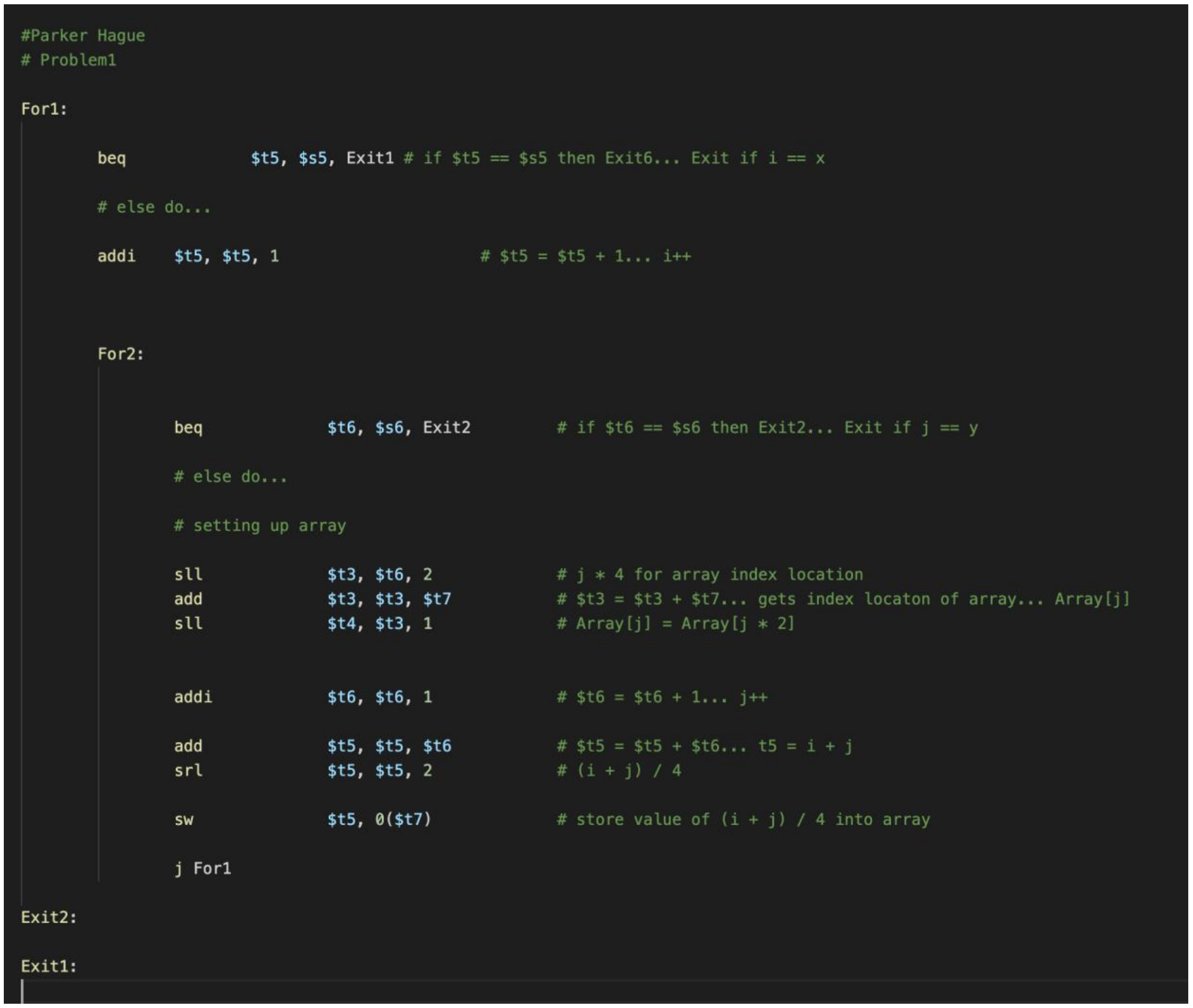
Parker Hague
phague@okstate.edu

CS3443
Assignment02

02/23/2020

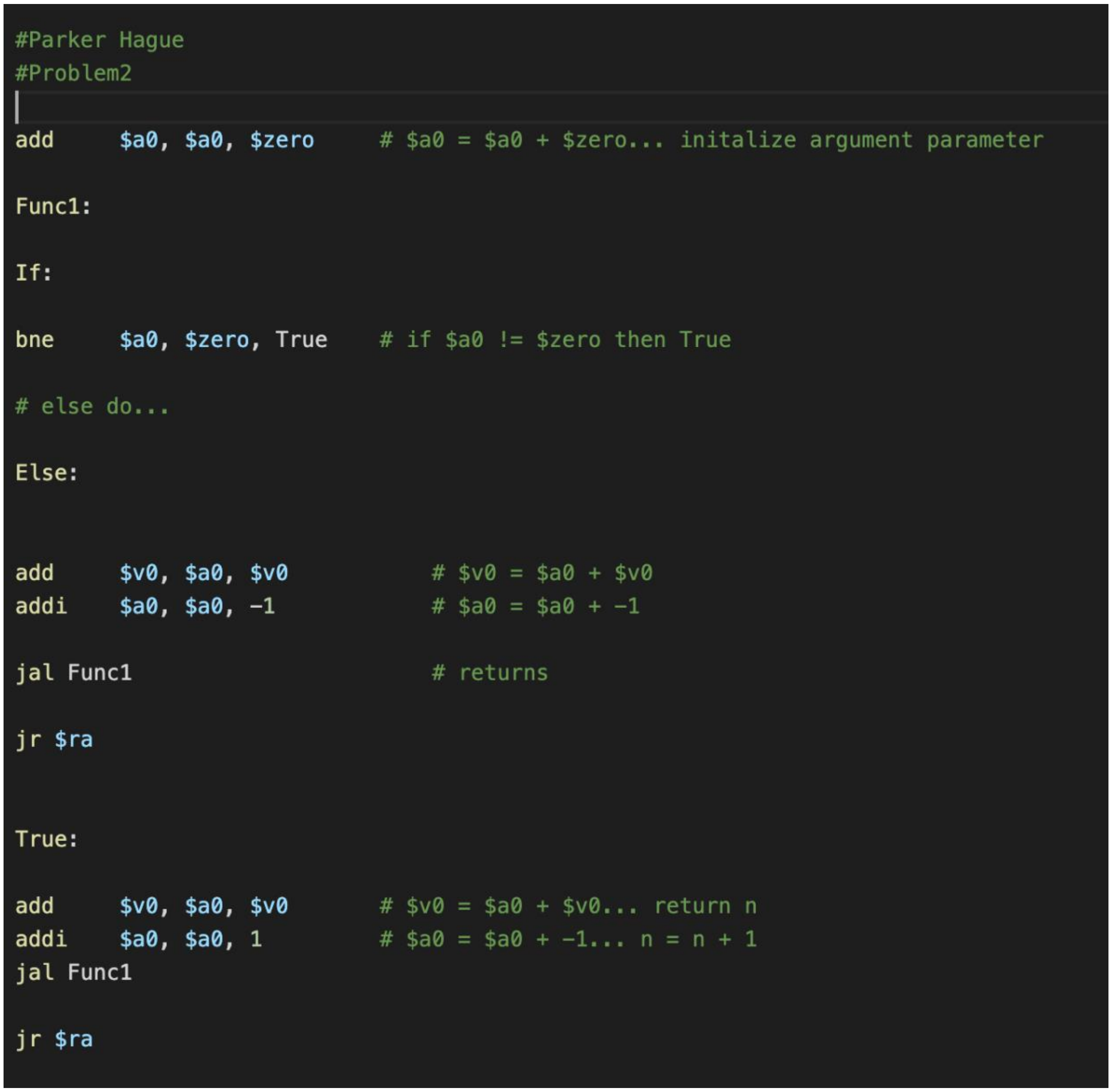
1.



```
#Parker Hague
# Problem1

For1:

        beq             $t5, $s5, Exit1 # if $t5 == $s5 then Exit6... Exit if i == x

        # else do...

        addi    $t5, $t5, 1                     # $t5 = $t5 + 1... i++


        For2:


                beq             $t6, $s6, Exit2         # if $t6 == $s6 then Exit2... Exit if j == y

                # else do...

                # setting up array

                sll             $t3, $t6, 2             # j * 4 for array index location
                add             $t3, $t3, $t7           # $t3 = $t3 + $t7... gets index locaton of array... Array[j]
                sll             $t4, $t3, 1             # Array[j] = Array[j * 2]


                addi            $t6, $t6, 1             # $t6 = $t6 + 1... j++

                add             $t5, $t5, $t6           # $t5 = $t5 + $t6... t5 = i + j
                srl             $t5, $t5, 2             # (i + j) / 4

                sw              $t5, 0($t7)             # store value of (i + j) / 4 into array

                j For1

Exit2:

Exit1:
```

Parker Hague CS3443
phague@okstate.edu Assignment02 02/23/2020

2.



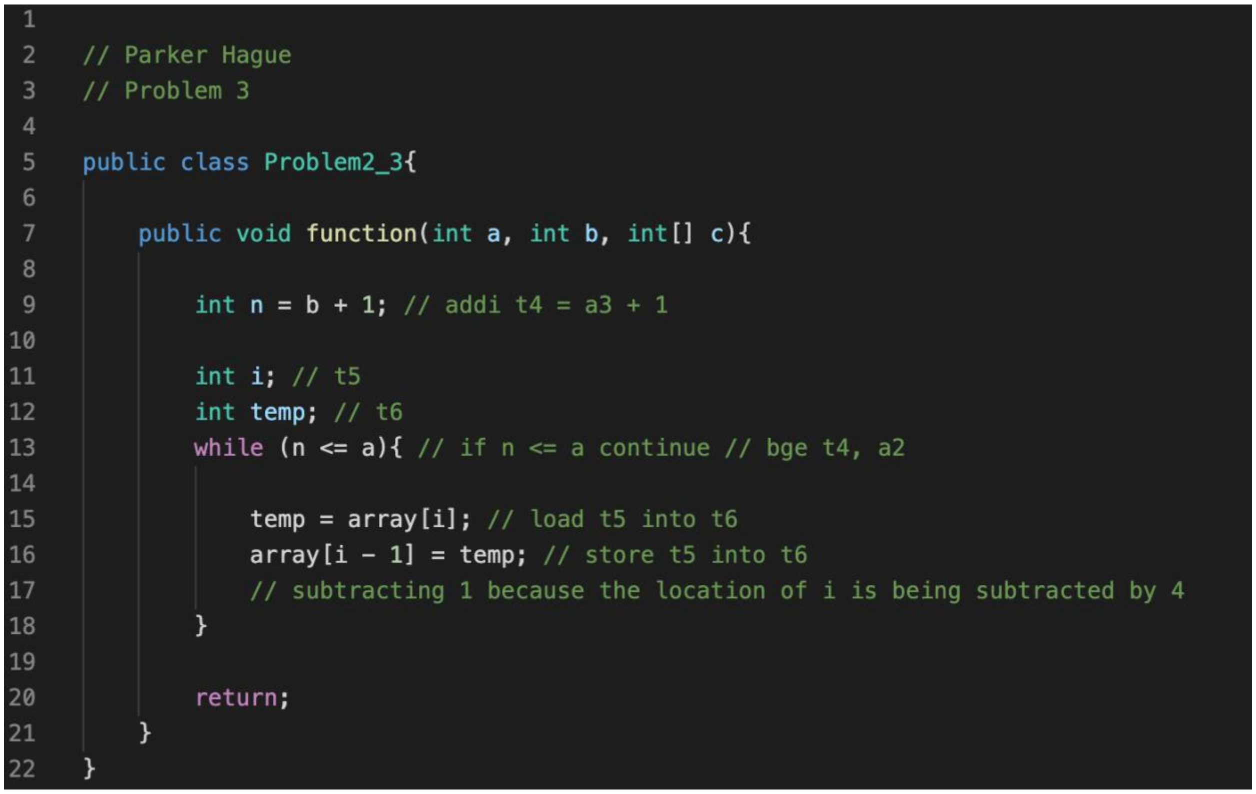
```
#Parker Hague
#Problem2

add     $a0, $a0, $zero     # $a0 = $a0 + $zero... initalize argument parameter

Func1:

If:

bne     $a0, $zero, True    # if $a0 != $zero then True

# else do...

Else:


add     $v0, $a0, $v0           # $v0 = $a0 + $v0
addi    $a0, $a0, -1            # $a0 = $a0 + -1

jal Func1                       # returns

jr $ra


True:

add     $v0, $a0, $v0       # $v0 = $a0 + $v0... return n
addi    $a0, $a0, 1         # $a0 = $a0 + -1... n = n + 1
jal Func1

jr $ra
```

Parker Hague
phague@okstate.edu

CS3443
Assignment02

02/23/2020

3.



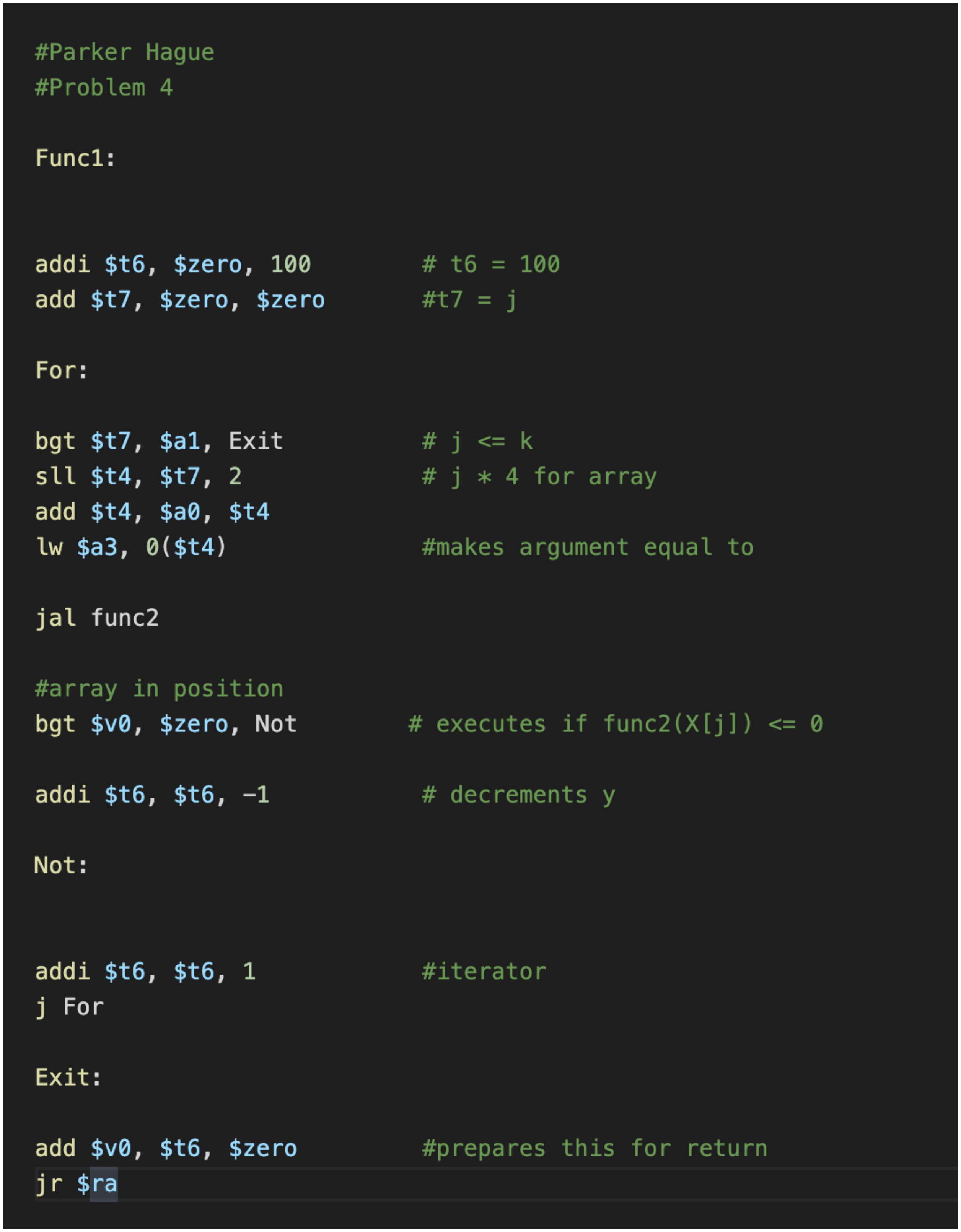
```java

// Parker Hague
// Problem 3

public class Problem2_3{

    public void function(int a, int b, int[] c){

        int n = b + 1; // addi t4 = a3 + 1

        int i; // t5
        int temp; // t6
        while (n <= a){ // if n <= a continue // bge t4, a2

            temp = array[i]; // load t5 into t6
            array[i - 1] = temp; // store t5 into t6
            // subtracting 1 because the location of i is being subtracted by 4
        }

        return;
    }
}
```

Parker Hague
phague@okstate.edu

CS3443
Assignment02

02/23/2020

4.



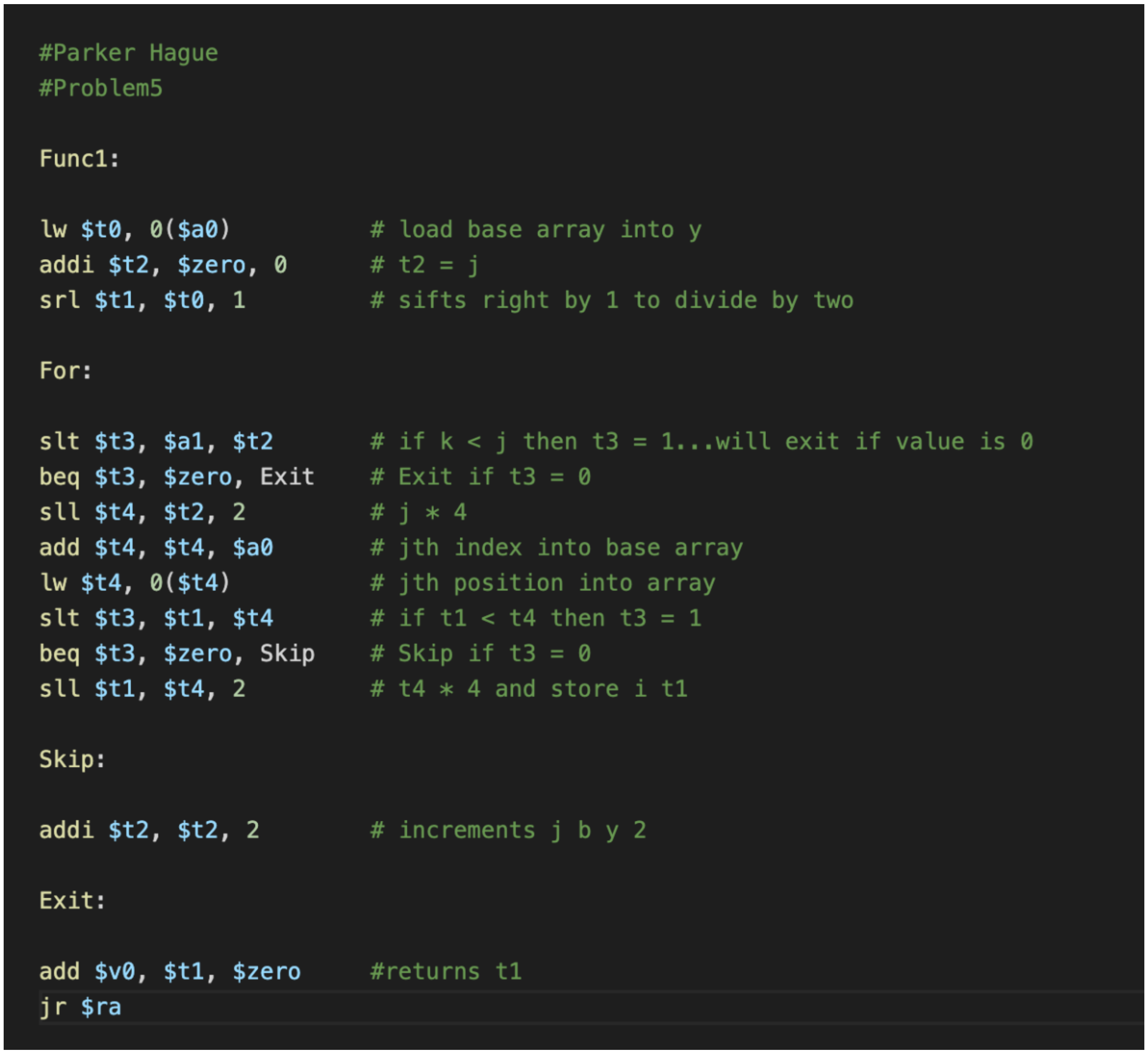
```
#Parker Hague
#Problem 4

Func1:


addi $t6, $zero, 100         # t6 = 100
add $t7, $zero, $zero        #t7 = j

For:

bgt $t7, $a1, Exit           # j <= k
sll $t4, $t7, 2              # j * 4 for array
add $t4, $a0, $t4
lw $a3, 0($t4)               #makes argument equal to

jal func2

#array in position
bgt $v0, $zero, Not         # executes if func2(X[j]) <= 0

addi $t6, $t6, -1            # decrements y

Not:


addi $t6, $t6, 1             #iterator
j For

Exit:

add $v0, $t6, $zero          #prepares this for return
jr $ra
```

Parker Hague
phague@okstate.edu

CS3443
Assignment02

02/23/2020

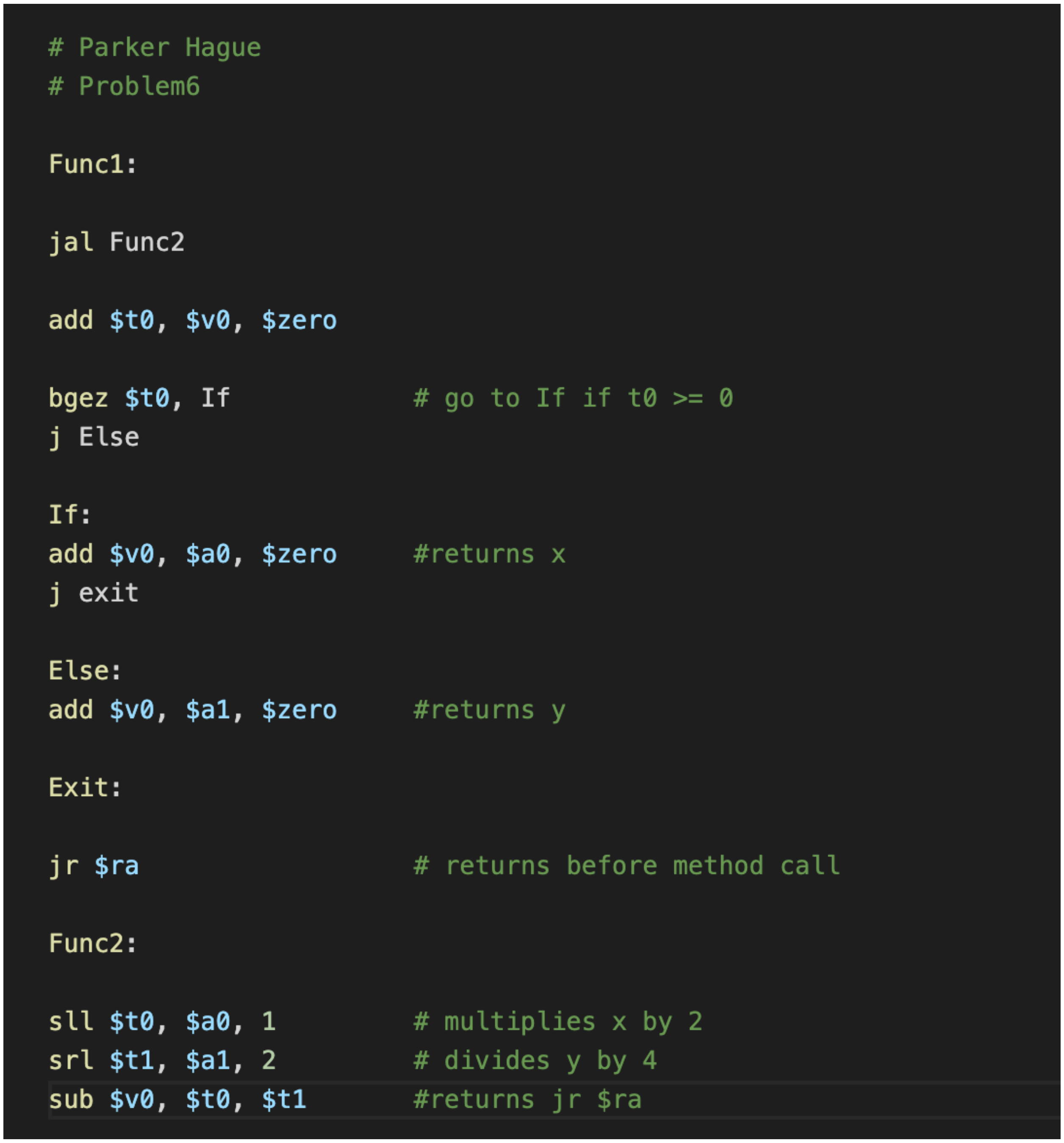
5.



```
#Parker Hague
#Problem5

Func1:

lw $t0, 0($a0)          # load base array into y
addi $t2, $zero, 0      # t2 = j
srl $t1, $t0, 1         # sifts right by 1 to divide by two

For:

slt $t3, $a1, $t2       # if k < j then t3 = 1...will exit if value is 0
beq $t3, $zero, Exit    # Exit if t3 = 0
sll $t4, $t2, 2         # j * 4
add $t4, $t4, $a0       # jth index into base array
lw $t4, 0($t4)          # jth position into array
slt $t3, $t1, $t4       # if t1 < t4 then t3 = 1
beq $t3, $zero, Skip    # Skip if t3 = 0
sll $t1, $t4, 2         # t4 * 4 and store i t1

Skip:

addi $t2, $t2, 2        # increments j b y 2

Exit:

add $v0, $t1, $zero     #returns t1
jr $ra
```

Parker Hague CS3443
phague@okstate.edu Assignment02 02/23/2020

6.



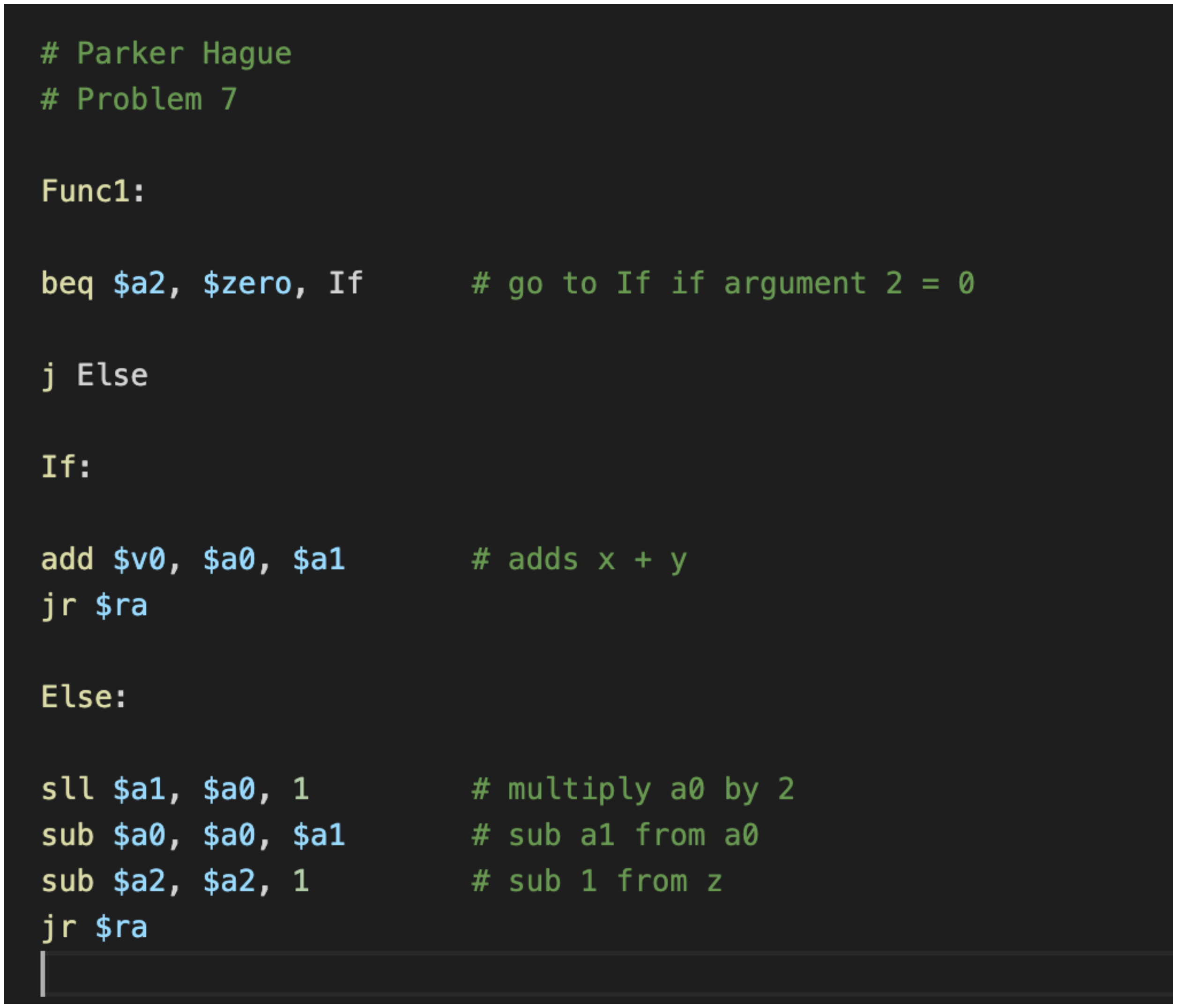
```
# Parker Hague
# Problem6

Func1:

jal Func2

add $t0, $v0, $zero

bgez $t0, If            # go to If if t0 >= 0
j Else

If:
add $v0, $a0, $zero     #returns x
j exit

Else:
add $v0, $a1, $zero     #returns y

Exit:

jr $ra                  # returns before method call

Func2:

sll $t0, $a0, 1         # multiplies x by 2
srl $t1, $a1, 2         # divides y by 4
sub $v0, $t0, $t1       #returns jr $ra
```

Parker Hague
phague@okstate.edu

CS3443
Assignment02

02/23/2020

7.



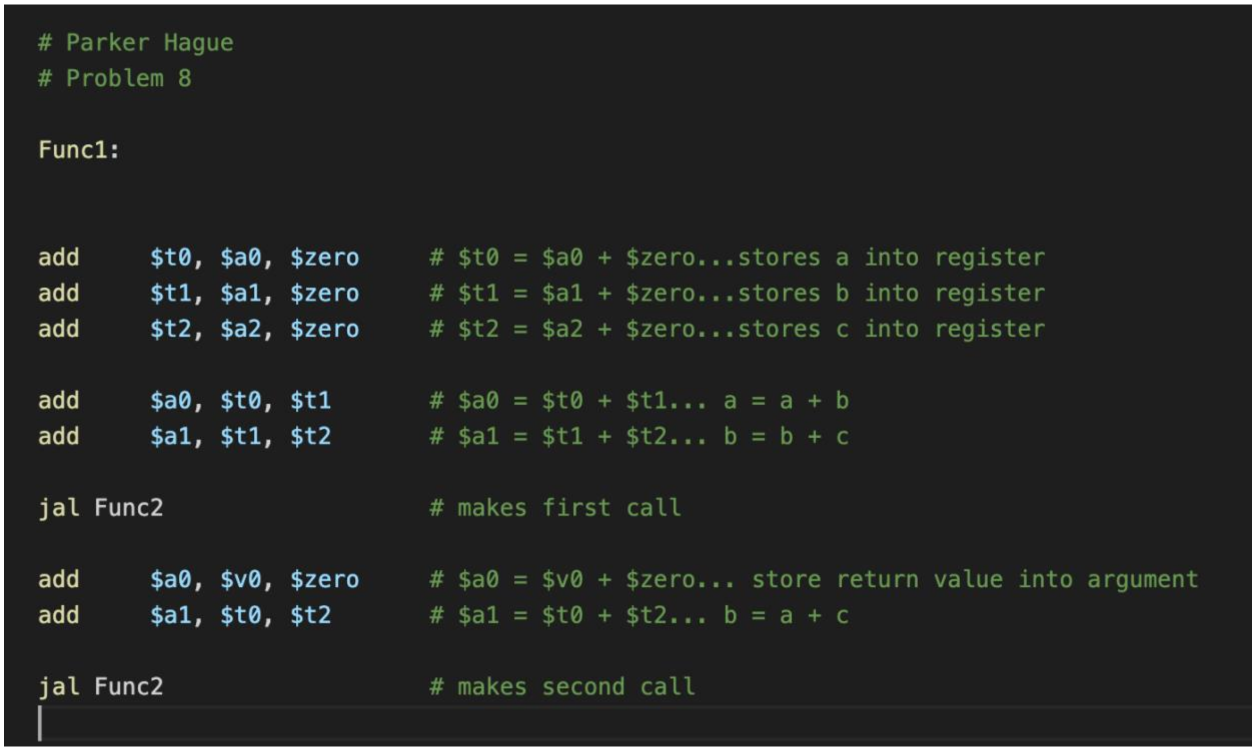
```
# Parker Hague
# Problem 7

Func1:

beq $a2, $zero, If       # go to If if argument 2 = 0

j Else

If:

add $v0, $a0, $a1        # adds x + y
jr $ra

Else:

sll $a1, $a0, 1          # multiply a0 by 2
sub $a0, $a0, $a1        # sub a1 from a0
sub $a2, $a2, 1          # sub 1 from z
jr $ra
```

Parker Hague
phague@okstate.edu

CS3443
Assignment02

02/23/2020

8.



```
# Parker Hague
# Problem 8

Func1:


add     $t0, $a0, $zero     # $t0 = $a0 + $zero...stores a into register
add     $t1, $a1, $zero     # $t1 = $a1 + $zero...stores b into register
add     $t2, $a2, $zero     # $t2 = $a2 + $zero...stores c into register

add     $a0, $t0, $t1       # $a0 = $t0 + $t1... a = a + b
add     $a1, $t1, $t2       # $a1 = $t1 + $t2... b = b + c

jal Func2                   # makes first call

add     $a0, $v0, $zero     # $a0 = $v0 + $zero... store return value into argument
add     $a1, $t0, $t2       # $a1 = $t0 + $t2... b = a + c

jal Func2                   # makes second call
```