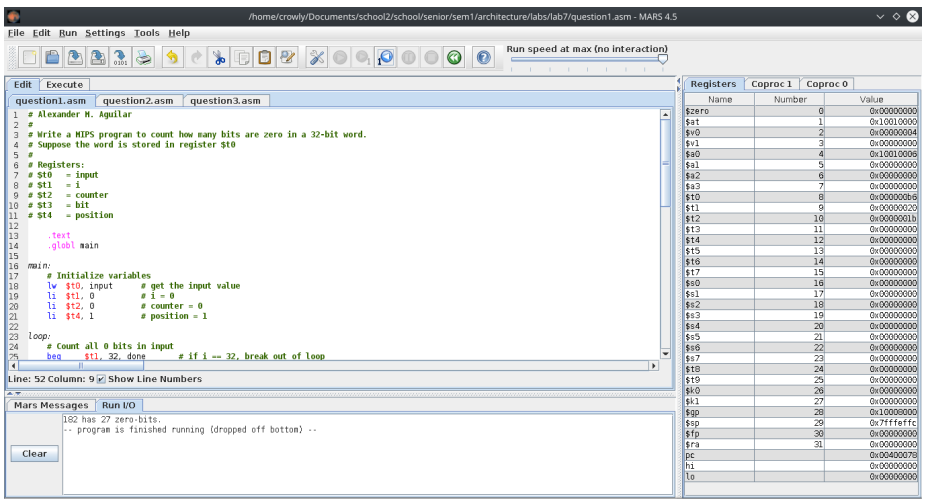
MIPS Lab Submission

Alexander M. Aguilar

April 22, 2020

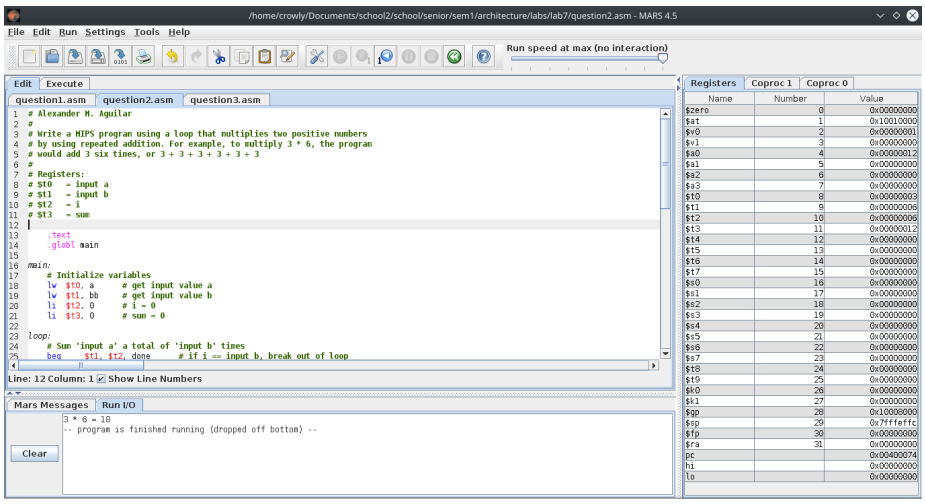
1 First Question




```
# Alexander M. Aguilar
#
# Write a MIPS program to count how many bits are zero in a 32-bit word.
# Suppose the word is stored in register $t0
#
# Registers:
# $t0     = input
# $t1     = i
# $t2     = counter
# $t3     = bit
# $t4     = position

  .text
  .globl main
```

```
main:
    # Initialize variables
    lw  $t0, input  # get the input value
    li  $t1, 0      # i = 0
    li  $t2, 0      # counter = 0
    li  $t4, 1      # position = 1

loop:
    # Count all 0 bits in input
    beq     $t1, 32, done   # if i == 32, break out of loop
    nop                     # delay

    and     $t3, $t0, $t4   # bit = input & position
    bnez    $t3, endif      # if (bit != 0), goto endif
    nop                     # delay
    addiu   $t2, $t2, 1     # counter++

endif:
    addiu   $t1, $t1, 1     # i++
    sll     $t4, $t4, 1     # position << 1

    j       loop            # loop again
    nop                     # delay

done:
    # Finished, print results
    li      $v0, 1
    move    $a0, $t0
    syscall

    li      $v0, 4
    la      $a0, hout
    syscall

    li      $v0, 1
    move    $a0, $t2
    syscall

    li      $v0, 4
    la      $a0, bout
    syscall

        .data
hout:   .asciiz " has "
bout:   .asciiz " zero-bits."
input:  .word 182               # has 27 zero-bits
                                # 00000000000000000000000010110110 in binary
```

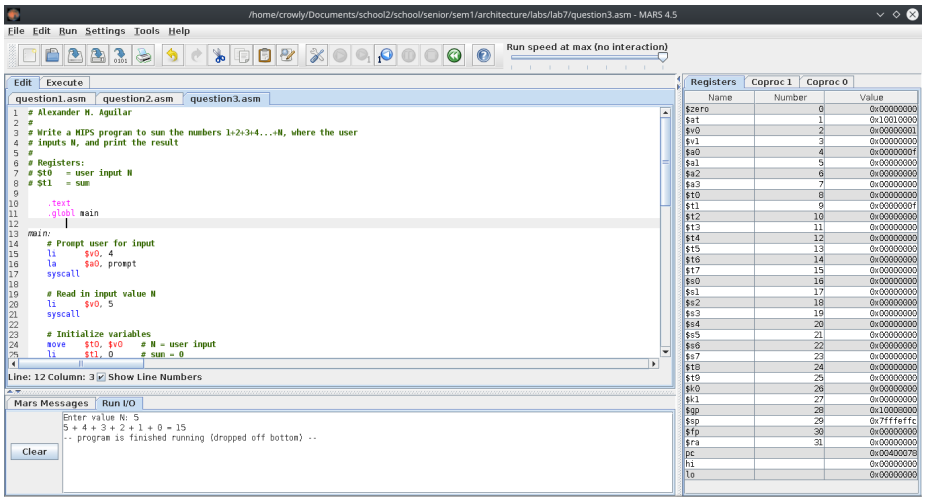
2 Second Question




```
# Alexander M. Aguilar
#
# Write a MIPS program using a loop that multiplies two positive numbers
# by using repeated addition. For example, to multiply 3 * 6, the program
# would add 3 six times, or 3 + 3 + 3 + 3 + 3 + 3
#
# Registers:
# £t0     = input a
# £t1     = input b
# £t2     = i
# £t3     = sum

    .text
    .globl main

main:
    # Initialize variables
    lw  $t0, a      # get input value a
    lw  $t1, bb     # get input value b
    li  $t2, 0      # i = 0
    li  $t3, 0      # sum = 0

loop:
```

```
    # Sum 'input a' a total of 'input b' times
    beq     $t1, $t2, done      # if i == input b, break out of loop
    nop                         # delay

    addiu   $t2, $t2, 1         # i++
    addu    $t3, $t3, $t0       # sum += input a

    j       loop                # loop again
    nop                         # delay

done:
    # Finished, print results in the format of a * b = sum
    li      $v0, 1
    move    $a0, $t0
    syscall

    li      $v0, 4
    la      $a0, mout
    syscall

    li      $v0, 1
    move    $a0, $t1
    syscall

    li      $v0, 4
    la      $a0, eout
    syscall

    li      $v0, 1
    move    $a0, $t3
    syscall

        .data
mout:   .asciiz " * "       # multiplication string
eout:   .asciiz " = "       # equals string
a:      .word   3           # input a
bb:     .word   6           # input b
```

3 Third Question




```
# Alexander M. Aguilar
#
# Write a MIPS program to sum the numbers 1+2+3+4...+N, where the user
# inputs N, and print the result
#
# Registers:
# £t0     = user input N
# £t1     = sum

    .text
    .globl main

main:
    # Prompt user for input
    li      $v0, 4
    la      $a0, prompt
    syscall

    # Read in input value N
    li      $v0, 5
    syscall

    # Initialize variables
    move    $t0, $v0    # N = user input
    li      $t1, 0      # sum = 0

loop:
    # Print current value of N
    move    $a0, $t0
```

```
    li      $v0, 1
    syscall

    # Add numbers 1+2+3+4...+N
    beqz    $t0, done       # if N == 0, break out of loop
    nop                     # delay

    # Print plus sign
    li      $v0, 4
    la      $a0, pout
    syscall

    addu    $t1, $t1, $t0   # sum += N
    subiu   $t0, $t0, 1     # N--
    j       loop            # loop again
    nop                     # delay

done:
    # Finished, print results
    li      $v0, 4
    la      $a0, eout
    syscall

    move    $a0, $t1
    li      $v0, 1
    syscall

        .data
prompt: .asciiz     "Enter value N: "
pout:   .asciiz     " + "
eout:   .asciiz     " = "
```