

1.

```

begin:  addi  $t0, $zero, 0      # $t0 = 0
        addi  $t1, $zero, 1      # $t1 = 1
loop:   slt   $t2, $a0, $t1      #   ↓
        bne   $t2, $zero, finish # while($a0 < $t1){
        add   $t0, $t0, $t1      # $t0 += $t1: sums $t1, so all odd numbers
        addi  $t1, $t1, 2        # $t1 += 2 :starts at 1 and counts odd numbers
        j     loop              # }
        add   $v0, $t0, 0        # return $t0

```

this program sums all odd numbers that are less than the input

2.

```

loop:   add   $t0, $s3, $s3
        add   $t0, $t0, $t0
        add   $t0, $t0, $s5
        lw    $t0, 0($t0)
        add   $s1, $s1, $t0
        add   $s3, $s3, $s4
        bne   $s2, $s3, loop

```

3.

\$t0 contains the largest value from $b[0] \dots b[9]$
 \$t2 contains $b[9]$

4.

```

Loop:   lw     $v1, 0($a0)        # Read next word from source
        sw     $v1, 0($a1)        # Write to destination
        beq    $v1, $zero, end    # end if word copied is = 0
        addi   $a0, $a0, 4        # Advance pointer to next source
        addi   $a1, $a1, 4        # Advance pointer to next destination
        addi   $v0, $v0, 1        # Increment count of words copied
        j      Loop
end:

```

5.

a)

```
begin:  addi  $t1, $a0, 400    # sets t1 to be 100 memory addresses ahead of a0
loop:   lw    $t2, 0($a1)      # loads b[i]
        add   $t2, $t2, $s0    # $t2 = b[i] + c
        sw    $t2, 0($a0)      # a[i] = $t2
        addi  $a0, $a0, 4      # increments the address for a
        addi  $a1, $a1, 4      # increments the address for b
        slt   $t2, $t1, $a0    # $t2 is 1 if $a0 < $t1 (i<100)
        beq   $t2, $zero, loop # loops while $t2 = 0 (i=100)
        addi  $a0, $a0, -404    # resets the value of a0 to be a[0]
        addi  $a1, $a1, -404    # resets the value of a1 to be b[0]
```

b)

$1 + 7 * 101 + 2 = 710$ assignments

c)

$(i \leq 100) = 101$ jumps

6.

a)

$7 * 9 + 5 = 68$

b)

```
        add   $t1, $s4, $s4    # t1 = 2j
        add   $t1, $t1, $t1    # t1 = 4j
        add   $t2, $s3, $s6    # t2 address of a[i]
loop:    lw    $t0, 0($t2)      # loads a[i + m*j]
        add   $t2, $t2, $t1    # t2 += 4j
        beq   $t0, $s5, loop   # loop if a[i + m*j] = k
```

$3 + 3 * 10 = 33$ instructions executed