

## 实验报告

### 1.Report name:Lab04

### 2.Purpose:

实验共有两个部分:1) 对于 16 个学生的分数的数组进行排序;2) 统计学生中获得 A 与 B 的人数;

a. 其中学生的分数储存在 x4000 开始的 16 个地址中, 要求将排序后的结果存储在 x5000 开始的地址中

b. 对于学生分数达到 85 分及以上, 并且排名前 25%, 则被评为 A; 对于没有获得 A 的学生, 但分数在 75 及以上, 并且排名前 50%, 则被评为 B; 要求将达到 A,B 的人数分别储存在 x5100 和 x5101 中;

### 3.Principles:

1) 采取选择排序, 构成两个循环的嵌套, 分别取 R0,R1 作为两个指针, 在 R0 从前向后历遍时,R1 从 R0+1 处开始向后查找;

2) 选择其余中最小的值, 记录该地址 R6 与最小值 R2,R1 结束一次遍历后执行写入;R7 指向写入地址, 将 R2 写入, 并将 R0 指向的值交换到 R6 对应的最小值处, 便于后续比较;

3) 当比较结束时, 此时 x5000 到 x500F 已经储存从小到大的分数, 从后向前开始计算获得 A,B 的人数;

4) 首先循环四次, 找到满足 A 的人数, 对于不满足 A 的分数判断是否满足 B; 结束后再次循环四次, 计算满足 B 的人数, 分别存储在 R7,R6 中;

### 4.Procedure:

```
.ORIG x3000
```

```
LD R0, DATA
```

```
LD R7, DATAWR
```

```
ADD R0, R0, #-1
```

```
; 选择排序
```

```
CHOOSE ADD R0, R0, #1
```

```
LD R3, DATAOVER
```

```
NOT R4, R3
```

```
ADD R4, R4, #1
```

```
ADD R5, R0, R4
```

```
BRzp NEW
```

```
; 与 R3 比较判断是否选择结束
```

```
ADD R1, R0, #0
```

```
LDR R2, R0, #0
```

```
; 选择该轮循环需要比较的值
```

```
ADD R6, R0, #0
```

；初始化最小值地址

```
AGAIN ADD R1, R1, #1
      LD R3, DATAOVER
      NOT R4, R3
      ADD R4, R4, #1
      ADD R5, R1, R4
      BRzp WRITED
      ; 与 R3 比较判断是否结束该次循环
      LDR R3, R1, #0
```

```
COMPARE NOT R4, R2
      ADD R4, R4, #1
      ADD R5, R3, R4
      ; 比较 R2, R3
      BRzp AGAIN
      ADD R2, R3, #0
      ; 若 R3 小于最小值 R2, 更新最小值
      ADD R6, R1, #0
      ; 记录最小值地址, 以便交换
      BRnzp AGAIN
```

```
WRITED STR R2, R7, #0
      ADD R7, R7, #1
      ; 存储地址的指针向后偏移一位
      LDR R2, R0, #0
      STR R2, R6, #0
      ; 把 R0 指向的值交换到最小值处, 以便后续比较
      BRnzp CHOOSE
```

；开始计算获得 A,B 的人数

```
NEW LD R0, DATAWROVER
      AND R7, R7, #0
      AND R6, R6, #0
      ADD R2, R7, #15
      ADD R2, R2, #15
      ADD R2, R2, #15
```

```
ADD R2, R2, #15
ADD R2, R2, #15
ADD R2, R2, #10
;R2=85
NOT R2, R2
ADD R2, R2, #1
ADD R3, R7, #15
ADD R3, R3, #15
ADD R3, R3, #15
ADD R3, R3, #15
ADD R3, R3, #15
ADD R3, R3, #15
;R3=75
NOT R3, R3
ADD R3, R3, #1
ADD R4, R7, #4
```

```
COUNTA LDR R1, R0, #0
      ADD R4, R4, #-1
      BRn COUNTB
      ADD R5, R1, R2
      BRn COUNTAB
      ADD R7, R7, #1
      ; 对于分数在 85 以上且位于前 25%
```

```
COUNTAN ADD R0, R0, #-1
      BRnzp COUNTA
```

```
COUNTAB ADD R5, R1, R3
      BRn COUNTAN
      ADD R6, R6, #1
      BRnzp COUNTAN
      ; 对于分数在 75 以上 85 以下且位于前 25%
```

```
COUNTB ADD R4, R4, #5
COUNTT LDR R1, R0, #0
      ADD R4, R4, #-1
      BRn OVER
```

```
ADD R5, R1, R3
BRn COUNTBN
ADD R6, R6, #1
; 对于分数在 75 以上位于前 25 50%
```

```
COUNTBN ADD R0, R0, #-1
BRnzp COUNTT
```

```
OVER STI R7, NUMA
STI R6, NUMB
HALT
```

```
DATA .FILL x4000
DATAOVER .FILL x4010
DATAWR .FILL x5000
DATAWROVER .FILL x500F
NUMA .FILL x5100
NUMB .FILL x5101
.END
```

### 5.Result of test:

根据自测网站，评测结果如下：

#### 汇编评测

3 / 3 个通过测试用例

- 平均指令数: 1872
- 通过 100:95:90:85:80:60:55:50:45:40:35:30:25:20:10:0, 指令数: 1930, 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: 1843, 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: 1843, 输出: 9,10,11,21,22,33,44,53,55,57,66,77,88,97,98,99,4,1