

# TCPL第4-6课作业检查

## class4

### cw1-insertsort.c

1. In:8 s[i] != '\0' 多余。
2. 排序遇到已经有序应该立即退出循环。
3. 包装成小函数: sort, print, check, swap。
4. 应该检查排序结果是否正确。
5. 输出时最后不能有多余的逗号。
6. 数组成员个数要用 countof。

### cw2-factorialsum.c

1. N 的取值范围要明确。
2. 输入参数要判断, 不能依赖 assert。
3. 输出的行尾要有\n。
4. 批量测试需要借助脚本 go.sh, input\*.txt, expected\*.txt。

### cw3-gcdlcm.c

1. M 和 N 的取值需要检查。

## ex2-intoverflow.c

1. `#include <limits.h>` 中有各种数据类型的最大值和最小值的宏定义，例如：  
`INT_MAX, INT_MIN`
2. 注意很多数据类型的值域是不对称的，例如 `signed char` 为 `[-128, 127]`。

## hw4-printline.c

1. 全局变量一般的建议是少用。要用的时候采用比较特殊的命令规则，例如：  
`char g_line[MAX_LINE]; // 也就是用一个特殊的前缀`
2. 虽然书上经常用 `int` 表示字符串的长度，现代的 C/C++ 更推荐用 `size_t`。它实际是个无符号的整数，例如：  
`typedef unsigned long size_t;`

## cointoss.c

1. 整数会溢出，改用128位整数。如果要彻底避免溢出，需要找个不限定精度的函数库。或用Matlab、Python。

2. 数1的算法需要简化。改用移位。
3. 注释后的代码：<https://github.com/zedware/notebook/blob/master/samples/tcpl/src/misc/cointoss.c>

## class5

### cw1-narcissus.c

1. 循环变量最好在 for 中定义，例如：

```
for (int j = 0; j < 3; j++) {
```
2. 重复的表达式最好用临时变量存起来，增加可读性，避免写错。例如：

```
int r = m % 10;
s += r * r * r;
```
3. 用 +=, /= 等代替  $x = x + y$ ;  $x = x / y$ ，例如：

```
m /= 10;
```

### cw2-perfect.c

1. 避免使用 hardcode，例如：

```
for (; n <= 1000; n++) {
```

 中的1000也应该换成 MAXFACTOR。
2. 建议采用 bool 表示函数 IsXxx 的返回

值，例如：`bool IsPerfect(int n)`。

## **cw4-shortestline.c**

1. 避免使用 `hardcode`，例如下面的1000都应该替换为MAXLINE：  
31: `int min = 1000;`  
32: `int second_min = 1000;`
2. **ln:14** `line[i] = '\0'`; 应该后移到右花括号之外，否则在输入没有遇到`\n`时字符串将没有`'\0'`结尾。
3. 注意测试，**ln:44** `} else if (len < second_min) {` 对次短行的处理不对。

```
$ ./a.out
```

```
aaa
```

```
b
```

```
The shortest line is: b
```

```
length=2
```

```
The second shortest line is: (p???)
```

```
length=1000
```

正确的逻辑应该是，如果长度小于`min`，则将`min`及其对应的行保存为次短行；并

将当前行保存为最短行。

还需要考虑特殊情况：

1. 没有任何输入，导致最短行和次短行都不存在。
2. 有输入，但是只有最短行，而没有次短行。

也就是min或second\_min的取值为MAXLINE时，就表示对应的最短行或次短行不存在。

## **cw5-varANSI.c**

1. 建议采用 bool 表示函数 IsXxx 的返回值，例如：bool isLegalANSI(...)。
2. 对输入的字符串数组要检查其是否为 NULL或空串，例如：

```
if (s == NULL || s[0] == '\0') {
```

如果程序自己用的函数，可以确保s不为NULL或空串，也最好用assert予以检查，以防万一用错了。

```
    assert(s != NULL);  
    assert(s[0] != '\0');
```

## **class6**

## **ex7-atoi.c**

1. 跟标准库里头的atoi对比一下，看看有哪些地方不同。
2. 遇到不是数字的字符怎么办？
3. 遇到正负符号怎么办？

## **ex5-binsearch.c**

1. 跟标准库中的bsearch对比一下，看看有哪些不同。
2. 如果有重复的记录，二分查找返回的是哪个呢？
3. 如果有重复记录，怎么把它们都找出来？
4. 怎么更好的测试二分查找？

## **ex6-switch1.c**

1. case里头有没有break的区别？
2. 有没有default语句的区别？