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**  **10_4**  **实训2**  **10_2**  **实训3**  **10_3**  **实训4**  **10_1** |

|  |
| --- |
| **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)**    **3)**    **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**    **实训4**  **初始化信息**  Screenshot 2024-11-26 161955  **排序结果** |

|  |
| --- |
| **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setcpy\_s即可。**  **实训4**  **改为**  **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。** |