

# TCPL第13-15课作业检查

## class13

### ex18-alloc.c

1. 规范的做法是，`alloc()` 分配不到内存时，应该返回 `NULL` 而不是 `0`。
2. `afree()` 有个限制，必须是最后一次 `alloc()` 返回的指针？
3. `afree()` 怎么检查调用者的错误，例如输入了 `NULL`，输入了不是 `alloc()` 返回的指针？

### ex25-patternarg.c

1. `if ((strstr(line, *argv) != NULL) != except) {`，红色的部分是多余的？
2. `while (c = *++argv[0])`，会导致警告，可以修改为 `while ((c = *++argv[0]) != '\0')`
3. 程序的参数 `-x` 和 `-n` 可以分别出现 0 次或多次。严格一点应该限制只能出现 0 次或 1 次。

```
$ ./a.out abc
$ ./a.out -x -n abc
$ ./a.out -n -x -x -n abc
```

## ex27

1. `int (*comp)(void *, void *)` 或 `int (*)(void *, void *)` 之类的写法比较长，而且不好理解。

用 `typedef` 有时会更短更好记，例如：

```
typedef int (*Cmp)(void *, void *);
```

实际上更准确的应该是：

```
typedef int (*Cmp)(const void *, const void *);
```

因为比较函数只会读，不会修改数据。

```
$ man 3 qsort
void qsort(void *base, size_t nel, size_t
width, int (*cmp)(const void *, const void *));
```

## hw1

### 1. 有几个警告需要消除：

```
$ gcc -Wall *.c
```

```
hw1-modyfiedpattern.c:10:7: warning:
assigning to 'char *' from 'const char *'
discards qualifiers [-Wincompatible-
pointer-types-discards-qualifiers]
```

```
    p = word;
```

```
    ^ ~~~~
```

```
hw1-modyfiedpattern.c:29:11: warning:
assigning to 'char *' from 'const char *'
discards qualifiers [-Wincompatible-
pointer-types-discards-qualifiers]
```

```
    p = word;
```

```
    ^ ~~~~
```

```
hw1-modyfiedpattern.c:59:18: warning:
using the result of an assignment as a
condition without parentheses [-
Wparentheses]
```

```
    while (c = *++argv[0]) {
```

```
        ^^ ^ ~~~~~
```

### 2. 函数 word\_in\_string() 功能不正确，注

释没有同步修改？

3. `if ((word_in_string(line, *argv) - wordcount >= 0) != except) {`，红色的部分是多余的？
4. `printf("Usage: find -x -n pattern\n");` 与实际的代码不匹配。