

## 第十四章习题参考答案

包含题目：习题 14.7 – 14.10

**题目 14.7** 假设位于存储单元 NUM 中的值可以是任意一个大于 2 的 32 位的正整数。如下程序实现了什么？

```

1      .data    x30000000
2  NUM:      .word    #201
3  RESULT:   .space   #4
4  ;
5      .text    x40000000
6      .global  main
7  main:     addi     r1, r0, #0
8           addi     r2, r0, #2
9           lw       r3, NUM(r0)
10 ;
11 LOOP:     jal      MOD
12           beqz     r6, EXIT0
13           addi     r2, r2, #1
14           seq      r6, r2, r3
15           bnez     r6, EXIT1
16           j        LOOP
17 ;
18 EXIT1:     addi     r1, r1, #1
19 EXIT0:     sw       RESULT(r0), r1
20           trap     x00
21 ;
22 MOD:       addi     r4, r3, #0
23 AGAIN:     sub      r4, r4, r2
24           slei     r6, r4, #0
25           beqz     r6, AGAIN
26           jr       r31

```

**解答** 将上述汇编代码写成伪代码如下

```

1  r1 = 0; r2 = 2; r3 = NUM;
2  Do{
3      MOD()
4      If (r6 == 0)
5          break;
6      r2 = r2 + 1;
7      r6 = r2 == r3;
8      If (r6 == 1) {
9          r1 = r1 + 1;
10         break;
11     }
12 } while(1);
13 RESULT = r1;
14 MOD(){

```

```

15     r4 = r3;
16     do{
17         r4 = r4 - r2;
18         r6 = r4 <= 0;
19     } while(r6 == 0);
20 }

```

由于调用子例程 MOD 结束后 r6 始终是 1，所以不会从 EXIT0 直接退出，实现了  $r1 = 1$ 。

**题目 14.8** 假设从存储单元 DATA 开始存储的 10 个值可以是任意的 10 个 32 位的整数。如下程序实现了什么？

```

1      .data    x30000000
2  DATA:      .word    #3, #14, #35, #47, #5, #20, #12, #14, #6, #22
3  SaveR31:    .space    #4
4
5      .text    x40000000
6      .global  main
7  main:      addi     r1, r0, DATA
8             addi     r2, r0, #9
9  OutLoop:   beqz     r2, EXIT
10            addi     r3, r2, #0
11  InnerLoop: lw      r4, 0(r1)
12            lw      r5, 4(r1)
13            jal      CMP
14            addi     r1, r1, #4
15            subi     r3, r3, #1
16            bnez     r3, InnerLoop
17            slli     r6, r2, #2
18            sub      r1, r1, r6
19            subi     r2, r2, #1
20            j        OutLoop
21  EXIT:      trap     x00
22
23  CMP:       sw       SaveR31(r0), r31
24            slt      r6, r4, r5
25            bnez     r6, Return
26            jal      SWAP
27  Return:    lw       r31, SaveR31(r0)
28            ret
29
30  SWAP:      sw       4(r1), r4
31            sw       0(r1), r5
32            ret

```

**解答** 将上述代码翻译如下

```

1  r1 = [DATA]; r2 = 9;
2  While (r2 != 0){
3      r3 = r2;

```

```

4      Do{
5          r4 = r1[0];
6          r5 = r1[1];
7          CMP();
8          r1 = r1 + 4;
9          r3 = r3 - 1;
10     } while(r3 != 0);
11     r6 = r2 * 4;
12     r1 = r1 - r6;
13     r2 = r2 - 1;
14 }
15 CMP(){
16     r6 = r4 < r5;
17     if (r6 == 0){
18         SWAP();
19     }
20 }
21 SWAP(){
22     r1[1] = r4;
23     r1[0] = r5;
24 }

```

比较相邻两个数，前者不小于后者就交换，每次 OutLoop 选出一个最大数，实现排序。

**题目 14.9** 如下程序将存储单元 NUM 中的数值显示在屏幕上，但是不能正常工作。请找出原因并修复。

```

1      .data    x30000000
2 NUM:   .word   #8
3
4      .text    x40000000
5      .global  main
6 main:   jal     SubA
7         trap   x07
8         trap   x00
9
10 SubA:   lw      r1, NUM(r0)
11         jal     SubB
12         ret
13
14 SubB:   addi    r4, r1, x30
15         ret

```

**解答** 嵌套子程序中上一层必须保存 r31，防止被下一层修改。可以通过在调用 jal SubB 之前保存 r31，调用之后恢复 r31 来修复。

**题目 14.10** 如下程序实现了什么？

```

1      .data    x30000000
2 STACK: .space #100

```

```

3 PROMPT: .ascii "Please enter your string: "
4 ;
5     .text    x40000000
6     .global  main
7 main:  addi   r4, r0, PROMPT
8        trap  x08
9        addi   r3, r0, STACK
10       addi   r5, r3, #100
11       addi   r29, r5, #0
12 Input: seq    r2, r29, r3
13       bnez   r2, Output
14       trap  x06
15       trap  x07
16       seqi   r2, r4, #10
17       bnez   r2, Output
18       jal    PUSH
19       j      Input
20 ;
21 Output: seq    r2, r29, r5
22        bnez   r2, DONE
23        jal    POP
24        trap  x07
25        j      Output
26 ;
27 DONE:  trap  x00
28 ;
29 PUSH:  subi   r29, r29, #4
30        sw     0(r29), r4
31        ret
32 ;
33 POP:   lw     r4, 0(r29)
34        addi   r29, r29, #4
35        ret

```

解答 将上述代码翻译如下

```

1  printf("Please enter your string:");
2  r3 = [STACK]; r5 = r3 + 100; r29 = r5;
3  While (1){
4      r2 = r29 == r3;
5      if (r2 == 1)
6          Output();
7      scanf();
8      printf();
9      r2 = r4 == 10;
10     if (r2 == 1)
11         Output();
12     PUSH();
13 }

```

```
14 Output(){
15     r2 = r29 == r5;
16     if (r2 == 1)
17         exit;
18     POP();
19     printf();
20     Output();
21 }
22 PUSH(){
23     r29 = r29 - 4;
24     r29[0] = r4;
25 }
26 POP(){
27     r4 = r29[0];
28     r29 = r29 + 4;
29 }
```

带回显地输入以换行符为结束标记的字符串（最长 25），再以反序输出到屏幕上。