

# C语言实验七：结构体与输入输出重定向

## 必做题要求（勿忘提交截图，尤其是做了选做题的同学）

1. 课程内完成：OJ系统的AC~AK题
2. 提交要求：提交包含以上题目编号的截图，截图右上角必须包含你的学号信息

## 选做题要求

- 按以下要求提交，否则不计分。需提交：
  1. 相应函数实现的截图；
  2. 用相应测试例测试的运行结果截图。
- 若题目要求使用动态数组实现却未合理释放内存，则该题不计分。
- 完成奖励：
  - 9点前全部做完并检查通过后则可以提前离开课堂
  - 晚上9:15前完成并提交：题目1加0.3分，题目2加0.2分（加满为止，只通过部分测试例的不加分）

## 题目1：结构体数组中按条件筛选并计算收入总和

函数原型：

```
1 double SumIncomeByLastname(struct Person *people, int n, const char *lastname);  
2 double SumIncomeByIDList(struct Person *people, int n, int *id_list, int id_count);
```

功能描述：

定义结构体 `Person` 如下（测试代码中已包含）：

```
1 struct Person {  
2     int id;  
3     char name[30];  
4     double income;  
5 };
```

编写两个函数：

1. `SumIncomeByLastname`：

在 `people` 数组（共 `n` 人）中，找出所有 姓氏（即姓名的第一个单词）等于 `lastname` 的人，返回他们的收入总和。

示例：若 `name = "Zhang San"`，则姓氏为 `"Zhang"`。

2. `SumIncomeByIDList`：在 `people` 数组中，找出所有 `id` 出现在 `id_list` (长度为 `id_count`) 中的人员，返回他们的收入总和。
3. 从服务器获取输入重定向文件，完成后导出名为 `income.out` 的输出重定向文件，并用以下命令验证：

```
1 curl -F "file=@income.out" http://10.131.21.39:9000
```

#### 示例：

假设输入人员数据为：

```
1 1001 Zhang San 5000.0
2 1002 Li Si 6000.0
3 1003 Zhang Wu 7000.0
4 1004 Wang Liu 8000.0
```

- 调用 `SumIncomeByLastname(people, 4, "Zhang")` → 返回  $5000.0 + 7000.0 = 12000.0$
- 若 `id_list = {1002, 1004}`，调用 `SumIncomeByIDList(..., id_list, 2)` → 返回  $6000.0 + 8000.0 = 14000.0$

#### 参数说明：

- `people`：指向结构体数组的指针
- `n`：数组中人员数量
- `lastname`：要匹配的姓氏字符串（保证不含空格）
- `id_list`：待查找的 ID 数组（元素顺序任意）
- `id_count`：`id_list` 中的 ID 个数

#### 返回值：

- 若无匹配人员，返回 `0.0`；
- 否则返回匹配人员的收入总和（`double` 类型）。

#### 测试代码与输入重定向文件的下载方式：

```
1 wget http://10.131.21.39:8000/CalculateIncome.c
2 wget http://10.131.21.39:8000/people.in
```

## 题目2：完成OJ系统所有题目