Jinwoo Kim
ComS 321
HW 6.
2/27/2018

+9)

1.

```
begin: addi  $t0, $zero, 0   # t0 = 0.
       addi  $t1, $zero, 1   # t1 = 1.
loop:  slt   $t2, $a0, $t1   # if n > $t1, then $t2 = 0. else $t2 = 1
       bne   $t2, $zero, finish # if $t2 ≠ 0, jump to finish.
       add   $t0, $t0, $t1   # $t0 = $t0 + $t1
       addi  $t1, $t1, 2     # $t1 = $t1 + 2
       j     loop            # jump to the instr. at the label loop.
finish add   $v0, $t0, $zero # $v0 = $t0.
```

```
int t0 = 0;
int t1 = 1;
int n;                    n=1        -> 0
while (n > t1)            n=2(1)     -> 1
   t0 = t0 + t1;   1 4    n=3(1.5)   -> 1
   t1 = t1 + 2;    3      n=4(2)     -> 4
return t0.                   5(2.5)  -> 4
                             6(3)    -> 9
                             7(3.5)  -> 9
```

Def:

floor[n] : the largest integer ≤b, e.g., [5.1] = 5

∴ $(\text{floor}[\frac{n}{2}])^2$

∴ When we consider the pattern, the program takes floor[$\frac{n}{2}$] then squre it
= $(\text{floor}[\frac{n}{2}])^2$

2.

```
loop : add  $s1, $s1, 0($s5)
       add  $s3, $s3, $s4
       addi $s5, $s5, 4
       bne  $s3, $s2, loop.
```

3. We can count the minimum elements within B[0] to B[9] with t_0.
Moreover, t_1 saves value of B[0].

4.

	# of order
Bugs : v_0 is not initialized	1
It can count word at zero position.	2
a_0 should be increased by 4.	3
also, a_1 should be increased by 4	4

```
loop : addi  $V0, $0, $0      # Initialize $V0
       lw    $V1, 0($a0)      # Load words.
       addi  $V0, $V0, 4      # Increment count of words copy
       sw    $V1, 0($a1)      # Write to destination.
       addi  $a0, $a0, 4      # Advance pointer to next source
       addi  $a1, $a1, 4      # Advance pointer to next destination.
       bne   $V1, $0, loop.   # loop if word copied is ≠ 0.
       addi  $V0 $V0, -1      # do not print words at 0.
```

89

5.

a)

```
       addi  $t0, $0, $0       # clear $t0
       addi  $s0, $0, 100      # set limit
loop:  lw    $t1, 0($a1)       # $t1 = b[i]
       add   $t1,$t1,$s0       # $t1 = a[i]
       sw    $t1, 0($a0)       # store base address of a[i]
       addi  $a0, $a0, 4       # next address of a[i]
       addi  $a1, $a1, 4       # next address of b[i]
       addi  $t0, $t0,1        # $t0 = $t0 +1
       beq   $t0,$s0,finish    # if ($t0 =100) finish
       j     loop
finish
```

b) 2 times outside of loop.

till "j loop" we can count 100 times.

but till "beq" we can count 101 times.

2×1 + 7×101 + 1×100 = 809

c) 202 times.

6 - (a).

(7 x 10) : till "J Loop"

+ 5 : last iteration.

$(7\times10)+5=\boxed{75.}$

6 - (b)

```
       add $t2, $s4, $s4      # t2 ← 2xj
       add $t2, $t2, $t2      # t2 ← 4xj
       add $t1, $s3, $s3      # t1 ← 2xi
       add $t1, $t1, $t1      # t1 ← 4xi
       add $t1, $t1, $s6      # t1 ← address of a[i]
       lw  $t0, 0($t1)
       bne $t0, $s5, Exit     # check condition first time.
loop:  add $t1, $t1, $t2      # i = i+j
       lw  $t0, 0($t1)
       beq $t0, $s5, loop     # if a[i]=k, go to loop.
Exit:
```

Number of instructions executed = 7 + (3x10) = 37