

```
1: #####
2: # Ethan West, CS 2318-253, Assignment 2 Part 1 Program D #
3: #####
4: # MIPS assembly program that lets user calculate weighted average score as
5: # follows:
6: # - Prompt the user to enter the integer scores for Exam 1, Exam 2 and
7: #   Final Exam, read the scores, compute the weighted average score (using
8: #   the following formula), and display a labeled output about the weighted
9: #   average score.
10: #
11: # - Formula for computing weighted average score:
12: #   avgScore = (512/2222)e1Score + (555/2048)e2Score + (finScore/2)
13: #
14: # IMPORTANT (for the purpose of this exercise, be sure to observe the following):
15: # - You MUST perform (in the appropriate order, of course) ALL the
16: #   additions, multiplications and divisions shown in the given formula.
17: #   (You should NOT resort to simplifying the formula in some way, perhaps
18: #   to make the computation more efficient.)
19: # - You MUST use bit-shifting to effect multiplications and divisions
20: #   involving powers of 2.
21: # - Note that 2, 512 and 2048 correspond to some powers of 2 (but not
22: #   555 and 2222).
23: # - You are NOT to replace 555 and 2222 (that are not powers of 2) with
24: #   their "sum-of-powers-of-2" equivalents.
25: #
26: # - When evaluating the first and second terms on the right hand side
27: #   (i.e., the Exam 1 and Exam 2 contributions, respectively), assume it is
28: #   the intent to simply discard the fractional portion when a division is
29: #   performed.
30: #
31: # - However, you MUST perform (in each case) the division after (NOT before) the
32: #   multiplication (otherwise, accuracy may be unnecessarily lost).
33: #
34: # - For any multiplication and division operation that cannot be effected with
35: #   simple (one-time) bit-shifting, you MUST use another "true" instruction (NOT
36: #   a pseudoinstruction) instead.
37: #
38: # - Note that (for multiplication) mulo Rdest, Rsrc1, Rsrc2 and mulou Rdest,
39: #   Rsrc1, Rsrc2 are pseudoinstructions (and should not be used here).
40: #
41: # - Note that (for division) div Rdest, Rsrc1, Rsrc2 and divu Rdest, Rsrc1,
42: #   Rsrc2 are pseudoinstructions (and should not be used here).
43: #
44: # - CAUTION: Too many past students regretted having points taken off for
45: #   not labeling output.
46: ##### data segment #####
47: .data
48: examOnePrompt: .ascii "Enter an integer value for exam one score: "
49: examTwoPrompt: .ascii "Enter an integer value for exam two score: "
50: examFinalPrompt: .ascii "Enter an integer value for final exam score: "
51: averageOutput: .ascii "Your wieghted average is: "
```

```
52:          .text
53:          .globl main
54: main:
55:          #BEGIN_(exam score intake)
56:          li $v0, 4          #Intake and store first exam score
57:          la $a0, examOnePrompt
58:          syscall
59:          li $v0, 5
60:          syscall
61:          move $t0, $v0
62:
63:          li $v0, 4          #Intake and store second exam score
64:          la $a0, examTwoPrompt
65:          syscall
66:          li $v0, 5
67:          syscall
68:          move $t1, $v0
69:
70:          li $v0, 4          #Intake and store final exam score
71:          la $a0, examFinalPrompt
72:          syscall
73:          li $v0, 5
74:          syscall
75:          move $t2, $v0
76:          #END_(exam score intake)
77:
78:          #BEGIN_(calculating exam 1 contribution)
79:          sll $t0, $t0, 9
80:          li $t3, 2222
81:          div $t0, $t3
82:          mflo $t0
83:          #END_(calculation exam 1 contribution)
84:
85:          #BEGIN_(calculating exam 2 contribution)
86:          li $t3, 555
87:          mult $t1, $t3
88:          mflo $t1
89:          srl $t1, $t1, 11
90:          #END__(calculating exam 2 contribution)
91:
92:          #BEGIN_(calculating final exam contribution)
93:          srl $t2, $t2, 1
94:          #END_(calculating final exam contribution)
95:
96:          #BEGIN_(final addition of all contributions)
97:          add $t0, $t0, $t1
98:          add $t0, $t0, $t2
99:          #END_(final addition of all contributions)
100:
101:          #BEGIN_(output final average)
102:          li $v0, 4
```

```
103:         la $a0, averageOutput
104:         syscall
105:         li $v0, 1
106:         move $a0, $t0
107:         syscall
108:         #END_(output final average)
109:
110:         li $v0, 11
111:         li $a0, '\n'
112:         syscall
113:
114:         li $v0, 10      # graceful exit
115:         syscall
```