```
#############################################################################
# Noah del Angel, CS 2318-002, Assignment 2 Part 1 Program D
#############################################################################
# MIPS assembly program that lets user calculate weighted average score
# follows:
#
# Prompt the user to enter the integer scores for Exam 1, Exam 2 and Final Exam,
# read the scores, compute the weighted average score (using the following
# formula), and display a labeled output about the weighted average score.
#
# IMPORTANT (for the purpose of this exercise, be sure to observe the
# following):
#
#               You MUST perform (in the appropriate order, of course) ALL the
#               additions, multiplications and divisions shown in the given formula.
#               (You should NOT resort to simplifying the formula in some way, perhaps to
#               make the computation more efficient.)
#               You MUST use bit-shifting to effect multiplications and divisions involving
#               powers of 2.
#                 Note that 2, 128 and 1024 correspond to some powers of 2 (but not 307 amd
#                 637).
#
#                 You are NOT to replace 307 and 637 (that are not powers of 2) with their
#                 "sum-of-powers-of-2" equivalents.
#
#               Assume it is the intent to simply discard the fractional portion when a
#               division is performed. When evaluating the first and second terms on the right
#               hand side (i.e., the Exam 1 and Exam 2 contributions, respectively), however,
#               you MUST perform (in each case) the division after (NOT before) the
#               division after (NOT before) the multiplication (otherwise, accuracy may be
#               unnecessarily lost).
#
#               For any multiplication and division operation that cannot be effected with
#               simple (one-time) bit-shifting, you MUST use another "true" instruction (NOT
#               a pseudoinstruction) instead.
#                 Note that mulo Rdest, Rsrc1, Rsrc2 and mulou Rdest, Rsrc1, Rsrc2 are
#                 pseudoinstructions.
#                 Note that div Rdest, Rsrc1, Rsrc2 and divu Rdest, Rsrc1, Rsrc2 are
#                 pseudoinstructions.
#
# CAUTION:
# Too many past students regretted having points taken off for not labeling
# output.
############################## data segment ###################################
                .data
e1Propmt:       .asciiz "Enter Exam 1 Score: "
e2Prompt:       .asciiz "Enter Exam 2 Score: "
finalPrompt:    .asciiz "Enter Final Exam: "
averageScore:   .asciiz "The Average Score is: "

                .text
                .globl main
```

```
main:
                #Print Exam 1 label
                li $v0, 4
                la $a0, e1Propmt
                syscall

                #Get input for exam 1
                li $v0, 5
                syscall
                move $t1, $v0

                #Print Exam 2 label
                li $v0, 4
                la $a0, e2Prompt
                syscall

                #Get input for exam 2
                li $v0, 5
                syscall
                move $t2, $v0

                #Print Final Exam Label
                li $v0, 4
                la $a0, finalPrompt
                syscall

                #Get input for Final Exam
                li $v0, 5
                syscall
                move $t3, $v0

                #Preform mathmatical operations on exam 1
                sll $t4, $t1, 7     #$t4 = exam1 * 128
                li $t0, 637         #$0 has 637
                div  $t4, $t0       #exam 1 * 128/647
                mflo $t4            #Move from lo

                #Preform mathmatical operations on exam 2
                li $t0, 307         #$t0 has 307
                mult $t2, $t0       #$t2 has exam 2 * 307
                mflo $t5            #$t5 has exam 2 * 307
                srl $t5, $t5,10     #$t5 has exam 2 * 307 / 1024

                #Preform mathmatical operatiosn on final exam
                srl $t6, $t3,1      #$t6 has final exam / 2

                #Combine all operations
                add $t0, $t4, $t5   #$t0 has exam 1 & 2
                add $t0, $t0, $t6   #$t0 has exam 1 & 2 & final


                #Output average score label
```

```
li $v0, 4
la $a0, averageScore
syscall

#Output average score
li $v0, 1
move $a0, $t0
syscall

#Graceful exit
li $v0, 10
syscall
```