**实验报告**

**Lab06 C Programming**

PB19030861 王湘峰

## 实验要求

1.将lab02至lab05用C语言实现

2.思考并写出低级语言和高级语言的区别

## 实验设计

对于实验2，采用辗转相处法求最小公约数，其中利用C语言的“%”进行带余除法。

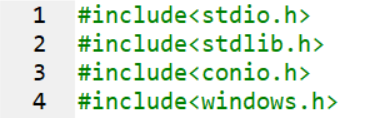
对于实验3，采用冒泡排序对链表进行排序，数据结构上使用结点的结构体实现。

对于实验4，利用C语言的标准输入输出函数进行读取操作。

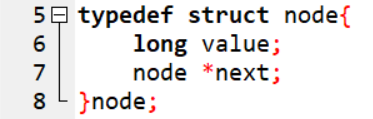
对于实验5，主要运用C语言windows.h中的kbhit（）函数进行中断检测。

## 代码部分

头函数：

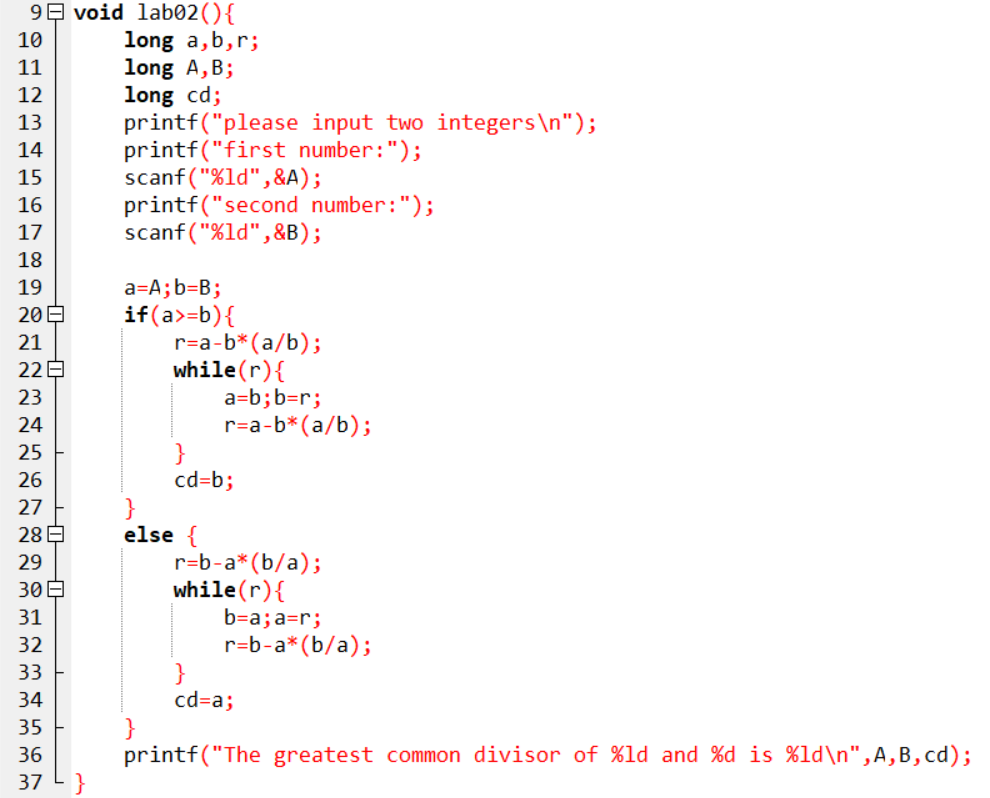


结构体：

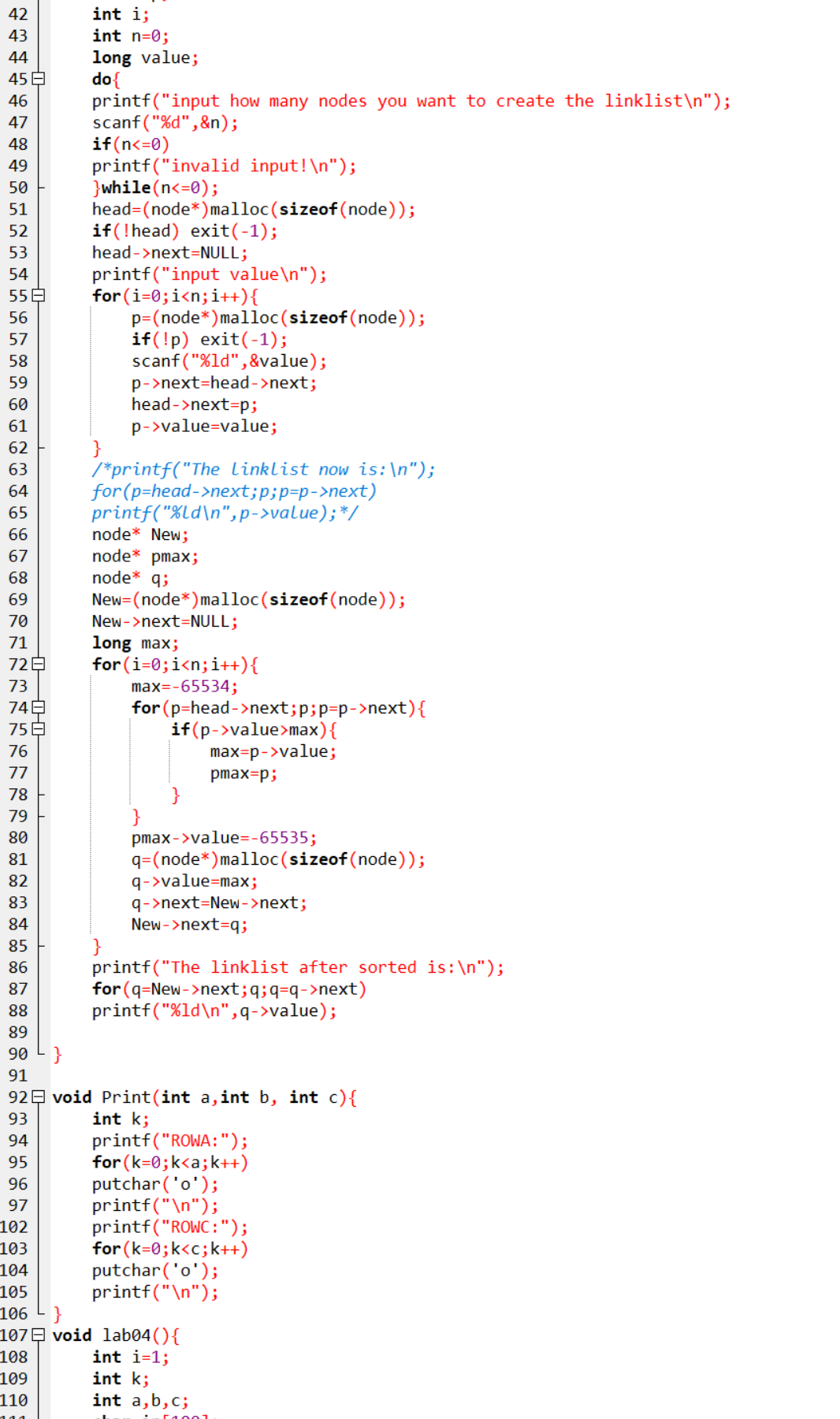


子函数：

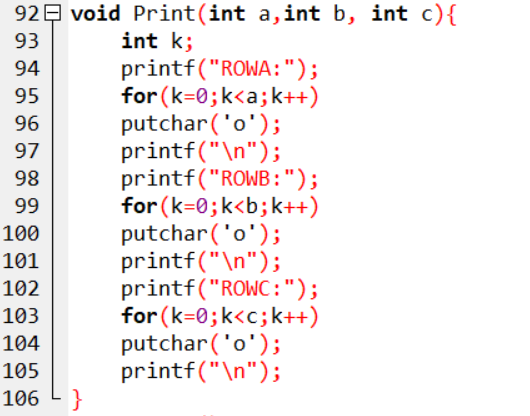
Lab02：



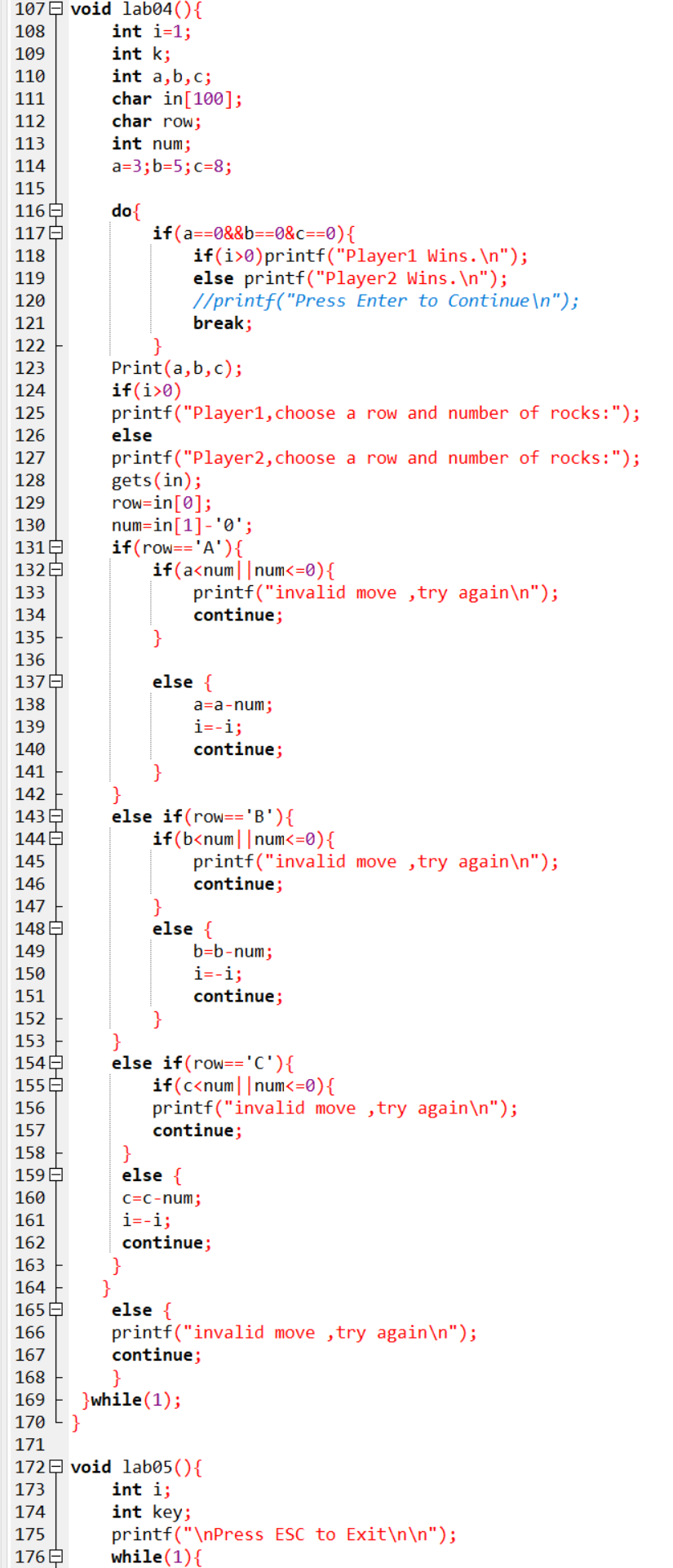
Lab03：



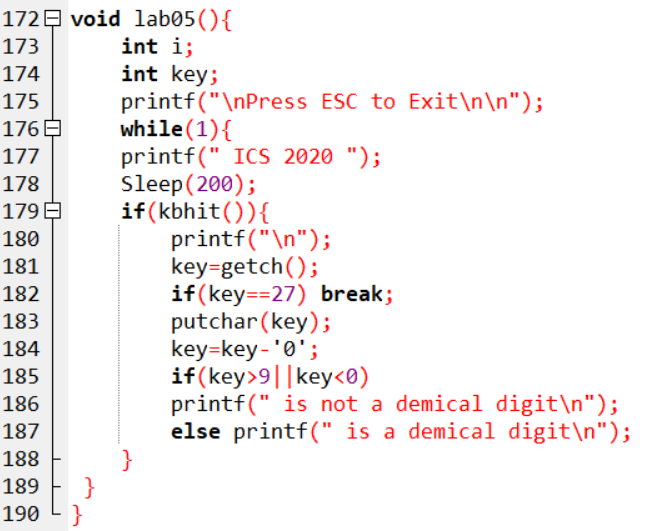
Print:



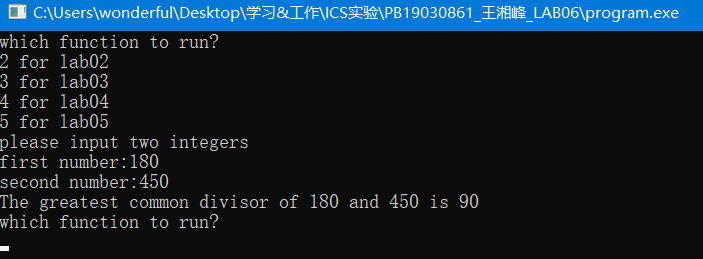
Lab 04:



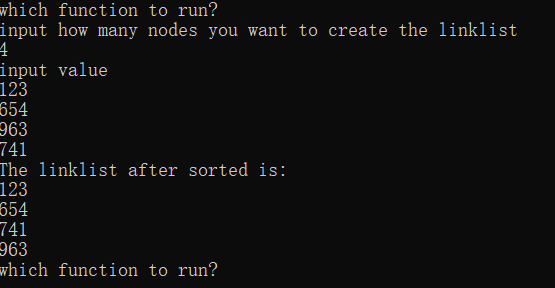
Lab05:



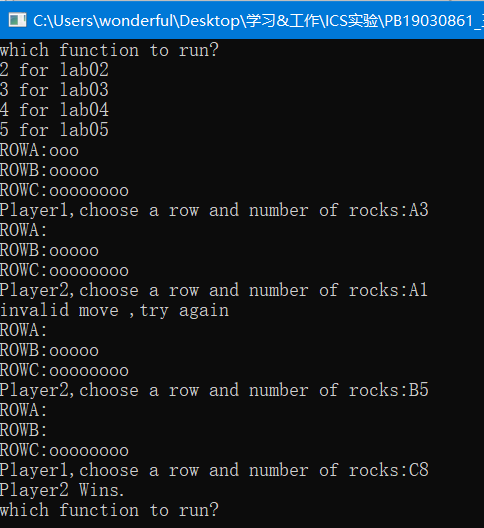
## 结果测试



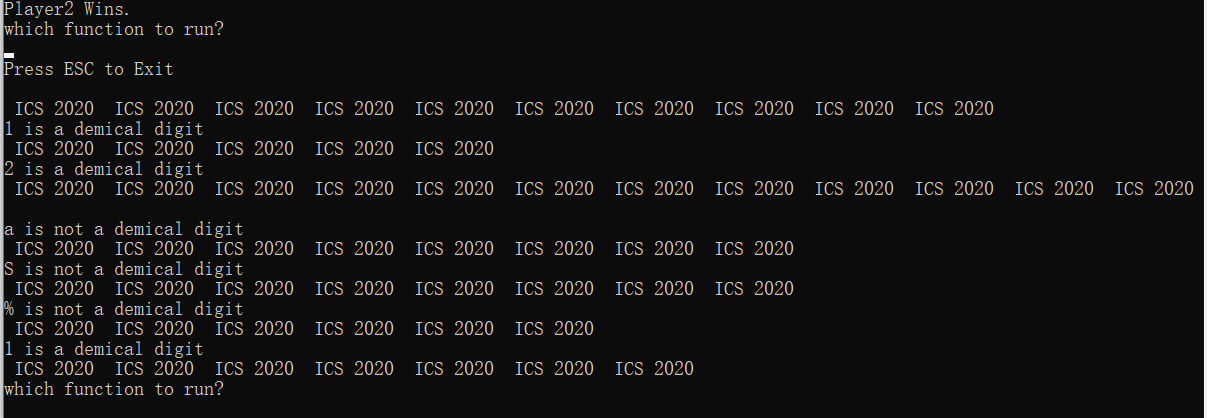
（test of lab02）



（test of lab03）



（test of lab04）



（test of lab05）

## 思考

高级语言例如C、C++、Python、Java等和低级语言相比，使用更多的指令（函数）实现，更利于程序的理解和记忆。在程序的行文上更接近人类的自然语言的习惯，因此在**编程时**更高效。最重要的是，高级语言易于移植，代码的复用性强。

低级语言如汇编语言、机器语言的优点是在**运行时**更高效。通过对汇编语言的直接编写，可大大简短机器执行指令的时间。而通过汇编器汇编的低级语言往往不够简洁，代码执行重复低效（相比于人类直接汇编）。但其缺点是程序较难思考、错误较为晦涩以及难以记忆，移植性也较弱。

综合来看，二者更有千秋。在实际问题的解决过程中，可根据具体情况综合使用二者。例如，对于汇编语言较为复杂的程序可以用高级语言的函数调用实现；而对那些对运算速度有较高要求的程序，则可人工编写汇编程序，大大缩短程序执行的速度。