



哈尔滨工业大学 (深圳)
HARBIN INSTITUTE OF TECHNOLOGY

实验报告

开课学期: 2022 秋季
课程名称: 高级语言程序设计
实验名称: 记账系统 2.0 实验
实验性质: 设计型
实验学时: 4 地点: T2- 608
学生班级: 计科 7 班
学生学号: 220110713
学生姓名: 周健毅
评阅教师:
报告成绩:

实验与创新实践教育中心制

2022 年 10 月

一、项目名称

《记账系统 2.0》

二、系统设计

1. 描述选择使用的数据结构（例如数组、结构体等）；

结构体:定义了一个 **People** 类型的结构体,并用 **typedef** 将其命名为 **P**,该结构体中包含用户的 **Id**,名字,收入和支出的信息。其中用户的 **Id** 和名字用字符数组来存储。这里有个关键点,为什么用字符数组来存储而不用整形来存储 **Id** 呢? 因为,当 **Id** 的首位为 **0** 时,整形类型的数据会忽略这个 **0**,造成储存的 **Id** 信息和原本的 **Id** 信息不同。而字符数组就可以保存下首位的 **0**,从而保证数据不会丢失。

```
typedef struct People//定义一个结构体,包含用户的id,姓名,收入,支出的信息
{
    char Id[6];
    char name[11];
    int income;
    int expense;
}P;
```

数组:

- ①: 结构体内的字符型数组 `char Id []`
- ②: 结构体中的字符型数组 `char name []`
- ③: 结构体数组: `P peoplearr`, 该数组用来存放各个用户的信息

`P peoplearr[10];`//定义一个结构体数组,储存所有用户的信息

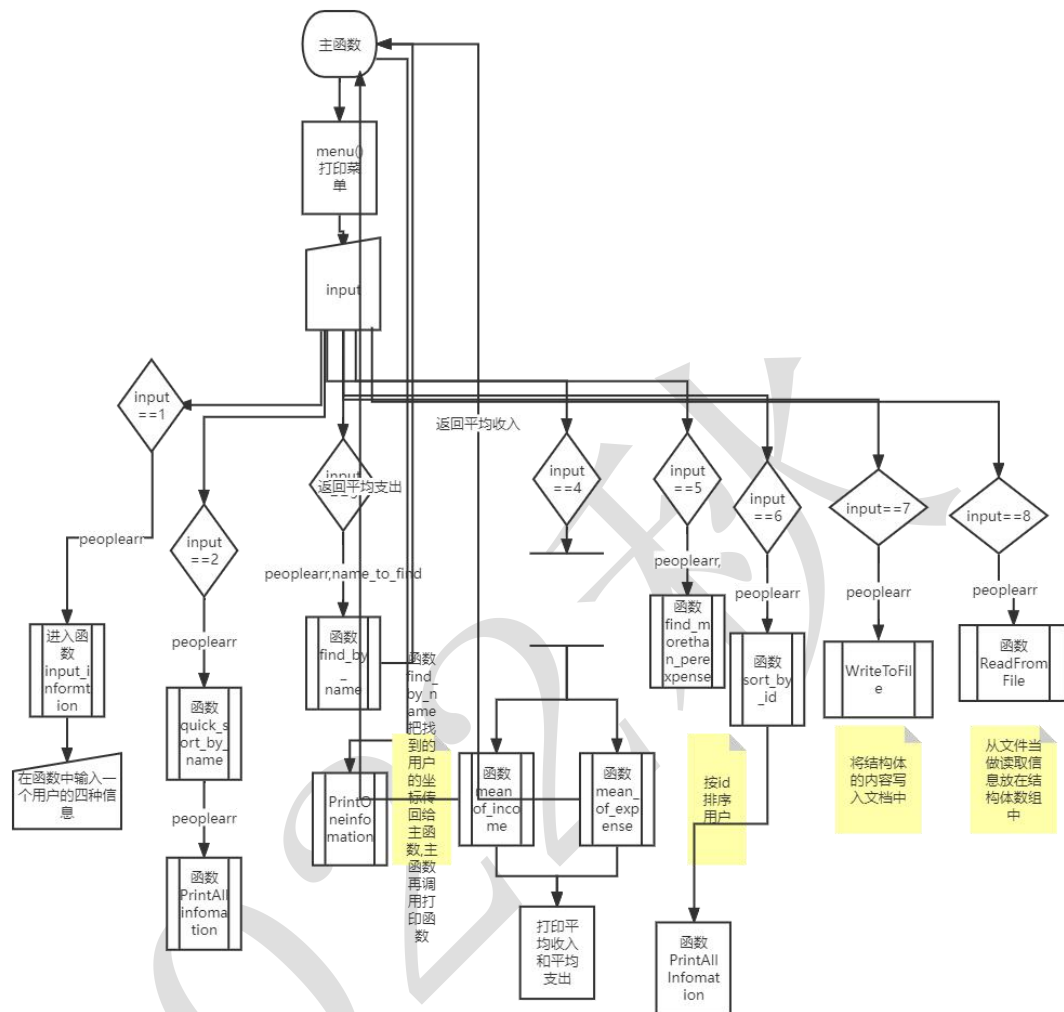
④一个字符型数组 `char name_to_find`, 该数组用来存放要搜索的用户姓名(后面有函数需要实现该功能)

- ⑤
- ⑥

2. 总体结构设计: **main** 函数与各子函数之间的调用和返回关系,程序运行流程等,可以用流程图方式描述。

```
void menu(); //打印菜单,提醒用户进行选择
int find_by_name(char* name, P peo[]); //按名字找到用户
double mean_of_income(P peo[]); //计算平均收入
double mean_of_expend(P peo[]); //计算平均支出
void find_morethan_perexpense(P peo[]); //找出支出大于平均的人
void input_information(P peo[10]); //输入用户信息
void PrintAllInformation(P peo[]); //打印所有信息
void PrintoneInformation(P peo[], int position); //打印某一个人的信息
void sort_by_id(P peo[]); //按id来排序用户
int sum_of_peo; //定义一个全局变量,代表所有用户人数
void quick_sort_by_name(P peo[], int L, int R); //用快速排序,排序用户名
void WriteToFile(P peo[]); //将信息读写到文件中
void ReadFromFile(P peo[]); //从文件中读取信息
```

以上为该实验的函数声明,包含了程序需要实现的所有功能
主函数到子函数的关系,以及各子函数之间的关系如下图所示:



整体上是一个多路选择的 switch 结构,有用户输入数字来选择要执行哪个函数。

三、 函数设计

1. 函数接口的定义: 函数原型定义、参数及返回值含义;

```
void menu(); //打印菜单,提醒用户进行选择
int find_by_name(char* name, P peo[]); //按名字找到用户
double mean_of_income(P peo[]); //计算平均收入
double mean_of_expense(P peo[]); //计算平均支出
void find_morethan_perexpense(P peo[]); //找出支出大于平均的人
void input_information(P peo[10]); //输入用户信息
void PrintAllInformation(P peo[]); //打印所有信息
void PrintoneInformation(P peo[], int position); //打印某一个人的信息
void sort_by_id(P peo[]); //按id来排序用户
int sum_of_peo; //定义一个全局变量,代表所有用户人数
void quick_sort_by_name(P peo[], int L, int R); //用快速排序,排序用户名
void WriteToFile(P peo[]); //将信息读写到文件中
void ReadFromFile(P peo[]); //从文件中读取信息
```

以上为我在程序中对各个函数的声明,共 12 个函数,下面对各个函数的定义原型,参数和返回类型以及返回值法的含义进行说明:

(由于 sum_of_peo 这个变量几乎在每个函数中都会使用到,所以我把它定义为全局变量)

①menu 函数

```
//函数功能:打印菜单,让用户知道各个数字代表的含义
void menu()
{
    printf(_Format: "1.Input record\n");
    printf(_Format: "2.Sort and list records in order by user name\n");
    printf(_Format: "3.Search records by user name\n");
    printf(_Format: "4.Calculate and list per capita income and expenses\n");
    printf(_Format: "5.List records which have more expense than per capita expenses\n");
    printf(_Format: "6.List all records\n");
    printf(_Format: "7.Write to file\n");
    printf(_Format: "8.Read from file\n");
    printf(_Format: "0.Exit\n");
    printf(_Format: " Please enter your choice:\n");
}
```

该函数非常简单,无接口参数和返回值,返回类型为 void

②input_information 函数

```
//函数功能:输入用户信息 入口:结构体数组的地址
void input_information(P peo[])
{
    int i = sum_of_peo, ret;
    printf(_Format: "Please input the Id,name,income and expense of user %d\n", sum_of_peo + 1);
    while (ret = scanf(_Format: "%s%d%d", &peo[i].Id, &peo[i].name, &peo[i].income, &peo[i].expense) < 4)
    {
        printf(_Format: "There are some erro of your input,please input again\n");
        fflush(_Stream: stdin);
    }
    sum_of_peo++;
}
```

该函数传入结构体数组的地址,返回类型为 void

③quick_sort_by_name 函数

```
//函数功能:按名字顺序排序 入口:结构体数组的名字,左端点,右端点
void quick_sort_by_name(P peo[], int L, int R)
{
    P pivot;
    int left = L, right = R;
    if (left >= right)
        return;
    pivot = peo[left];
    while (left < right)
    {
        while (left < right && strcmp(_Str1:peo[right].name, _Str2:pivot.name) > 0)
        {
            right--;
        }
        if (left < right && strcmp(_Str1:peo[right].name, _Str2:pivot.name) < 0)
        {
            peo[left] = peo[right];
        }
        while (left < right && strcmp(_Str1:peo[left].name, _Str2:pivot.name) < 0)
        {
            left++;
        }
        if (left < right && strcmp(_Str1:peo[left].name, _Str2:pivot.name) > 0)
        {
            peo[right] = peo[left];
        }
    }
    peo[right] = pivot;
    quick_sort_by_name(peo, L, left - 1);
    quick_sort_by_name(peo, left + 1, R);
}
```

该函数返回类型为 **void**,接口参数为结构体数组的地址,快速排序中左右端点的数值

④find_by_name 函数

```
//函数功能:按名字搜索用户 入口:用户名, 结构体数组名 返回值:存在时该用户的位置下标,不存在返回-1
int find_by_name(char* name, P peo[])
{
    quick_sort_by_name(peo, L:0, R:sum_of_peo - 1);
    //先把名字逆序排序,再二分查找
    int left = 0, right = sum_of_peo - 1, mid;
    while (left <= right)
    {
        mid = (left + right) / 2;
        if (strcmp(_Str1:name, _Str2:peo[mid].name) < 0)
            left = mid + 1;
        else if (strcmp(_Str1:name, _Str2:peo[mid].name) > 0)
            right = mid - 1;
        else if (strcmp(_Str1:name, _Str2:peo[mid].name) == 0)
            return mid;
    }
    return -1;
}
```

⑤PrintoneInformation 函数

```
//函数功能:打印某个用户的信息 函数入口:结构体数组,用户下标的位置
void PrintoneInformation(P peo[], int position)
{
    printf(_Format: "%s\t%s\t%d\t%d\n", peo[position].Id, peo[position].name, peo[position].income, peo[position].expense);
}
```

该函数返回类型为 **void**

⑥PrintAllInformation 函数

```
//函数功能:打印所有用户的信息 函数入口:结构体数组名
void PrintAllInformation(P peo[])
{
    printf(_Format: "ID\tUserName\tIncome\tExpenses\n");
    int i;
    for (i = 0; i < sum_of_peo; i++)
    {
        PrintoneInformation(peo, position:i);
    }
}
```

⑦和⑧

两个函数的返回类型分别代表平均收入和平均支出,由于是平均值,所以返回类型为 **double 型**

```
//函数功能:计算所有用户的平均支出 函数入口参数:结构体数组名
double mean_of_expend(P peo[])
{
    double sum_of_expense = 0;
    int i;
    for (i = 0; i < sum_of_peo; i++)
    {
        sum_of_expense += peo[i].expense;
    }
    return sum_of_expense / sum_of_peo;
}

//函数功能:计算用户的平均收入 函数入口参数:结构体数组名 返回平均收入的值
double mean_of_income(P peo[])
{
    double sum_of_income = 0;
    int i;
    for (i = 0; i < sum_of_peo; i++)
    {
        sum_of_income += peo[i].income;
    }
    return sum_of_income / sum_of_peo;
}
```


两个函数相类似,都是传入结构体数组的地址(即结构体数组名),然后计算收入,支出的平均值

⑨find_morethan_perexpense 函数

接口参数:函数体数组的地址 返回类型为 void

```
//函数功能:打印找到所有大于平均收入的用户的信息
void find_morethan_perexpense(P peo[])
{
    int i, j = 0;
    double perexpense = mean_of_expend(peo);
    for (i = 0; i < sum_of_peo; i++)
    {
        if (peo[i].expense > perexpense)
            PrintoneInformation(peo, position: i);
    }
}
```

⑩sort_by_id 函数

//函数功能:按学号进行选择排序

//参数接口:结构体数组地址

//返回类型为void

```
void sort_by_id(P peo[])
{
    int i, j, k;
    P temp;
    int n = sum_of_peo;
    for (i = 0; i < n - 1; i++)
    {
        k = i;
        for (j = i + 1; j < n; j++)
        {
            if (strcmp(_Str1: peo[j].Id, _Str2: peo[k].Id) < 0)
            {
                k = j;
            }
            if (k != i)
            {
                temp = peo[i];
                peo[i] = peo[k];
                peo[k] = temp;
            }
        }
    }
}
```

⑪

//函数功能:把结构体数组中的内容写到文件中保存起来

//函数接口参数:结构体数组的地址

//返回类型为void

```

void WriteToFile(P peo[])
{
    FILE* fp = fopen(_FileName: "system.txt", _Mode: "w");
    if (fp == NULL)
    {
        printf(_Format: "Fail to open system.txt\n");
        return ;
    }
    fprintf(_Stream: fp, _Format: "%d\n", sum_of_peo);
    int i = 0;
    for(i=0;i<sum_of_peo;i++)
    {
        fprintf(_Stream: fp, _Format: "%s\t%s\t\t%d\t\t%d\n",
            peo[i].Id, peo[i].name, peo[i].income, peo[i].expense);
    }
    fclose(_Stream: fp);
    fp = NULL;
    return;
}

```

⑫

//函数功能:从文件中读取信息放到结构体数组中

//函数接口参数:结构体数组的地址

//函数返回类型:void

```

void ReadFromFile(P peo[])
{
    FILE* pf=fopen(_FileName: "system.txt", _Mode: "r");
    if (pf == NULL)
    {
        printf(_Format: "fail to open the file\n");
        return;
    }
    int i = 0;
    fscanf(_Stream: pf, _Format: "%d\n", &sum_of_peo);
    for (i = 0; i < sum_of_peo; i++)
    {
        fscanf(_Stream: pf, _Format: "%s%s%d%d", &peo[i].Id, &peo[i].name, &peo[i].income, &peo[i].expense);
    }
    fclose(_Stream: pf);
    pf = NULL;
    return;
}

```

2. 函数实现的功能、算法及函数逻辑流程的描述。

①

```

void menu(); //打印菜单,提醒用户进行选择

```

该函数的算法简单,只是普通的打印语句

②input_information 函数

//函数功能:输入用户信息 入口:结构体数组的地址

```
void input_information(P peo[])
{
    int i = sum_of_peo, ret;
    printf(_Format: "Please input the Id,name,income and expense of user %d\n", sum_of_peo + 1);
    while (ret = scanf(_Format: "%s%d", &peo[i].Id, &peo[i].name, &peo[i].income, &peo[i].expense) < 4)
    {
        printf(_Format: "There are some erro of your input,please input again\n");
        fflush(_Stream: stdin);
    }
    sum_of_peo++;
}
```

该函数是接收用户输入的数据,采用 scanf 进行格式化输入,其中利用 scanf 函数的返回值和 while 循环对用户输入的数据进行判断是否合法

③quick_sort_by_name 函数,该函数利用快速排序,按照名字的顺序对结构体数组中的成员进行排序。算法:按照分而治之的思想,把整个要排序的数组进行细分。

详细思路:先选取一个中间变量 pivot,并定义 left 和 right 变量为最左最右端的值,我这里都是选取最左边的元素作为 pivot,当右边的变量比 pivot 大时,说明此时已经满足我的所需,所以只要让 right--,不用进行其它操作。当右边的变量小于 pivot 时,只需把该量放到 left 指向的位置并对 left 做相类似的操作,直到 left>right 时,说明已排序完成。

//函数功能:按名字顺序排序 入口:结构体数组的名字,左端点,右端点

```
void quick_sort_by_name(P peo[], int L, int R)
{
    P pivot;
    int left = L, right = R;
    if (left >= right)
        return;
    pivot = peo[left];
    while (left < right)
    {
        while (left < right && strcmp(_Str1: peo[right].name, _Str2: pivot.name) > 0)
        {
            right--;
        }
        if (left < right && strcmp(_Str1: peo[right].name, _Str2: pivot.name) < 0)
        {
            peo[left] = peo[right];
        }
        while (left < right && strcmp(_Str1: peo[left].name, _Str2: pivot.name) < 0)
        {
            left++;
        }
        if (left < right && strcmp(_Str1: peo[left].name, _Str2: pivot.name) > 0)
        {
            peo[right] = peo[left];
        }
    }
    peo[right] = pivot;
    quick_sort_by_name(peo, L, 0, right - 1);
    quick_sort_by_name(peo, left + 1, R);
}
```


④find_by_name 函数，函数的基本逻辑为，先按名字对数组进行排序，再采用二分查找的方法找到对应人，如果找到了就返回该用户的坐标，没找到就返回-1。其中关键点在于二分查找时，不能像普通的整形一样直接比较大小，要用 strcmp 函数来比较两个字符串的大小，不断的缩小范围，left>=right 为止

```
//函数功能:按名字搜索用户 入口:用户名, 结构体数组名 返回值:存在时该用户的位置下标, 不存在返回-1
int find_by_name(char* name, P peo[])
{
    quick_sort_by_name(peo, L:0, R:sum_of_peo - 1); //先把名字排序, 再二分查找
    int left = 0, right = sum_of_peo - 1, mid;
    while (left <= right)
    {
        mid = (left + right) / 2;
        if (strcmp(_Str1:name, _Str2:peo[mid].name) < 0)
            left = mid + 1;
        else if (strcmp(_Str1:name, _Str2:peo[mid].name) > 0)
            right = mid - 1;
        else if (strcmp(_Str1:name, _Str2:peo[mid].name) == 0)
            return mid;
    }
    return -1;
}
```

⑤和⑥

这两个为打印函数,较为简单,其中 PrintoneInformation 函数只需要直接打印出某坐标处的用户,PrintAllInfomation 函数用一个 for 循环来遍历整个数组,从而打印整个数组.

```
//函数功能:打印所有用户的信息 函数入口:结构体数组名
void PrintAllInformation(P peo[])
{
    printf(_Format:"ID\tUserName\tIncome\tExpenses\n");
    int i;
    for (i = 0; i < sum_of_peo; i++)
    {
        PrintoneInformation(peo, position:i);
    }
}

//函数功能:打印某个用户的信息 函数入口:结构体数组, 用户下标的位置
void PrintoneInformation(P peo[], int position)
{
    printf(_Format:"%s\t%s\t\t%d\t%d\n", peo[position].Id, peo[position].name, peo[position].income, peo[position].expense);
}
```

⑦和⑧

这两个函数的功能为求平均值,用传进来的结构体数组,求支出或收入的和,再除以总人数即为平均值.

```

//函数功能:计算所有用户的平均支出 函数入口参数:结构体数组名
double mean_of_expend(P peo[])
{
    double sum_of_expense = 0;
    int i;
    for (i = 0; i < sum_of_peo; i++)
    {
        sum_of_expense += peo[i].expense;
    }
    return sum_of_expense / sum_of_peo;
}

//函数功能:计算用户的平均收入 函数入口参数:结构体数组名 返回平均收入的值
double mean_of_income(P peo[])
{
    double sum_of_income = 0;
    int i;
    for (i = 0; i < sum_of_peo; i++)
    {
        sum_of_income += peo[i].income;
    }
    return sum_of_income / sum_of_peo;
}

```

⑨find_morethan_perexpense函数,在该函数中调用 mean_of_expend 函数来计算所有用户的支出平均值,再用 for 循环来遍历整个结构体数组中的所有用户的 expense 的值,找到比平均值大的用户再打印即可

```

//函数功能:打印找到所有大于平均收入的用户的信息
void find_morethan_perexpense(P peo[])
{
    int i, j = 0;
    double perexpense = mean_of_expend(peo);
    for (i = 0; i < sum_of_peo; i++)
    {
        if (peo[i].expense > perexpense)
            PrintoneInformation(peo, position: i);
    }
}

```

⑩sort_by_id 函数

该函数的功能为,按照 id 对用户进行排序,在这里,我选用了选择排序法,基本逻辑为,先假定最小的数就是第 i 个数,另最小数的坐标为 k,每次都把第 k 个数,与后面的数进行比较,当找到比第 k 个数小的数 arr[j]时,就另 k=j,然后继续与后面的数比较.所以,当比较完整数组中的 id 后总能找到一个最小的数,然后把它放到第一位.接着,再去找第 2 小,第 3 小的数,依次放到第 2,第 3 位

```

//函数功能:按学号进行选择排序
//参数接口:结构体数组地址
//返回类型为void
void sort_by_id(P peo[])
{
    int i, j, k;
    P temp;
    int n = sum_of_peo;
    for (i = 0; i < n - 1; i++)
    {
        k = i;
        for (j = i + 1; j < n; j++)
        {
            if (strcmp(_Str1:peo[j].Id, _Str2:peo[k].Id) < 0)
            {
                k = j;
            }
        }
        if (k != i)
        {
            temp = peo[i];
            peo[i] = peo[k];
            peo[k] = temp;
        }
    }
}

```

⑪WriteToFile 函数

逻辑：先打开文件，再利用 fprintf 函数把数组中的内容写入到文件中

//函数功能:把结构体数组中的内容写到文件中保存起来

//函数接口参数:结构体数组的地址

//返回类型为void

```

void WriteToFile(P peo[])
{
    FILE* fp = fopen(_FileName:"system.txt", _Mode:"w");
    if (fp == NULL)
    {
        printf(_Format:"Fail to open system.txt\n");
        return ;
    }
    fprintf(_Stream:fp, _Format:"%d\n", sum_of_peo);
    int i = 0;
    for(i=0;i<sum_of_peo;i++)
    {
        fprintf(_Stream:fp, _Format:"%s\t%s\t\t%d\t%d\n",
            peo[i].Id, peo[i].name, peo[i].income, peo[i].expense);
    }
    fclose(_Stream:fp);
    fp = NULL;
    return;
}

```

⑫ReadFromFile 函数

函数实现逻辑：先用只读的形式打开文件“system.txt”，再读取文件中的第一个数字，及用户的总数，再根据用户总数用 for 循环来格式化的把文件中的内容写入到数组中

```

//函数功能:从文件中读取信息放到结构体数组中
//函数接口参数:结构体数组的地址
//函数返回类型:void
void ReadFromFile(P peo[])
{
    FILE*pf=fopen(_FileName:"system.txt", _Mode:"r");
    if (pf == NULL)
    {
        printf(_Format:"fail to open the file\n");
        return;
    }
    int i = 0;
    fscanf(_Stream:pf, _Format:"%d\n", &sum_of_peo);
    for (i = 0; i < sum_of_peo; i++)
    {
        fscanf(_Stream:pf, _Format:"%s%d%d", &peo[i].Id, &peo[i].name, &peo[i].income, &peo[i].expense);
    }
    fclose(_Stream:pf);
    pf = NULL;
    return;
}

```

四、系统运行结果

1. 测试数据列表，注意数据集的完备性（正常值、边界值、异常值）；

用户 ID	姓名	收入	支出
10001	zero	6000	1500
10023	Aef	10000	3000
20011	eric001	20000	10000
20012	ffff	15000	0
30004	abc	8000	14000
边界值			
异常值			
...

按照上述正常值输入后，各功能的实现如下：

选 6:

```

0. EXIT
Please enter your choice:
6
ID      UserName      Income  Expenses
10001   zero            6000    1500
10023   Aef             10000    3000
20011   eric001        20000    10000
20012   ffff           15000     0
30004   abc             8000    14000
1. Input record

```

选 5:

```

Please enter your choice:
5
ID      UserName      Income  Expenses
20011   eric001           20000   10000
30004   abc               8000    14000
1. Input record

```

选 4:

```

Please enter your choice:
4
Per capita income:      11800.000000
Per capita expenses:    5700.000000
1. Input record

```

选 3:

```

Please enter your choice:
3
请输入你要查找人的姓名:
Aef
ID      UserName      Income  Expenses
10023   Aef              10000   3000

```

选 2:

```

Please enter your choice:
2
ID      UserName      Income  Expenses
10023   Aef              10000   3000
30004   abc              8000    14000
20011   eric001          20000   10000
20012   ffff             15000    0
10001   zero             6000    1500
1. Input record

```

选 7:

```

Please enter your choice:
7
读取成功

```

选 8:

```

0. Exit
Please enter your choice:
8
ID      UserName      Income  Expenses
10023   Aef              10000   3000
30004   abc              8000    14000
20011   eric001          20000   10000
20012   ffff             15000    0
10001   zero             6000    1500
1. Input record

```

当输入异常值时:

```

0. Exit
Please enter your choice:
1
Please input the Id,name,income and expense of user 6
12345 aaa add2 dd13
There are some erro of your input,please input again
12333 32ds 13323 23233
1. Input record

```


如这个,我在输入用户的收入时输入了字母,此时程序会提醒我输入有误,请重新输入。直到输入正确才录入信息

```
0. Exit
Please enter your choice:
1
Please input the Id,name,income and expense of user 1
123455 dsa 13323 23232
There are some erro of your input,please input again
12343 dad 32332 32323
1. Input record
```

Id 规定为 5 位,当我输入 6 位 id 时,会提醒 " there are some erro... ",让用户重新输入,直到符合为止。

```
0. Exit
Please enter your choice:
1
Please input the Id,name,income and expense of user 1
123455 dsa 13323 23232
There are some erro of your input,please input again
12343 dad 32332 32323
1. Input record
```

```
1. Input record
2. Sort and list records in order by user name
3. Search records by user name
4. Calculate and list per capita income and expenses
5. List records which have more expense than per capita expenses
6. List all records
7. Write to file
8. Read from file
9. Exit
Please enter your choice:
2
There are some erro of your choice,please choose again.
3
There are some erro of your choice,please choose again.
4
There are some erro of your choice,please choose again.
5
There are some erro of your choice,please choose again.
6
There are some erro of your choice,please choose again.
```

当我未选 1 或 8 来录入信息就直接选其他选择时,程序会告诉我出现了错误,让我重新选择。而当我已经选了 1 或 8 来录入信息时,就可以正常的选择其他选项了(如下图)

```

Please enter your choice:
8
ID      UserName      Income  Expenses
10023   Aef                10000   3000
30004   abc                8000    14000
20011   eric001           20000   10000
20012   ffff              15000    0
10001   zero              6000    1500
1. Input record
2. Sort and list records in order by user name
3. Search records by user name
4. Calculate and list per capita income and expenses
5. List records which have more expense than per capita expenses
6. List all records
7. Write to file
8. Read from file
0. Exit
Please enter your choice:
4
Per capita income:      11800.000000
Per capita expenses:    5700.000000
1. Input record

```

以上即为我在此次实验做的报告，在写报告的过程中也思考和收获了许多。

注：

- (1) ID 由 5 位数字组成；
- (2) 用户名由字母和数字组成，长度不超过 10，区分字母大小写。

2. 测试结果描述或截图

- (1) 需体现正常值、边界值、异常值输入后的输出；
- (2) 需体现非常规流程处理能力（例如未录入数据，用户先选择排序等其他功能...）

五、实验课总结和建议

总结：

- ①c 语言是一门实操性比较强的学科，在学习的过程中，只有不断通过做实验题,多敲代码,才能真正的把握知识.
- ②写代码时要有大局观,在处理复杂问题时,先写出主体,知道自己需要什么功能的函数,再慢慢实现,这样往往能提高效率.
- ③在写程序的过程当中,要时刻注意边界值的输入情况,很多时候程序的错误往往出现在边界值这一块.
- ④要注意加强代码的鲁棒性,如利用 `scanf` 的返回值来判断用户的输入是否规范.

建议：

希望学校能升级一下我们的 oj 平台,因为有时候看着输出明明一模一样却就是通过不了.

签名：周健毅

2022 年 10 月 31

日

2022秋