

پاسخ تمرین فصل دوم - دستورات: زبان کامپیوتر

زهره سورانی
مهدی حقوردی

فهرست مطالب

۲	۱	C به Assembly
۲	۲	عملیات‌های منطقی
۲	۳	عملیات‌های شرطی (Branches)
۲	۱.۳	مقدار نهایی چیست؟
۳	۲.۳	کد اسمبلی را بنویسید
۳	۴	توابع
۳	۱.۴	تابع positive
۴	۲.۴	In-line کردن تابع
۴	۳.۴	چندین بار صدا زده شدن تابع
۵	۵	توابع بازگشتی
۵	۱.۵	تابع فاکتوریل
۶	۲.۵	تابع فیبوناچی

۱ C به Assembly

```
1 copy: move    $t0, $0
2 loop: beq     $t0, $a2, done
3       sll     $t1, $t0, 2
4       add     $t2, $t1, $a1
5       lw      $t2, 0($t2)
6       add     $t1, $t1, $a0
7       sw      $t2, 0($t1)
8       addi    $t0, $t0, 1
9       b       loop
10 done: jr      $ra
```

۲ عملیات‌های منطقی

۱. 0xBABEFEF8

۲. 0xAAAAAAAA0

۳. 0x00005545

۳ عملیات‌های شرطی (Branches)

۱.۳ مقدار نهایی چیست؟

۱. ۲۰

۲. ۲۰

۲.۳ کد اسمبلی را بنویسید

```
1      addi    $t0, $0, 0
2      beq     $0, $0, TEST1
3  LOOP1: addi    $t1, $0, 0
4      beq     $0, $0, TEST2
5  LOOP2: add     $t3, $t0, $t1
6      sll     $t2, $t1, 4
7      add     $t2, $t2, $s2
8      sw      $t3, ($t2)
9      addi    $t1, $t1, 1
10     TEST2: slt     $t2, $t1, $s1
11      bne     $t2, $0, LOOP2
12      addi    $t0, $t0, 1
13     TEST1: slt     $t2, $t0, $s0
14      bne     $t2, $0, LOOP1
```

۴ توابع

۱.۴ تابع positive

```
1  positive:
2      addi    $sp, $sp, -4
3      sw      $ra, 0($sp)
4      jal     addit
5      addi    $t1, $0, 1
6      slt     $t2, $0, $v0
7      bne     $t2, $0, exit
8      addi    $t1, $0, $0
9  exit:  add     $v0, $t1, $0
10     lw      $ra, 0($sp)
11     addi    $sp, $sp, 4
12     jr      $ra
13  addit: add     $v0, $a0, $a1
14     jr      $ra
```

۲.۴ In-line کردن تابع

```
1 positive:
2     add    $t0, $a0, $a1
3     addi   $v0, $0, 1
4     slt    $t2, $0, $t0
5     bne    $t2, $0, exit
6     addi   $v0, $0, $0
7 exit:
8     jr     $ra
```

۳.۴ چندین بار صدا زده شدن تابع

```
1 f: addi    $sp, $sp, -12
2     sw     $ra, 8($sp)
3     sw     $s1, 4($sp)
4     sw     $s0, 0($sp)
5     move   $s1, $a2
6     move   $s0, $a3
7     jal    func
8     move   $a0, $v0
9     add    $a1, $s0, $s1
10    jal    func
11    lw     $ra, 8($sp)
12    lw     $s1, 4($sp)
13    lw     $s0, 0($sp)
14    addi   $sp, $sp, 12
15    jr     $ra
```

۵ توابع بازگشتی

۱.۵ تابع فاکتوریل

```
1 FACT:  addi $sp, $sp, -8
2         # make room in stack for 2 more items
3
4         sw $ra, 4($sp)
5         # save the return address
6
7         sw $a0, 0($sp)
8         # save the argument n
9
10        slti $t0, $a0, 1
11        # $t0 = $a0 x 2
12
13        beq $t0, $0, L1
14        # if $t0 = 0, goto L1
15
16        add $v0, $0, 1
17        # return 1
18
19        add $sp, $sp, 8
20        # pop two items from the stack
21
22        jr $ra
23        # return to the instruction after jal
24
25 L1:     addi $a0, $a0, -1
26        # subtract 1 from argument
27
28        jal FACT
29        # call fact(-n1)
30
31        lw $a0, 0($sp)
32        # just returned from jal: restore n
33
34        lw $ra, 4($sp)
35        # restore the return address
```

```

36
37     add $sp, $sp, 8
38     # pop two items from the stack
39
40     mul $v0, $a0, $v0
41     # return n*fact(-n1)
42
43     jr $ra
44     # return to the caller

```

۲.۵ تابع فیبوناچی

```

1  FIB:  addi  $sp, $sp, -12
2        sw   $ra, 8($sp)
3        sw   $s1, 4($sp)
4        sw   $a0, 0($sp)
5        slti $t0, $a0, 3
6        beq  $t0, $0, L1
7        addi $v0, $0, 1
8        j    EXIT
9  L1:   addi  $a0, $a0, -1
10       jal   FIB
11       addi  $s1, $v0, $0
12       addi  $a0, $a0, -1
13       jal   FIB
14       add   $v0, $v0, $s1
15  EXIT:  lw   $a0, 0($sp)
16       lw   $s1, 4($sp)
17       lw   $ra, 8($sp)
18       addi  $sp, $sp, 12
19       jr    $ra

```
