

## Lab4 实验报告

### 一．实验目的

从连续的 16 个地址中读取 16 个学生的成绩，把它们排序后按照从小到大的顺序输出到另一处连续的 16 个地址中。根据成绩统计 16 人中获得 A 等和 B 等的人数，具体规则为：若成绩在所有人的前 25%并且不低于 85 分，则可以获得 A；若成绩在所有人的前 50%并且不低于 75 分，则可以获得 B。将获得 A 和 B 的人数分别输出到指定的地址。

### 二．实验原理

#### 1．读取存入

用 STR 和 LDR 指令将 x4000 及后续地址中的内容读取到寄存器并存入到 x5000 及后续地址中

#### 2．排序

排序过程采用冒泡排序法，R1 和 R2 分别作为冒泡排序中的 i 和 j ( $i=0; i < len; i++$ ) ( $j=0; j < len-1-i; j++$ ) 循环读取 R2 储存的地址指向的内存的内容以及此地址后一位的内存内容，比较大小，完成所有的排序，并按照从小到大存入从 x5000 开始的连续内存中。

#### 3．统计等级

从 x500f 开始，依次读取它及三个连续减 1 的地址中的内容，比较单个成绩与 85 的大小（通过成绩加上-85 的方式），做对应操作：

- (1) 若当前处理的数大于等于 85，计数加一，跳转处理下一个
- (2) 若此时还没有读到第四个数（正在处理第 1, 2, 3 个数），但是已经小于 85，那么直接结束 A 的判断，输出 A 的个数并进入 B 等级的判断
- (3) 若此时已经读到第四个数，完成了判断，输出 A 的个数并进入 B 等级的判断  
接下来进入 B 等级的判断，继续用上面 A 等级判断结束后的地址指针
- (1) 若当前处理的数大于等于 85，计数加一，跳转处理下一个
- (2) 若此时还没有读到第八个数，但是已经小于 75，那么直接结束 B 的判断，输出 B 的个数
- (3) 若此时已读到第八个数，输出 B 的个数

### 三．实验过程

实验中循环的边界条件需要注意，如果边界被错误地扩大了，比如冒泡排序时多比较了一位，那么就会导致有不需要地 0 被计入排序。这样会导致程序出错，无法正确比较。本来想写成在 x4000 及后续地址中先完成排序，再把已经排序完成的数据存入 x5000 及后续地址。但后来考虑到修改原始数据不知道测试时（例如使用 lc-3 工具或者测试网站）会不会影响结果，所以改为了先存入再排序（本来也应该这样做）。

冒泡排序我在直接用地址指针还是每次用 LDR 时都读取初始地址加寄存器记录的次数中纠结了以下，但是后者需要调用的寄存器太多，会导致后面统计 B 等级时寄存器不够用，所以最后就用地址指针，在操作中直接对地址指针作加减操作。

然后在统计等级时，因为我判断是否读取过四次、八次都是在子程序的最开头，但是对 (2) 的判断在子程序的末尾，所以对于如上的 (2)、(3)，虽然都是输出 A 的个数并转入 B 的判断，但是这两种情况下的地址指针一个是可以直接在下一步使用，一个是需要作加 1 操作才能使用，写的时候没有注意这一点。导致 A 等级有四个的样例都可以通过，而其他案例不能通过。后面另外补充了适合情况(2)的子程序后，输出正常。

### 四．实验结果

## 汇编评测

### 3 / 3 个通过测试用例

- 平均指令数: 1804.3333333333333
- 通过 100:95:90:85:80:60:55:50:45:40:35:30:25:20:10:0, 指令数: 1886, 输出: 0,10,20,25,30,35,40,45,50,55,60,80,85,90,95,100,4,1
- 通过 95:100:0:50:45:40:80:65:70:75:35:20:25:15:10:90, 指令数: 1767, 输出: 0,10,15,20,25,35,40,45,50,65,70,75,80,90,95,100,3,2
- 通过 88:77:66:55:99:33:44:22:11:10:9:98:97:53:57:21, 指令数: 1760, 输出: 9,10,11,21,22,33,44,53,55,57,66,77,88,97,98,99,4,1

## 汇编评测

### 3 / 3 个通过测试用例

- 平均指令数: 1820.3333333333333
- 通过 66:70:46:84:80:60:55:50:45:40:35:30:25:20:10:0, 指令数: 1835, 输出: 0,10,20,25,30,35,40,45,46,50,55,60,66,70,80,84,0,2
- 通过 66:70:46:85:71:60:55:50:45:40:35:30:25:20:10:0, 指令数: 1827, 输出: 0,10,20,25,30,35,40,45,46,50,55,60,66,70,71,85,1,0
- 通过 66:70:46:44:23:60:55:50:45:40:35:30:25:20:10:0, 指令数: 1799, 输出: 0,10,20,23,25,30,35,40,44,45,46,50,55,60,66,70,0,0