

C++程序设计入门（上）

第三周编程作业

1.题目内容：（难度：中等）

本题要求实现两个重载的 `swap` 函数，每个 `swap` 函数都可交换 3 个整数 `a`, `b`, `c` 的值。将 `a` 的值存入 `b`, `b` 的值存入 `c`, `c` 的值存入 `a`, 并且返回三个整数中最大的数的值。例如 `a,b,c` 的值为 `1,2,3`, 则交换后, `a,b,c` 的值为 `3,1,2`, 所返回的值为 `3`。

- 1) 在主函数中一次读入三个整数的值存入 `a`、`b`、`c` 中；
- 2) 以 `a`、`b`、`c` (或者其地址)作为实参，连续调用两个不同 `swap` 函数；
- 3) 最后输出 `a`、`b`、`c` 三个整数中最大的值，以及三个整数；
- 4) 变量定义时均应赋初值 0。
- 5) 不得使用 C 语言中的输入输出函数

函数接口定义：

```
int swap ( int& x, int& y, int& z); //三个参数均为引用类型
```

```
int swap ( int* x, int* y, int* z); //三个参数均为指针类型
```

任一个 `swap()`在交换三个整数的值后，返回 `x,y,z` 中最大的数的值

每个 `swap` 函数都可交换 3 个整数 `a`, `b`, `c` 的值。将 `a` 的值存入 `b`, `b` 的值存入 `c`, `c` 的值存入 `a`

输入格式：

空格分隔的三个整数

整数 1 整数 2 整数 3

（这三个整数按照输入次序存入 `a`、`b`、`c` 三个变量）

输出格式：

空格分隔的四个整数

最大数 `a` 变量的值 `b` 变量的值 `c` 变量的值

（注意：`a`、`b`、`c` 三个变量中的值是经过 `swap` 函数交换过的值）

输入样例：

1 2 3

（备注：程序中，1 存入变量 `a`, 2 存入变量 `b`, 3 存入变量 `c`）

输出样例：

3 2 3 1

（备注 1：最大值 3 首先输出）

（备注 2：第一次调用 `swap` 前，`a` 值为 1；`b` 值为 2；`c` 值为 3；第一次调用 `swap` 后，`a` 值为 3，`b` 值为 1，`c` 值为 2）

（备注 3：第二次调用 `swap` 后，`a` 值为 3，`b` 值为 1，`c` 值为 2）

测试例说明:

- 1) 输入的值超过 `int` 范围时, 由编译器自动处理, 无需在程序中做判别。
- 2) 输入的值非 `int` 时, 由编译器自动处理, 无需再程序中做判别。
- 3) 要考虑 3 个输入的值有两个或者三个相同的情况 (是否需要专门写相关代码处理这种情况, 你需要思考决定)

2.题目内容: (难度: 易)

编程序, 学习使用类型转换运算符 `static_cast`

编程序, 读入两个数, 然后计算两个数的除法结果。读入的第一个数 `a` 是整数; 读入的第二个数 `b` 是双精度浮点数:

- 1) 将 `b` 转换为整数, 然后计算整数除法 `a/b`, 结果记为 `x`;
- 2) 将 `a` 转换为双精度浮点数, 然后计算双精度浮点数除法 `a/b`, 结果记为 `y`
- 3) 将 `a` 转换为双精度浮点数, 将 `b` 转换为整数然后再转换为双精度浮点数, 然后计算双精度浮点数除法 `a/b`, 结果记为 `z`

输出 `x`、`y`、`z` 的值。

本程序无需考虑除数为 0、结果溢出等异常情况。

- 注意: 1) 以上所有转换都使用 `static_cast` 运算符
2) 如果除法结果为浮点数, 则输出时精确到小数点后 3 位数字

输入格式:

第一个数为整数, 第二个数为双精度浮点数;
两个数之间使用空格分隔

输出格式:

按序输出 `x`、`y`、`z` 的值, 相邻两个值之间使用 1 个空格分隔

注 1: 如果输出值为浮点数, 则需要使用 `std::fixed` 与 `std::setprecision` 函数设置小数点后位数为 3 位。

这两个 `std` 成员可以自行搜索或者到 cppreference.com 检索 (英文)

注 2: 使用 `std::setprecision` 函数需要包含 `<iomanip>` 头文件

输入样例:

```
4 2.5
```

输出样例:

```
2 1.600 2.000
```

3.题目内容：（难度：极易）

编写一个 `swap` 函数，将数组元素作为函数参数，实现数组元素内容的交换

- 1) 在 `main` 函数中定义一个整型数组 `a[]`，其中存有 5 个元素：1,2,3,4,5
- 2) 定义函数 `void swap(int &x, int &y);` `swap` 函数将其参数 `x` 与 `y` 中的值互换
- 3) 在主函数中调用 `swap` 函数；使得数组 `a[]` 中的第 2 个元素与第 4 个元素的值互换
- 4) 调用 `swap` 之后，主函数再按照顺序输出数组 `a[]` 中的所有元素的值

特别注意：1) 数组中的“第 n 个”元素是指从 1 开始的，符合普通人类认知的计数方法

2) 例如对于数组 `int arr[] = {9,8,7,6};`

“第 1 个元素”是 `a[0]`，值 9；

“第 2 个元素”是 `a[1]`，

“第 n 个元素”是 `a[n-1]`，其中 $n \geq 1$

3) 我们称：数组中的“第 m 个元素”是下标为 $m-1$ 的元素，即 `arr[]` 的第 m 个元素是 `arr[m-1]`

4) 我们称：数组中的“编号为 n 元素”是下标为 n 的元素，即 `arr[]` 的第 n 号元素是 `arr[n]`

想象一下，当你学完 C 语言，放学回家跟自己老娘说：“母上大人，你给我带的第 0 个茶叶蛋齁死了”；你母上大人会不会去找鸡毛掸子准备揍你？(¬▽¬) 切~~

输出格式：

注意：绝对绝对绝对【不要】输出任何无关的信息!!!

例如 `cout << "The numbers are:"` 这种语句不要出现在本作业中!!!

也不要无聊滴加一些空格、换行，例如 `cout << ' '; cout << endl;`

连续输出 5 个元素。元素之间紧挨彼此，中间没有分隔符；

输出之后没有换行符

输出样例：

注意：下面的输出样例与程序输出不同，不能用于测试程序

74589