lab 4 report

PB21081601 张芷苒

实验要求

使用 LC-3 汇编命令实现对 16 个人的成绩的升序排列，并求出这 16 个人中获得评级 A，B 的数量。 已知当该同学的成绩为 85 分及以上，且其排名为 25% 时，评级为 A。 当该同学的成绩低于 85 分但大于等于 75 分，且其排名为 50% 时，评级为 B。并且

- 16 个人的成绩存储在以x4000开始的连续内存空间中
- 每个人的成绩都在 0-100 之间，且每个人的成绩都不相同
- 排序后的成绩存储在以x5000开始的连续内存空间中
- 评级 A, B 的数量存储在内存位置x5100, x5101中

实验原理

使用选择排序的方式，既要对 16 个人的程 序进行升序排列，又要统计评级 A，B 的数量，已知 A，B 既有分数限制又有排名限制，所以如果在升序排序的同时判断评级，则会因无法判断其排名而得到正确结果。所以我们采用降序排序的同时判断评级，最后逆序存储序列。

实验步骤

选择排序的实现

对于通常情况下的选择排序，第二次循环起始位置应该为初始 max 的下一个位置，但若果这样做，会导致两个循环的判断终止条件不同，为了降低代码复杂度，设定第二次循环的起始位置为 max。

判断评级

设 num_a, num_b 为此时评级为 A, B 的数量。

1. num_a += 1 (score >= 85, num_a < 4)
2. num_b += 1 (score >= 85, num_a + num_b < 8, num_a >= 4)
3. num_b += 1 (75 <= score <=85, num_a + num_b < 8)

倒序存储

对降序排序的序列进行倒序存储，得到的即为正向的升序排列。

代码解释

1 初始化变量

将 R0 作为 16 个成绩的指针，R1 为循环的最后一个位置，R6, R7 存储评级 A，B 的数量。

```
LD R0, SCORE
ADD R1, R0, #15
AND R6, R6, #0
AND R7, R7, #0
```

2 外部循环

选择排序的外部循环，初始化 max, j 后进入内部循环，从内部循环返回后，进入 JUDGE 判断评级，然后将局部最大值写入相应位置，逆序存储结果，循环结束时，进入 STORE 存储评级结果。

```
LOOPA
AND R2, R2, #0
ADD R2, R0, #0 ; max = i
AND R3, R3, #0
ADD R3, R0, #0 ; j = i
BRnzp LOOPB
RETB
LDR R4, R2, #0
LDR R5, R0, #0
STR R4, R0, #0
STR R5, R2, #0 ; swap(score[i], score[max])
LD R3, RESULTS
LD R2, SCORE
ADD R3, R2, R3
NOT R2, R0
ADD R2, R2, #1
ADD R2, R2, R1
ADD R3, R3, R2
STR R4, R3, #0 ; sortScore[16 - i] = score[i]
BRnzp JUDGE
RETJ
ADD R0, R0, #1
NOT R4, R0
ADD R4, R4, #1
ADD R4, R1, R4 ; i < 16 ?
```

```
BRn STORE
BRnzp LOOPA
```

3 内部循环

选择排序的内部循环，若当前指针指向的值大于 max 指向的值，则更新 max。

```
LOOPB
LDR R4, R2, #0
LDR R5, R3, #0
NOT R4, R4
ADD R4, R4, #1
ADD R4, R5, R4 ; score[max] < score[j] ?
BRnz #2
AND R2, R2, #0 ; max = j
ADD R2, R3, #0
ADD R3, R3, #1
NOT R4, R3
ADD R4, R4, #1
ADD R4, R1, R4 ; j < 16 ?
BRzp LOOPB
BRnzp RETB
```

4 判断评级

若成绩不低于 85 且排名不低于 25%，则评级 A；若成绩不低于 85 且排名低于 25% 高于 50%，则评级 B; 若成绩低于 85 不低于 75 且排名低于 25% 高于 50%，则评级 B。

```
JUDGE
LDR R4, R0, #0
LD R5, SCOREMARKA
ADD R4, R4, R5 ; score[i] >= 85?
BRn #4
ADD R5, R6, #-4 ; num_a < 4?
BRz #4
ADD R6, R6, #1
BRnzp RETJ
ADD R4, R4, #10 ; score[i] >= 75?
BRn RETJ
ADD R5, R6, R7
ADD R5, R5, #-8 ; num_a + num_b < 8 ?
BRz RETJ
ADD R7, R7, #1
BRnzp RETJ
```

5 存储结果

分别将评级 A，B 的数量存储到对应地址。

```
STORE
STI R6, RESULTA
STI R7, RESULTB
```

实验结果

汇编评测

3 / 3 个通过测试用例

- 平均指令数: 2036.3333333333333
- 通过 100:95:90:85:80:60:55:50:45:40:35:30:25:20:10:0, 指令数: 2010, 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: 2043, 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: 2056, 输出: 9,10,11,21,22,33,44,53,55,57,66,77,88,97,98,99,4,1