

TCPL第10-12课作业检查

class10

cw2.c

1. ln:127-128 为什么对左移右移操作符要多吃掉右边两个字符呢？

```
126 ✓      if ((c == '<') | (c == '>')) {  
127          d = getchar();  
128          d = getchar();  
129 ✓          if (d != EOF) {  
130              ungetch(d);  
131          }  
132          return c;  
133      }
```

2. pop() 函数返回值是个整数，所以
ln:109 返回 0.0 应该改成 0。

```
103  /* pop:  pop and return top value from stack */
104  int pop(void) {
105      if (sp > 0)
106          return val[--sp];
107      else {
108          printf("error: stack empty\n");
109          return 0.0;
110      }
111  }
```

如果再仔细思考一下，会发现 pop() 遇到错误后，调用者不能根据其返回值进行处理。

所以，简单一点，遇到错误 pop() 可以直接调用 abort() 退出程序。

3. 函数 gettop() 以及 atof() 可以处理小数，但是后面 push(), pop() 以及运算都只处理了整数。

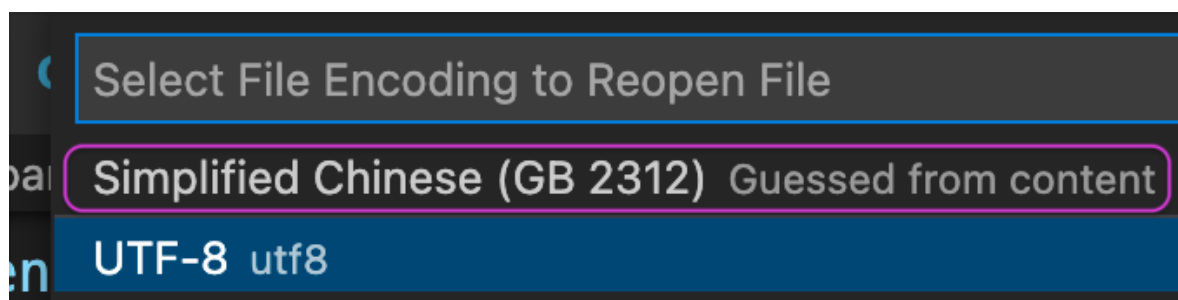
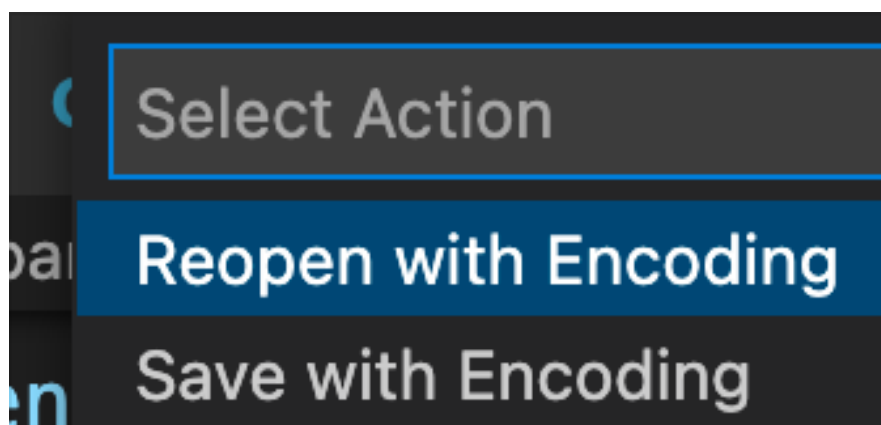
```
1 2 +
      3
0.1 0.2 +
      0
0.1 0.2 -
      0
0.1 1.2 *
      0
0.1 1.2 /
      0
0.1 2 <
      0
```

cw3.c

1. 画棋盘的 .c 文件的字符集是 GB2312, 需要在 VScode 中用 GB2312 打开。

Ln 36, Col 30 Tab Size: 4 **UTF-8** LF C

点击 UTF-8，在出现的菜单中选择：



但是输出的时候，因为操作系统默认的字符集是 UTF-8，会乱码。要么修改终端的默认字符集，要么把文件编码转换为 UTF-8。下面我们尝试把文件编码转换为 UTF-8，以保持跟系统默认的一致，避免其他各种潜在的问题。

1) 把原先的文件的字符集从 GB2312 转换

为 UTF-8

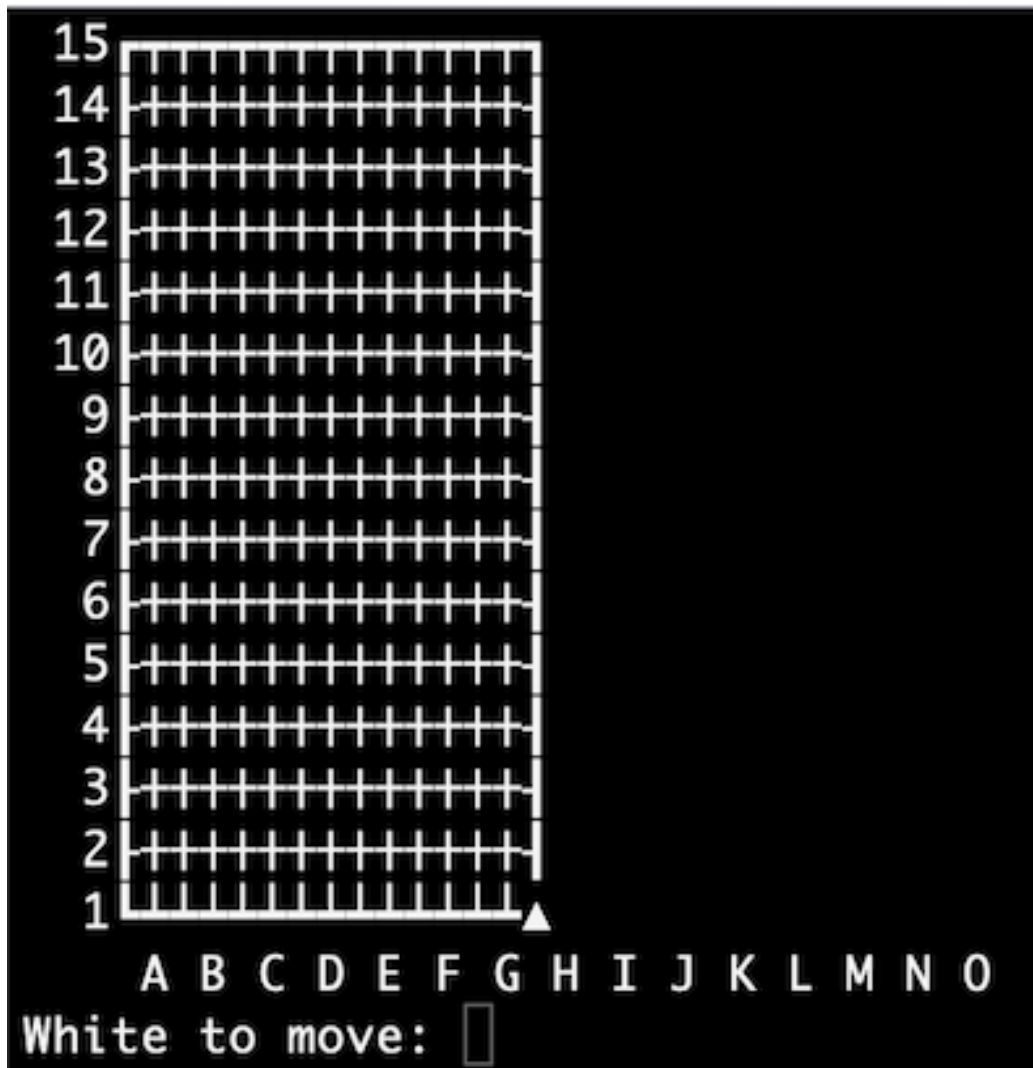
```
$ iconv -f GB2312 -t UTF-8 drawBoard-  
s1.c > draw.c
```

2) 注意，还需要修改代码中每个字符占用的字节数（从2改为3）

```
#define CHARSIZE 3 // UTF-8 编码时每个  
棋盘线的字符需要 3 个字节
```

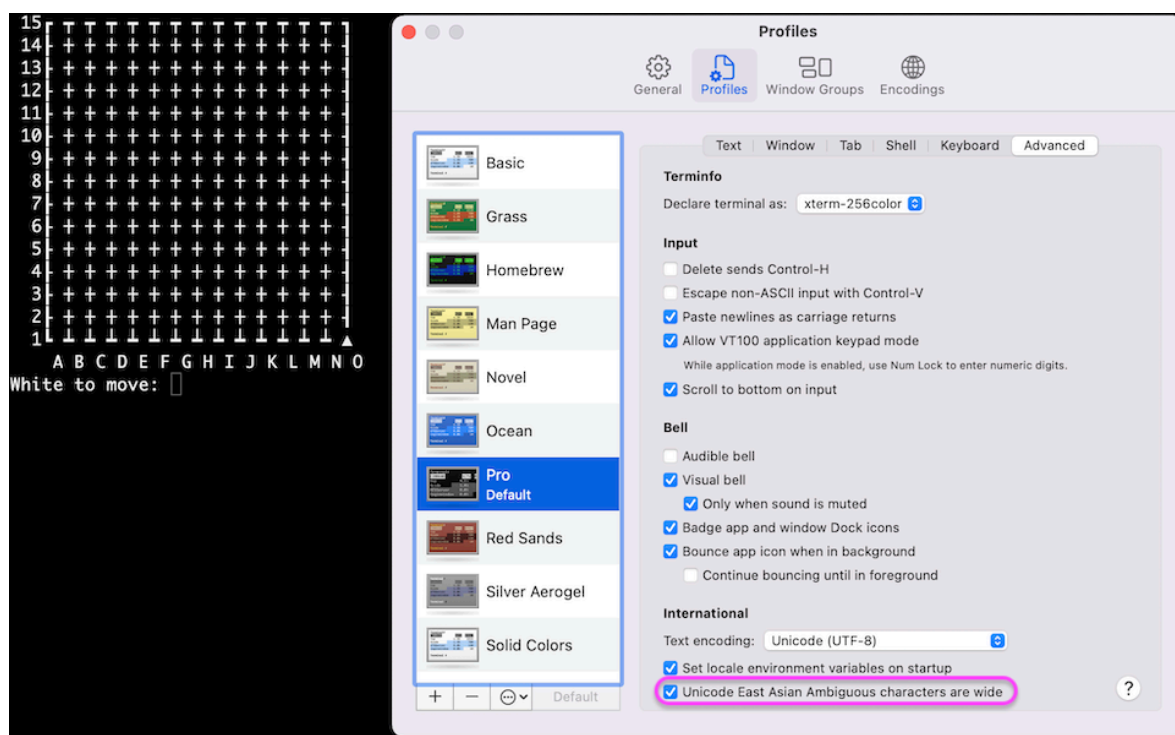
3) 运行的屏幕输出为：

```
$ gcc -Wall draw.c  
$ ./a.out
```



显然，输出的字符宽度不对，只有期望的一半。需要修改 Terminal 终端程序的设置。

4) 选中 Unicode East Asian Ambiguous characters are wide:



Ubuntu 的 Terminal 中也有类似的设置。

2. 函数命名风格不统一

- 1) 大部分是这种风格：initRecordBorard（Java 或某些 C/C++ 风格），注意最后的单词拼写错啦。
- 2) 少部分是这种风格：detect_whitewin（K&R C 跟这个类似）。这个程序可以统一成：detectWhiteWin。
- 3) getposition，按照 K&R C 风格，应该写成 get_position。这个程序可以写成：getPosition。

3. getPosition() 的改进

- 1) 它的参数已经定义为全局变量了，可以

去掉。当然更好的办法是不用全局变量。

2) 每次获取用户输入的 x 和 y 之前，需要将 x 初始化为非法的值，例如 `x[0] = '?'`。

这样在用户没有输入合法的值时，可以容易判断出来。

3) 试试在出现提示信息“Black to move:”时，直接按下 Ctrl+D 结束输入的效果。

4) 函数中规范化输入的 `x[0]` 为大写字母了，所以调用 `getPosition` 之后不再需要查小写字母了。

4. 代码中的很多处硬编码 15 应该换用 `SIZE` 宏。

5. 函数 `recordtoDisplayArray()`

1) 名字应该规范换为 `recordToDisplayArray`。

2) 代码中的四个 `if else` 几乎一样，可以想办法用一个函数替代。

```
updateDisplayArray(int i, int j, char  
*playPic) { }
```

6. 全局变量名字统一用大写字母打头。

7. 硬编码的 1 和 2 都用枚举类型 `Mode` 和 `Side` 的成员取代。

8. 全局变量 `x` 和 `y` 数组不需要，用局部变量 `x`

和y取代，不需要数组。

ex9.c

1. 文件名中的 recursive 拼错了。
2. 注意整数值域的不对称导致的问题。

```
[ $ ./a.out
please input an integer: -2147483647
-2147483647
[ $ ./a.out
please input an integer: -2147483648
--214748364C
```

- 1) if ($n < 0$) 应该放在 printf() 函数之外，调用最多一次。
- 2) $n = -n$ 在遇到 INT_MIN 时会溢出。
INT_MIN 在 <limits.h> 中定义，32 位整数就是 -2147483648。

ex10-qsort.c

1. 注意跟标准库里的 qsort() 函数对比一下。man qsort。
2. 如果输入的数据中有重复的数字，例如两个10，排序后它们的顺序跟原始的顺

序是不是一致的呢？

若排序算法可以保持相同值的原始顺序不变，则称为稳定排序，否则称为不稳定排序。

3. 算法中如果数据本来就是有序的，执行的速度会快一点吗？是不是做了很多无用比较或交换？

class11

1. 开始学习指针、宏了。

class12

1. 开始学习指针版本的 strcpy 等。

2. 注意字符数组和字符串的差别。