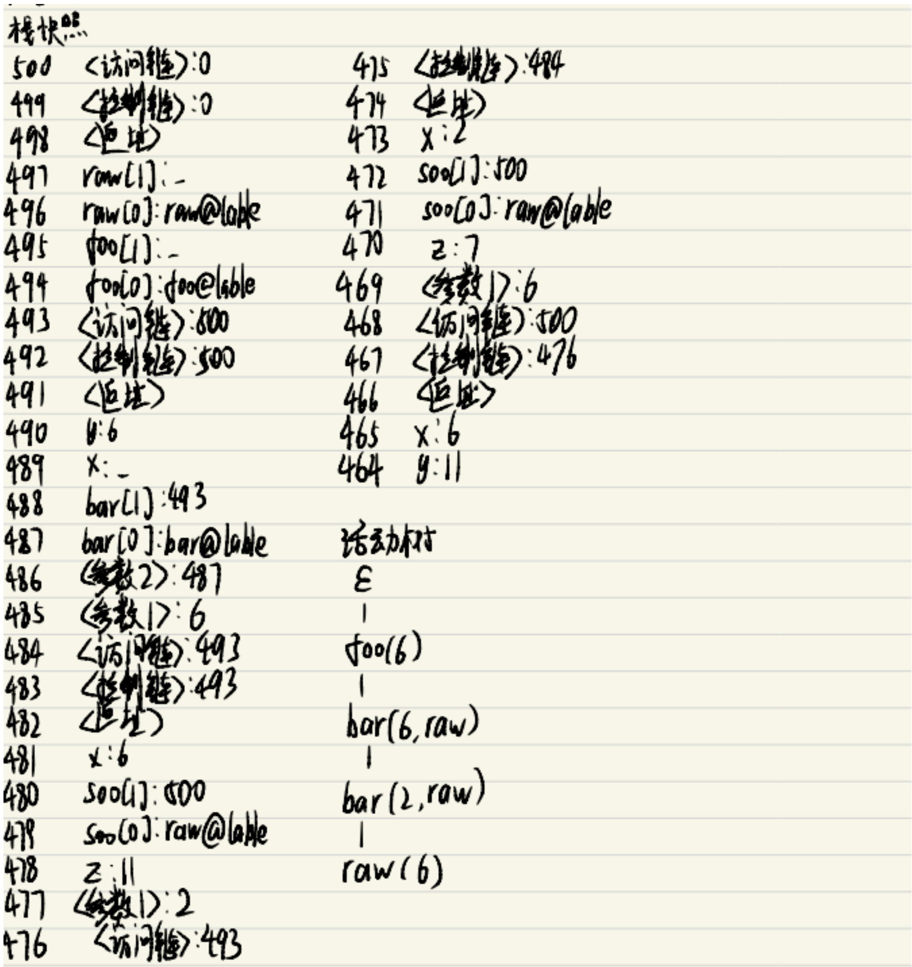
形式语言与编译作业十 参考答案

2025 春

1 习题 01

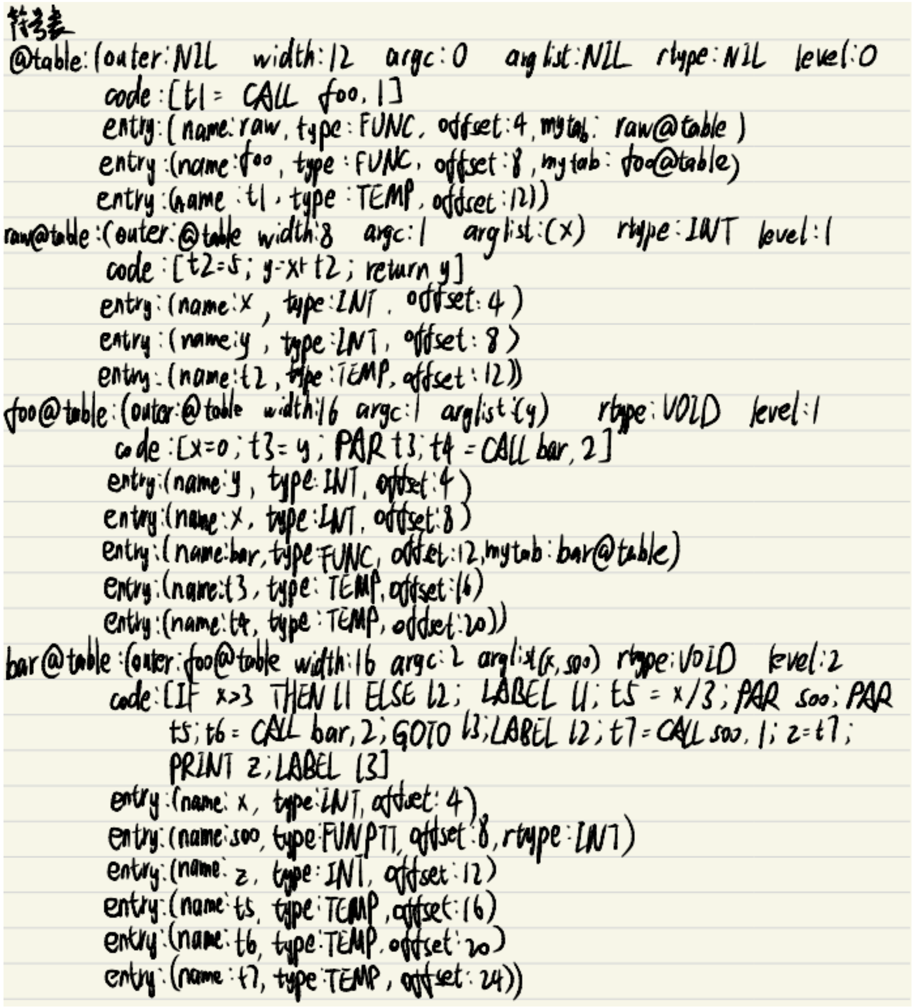


栈快照

500	〈访问链〉:0
499	〈控制链〉:0
498	〈返址〉
497	raw[1]:_
496	raw[0]:raw@lable
495	foo[1]:_
494	foo[0]:foo@lable
493	〈访问链〉:500
492	〈控制链〉:500
491	〈返址〉
490	y:6
489	x:_
488	bar[1]:493
487	bar[0]:bar@lable
486	〈参数2〉:487
485	〈参数1〉:6
484	〈访问链〉:493
483	〈控制链〉:493
482	〈返址〉
481	x:6
480	soo[1]:500
479	soo[0]:raw@lable
478	z:11
477	〈参数1〉:2
476	〈访问链〉:493
475	〈控制链〉:484
474	〈返址〉
473	x:2
472	soo[1]:500
471	soo[0]:raw@lable
470	z:7
469	〈参数1〉:6
468	〈访问链〉:500
467	〈控制链〉:476
466	〈返址〉
465	x:6
464	y:11

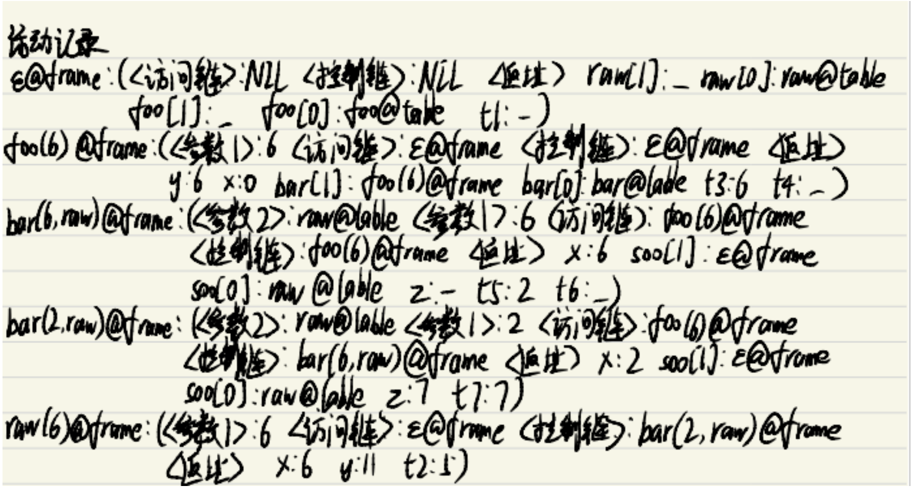
活动树

ε
|
foo(6)
|
bar(6,raw)
|
bar(2,raw)
|
raw(6)



符号表

```
@table:(outer:NIL  width:12  argc:0  arglist:NIL  rtype:NIL  level:0
        code:[t1 = CALL foo, 1]
        entry:(name:raw, type:FUNC, offset:4, mytab: raw@table)
        entry:(name:foo, type:FUNC, offset:8, mytab: foo@table)
        entry:(name:t1, type:TEMP, offset:12))
raw@table:(outer:@table width:8  argc:1  arglist:(x)  rtype:INT  level:1
        code:[t2=5; y=x+t2; return y]
        entry:(name:x , type:INT , offset:4 )
        entry:(name:y , type:INT , offset:8 )
        entry:(name:t2 , type:TEMP, offset:12))
foo@table:(outer:@table  width:16 argc:1  arglist:(y)  rtype:VOID  level:1
         code:[x=0; t3=y; PAR t3; t4 = CALL bar, 2]
         entry:(name:y , type:INT, offset:4 )
         entry:(name:x , type:INT, offset:8 )
         entry:(name:bar, type:FUNC, offset:12, mytab:bar@table)
         entry:(name:t3 , type:TEMP, offset:16)
         entry:(name:t4, type:TEMP, offset:20))
bar@table:(outer:foo@table width:16 argc:2 arglist(x, soo) rtype:VOID  level:2
         code:[IF x>3 THEN l1 ELSE l2;  LABEL l1; t5 = x/3; PAR soo; PAR
              t5; t6 = CALL bar, 2; GOTO l3; LABEL l2; t7 = CALL soo, 1; z=t7;
              PRINT z; LABEL l3]
         entry:(name: x , type:INT, offset: 4)
         entry:(name:soo, type:FUNPTT, offset:8, rtype:INT)
         entry:(name: z , type:INT, offset:12)
         entry:(name:t5, type:TEMP , offset:16)
         entry:(name:t6, type:TEMP, offset:20)
         entry:(name:t7, type:TEMP , offset:24))
```



活动记录

ε@frame:(<访问链>:NIL <控制链>:NIL <返址> raw[1]:_ raw[0]:raw@table
foo[1]:_ foo[0]:foo@table t1:-)

foo(6)@frame:(<参数1>:6 <访问链>:ε@frame <控制链>:ε@frame <返址>
y:6 x:0 bar[1]:foo(6)@frame bar[0]:bar@lable t3:6 t4:_)

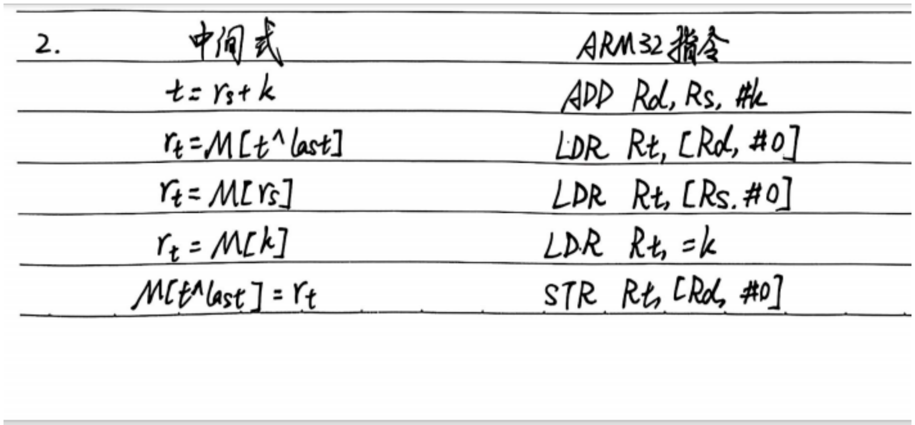
bar(6,raw)@frame:(<参数2>:raw@lable <参数1>:6 <访问链>:foo(6)@frame
<控制链>:foo(6)@frame <返址> x:6 soo[1]:ε@frame
soo[0]:raw@lable z:- t5:2 t6:_)

bar(2,raw)@frame:(<参数2>:raw@lable <参数1>:2 <访问链>:foo(6)@frame
<控制链>:bar(6,raw)@frame <返址> x:2 soo[1]:ε@frame
soo[0]:raw@lable z:7 t7:7)

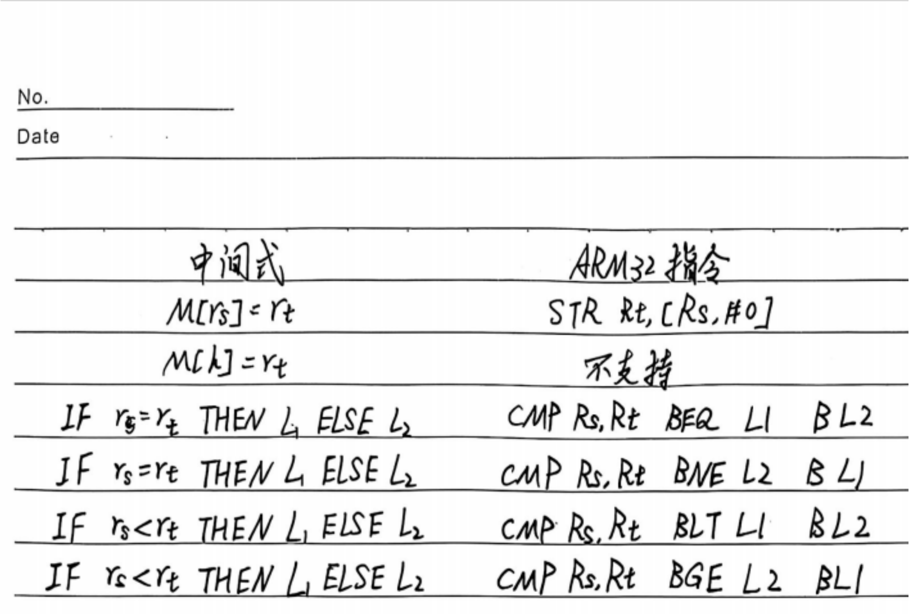
raw(6)@frame:(<参数1>:6 <访问链>:ε@frame <控制链>:bar(2,raw)@frame
<返址> x:6 y:11 t2:5)

说明：请参考书本 P356，注意有如下易错扣分点：（1）注意栈地址是递减的（2）栈快照需要给出对应引用关系，即表格第三列的行号数字（3）不要漏写“返址”行（4）每个 INT 对应 4 字节，即 4 offset

2 习题 02



2.	中间式	ARM32指令
	$t = r_s + k$	ADD Rd, Rs, #k
	$r_t = M[t \wedge last]$	LDR Rt, [Rd, #0]
	$r_t = M[r_s]$	LDR Rt, [Rs. #0]
	$r_t = M[k]$	LDR Rt, =k
	$M[t \wedge last] = r_t$	STR Rt, [Rd, #0]



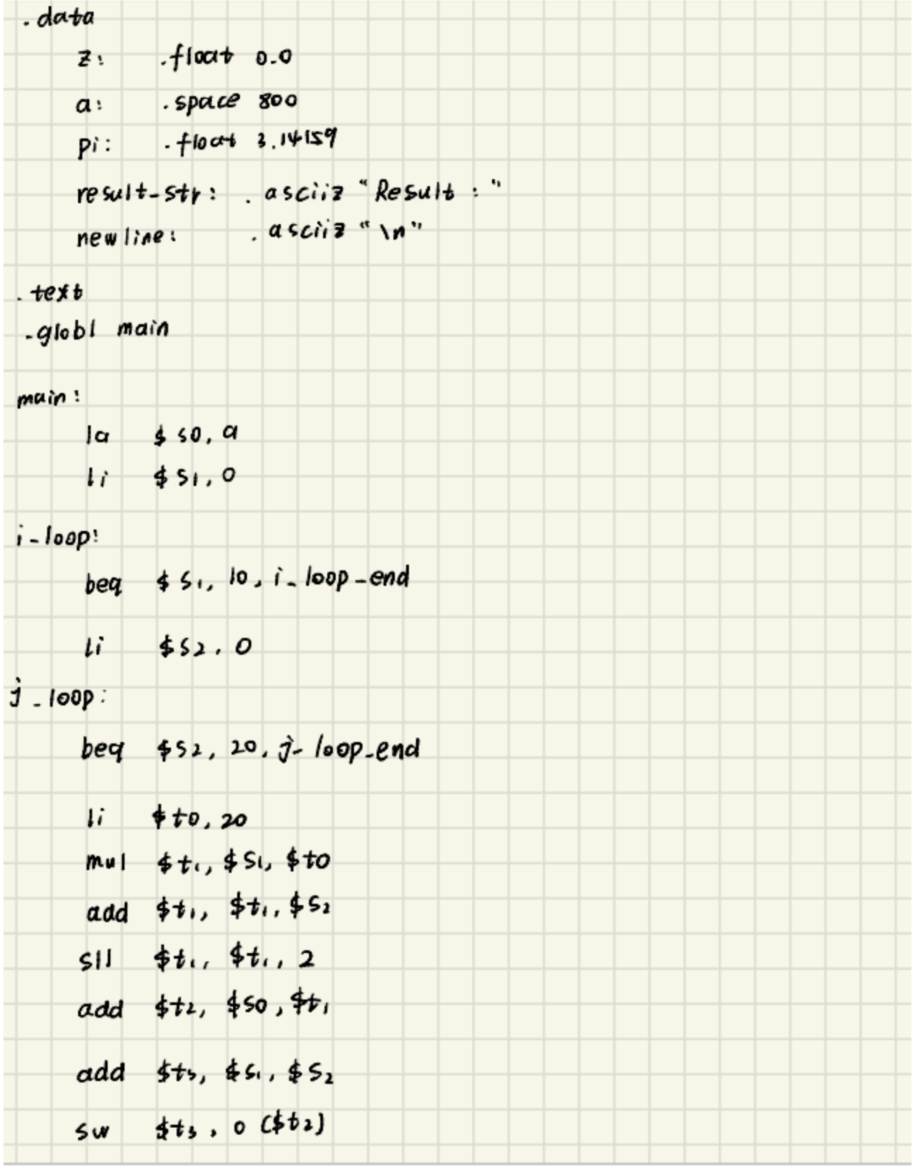
No.

Date

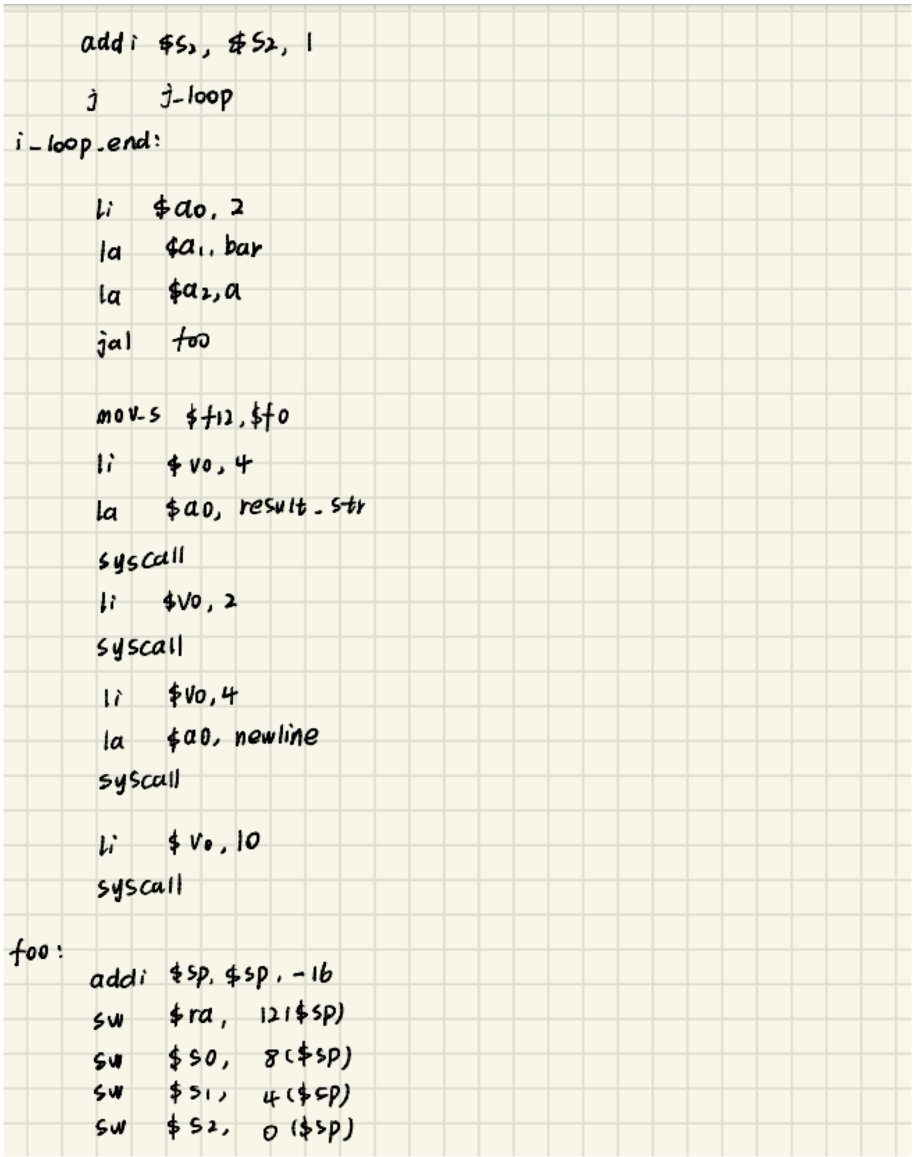
中间式	ARM32指令
$M[r_s] = r_t$	STR Rt, [Rs, #0]
$M[k] = r_t$	不支持
IF $r_s = r_t$ THEN L_1 ELSE L_2	CMP Rs, Rt BEQ L1 B L2
IF $r_s = r_t$ THEN L_1 ELSE L_2	CMP Rs, Rt BNE L2 B L1
IF $r_s < r_t$ THEN L_1 ELSE L_2	CMP Rs, Rt BLT L1 B L2
IF $r_s < r_t$ THEN L_1 ELSE L_2	CMP Rs, Rt BGE L2 B L1

说明：答案不唯一，仅供参考

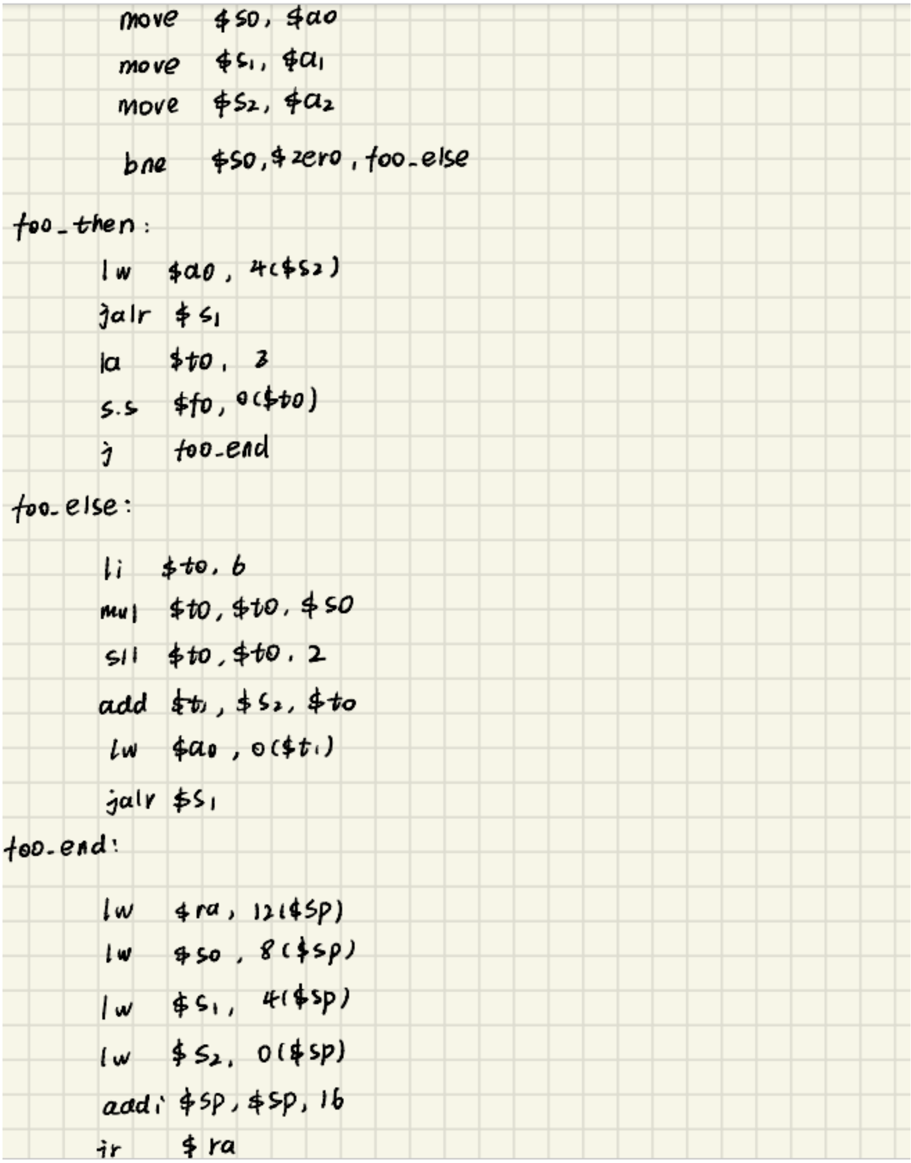
3 习题 03



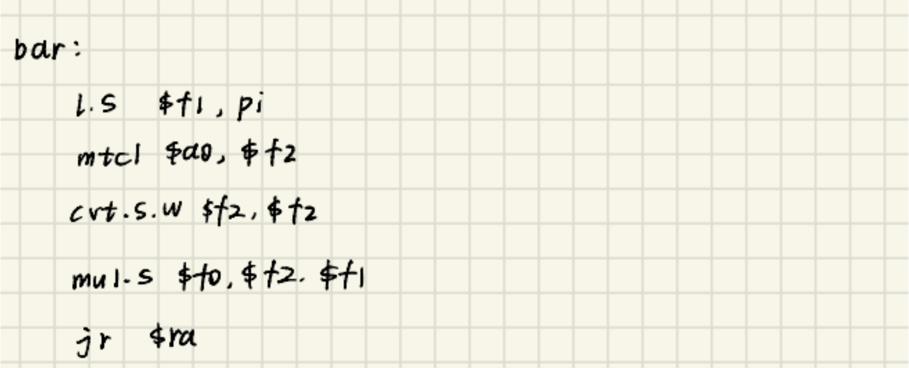
```
.data
    z:      .float 0.0
    a:      .space 800
    pi:     .float 3.14159
    result-str:  .asciiz "Result : "
    newline:     .asciiz "\n"

.text
.globl main

main:
    la   $s0, a
    li   $s1, 0
i-loop:
    beq  $s1, 10, i_loop-end

    li   $s2, 0
j_loop:
    beq  $s2, 20, j-loop_end

    li   $t0, 20
    mul  $t1, $s1, $t0
    add  $t1, $t1, $s2
    sll  $t1, $t1, 2
    add  $t2, $s0, $t1

    add  $t3, $s1, $s2
    sw   $t3, 0($t2)
```



```
    addi  $s2, $s2, 1
    j     j_loop
i_loop_end:

    li    $a0, 2
    la    $a1, bar
    la    $a2, a
    jal   foo

    mov.s $f12, $f0
    li    $v0, 4
    la    $a0, result_str
    syscall
    li    $v0, 2
    syscall
    li    $v0, 4
    la    $a0, newline
    syscall

    li    $v0, 10
    syscall

foo:
    addi  $sp, $sp, -16
    sw    $ra,  121$sp)
    sw    $s0,  8($sp)
    sw    $s1,  4($sp)
    sw    $s2,  0 ($sp)
```



```
        move  $s0, $a0
        move  $s1, $a1
        move  $s2, $a2

        bne   $s0,$zero,foo_else

foo_then:
      lw    $a0, 4($s2)
      jalr  $s1
      la    $t0, z
      s.s   $f0, 0($t0)
      j     foo_end
foo_else:

      li   $t0, 6
      mul  $t0, $t0, $s0
      sll  $t0, $t0, 2
      add  $t1, $s2, $t0
      lw   $a0, 0($t1)
      jalr $s1
foo_end:

      lw    $ra, 12($sp)
      lw    $s0, 8($sp)
      lw    $s1, 4($sp)
      lw    $s2, 0($sp)
      addi  $sp, $sp, 16
      jr    $ra
```



```
bar:
    l.s     $f1, pi
    mtc1  $a0, $f2
    cvt.s.w $f2, $f2
    mul.s  $f0, $f2. $f1
    jr    $ra
```

说明：答案不唯一，仅供参考