1. Download and install two MIPS simulators: Mipster and PCSpim.

2. Copy the following code fragment into Mipster window, run it for a positive integer number and observe the corresponding result.

```mips
.data
    out_string: .asciiz "\n Enter a positive integer number:" 

.text
main:
    addi $v0, $zero, 4
    la $a0, out_string  # load address (la) la is a pseudoinstruction
    syscall         # print out_string
    addi $v0,$zero,5
    syscall         # read an integer and
    add $s0,$v0,$zero  # copy it to $s0

    addi $t0,$zero,0
    addi $t1,$zero,1

loop:   slt $t2,$s0,$t1
    bne $t2,$zero, finish
    add $t0,$t0,$t1
    addi $t1,$t1,2
    j loop

finish:   add $s1,$t0,$zero

    addi $v0,$zero,1
    add $a0,$s1,$zero
    syscall         # print out the result
```

Add comments to the MIPS code and describe in one sentence what it computes. You should turn in a copy of the command window in which the program is run – the number you entered and the result should be visible in the command window.

3. Run the MIPS code in PCSpim for the same integer number and observe that PCSpim also produces the same result with Mipster. You should turn in a copy of PCSpim command window as well.
4. Consider the following MIPS code used to implement the instruction

```
.sllv      $s0,$s1,$s2
```

which uses the least significant 5 bits of the value in register $s2 to specify the amount register $s1 should be shift left:

```
.data
  mask: .word  0xfffffffff
  endl: .asciiz "\n"

.text
  main:
    addi       $s1,$zero,256
    addi       $s2,$zero,4
    sllv         $s0,$s1,$s2
    addi    $v0,$zero,1
    add   $a0,$s0,$zero
    syscall

    lw           $t0,mask
    lw           $s0,shifter
    and   $s0,$s0,$t0
    andi   $s2,$s2,0x1f
    sll   $s2,$s2,6
    or            $s0,$s0,$s2
    sw            $s0,shifter

  shifter:
    sll   $s0,$s1,0

    la          $a0,endl
    addi      $v0,$zero,4
    syscall

    addi    $v0,$zero,1
    add   $a0,$s0,$zero
    syscall
```

Add comments to the code and write a paragraph describing how it works. Note that the two lw instructions are pseudoinstructions that use a label to specify a memory address that contains a word of data to be loaded. Using Mipster and PCSpim, observe that sllv instruction can be implemented as shown above – turn in both Mipster and PCSpim command windows.