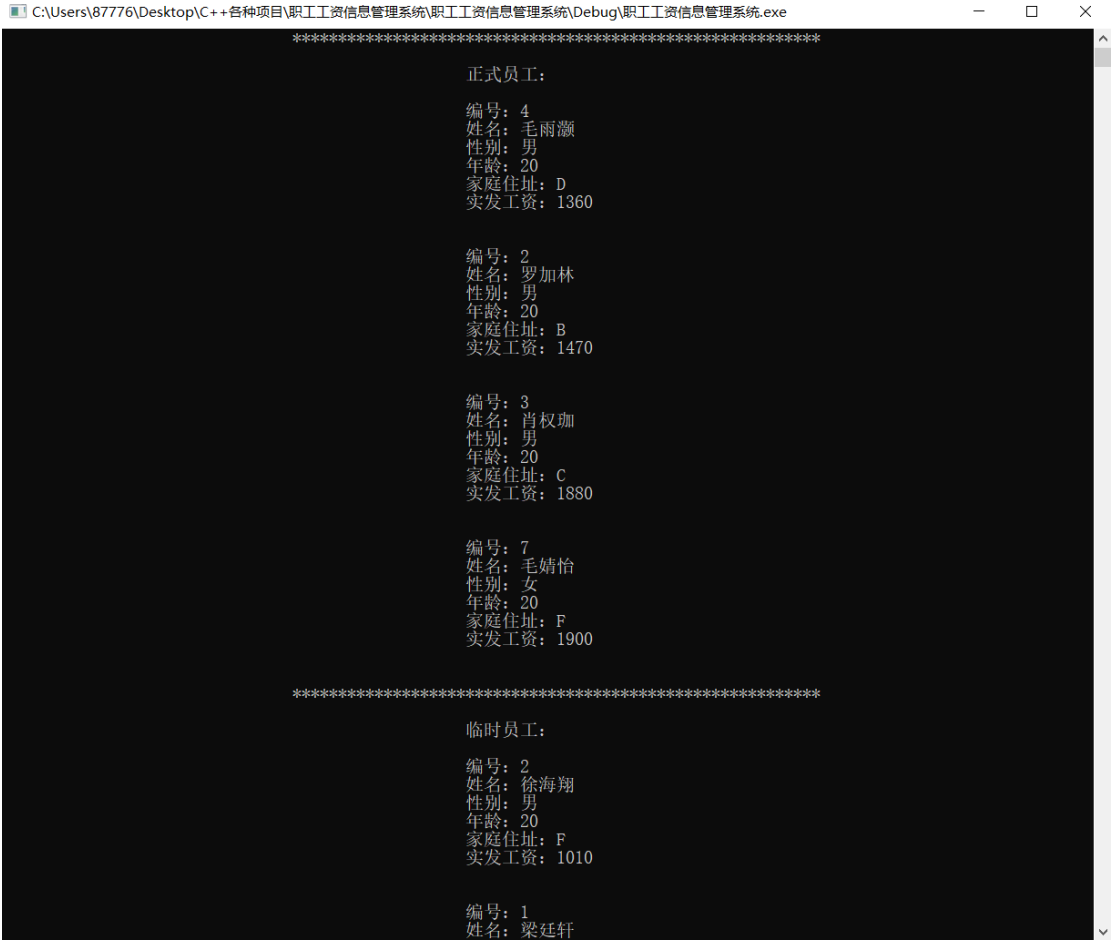
功能五修改数据：



功能六删除选择删除方式:



功能七显示所有员工:





### 3 心得体会

通过此次高级程序设计的代码编写,让我更加熟悉了用 C++语言完成一个程序的过程,对我以后的代码编写,起着非常重要的作用,让我对编程的过程有了更深刻的理解。

在代码编写的过程中,我也遇到了很多困难,也少不了查找和修改 bug 的过程,其中困扰我最长时间的 bug 是当时我在给链表分配空间的时候,大小分配成了链表指针,导致在运行程序的时候总是玄学报错,在单步调试的时候也查不出来,但是最后通过在网上查找资料 and 找同学帮助,才发现给链表分配内存的时候出错。还有一些 bug 比如忘记给链表的 next 指针设为 null,导致在循环最后报错等等。

在检查的过程中,老师也给我提了一些很有帮助的意见,站在客户的角度来让我修改我的程序,我觉得这对我来说是很重要的。对于我们现在编程来说,可能有一些点会注意不到,还没有养成站在客户角度想问题的习惯。比如在修改员工信息的时候,老师会告诉我应该先输出他要被修改的信息,还有比如在删除员工的时候,应该先输出所有员工的信息来供客户修改。还有一些程序运行的效率问题,就我现在而言,只是想着实现它,而很少考虑它的效率问题,比如在排序的过程中,我第一时间想到的是冒泡排序,而没有根据具体的情况来选择排序的方式,再比如老师提醒我横向输出比竖向输出的效率会高。以上的种种可能都是我不太会注意的点,老师能给我指出,这对我以后的程序编写会有更重要的作用,会让我考虑事情考虑的更加的全面。