HW#4

2.4

Sll $t0, $s0, 2 # $t0 = 4 \* f

Add $t0, $s6, $t0 # $t0 =& A[f]

Sll $t1, $s1, 2 # $t1 = 4 \* g

Add $t1, $s7, $t1 # $t1 =& B[g]

Lw $s0, 0($t0) # f’ = A[f]

Addi $t2, $t0, 4 # $t2 =& A[f + 1]

Lw $t0, 0($t2) # $t0 = A[f + 1]

Add $t0, $t0, $s0 # $t0 = A[f + 1] + f’ = A[f + 1] + A[f]

Sw $t0, 0($t1) # B[g] = = A[f + 1] + A[f]

C code:

B[g] = A[f + 1] + A[f];// set B[g] first, because f will change

F = A[f];

2.5

Assemble code:

Sll $t0, $s0, 2 # $t0 = 4 \* f

Add $t0, $s6, $t0 # $t0 =& A[f]

Sll $t1, $s1, 2 # $t1 = 4 \* g

Add $t1, $s7, $t1 # $t1 =& B[g]

Lw $s0, 0($t0) # f’ = A[f]

Lw $t0, 4($t0) # $t0 = A[f + 1]

Add $t0, $t0, $s0 # $t0 = A[f + 1] + f’ = A[f + 1] + A[f]

Sw $t0, 0($t1) # B[g] = = A[f + 1] + A[f]

2.6.1

// bubble sort

**int** Array[] = {2, 4, 3, 6, 1};  
**int** len = 5;  
**int** tmp = 0;  
**for** (**int** i = 0; i < len; i++){  
 **for** (**int** j = 0; j < len - i - 1; j++){  
 **if** (Array[j] > Array[j + 1]){  
 tmp = Array[j];  
 Array[j] = Array[j + 1];  
 Array[j + 1] = tmp;  
 }  
 }

}

2.6.2

.data

Array: .word 2, 4, 6, 3, 1

#############################################################################

.text

.globl main

main: la $s6, Array # set $s6 to Array

li $s1, 5 # len = 5

li $t0, 0 # $t0: offset of i

sll $t2, $s1, 2 # $t2: max offset of i

add $t3, $zero, $t2 # $t3: max offset of j

loopI: beq $t0, $t2, exitI # break loopI

li $t1, 0 # $t1: offset of j

addi $t3, $t3, -4 # $t3 = $t3 - 4

loopJ: beq $t1, $t3, exitJ

add $t4, $s6, $t1 # &= Array[j]

lw $t5, 0($t4)

lw $t6, 4($t4)

slt $t7, $t6, $t5 # Array[j + 1] < Array[j] ? 1 : 0

beq $t7, $zero, skip

sw $t5, 4($t4) # swap

sw $t6, 0($t4)

skip: addi $t1, $t1, 4

j loopJ

exitJ: addi $t0, $t0, 4

j loopI

exitI: