

四川大学电气工程学院

# 实验报告（本科）



提交人学号 2024141440057

提交人姓名 张美杰

专 业 电气类

日 期 2024/11/26

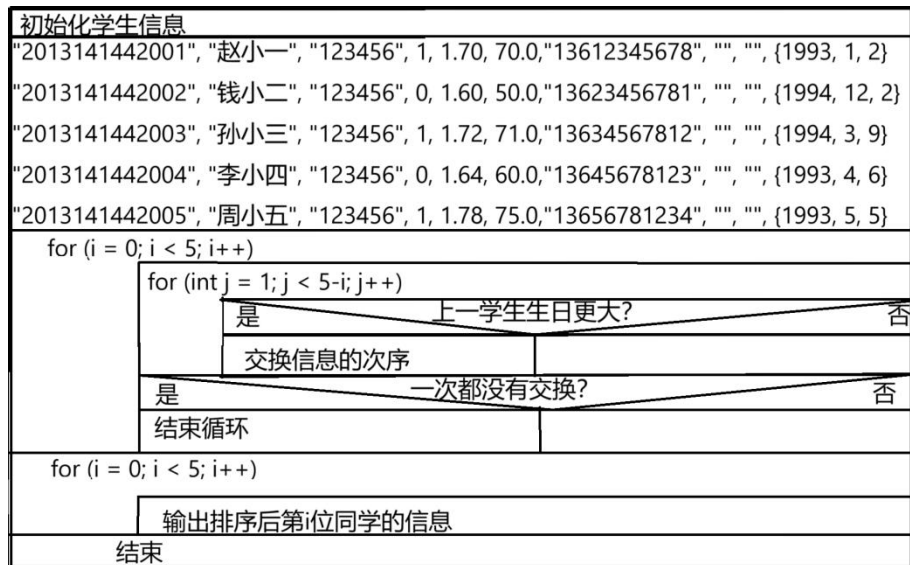
实验题目 函数（2）

组员（按贡献大小排序）：

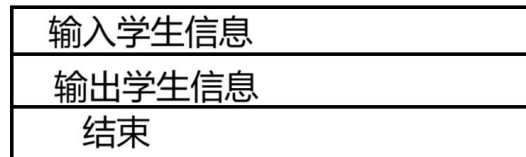
张美杰

# 算 法 描 述

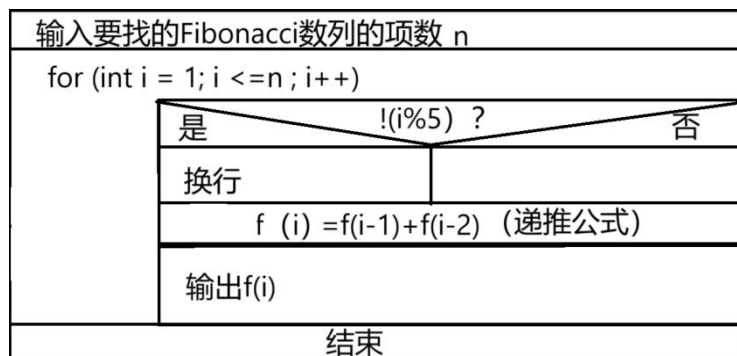
## 实训 1



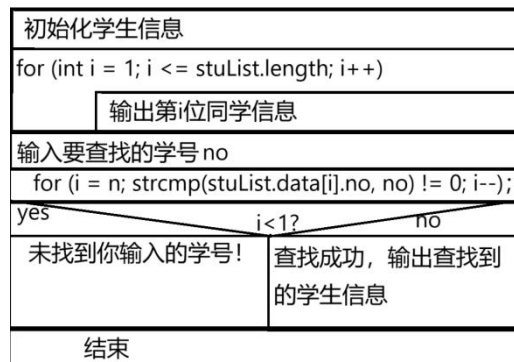
## 实训 2



## 实训 3

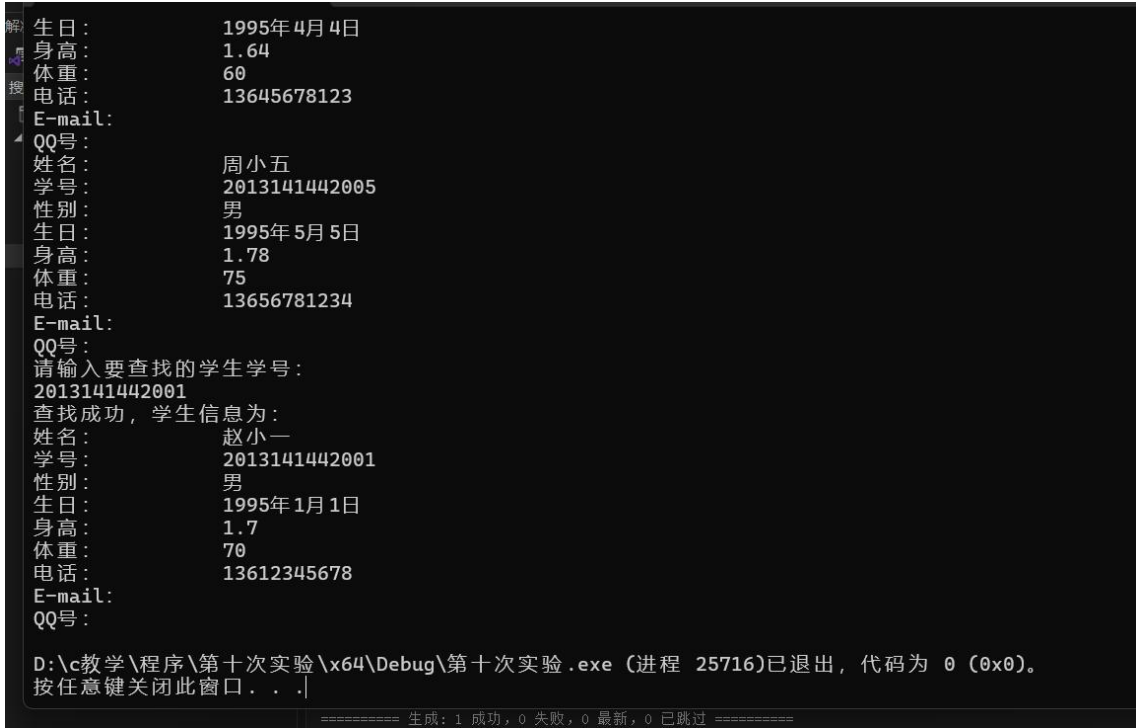


## 实训 4



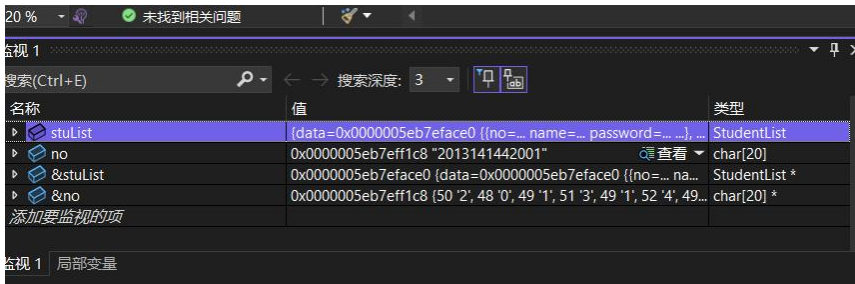
# 实验结果（实训 1~4 的执行界面截图）

## 实训 1 2)



## 3)

对于 stuList, 是用的传引用调用, 而实验 9 实训 1 为传值调用, 在这里 stulist 和 no 在函数体中的变化会返回给实参 stulist 和 no, 影响原本的值 (实验 9 实训 2 有提到), 表现为值和地址的值相同, 形参不单独占空间 (如下)。



## 实训 2

### 2)

```

Microsoft Visual Studio 调试
请输入学生信息：
姓名： 张美杰
学号： 2024141440057
密码： 123456
性别 (0为女, 1为男)： 1
生日：
年： 2006
月： 2
日： 16
身高： 170
体重： 120
电话： 12345678901
E_mail: 12345678901@gmail.com
QQ号： 12345678901
你输入的学生信息如下：
姓名： 张美杰
学号： 2024141440057
性别： 男
生日： 2006年2月16日
身高： 170
体重： 120
电话： 12345678901
E_mail : 12345678901@gmail.com
QQ号： 12345678901

D:\c教学\程序\第十次实验\x64\Debug\第十次实验.exe (进程 26128)已退出，代码为 0 (0x0)。
按任意键关闭此窗口...

```

### 3)

名称	值	类型
&stu	0x000000fd5550fb10 {no=0x000000fd5550fb10 "1" name=0x000...	StudentInfo *
no	0x000000fd5550fb10 "1"	char[20]
name	0x000000fd5550fb24 "1"	char[20]
password	0x000000fd5550fb38 "1"	char[20]
sexy	1	int
height	1.0000000000000000	double
weight	1.0000000000000000	double
telephone	0x000000fd5550fb60 "1"	char[20]
e_mail	0x000000fd5550fb74 "1"	char[40]
qq	0x000000fd5550fb9c "1"	char[20]
birthday	{year=1 month=1 day=1 }	Date
stu	{no=0x000000fd5550fb10 "1" name=0x000000fd5550fb24 "1" pa...	StudentInfo
no	0x000000fd5550fb10 "1"	char[20]
name	0x000000fd5550fb24 "1"	char[20]
password	0x000000fd5550fb38 "1"	char[20]
sexy	1	int
height	1.0000000000000000	double
weight	1.0000000000000000	double
telephone	0x000000fd5550fb60 "1"	char[20]
e_mail	0x000000fd5550fb74 "1"	char[40]
qq	0x000000fd5550fb9c "1"	char[20]
birthday	{year=1 month=1 day=1 }	Date
添加要监视的项		

此处为指针传递可以改变原来的值，而实验 9 实训 1 的不会。

### 4)

传值调用赋予形参值来运行函数，传引用调用将地址一并赋给形参，它们都能使函数正常根据实参的值运行，传值调用只调用形参的数值，形参单独在另一个地址储存，传引用调用将实参数值和地址一并赋给了形参，形参和实参在同一地址储存数值。此题目中，函数返回赋值形参，但实际的 stu 并未改变。

### 实训 3

```
Microsoft Visual Studio 调试 × + -
解 请输入要输出的数列的项数 (从1开始) : 40
排 前40项Fibonacci数列如下:
工  f(1)=1      f(2)=1      f(3)=2      f(4)=3      f(5)=5
f(6)=8      f(7)=13     f(8)=21     f(9)=34     f(10)=55
f(11)=89     f(12)=144   f(13)=233   f(14)=377   f(15)=610
f(16)=987    f(17)=1597  f(18)=2584  f(19)=4181  f(20)=6765
f(21)=10946  f(22)=17711 f(23)=28657 f(24)=46368 f(25)=75025
f(26)=121393 f(27)=196418 f(28)=317811 f(29)=514229 f(30)=832040
f(31)=1346269 f(32)=2178309 f(33)=3524578 f(34)=5702887 f(35)=9227465
f(36)=14930352 f(37)=24157817 f(38)=39088169 f(39)=63245986 f(40)=102334155

D:\c教学\程序\第十次实验\x64\Debug\第十次实验.exe (进程 27224)已退出, 代码为 0 (0x0)。
按任意键关闭此窗口. . . |

===== 生成: 1 成功, 0 失败, 0 最新, 0 已跳过 =====
===== 生成 于 15:17 完成, 耗时 00.708 秒 =====
```

## 实训 4

### 初始化信息

```
StudentInfo stuList[nStudent+1] =//学生信息初始化
{
    {},
    {"2013141442001", "赵小一", "123456", 1, 1.70, 70.0, "13612345678", "", "", {1993, 1, 2}},
    {"2013141442002", "钱小二", "123456", 0, 1.60, 50.0, "13623456781", "", "", {1994, 12, 2}},
    {"2013141442003", "孙小三", "123456", 1, 1.72, 71.0, "13634567812", "", "", {1994, 3, 9}},
    {"2013141442004", "李小四", "123456", 0, 1.64, 60.0, "13645678123", "", "", {1993, 4, 6}},
    {"2013141442005", "周小五", "123456", 1, 1.78, 75.0, "13656781234", "", "", {1993, 5, 5}}
};
```

### 排序结果

```
姓名: 赵小一
学号: 2013141442001
性别: 男
生日: 1993年1月2日
身高: 1.7
体重: 70
电话: 13612345678
E-mail:
QQ号:
姓名: 李小四
学号: 2013141442004
性别: 女
生日: 1993年4月6日
身高: 1.64
体重: 60
电话: 13645678123
E-mail:
QQ号:
姓名: 周小五
学号: 2013141442005
性别: 男
生日: 1993年5月5日
身高: 1.78
体重: 75
电话: 13656781234
E-mail:
QQ号:
姓名: 孙小三
学号: 2013141442003
性别: 男
生日: 1994年3月9日
身高: 1.72
体重: 71
电话: 13634567812
E-mail:
QQ号:
姓名: 钱小二
学号: 2013141442002
性别: 女
生日: 1994年12月2日
身高: 1.6
体重: 50
电话: 13623456781
E-mail:
QQ号:

D:\c教学\程序\第十次实验\x64\Debug\第十次实验.exe (进程 28296)已退出, 代码为 0 (0x0)。
按任意键关闭此窗口. . . |
```

## 总结（出错信息、出错原因、修改方法、体会等。这部分需包含每个组员自己的总结体会）

### 出错信息

#### 实训 1

代码: `strcpy(stuList.data[0].no, no);`

报错: 'strcpy': This function or variable may be unsafe. Consider using `strcpy_s` instead. To disable deprecation, use `_CRT_SECURE_NO_WARNINGS`. See online help for details.

#### 实训 4

### 冒泡排序功能出错

### 出错原因

#### 实训 1

在 Visual Studio 等现代编译器中, `strcpy` 被认为是不安全的(可能导致缓冲区溢出)。编译器会推荐使用更安全的函数, 如 `strcpy_s`。

#### 实训 4

判断是否交换的 `changed` 放入了内循环中, 导致排序提前结束。

### 修改方法

#### 实训 1

将 `strcpy` 修改为 `strcpy_s` 即可。

#### 实训 4

### 改为

```
46 void AgeOrder(StudentInfo* stu)
47 {
48     int i;
49     for (i = 0; i < 5; i++)
50     {
51         int exchange = 0;
52         for (int j = 1; j < 5-i; j++)
53         {
54             if ((stu[j].birthday.year * 10000 + stu[j].birthday.month * 100 + stu[j].birthday.day) > (stu[j+1].birthday.year * 10000 + stu[j+1].birthday.month * 100 + stu[j+1].birthday.day))
55             {
56                 stu[0]=stu[j];
57                 stu[j] = stu[j + 1];
58                 stu[j + 1] = stu[0];
59                 exchange = 1;
60             }
61         }
62         if (exchange == 0)
63         {
64             break;
65         }
66     }
67 }
```

### 体会（张美杰）

函数参数传递方式分为传值调用和传址调用。传值调用将实参的值复制给形参, 函数内部对形参的修改不影响实参, 适合传递简单变量。传址调用通过引用或指针将实参的地址传递给形参, 函数内部对形参的操作直接作用于实参, 适合处理大对象或需要修改实参的情况。两者区别在于是否直接操作实参及效率差异。前者使用较为复杂但不用考虑实参的安全问题, 后者使用简单但需要保证实参对应的值安全。

实训 3 代码实现了通过递归方式计算 Fibonacci 数列的前 n 项, 并输出结果。使用递归能更加简洁美观通俗易懂。

使用函数封装各个反复调用的相同代码模块可以简化代码, 使代码更有逻辑层次分明。读代码也更加简单了。