

# 深圳大学实验报告

课程名称：计算机系统(1)

实验名称：Grade Sorting with Arrays

学院：数学与统计学院

指导老师：俞航

报告人：王曦      组号：无

学号：2021192010      实验地点：家中

实验时间：2022年04月30日

提交时间：2022年04月30日

# 1. 实验目的

在x3200开始的连续16个字中给出相异的、无序的16个学生成绩(0 ~ 100间的整数),将它们降序排列的结果存放在x4000开始的连续16个字中,统计并输出16个学生中获得等级A、B的人数.要求程序从x3000开始.

等级规则:

- ①前25%且成绩 $\geq 85$ 分的学生得A.
- ②前50%中没得A且成绩 $\geq 75$ 分的学生得B.

# 2. 实验内容

## 2.1 基本思路

因为程序从x3000开始,而测试数据从x3200开始,则这题的代码量不会很大.

## 2.2 排序

因要求排序结果存在x4000开始的连续空间,则可以改变x3200开始的连续空间中的数据.

考虑暴力排序.做16次循环,每次循环遍历一遍x3200~x320F,更新出当前的最大值 $curmax$ ,遍历一遍后再遍历一遍x3200~x320F,将等于 $curmax$ 的值置为0,因为成绩是0 ~ 100间的整数,所以这样做是reasonable的.下面的代码中找到一个等于 $curmax$ 的值并置零后即返回,显然这也适用于有重复成绩的情况.每次循环更新出 $curmax$ 后将其依次放到x4000开始的连续空间.

## 2.3 输入

用一个指针指向测试数据起始地址x3200,每次用LDR接收该指针指向的数,接收结束后指针+1,循环16次即可更新出所有的 $curmax$ .

用一个指针指向存放排序结果的起始地址x4000, $idx$ 记录当前放到哪个下标,每次更新出当前的 $curmax$ 后用STR将其存放到 $(x4000+idx)$ 处,然后 $idx++$ .

## 2.4 输出

因16个学生的前25%、前50%不超过4、8人,则只需输出单个字符即可.为区分两个字符,可在第一个字符输出后输出一个空格(ASCII码十进制23)分隔.

## 2.5 代码

```

1  ;init
2      .ORIG x3000
3      AND R0, R0, #0    ;0
4      AND R7, R7, #0    ;idx
5
6  ;main
7  READ  LD R4, BEGIN    ;pointer

```

```

8      AND R1, R1, #0      ;curmax
9      AND R3, R3, #0      ;i
10     ADD R3, R3, #10
11     ADD R3, R3, #6       ;16numbers
12     LOOP1 LDR R5, R4, #0  ;get the number
13     NOT R5, R5
14     ADD R5, R5, #1
15     ADD R5, R5, R1
16     BRp OK              ;R1>R5
17     LDR R1, R4, #0      ;update curmax
18     OK    ADD R4, R4, #1
19     ADD R3, R3, #-1
20     BRp LOOP1
21
22     LD R4, BEGIN
23     AND R3, R3, #0      ;i
24     ADD R3, R3, #10
25     ADD R3, R3, #6       ;16numbers
26     LOOP2 LDR R5, R4, #0  ;get the number
27     NOT R5, R5
28     ADD R5, R5, #1
29     ADD R5, R5, R1
30     BRz YES
31     ADD R4, R4, #1
32     ADD R3, R3, #-1
33     BRp LOOP2
34
35     YES    LDR R5, R4, #0
36     LD R6, SORTED
37     ADD R6, R6, R7      ;destination
38     STR R5, R6, #0      ;place curmax
39
40     STR R0, R4, #0      ;set curmax to 0
41     ADD R7, R7, #1
42
43     ADD R6, R7, #-16
44     BRz GET
45     BRnzp READ
46
47     GET    AND R1, R1, #0      ;resa
48     AND R2, R2, #0      ;resB
49     LD R4, SORTED
50     AND R3, R3, #0      ;i
51     ADD R3, R3, #4       ;the front 25%
52     LOOP3 LD R6, EIGHT      ;85
53     LDR R5, R4, #0      ;get the number
54     NOT R5, R5
55     ADD R5, R5, #1
56     ADD R6, R5, R6
57     BRp CHECKB          ;R5<85
58     ADD R1, R1, #1      ;resa++
59     NOB    ADD R4, R4, #1
60     ADD R3, R3, #-1     ;i--
61     BRp LOOP3
62
63     ;write resA
64     LD R0, ASCII
65     ADD R0, R0, R1      ;resA

```

```

66         OUT
67         LD R0, SPACE
68         OUT
69
70         ;getB
71         LD R4, SORTED
72         ADD R4, R4, #4    ;start at NO.5
73         AND R3, R3, #0    ;i
74         ADD R3, R3, #4    ;the front 25% ~ 50%
75     LOOP4 LD R6, SEVEN    ;75
76         LDR R5, R4, #0    ;get the number
77         NOT R5, R5
78         ADD R5, R5, #1
79         ADD R6, R5, R6
80         BRp NOBB
81         ADD R2, R2, #1    ;resB++
82     NOBB  ADD R4, R4, #1
83         ADD R3, R3, #-1   ;i--
84         BRp LOOP4
85
86         ;write resB
87         LD R0, ASCII
88         ADD R0, R0, R2    ;resB
89         OUT
90         BRnzp DONE
91
92     CHECKB LD R6, SEVEN    ;75
93         ADD R6, R5, R6
94         BRp NOB
95         ADD R2, R2, #1    ;resB++
96         BRnzp NOB
97
98     DONE  HALT
99
100 BEGIN   .FILL x3200
101 SORTED  .FILL x4000
102 EIGHT   .FILL #85
103 SEVEN   .FILL #75
104 ASCII   .FILL x0030
105 SPACE   .FILL #32
106
107         .END

```

## 2.6 时空复杂度分析

设有 $n$ 个成绩,本例中 $n = 16$ .

外层循环枚举 $n$ 次,内层循环之一遍历一遍x3200~x320F更新出当前的 $curmax$ ,内层循环之二将当前的 $curmax$ 之一置零,最坏的情况是原序列倒序,则每次的 $curmax$ 都在当前序列的最后,则内层循环之二依次要枚举 $n, n-1, \dots, 1$ 次,时间复杂度 $O(n^2)$ .排序后统计前 $\frac{n}{2}$ 名中等级A、B的人数,时间复杂度 $O(n)$ .总时间复杂度 $O(n^2)$ .

排序结果存在x4000开始的 $n$ 个连续空间中,总空间复杂度 $O(n)$ .

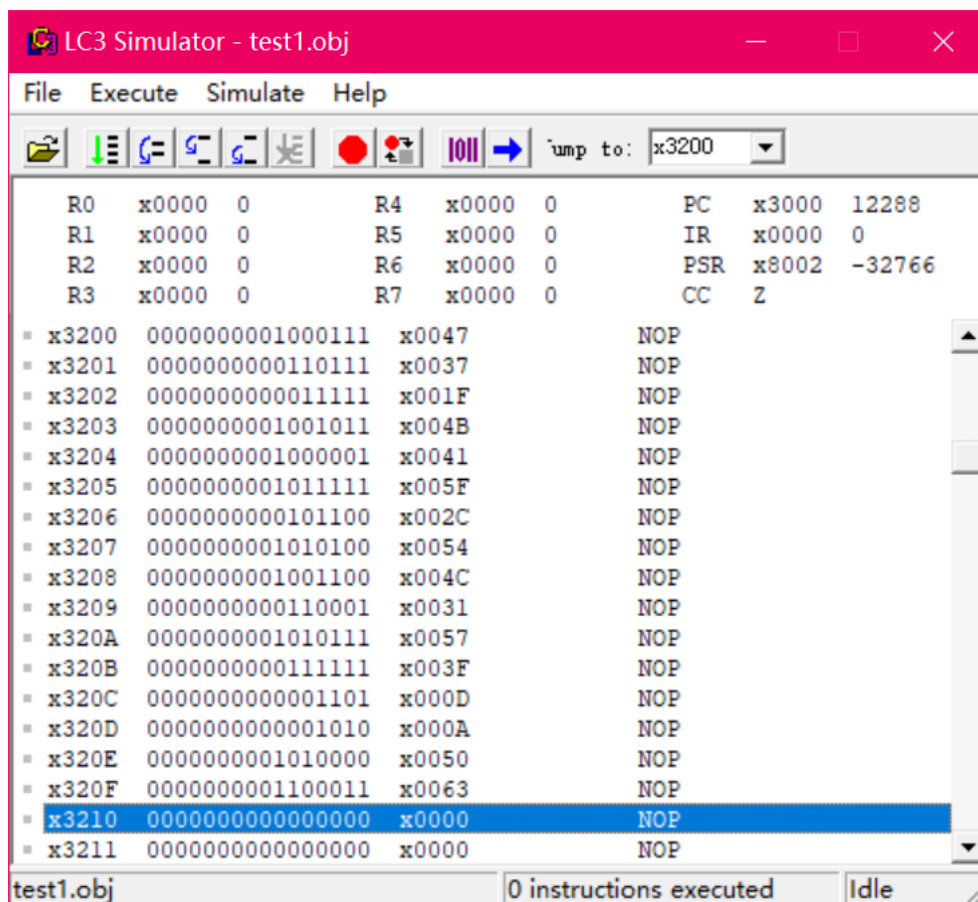
### 3. 实验结果

测试数据(左边为原数据,右边为排序结果):

十进制数	十六进制表示	十进制数	十六进制表示
71	x0047	99	x0063
55	x0037	95	x005F
31	x001F	87	x0057
75	x004B	84	x0054
65	x0041	80	x0050
95	x005F	76	x004C
44	x002C	75	x004B
84	x00534	71	x0047
76	x004C	65	x0041
49	x0031	63	x003F
87	x0057	55	x0037
63	x003F	49	x0031
13	x000D	44	x002C
10	x000A	31	x001F
80	x0050	13	x000D
99	x0063	10	x000A

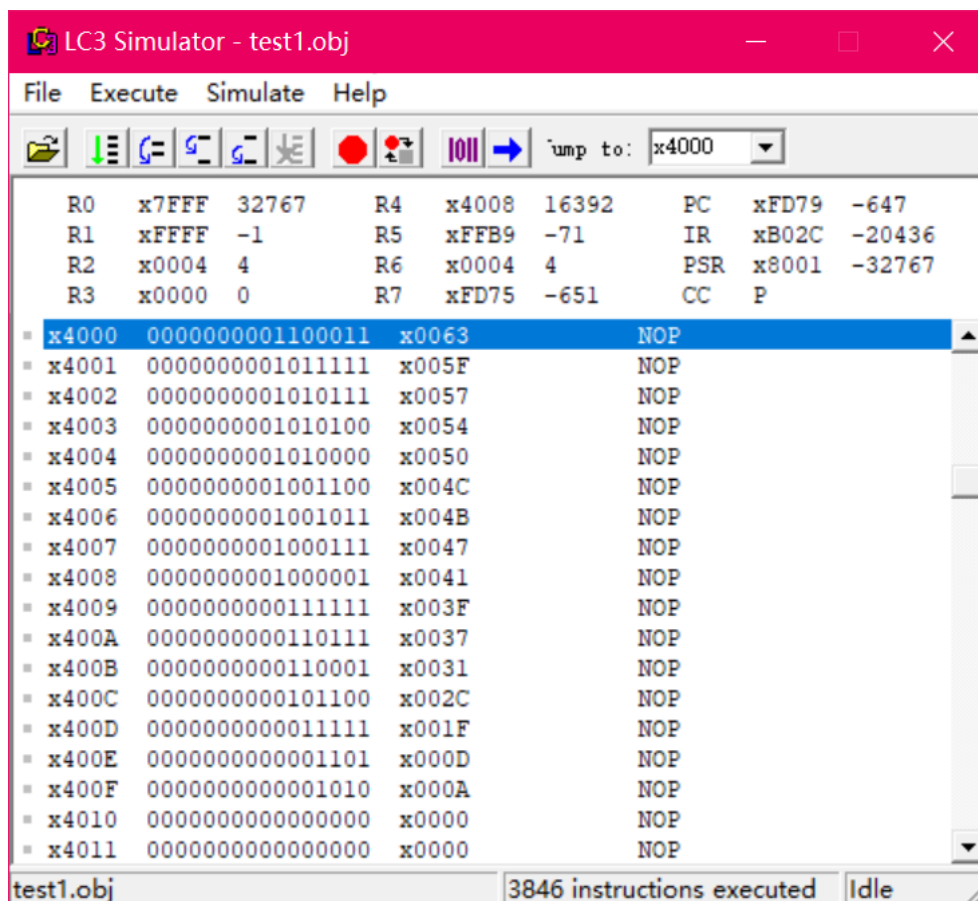
其中99、95、87得*A*,84、80、76、75得*B*.期望输出:3 4.

将原数据放入内存:



运行程序得到结果:

①排序结果:



②控制台输出:

```
LC3 Console
3 4
----- Halting the processor -----
```

结果与预期相符,Accepted!

## 4. 实验结论

1. 对存放在连续空间中的数据,可通过用指针指向其首地址,再指针不断+1来实现类似于遍历数组的操作.
2. 与一个固定值比较大小时,可先在一个地址用.FILL填充该比较数,使用时用LD将其读入到寄存器中,再将当前值与比较数作差(转化为补码做加法),根据结果的正负或零用BR指令跳转到相应的地方.
3. 适当分段、使用LABEL和添加COMMENTS可让代码思路更清晰.
4. 输出数字时,应将其先存到寄存器 $R_0$ 中, $R_0$ 加上'0'的ASCII码x0030后再用OUT输出,因为OUT只能输出 $R_0$ 中的值.
5. 使用寄存器前注意清空.
6. 程序出现bug时添加断点方便调试.