

# 暨南大学本科实验报告专用纸

课程名称 高级语言程序设计实验 成绩评定             
实验项目名称 验证实验 指导教师 张鑫源  
实验项目编号 ⑩ 实验项目类型            实验地点 家中  
学生姓名 位雨昕 学号 2019051112  
学院 智能科学与工程学院 系            专业 信息安全  
实验时间 2020 年 5 月 13 日 上午 ~ 5 月 13 日 下午 温度    °C 湿度   

## （一）实验目的

1. 进一步了解 Visual Studio 的使用以及 C 语言程序的结构；
2. 掌握 C 语言中的常用函数的使用方法；
3. 了解值传递和引用传递两种参数传递方式两种方式的机制；
4. 锻炼个人的编程操作能力。

## （二）实验内容和要求

内容：

设计实验探讨值传递和引用传递两种参数传递方式的差别，并结合实验结果在实验报告中对两种方式的机制做出说明。

示例：设计函数 f1 和 f2，分别采用值传递和引用传递两种方式实现实参与形参之间的数据传递，f1 和 f2 的功能均是实现形参数值的互换。

要求：

1. 设计合理的输出展示实验结果；
2. f1 和 f2 可以放在独立的源文件中。

## （三）主要仪器设备

仪器：计算机

实验环境：Visual Studio Community 2019

## （四）源程序

主程序.cpp:

```
#include<stdio.h>
extern void fun1(int x, int y);
extern void fun2(int &x, int &y);

int main()
{
    int a = 1;
    int b = 2;
    fun1(a, b);
    printf("a=%d, b=%d\n", a, b);
    printf("\n");
    fun2(a, b);
    printf("a=%d, b=%d\n", a, b);
    return 0;
}
```

fun1.cpp:

```
#include<stdio.h>
void fun1(int x, int y) //值传递
{

    int temp;
    temp = x;
    x = y;
    y = temp;
    printf("x=%d, y=%d\n", x, y);
}
```

fun2.cpp:

```
#include<stdio.h>
void fun2(int &x, int &y) //引用传递
{

    int temp;
    temp = x;
    x = y;
    y = temp;
    printf("x=%d, y=%d\n", x, y);
}
```

## （五）实验步骤与调试

实验步骤：

1. 对实验进行基本构思；
2. 启动 Visual Studio，创建新项目。将源程序写在新项目中；
3. 利用“本地 Windows 调试器”进行调试；
4. 进行多次调试，得出理想结果。

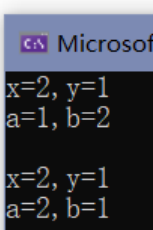
实验思路：

1. 新建三个源文件，一个用于写主程序，其余两个用以定义实现数值对调功能的函数；
2. 通过值传递实现实参与形参之间的数据传递的函数 fun1 在定义时形参使用(int x, int y)的形式；
3. 通过引用传递实现实参与形参之间的数据传递的函数 fun2 在定义时形参使用(int &x, int &y)的形式；（引用的声明方法：类型标识符 &引用名=目标变量名；）
4. 在主程序中调用外部函数 fun1、fun2，并用 extern 进行声明；
5. 在 fun1 与 fun2 中添加打印 x、y 值的语句，同时调用函数后增添打印 a、b 值的语句，可以更好地反映参数的值传递的情况。

## （六）实验结果与分析

实验结果：

```
int a = 1;
int b = 2;
fun1(a, b);
printf("a=%d,b=%d\n", a, b);
printf("\n");
fun2(a, b);
printf("a=%d,b=%d\n", a, b);
```



可以看到，通过值传递方式实现实参与形参之间数据传递的 fun1 并没有实现将 a、b 的值对调的功能。而 fun1 中的变量 x、y 的值却发生了对调。

fun2 中 x、y 的值发生了对调，且调用 fun2 也实现了将 a、b 的值对调的功能。

分析：

1. 在调用 fun1 函数时，将实参 a、b 的值传入形参 x、y。在传入 fun1 后其实隐含了

```
int x = a;  
int y = b;
```

这两行语句的操作（进行值传递）。在函数 fun1 内部操作的其实是定义 fun1 时定义的变量 x、y，而不是 main 函数中的变量 a、b。此后 fun1 内部没有任何对 main 函数中变量 a、b 进行任何操作。因此采用值传递的方式并没有实现对调 a、b 的值的功能；

```
x=2, y=1  
a=1, b=2
```

如图，只有 x、y 实现了对调。

2. 在调用 fun2 函数时，将实参 a、b 的值传入形参 x、y。在传入 fun2 后其实隐含了

```
int &x = a;  
int &y = b;
```

这两行语句的操作（进行引用传递）。这两句语句定义引用 x、y，是变量 a、b 的别名（相当于同一变量有两个名称）。调用 fun2 时函数会将 a、b 分别代替了 x、y，即 x、y 分别引用了 a、b 变量。引用就是某一变量（目标）的一个别名，对引用的操作与对变量直接操作完全一样。

函数 fun2 中操作的其实就是实参 a、b 本身，可以直接修改到 a、b 的值，采用引用传递的方式可以实现对调 a、b 的值的操作。

```
x=2, y=1  
a=2, b=1
```

如图，变量 x、y 及变量 a、b 均实现了对调。

3. 综上，值传递和引用传递两种参数传递方式的差别是值传递是将原变量的值传递给另一变量，重新分配存储空间；而引用传递是直接访问变量，不用分配自己的内存空间。值传递操作的是不同的变量，而引用传递操作的是同一变量。