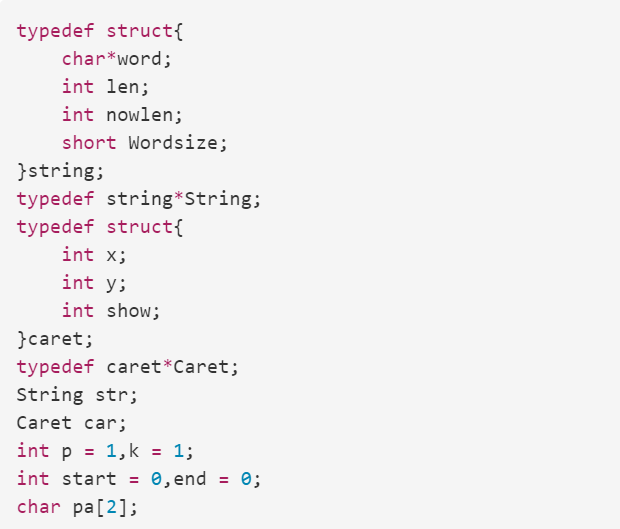
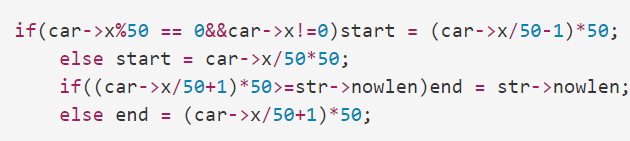


主函数：注册并初始化绘图和控制台窗口🡪程序初始化🡪监听键盘事件，进入消息循环



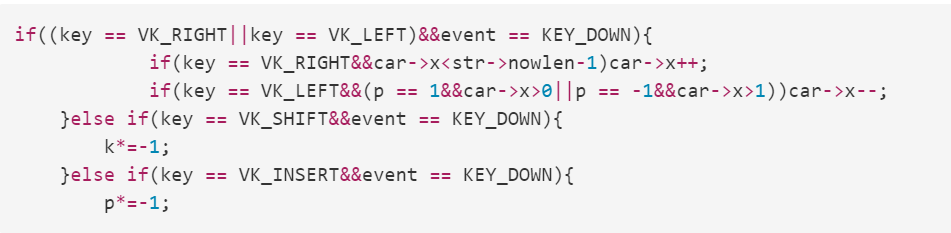
全局变量初始化：定义变长字符串结构对象string,光标对象caret；p/k指示shift/insert状态变化；start/end指示镜头显示范围；pa用于临时存储打印字符。



镜头位置控制：显示窗口宽600像素，字符宽12像素，镜头下最多显示50字符，第一个if判断是否翻页，第二个if控制未到达屏幕边缘的情况



删除：为新的字符串分配空间，按照光标“指针”指示位置进行删除，复制字符串，释放原有字符串空间，更新字符串状态。



指针移动，状态更新：移动光标指针，处理越界情况，当shift或insert键被按下，更新p\k状态。

实验体会和心得：

GUI程序与控制台程序的侧重点相对不同，GUI程序更加注重实际的呈现效果，不同的处理方式对于最终的效果有很大影响，比如在注册键盘回调函数时，先更新状态再绘图和先绘图再更新状态在直观的理解上应该没有太大差别，但经过试验后发现，先更新再绘图情况下，连续删除时会有明显的延迟，过程中画面会暂停更新。

其次是前端与后端的协调，因为键码值和ascii码值的不同，过程中采用了ctype.h中的toascii函数进行转化，但过程中会出现shift的状态切换，因此仍有许多部分需要手动进行修正。

个人认为GUI程序的用户体验是相当重要的，因此在光标的处理上，对insert的状态没有采用光标闪烁的机制，个人认为光标对要覆盖的字符的遮挡会降低用户体验。且本实验仍然存在bug，实际上光标的闪烁是有条件和时间限制的，6秒以上未进行操作，光标闪烁效果会停止，直到下次触发键盘事件。