LAB 01

——**Grade Sorting with Arrays**

一．实验要求：

Background:

当地一所高中的老师 Alga Rythm ，需要你的帮助来决定她的学生的成绩。她想根据学生的级别和课程的分数来打分。她的标准如下：

•如果学生在班上排名前 30%，并且获得 85 分或以上，则该学生应获得 A。

•如果学生不符合 A 的要求，但在班上排名前 50%，且得分在 75 分或以上，则该学生应获得 B。

•如果学生获得 59 分或以下，则应获得 D。

•所有剩余的学生都应获得 C。

她班上有 60 个学生，每个学生都有一个分数。

Your job: 用汇编语言编写一个程序，把成绩分配给 Rythm 女士的班级。你的课程必须对分数列表进行排序，并计算出应获得 As 和 Bs 的学生人数。你的程序应该从 x3000 开始.

Program input:

全班 60 名学生的未分类分数。每个分数是 0 到 100 之间的数字，表示为 16 位无符号整数。该列表存储在 60 个连续的内存位置（每个位置一个分数），从地址 x3200 开始。最后一个分数在位置 x322B。不要假设分数的顺序。你可以假设所有的分数都是唯一的（没有两个分数可以相同）。

Program output:

程序必须产生两个输出：

1. 60 名学生的排序得分。分数必须按降序排序，从地址 x4000 开始存储在连续的内存位置（每个位置一个分数）。位置 x4000 将包含最高分数。
2. 获得 As、Bs、Cs 和 Ds 的学生人数。As 的数量必须存储

在 x4100 中，Bs 的数量必须存储在 x4101 中，Cs 的数量必须存储在 x4102 中，Ds 的数量必须存储在 x4103 中。

二．设计思路：

先将成绩在旧地址（x3200-x323B）排序，之后在转移至 x4000 的过程中对各个等级的人数进行统计

三．关键代码讲解：

首先对数据进行排序。选择了冒泡排序算法，counter 与 BEGIN1 在后面被 .FILL 为了 60 与 x3200（原始数据存放地址）
其中 LOOP1 与 LOOP2 为控制循环的两个 label，LOOP3 为控制是否进行交换的 label

```
        ;First,sort the students by their grade

        LD R3,counter           ;R3 is the counter
LOOP1   LD R2,BEGIN1            ;R2 is pointer to character
        ADD R4,R3,#0
LOOP2   LDR R1,R2,#0
        LDR R0,R2,#1
        NOT R5,R0
        ADD R5,R5,#1
        ADD R5,R5,R1
        BRP LOOP3
        STR R1,R2,#1
        STR R0,R2,#0
LOOP3   ADD R2,R2,#1
        ADD R4,R4,#-1
        BRP LOOP2
        ADD R3,R3,#-1
        BRP LOOP1
```

之后将数据写入从 x4000 开始的连续 60 个内存空间中，并且记录成绩>=85，75 ~ 84，60 ~ 74，0 ~ 59 四段区间内的人数

首先是统计成绩大于等于 85 分的人数：R2 与 R5 分别初始化为 x3200 与 x4000，R3 装入总人数 60，SCOREA 被 .FILL 为-85，装入 R0.

代码如下：

```
        ;then store the data and classify the students

        LD R3,counter
        LD R2,BEGIN1
        LD R5,BEGIN2


        LD R0,SCOREA
        ADD R4,R3,#0
LOOP4   LDR R1,R2,#0
        STR R1,R5,#0
        ADD R6,R1,R0
        BRN COUNTA
        ADD R2,R2,#1
        ADD R5,R5,#1
        ADD R3,R3,#-1
        BRP LOOP4

COUNTA  NOT R1,R3
        ADD R1,R1,#1
        ADD R4,R4,R1
        LD R0,adress_a
        STR R4,R0,#0
```

每次放入的数据都与 85 比较。因为此时数据已经排序完毕，所以找到第一个小于 85 的成绩后，即可求得成绩大于等于 85 分的人数

余下分数段的统计方法类似。当数据全部放置到从 x4000 开始的新位置后，我们也相应得到了各分数段的人数，暂时先放到 x4100 ~ x4103 中

最后根据各分数段人数统计 A,B,C,D 四个等级的人数：

假设 4 个分数段（85 ~ 100、75 ~ 84、60 ~ 74、0 ~ 59）人数分别为 x，y，m，n：

（1）若 x <= 18,转（2）；

若 x > 18 则 y = y +（x - 18），x = 18，转（2）

（2）若 y <= 30 – x,结束；

若 y > 30 – x ，则 m = m + y –（30 – x），y = 30 – x，结束

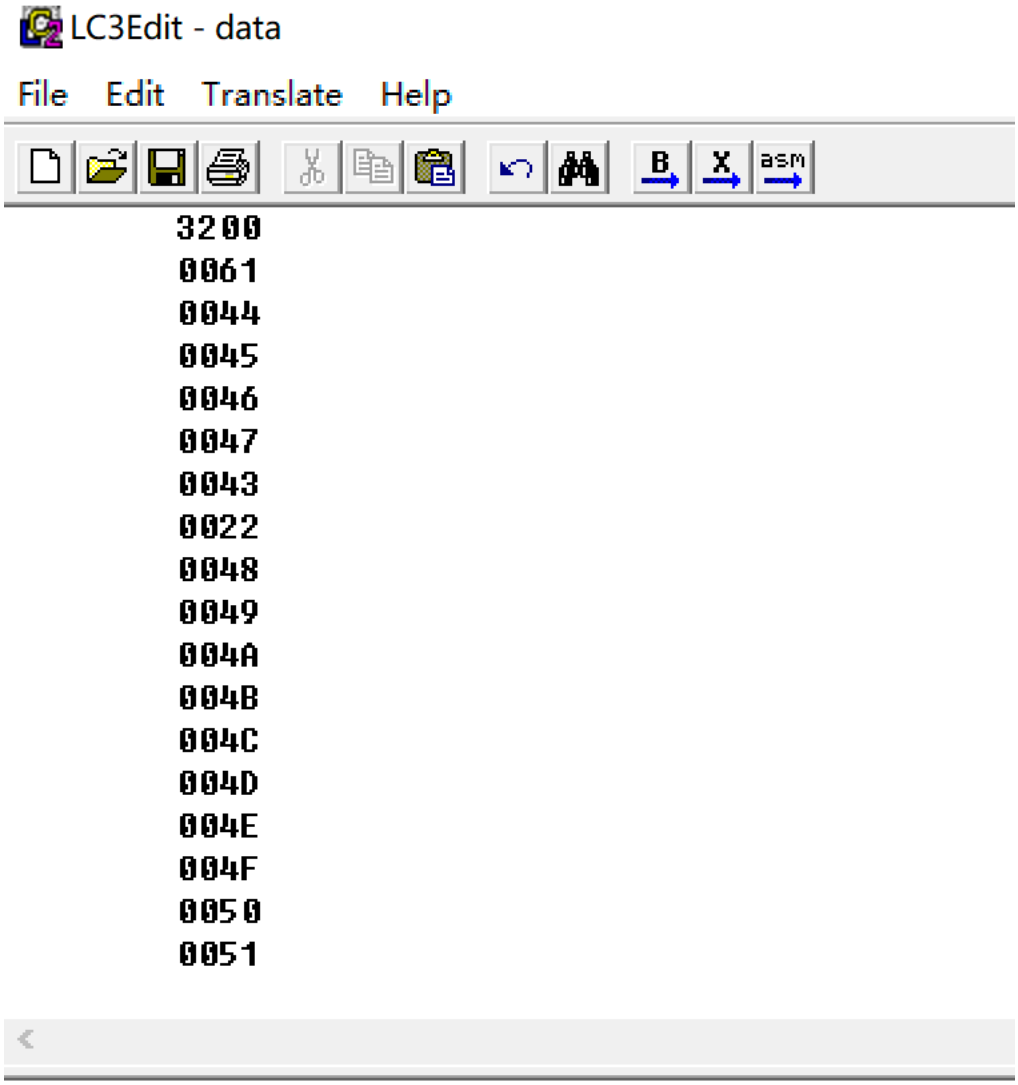
之后的 x，y，m，n 分别即为四个等级 As,Bs,Cs,Ds 的人数

具体实现代码如下：

```
        LDR R1,R0,#0
        LD R2,max_a
        NOT R1,R1
        ADD R1,R1,#1
        ADD R1,R2,R1
        BRN SITUATION1

LOOP8   LDR R1,R0,#1
        LDR R3,R0,#0
        LD R2,max_b
        NOT R1,R1
        NOT R3,R3
        ADD R1,R1,#1
        ADD R3,R3,#1
        ADD R2,R2,R3    ;R2 is max_b
        ADD R1,R2,R1
        BRN SITUATION2
LOOP9   HALT

SITUATION1      STR R2,R0,#0
                NOT R1,R1
                ADD R1,R1,#1
                LDR R2,R0,#1
                ADD R2,R1,R2
                STR R2,R0,#1
                BRNZP LOOP8

SITUATION2      STR R2,R0,#1
                NOT R1,R1
                ADD R1,R1,#1
                LDR R2,R0,#2
                ADD R2,R1,R2
                STR R2,R0,#2
                BRNZP LOOP9
```

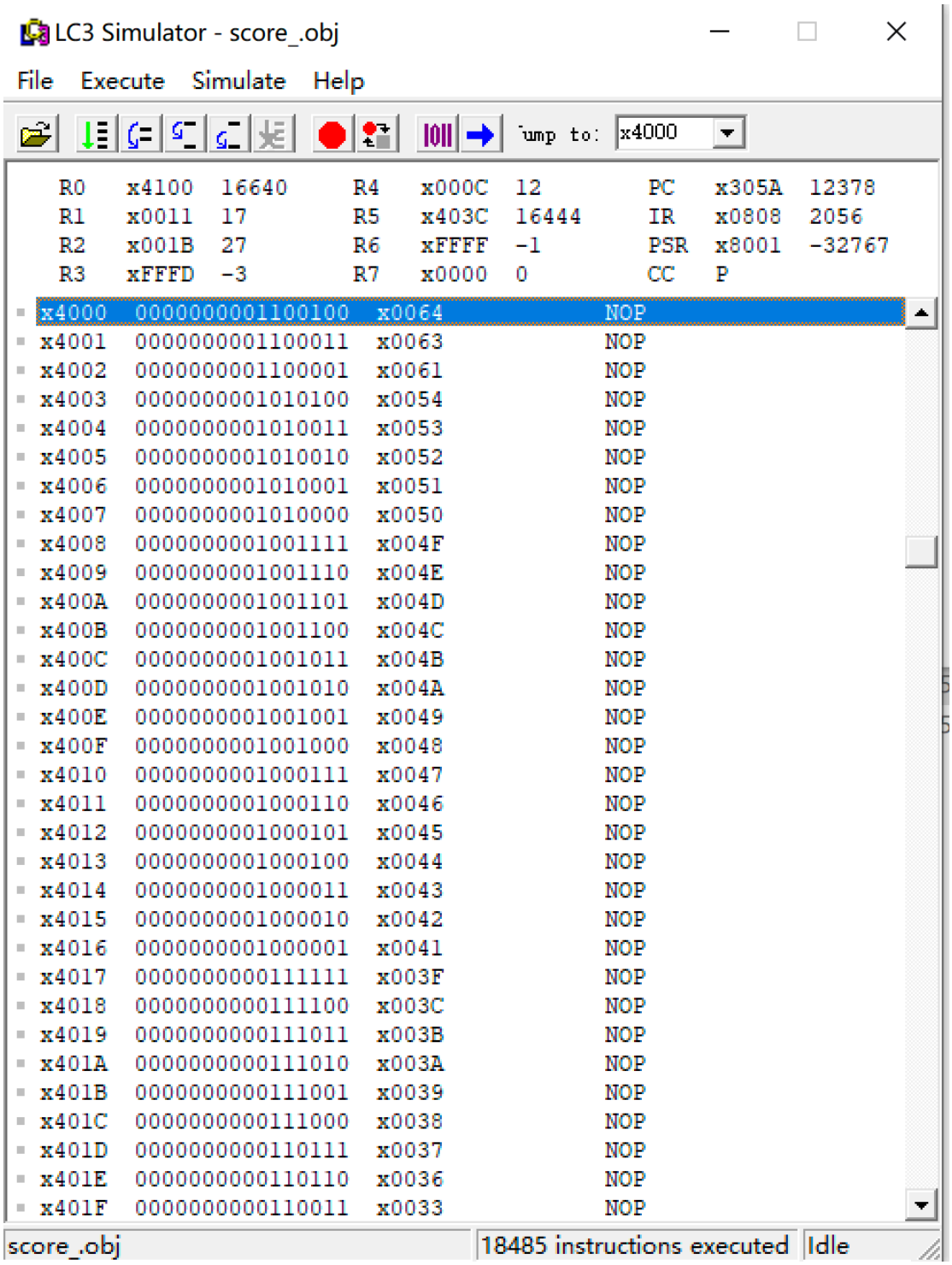
四．代码测试：

1. 首先输入一组随机生成的成绩：



之后在仿真器 simulator 中加载，并且通过程序进行处理，得到如下结果：

首先是分数的降序排序，从 x4000 开始，第一个分数最是高分 100（x0064）

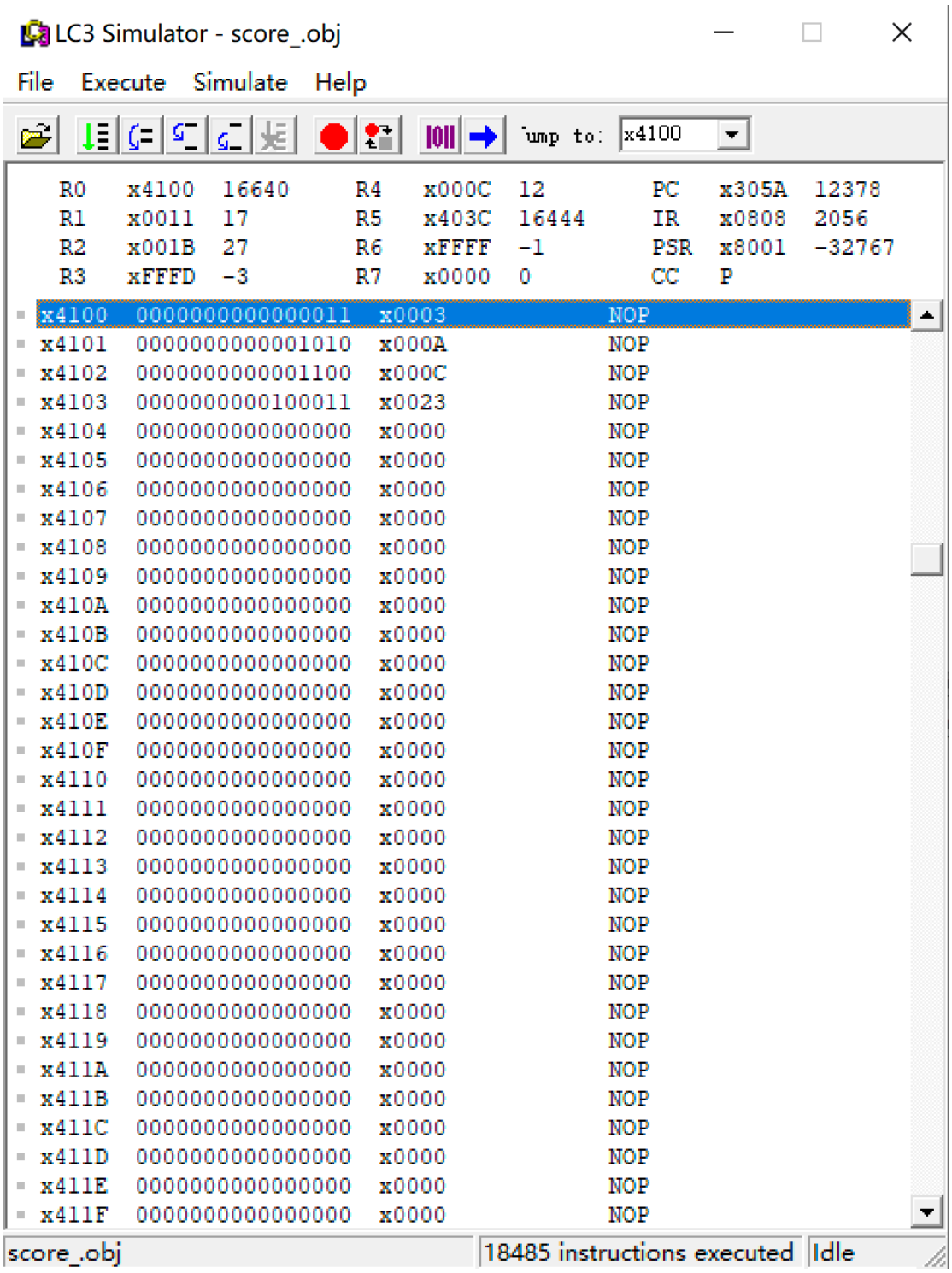


第二个输出是 As,Bs,Cs,Ds 四个等级的人数，分别占据 x4100 ~ x4103 四个位置：

A 等级人数为 x64，x63，x61 的 3 人

……

总人数为(x3+xA+xC+x23)=3+10+12+35=60 （人）



五．实验总结：

（1）熟悉了汇编语言的编写，以及计数法控制循环的具体实现

（2）编程中学习了 LC-3 中一系列伪操作的用法

六．附录：

“PB18071496_李昱祁_Lab01.asm”