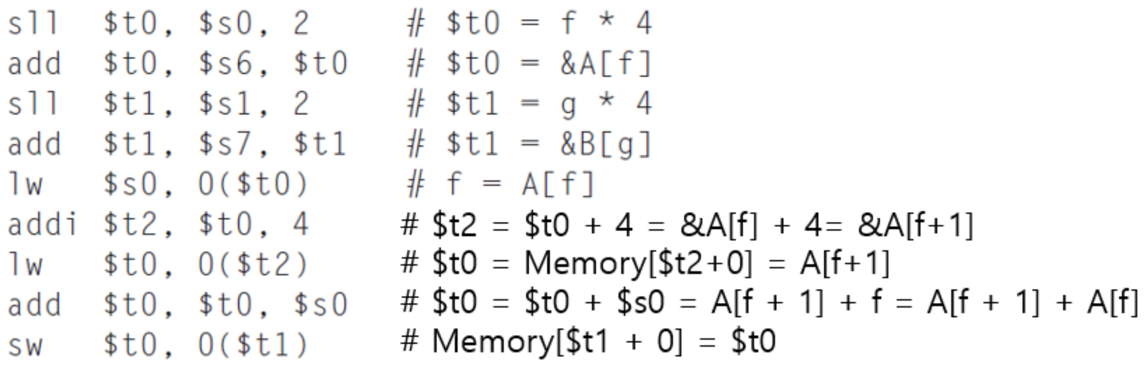
2021141460159-邓钰川-作业1-1

2.4



```
sll  $t0, $s0, 2       # $t0 = f * 4
add  $t0, $s6, $t0     # $t0 = &A[f]
sll  $t1, $s1, 2       # $t1 = g * 4
add  $t1, $s7, $t1     # $t1 = &B[g]
lw   $s0, 0($t0)       # f = A[f]
addi $t2, $t0, 4      # $t2 = $t0 + 4 = &A[f] + 4= &A[f+1]
lw   $t0, 0($t2)      # $t0 = Memory[$t2+0] = A[f+1]
add  $t0, $t0, $s0    # $t0 = $t0 + $s0 = A[f + 1] + f = A[f + 1] + A[f]
sw   $t0, 0($t1)      # Memory[$t1 + 0] = $t0
```

后面补了注释，直接一目了然

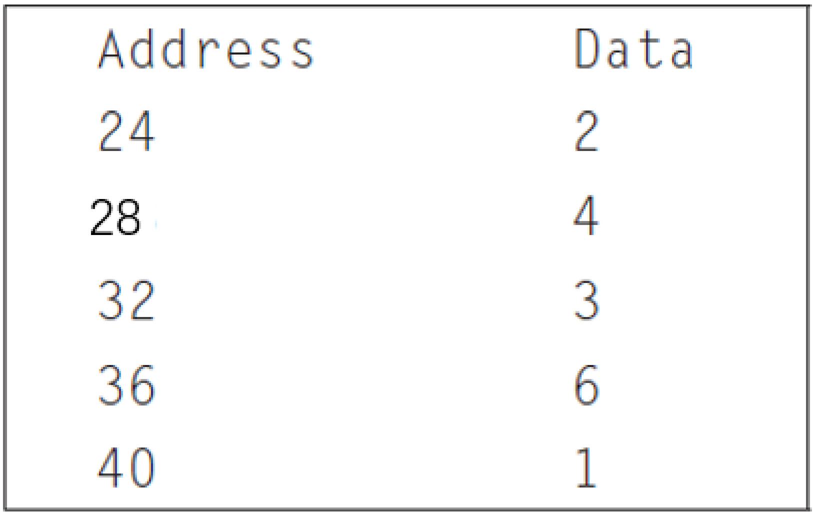
```c
B[g] = A[f]+A[1+f]
```

2.5

立即数加法可以调一下，在取数的地方处理一下就行

```
sll $s0, $s0, 2
sll $s1, $s1, 2
add $t0, $s6, $s0
add $t1, $s7, $s1
lw $s0, 0($t0)
lw $t0, 4($t0) #这里改了即可
add $t0, $t0, $s0
sw $t0, 0($t1)
```

2.6



Address	Data
24	2
28	4
32	3
36	6
40	1

40	1	4
24	2	0
32	3	2
28	4	1
36	6	3

这题38应该是28

Array[2]不动，共4个数位置交换

2.6.1

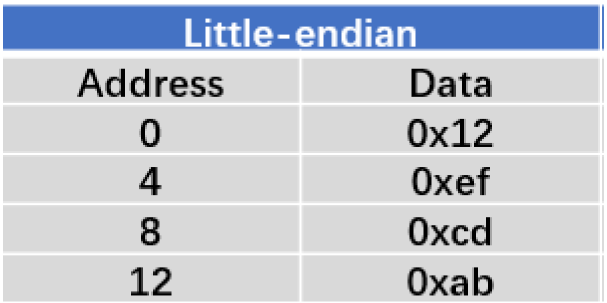
```c
tmp_1 = Array[0];
tmp_2 = Array[1];
Array[0] = Array[4];
Array[1] = tmp_1;
Array[4] = Array[3];
Array[3] = tmp_2;
```

2.6.2

```
lw $t0, 0($s6) #$t0=Array[0]
lw $t1, 4($s6) #$t1=Array[1]
sw 16($s6),0($s6) #Array[0]=Array[4]
sw $t0, 4($s6) #Array[1]=$t0=Array[0]
sw 12($s6),16($s6) #Array[4]=Array[3]
sw $t1, 12($s6) #Array[3]=$t1=Array[1]
```

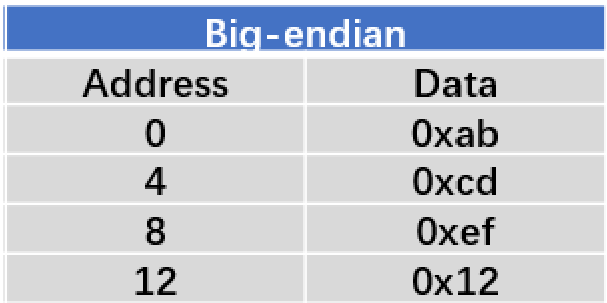
2.7

Little-Endian就是低位字节排放在内存的低地址端，高位字节排放在内存的高地址端。



Little-endian	
Address	Data
0	0x12
4	0xef
8	0xcd
12	0xab

Big-Endian就是高位字节排放在内存的低地址端，低位字节排放在内存的高地址端



Big-endian	
Address	Data
0	0xab
4	0xcd
8	0xef
12	0x12

2.9

说白了就是取A[i],A[j],要点就是4i与4j相当于左移两位
最后肯定是sw xx 32($s7)

```
sll $t0,$s3,2
add $t0,$t0,$s6
lw $t0,0($t0) #取A[i]
sll $t1,$s4,2
add $t1,$t1,$s6
lw $t1,0($t1) #取A[j]
add $t2,$t0,$t1
sw $t2,32($s7)
```

2.10

```
addi  $t0, $s6, 4  #$t0 = &A[1]
add  $t1, $s6, $0 # $t1 = &A[0]
sw   $t1, 0($t0) # A[1] = $t1=&A[0]
lw   $t0, 0($t0) # $t0 = A[1] =&A[0]
add $s0, $t1, $t0 # f = $t1 + $t0
```

这个题目比较绕，说白了就是A[1]=&A[0]最后让你把A[1]+&A[0]再换算一下

```C
f = &A[0]+&A[0];
```