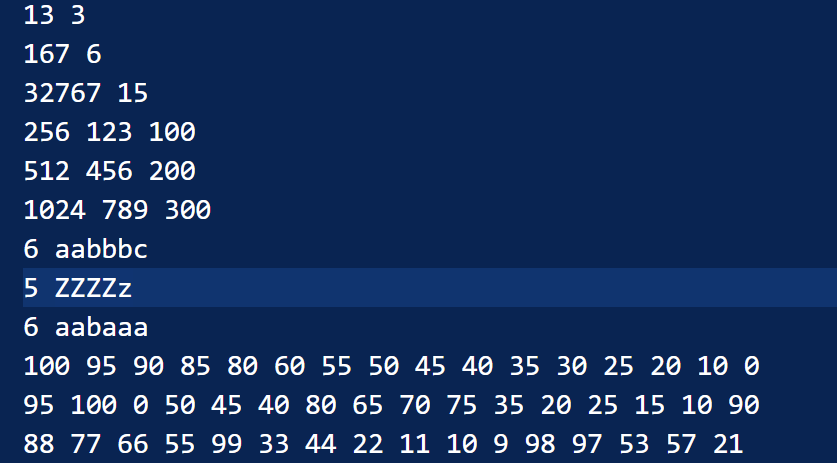
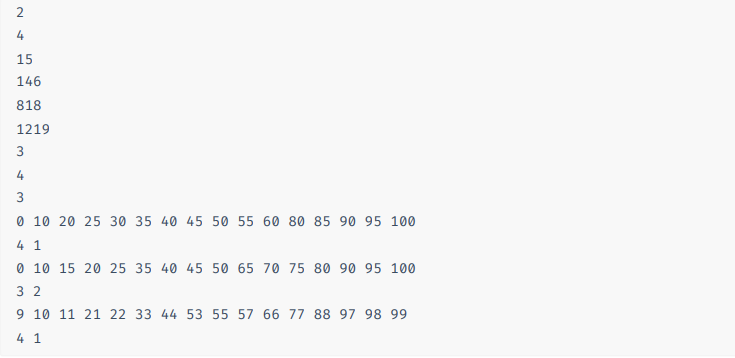
实验题目：Learn from the past

实验目的：本次实验需要用C语言写一个程序，程序的功能是将之前的实验1到实验4按照相同的思路使用C语言完成。

实验的部分测试数据如下：



output：

实验原理：

**代码文件与test.txt文件压缩一起提交**

**检查时test.txt文件地址请助教修改后运行**

实验一:1.根据要求不能使用lc-3中没有的操作，因此首先写出除法取整，取模，乘法，十进制转二进制存数组函数，便于直接使用；

2.lc-3中的移位操作通过直接从数组后B位开始判断，lc-3中判断高位是否为1的操作通过数组直接取出高位并直接比较实现；

实验二:1.同样取模函数自己编写调用；

2.使用while循环判断n—实现lc-3中的计数寄存器的值递减来判断需要进行的循环次数

实验三：1.lc-3的多次BR操作可以通过if—else语句轻松完成；

实验四：1.需要求得的结果较多，所以使用指针传递函数参数；

2.lc-3中较多的BR操作需要仔细判断先后顺序来保持代码BR跳转较少，使用if-else语句简单明了；

3.统计部分直接使数组从第九个开始判断，实现lc-3中结果地址指针从第九个开始的操作

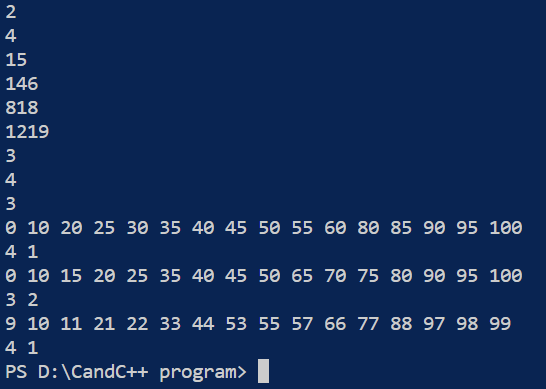
实验过程（debug部分）：

1.取模函数和除法函数出现了整除时错误的情况，原因是判断临界条件时没有算上等于；

2.lab2中开始没有使用f\_1\_t之类的临时存储变量，统一使用f\_1此类变量进行取模和更新，导致在被除数超过除数时，f\_1等中保存的是取模的值，并非本身，从而导致后续出错，添加临时储存变量即可解决；

3.lab4开始出现a，b人数都是0的情况，检查之后发现是（\*a）++和\*a++没有区分开，将传进去的地址加了，其值没有增加；

实验结果：

添加test.txt文件运行后，结果如下：

和给出的输出结果一致，正确

回答问题：

1.lc-3汇编语言和更高级语言的区别

感觉上最大区别在于操作的封装度，很多在高级语言例如C语言中可以轻易实现的操作，如取模，除法，乘法等在lc-3汇编语言中需要用较长一段代码来实现，另外如递归，循环等操作在lc-3汇编语言中实现起来不如高级语言中轻松，尤其是在判断条件很多时，如何合理安排跳转顺序在汇编中似乎成为一个比较值得思考的事情；

2.需要加入的操作

减法：很常用，每次都要取反加1再相加比较麻烦

取模：同减法，常用但每次使用需要一段代码

直接比较大小：每次使用BR操作都只能和0比较大小，比较麻烦

3.学习到的东西：

最主要是了解了高级语言的底层逻辑，大概知道计算机是如何“看懂”那些高级语言，将他们编译之后是怎么样的，又是如何执行这些代码的，学习lc-3能很好地帮助解决学习高级语言时知其然不知其所以然的情况