



به موارد زیر توجه کنید:

- ۱- حتما نام و شماره دانشجویی خود را روی پاسخنامه بنویسید.
- ۲- کل پاسخ تمرینات را در قالب یک فایل pdf با شماره دانشجویی خود نام گذاری کرده در سامانه CW بارگذاری کنید.
- ۳- این تمرین ۵۰ نمره دارد که معادل ۵،۰ نمره از نمره کلی درس است.
- ۴- در صورت مشاهده هر گونه مشابهت نامتعارف هر دو (یا چند) نفر کل نمره این تمرین را از دست خواهند داد.

- ۱- (۱۰ نمره) برنامه زیر به زبان اسمبلی MIPS-32 نوشته شده است. درباره این برنامه به این سوالات پاسخ دهید.
 - الف- رهنمون‌های (directives) این برنامه را مشخص کرده و کاربرد آنها را در فرایند ترجمه بنویسید.
 - ب- شبه دستورات (pseudo-instructions) برنامه را مشخص کنید و دستورات معادل هر کدام را بنویسید.
 - ج- جدول نمادهای (symbol table) این برنامه را رسم کنید. آدرس شروع برنامه و آدرس شروع داده‌ها را به ترتیب 0x00400000 و 0x100000000 در نظر بگیرید.

```
1 .text
2     la $s0, var1
3     la $s1, var2
4     la $s2, var3
5     la $s3, str
6     li $t0, 3                # initialize loop counter
7 loop:
8     lbu $t1, 0($s0)
9     sb $t1, 0($s3)          # str[i]='C'
10    lbu $t1, 0($s1)
11    sb $t1, 1($s3)          # str[i+1]='S'
12    lbu $t1, 0($s2)
13    sb $t1, 2($s3)          # str[i+2]='L'
14    addi $s3, $s3, 3        # &str += 3
15    subi $t0, $t0, 1        # decrement counter
16    beq $zero, $t0, end
17    j loop
18 end:
19    la $a0, str
20    addi $v0, $0, 4
21    syscall
22    li $v0, 10
23    syscall
24 .data
25    var1: .asciiz "C"
26    var2: .asciiz "S"
27    var3: .asciiz "L"
28    str: .space 10
```

پاسخ:

الف- خط ۱: text. نشان می‌دهد خطوط بعدی دستورالعمل است و باید به زبان ماشین تبدیل شود.
 خط ۲۴: data. نشان‌دهنده بخش تعریف متغیرها است و باید به اندازه آنها فضا در حافظه تخصیص داده شود.
 خطوط ۲۵ و ۲۶: .asciiiz. نشان‌دهنده تعریف رشته‌هایی است که با 0 تمام می‌شود.
 خط ۲۸: space. تعدادی فضای خالی در حافظه تخصیص می‌دهد.

ب-

```
line 6,22:      li R,val      addiu R, $zero,val
line 2,3,4,5,19:  la R,label      lui $at,0x1000
                  ori R,$at,offset
line 15:         subi $t0,$t0,1  addi $at,$zero,-1
                  Sub $t0,$t0,$at
```

ج-

Symbol	Address	Type
loop	0x00400024	label
end	0x00400050	label
var1	0x10000000	variable
var2	0x10000002	variable
var3	0x10000004	variable
str	0x10000006	variable

۲- (۱۰ نمره) برنامه زیر به زبان ماشین MIPS-32 داده شده است. این قطعه کد را با توجه به نحوه ترجمه دستورات در اسمبلی MIPS-32 به زبان اسمبلی بازگردانید. چنانچه جایی نیاز به قرار دادن برچسب (label) بود می‌توانید از هر اسمی برای آن استفاده کنید. تمامی کد را جزئی از text segment فرض کنید. همچنین فرض کنید متغیری استفاده یا تعریف نشده است.

Address	Code	Assembly Code
0x00400000	0x2008000c	addi \$t0,\$zero,12
0x00400004	0x01084820	add \$t1,\$t0,\$t0
0x00400008	0x01090018	mult \$t0,\$t1
0x0040000c	0x00005012	mflo \$t2
0x00400010	0x08100006	j mylabel
0x00400014	0x01495822	sub \$t3,\$t2,\$t1
0x00400018	0x2002000a	mylabel: addi \$v0,\$zero,10
0x004000c	0x0000000c	syscall

۳- (۱۰ نمره) برنامه زیر را که به زبان C نوشته شده است، ابتدا به کد اسمبلی MIPS تبدیل کنید و سپس آن را به کد ماشین ترجمه کنید. آدرس شروع برنامه و آدرس شروع داده‌ها را به ترتیب 0x00400000 و 0x100000000 در نظر بگیرید.

```
for (i=0 ; i<250 ; i+=2)
    arr[i+5] = i>>2
```

پاسخ:

```
.data
arr: .space 1020
.text
la $s0,arr          lui $at, 0x00001001 001111 00000 00001 00010 00000 000001
                    ori $s0, $at, 0      001101 00001 10000 00000 00000 000000
add $t0,$zero,$zero 000000 00000 00000 01000 00000 100000
addi $s1,$zero,250  001000 00000 10001 00000 00011 111010
loop:
bge $t0,$s1,break   slt $at,$t0,$s1    000000 01000 10001 00001 00000 101010
                    beq $at,$zero,break 000100 00001 00000 00000000000000111
addi $t1,$t0,5      001000 01000 01001 00000000000000101
sll $t1,$t1,2        000000 00000 01001 01001 00010 000000
add $t1,$s0,$t1      000000 10000 01001 01001 00000 100000
srl $t2,$t0,2        000000 00000 01000 01010 00010 000010
lw $t2, 0($t1)       100011 01001 01010 00000000000000000
addi $t0,$t0,2       001000 01000 01000 00000000000000010
j loop               000010 0000010000000000000000000100
break:
```

۴- (۲۰ نمره) فرض کنید هر یک از موارد زیر یک شبه‌دستور در زبان اسمبلی MIPS باشند. اسمبلر هر کدام از آنها را با چه دستوراتی جایگزین می‌کند؟

```
a. dbnz $t1,label      # decrement and branch if not equal to zero
b. ibeq $t1,$t2,label  # increment and branch if equal
c. ibeq $t1,100,label  # increment and branch if equal
d. xnor $t1,$t2,$t3    # set $t1 to ($t2 bitwise-xnor $t3)
e. xnor $t1,$t2,0x00FF # set $t1 to ($t2 xnor zero-extend 0x00FF)
f. xnor $t1,$t2,0xFF0000FF #set $t1 to ($t2 bitwise-xnor 0xFF0000FF)
```

پاسخ:

```
dbnz $t1,label # decrement and branch if not equal zero
addi $t1,$t1,-1
bne $t1,$zero,label

ibeq $t1,$t2,label # increment and branch if equal
addi $t1,$t1,1
beq $t1,$zero,label

ibeq $t1,100,label # increment and branch if equal
addi $t1,$t1,1
addi $at,$zero,100
beq $t1,$at,label

xnor $t1,$t2,$t3 # set $t1 to ($t2 bitwise-xnor $t3)
xor $at,$t2,$t3
nor $t1,$at,$at

xnor $t1,$t2,0x00FF # set $t1 to ($t2 xnor zero-extend 0x00FF)
addi $at,$zero,0x00FF
xor $at,$t2,$at
nor $t1,$at,$at

xnor $t1,$t2,0xFF0000FF #set $t1 to ($t2 bitwise-xnor 0xFF0000FF)
lui $at,0xFF00
ori $at,$at,0x00FF
xor $at,$t2,$at
nor $t1,$at,$at
```