

Steve Smith  
ELEC 385, Section 1  
Dr. Marino  
27 March 2015

## Homework #10

File Edit Run Settings Tools Help

Run speed at max (no interaction)

summation.asm

```
1 .data
2   arr: .float      82.6, -1.0, -6.7348483, 0.0, 100.9, 82.0116
3   counter: word    6
4   const: .float    82.0116
5   sum: .float      0.0
6 .text
7   la $t0, arr      # Initialize pointer
8   lw $t1, counter  # Load the counter into $t1
9   lwc1 $f12, sum    # Initialize the sum (which starts at 0.0) into $f12 (also the register printing a double reads from in future syscall)
10  lwc1 $f1, const   # Initialize the constant we're going to subtract by
11 loop:
12  lwc1 $f2, 0($t0)   # load the value from the array
13  sub.s $f3, $f2, $f1 # subtract the value from array by the constant, place in $f3
14  mul.s $f3, $f3, $f3 # (x-82.0116)^2
15  add.s $f12, $f12, $f3 # add to the running total
16  addi $t0, $t0, 4   # increment the address for the array index
17  addi $t1, $t1, -1  # decrement the counter
18  beq $t1, $0, exit # if the counter is 0, exit
19  j loop             # keep looping
20 exit:
21  li $v0, 2          # the syscall to print a float to the console
22  syscall             # print to screen
23  li $v0, 10         # load syscall to exit program
24  syscall             # exit program
25
```

Line: 3 Column: 18 ☒ Show Line Numbers

Mars Messages Run I/O

21849.877  
-- program is finished running --

Clear

Registers		
	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	10
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	268501016
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194384
hi		0
lo		0

File Edit Run Settings Tools Help

Run speed at max (no interaction)

summation.asm

```
1 .data
2   arr: .float      82.6, -1.3, 0.0, 82.0116
3   counter: word    4
4   const: .float    82.0116
5   sum: .float      0.0
6 .text
7   la $t0, arr      # Initialize pointer
8   lw $t1, counter  # Load the counter into $t1
9   lwc1 $f12, sum    # Initialize the sum (which starts at 0.0) into $f12 (also the register printing a double reads from in future syscall)
10  lwc1 $f1, const   # Initialize the constant we're going to subtract by
11 loop:
12  lwc1 $f2, 0($t0)   # load the value from the array
13  sub.s $f3, $f2, $f1 # subtract the value from array by the constant, place in $f3
14  mul.s $f3, $f3, $f3 # (x-82.0116)^2
15  add.s $f12, $f12, $f3 # add to the running total
16  addi $t0, $t0, 4   # increment the address for the array index
17  addi $t1, $t1, -1  # decrement the counter
18  beq $t1, $0, exit # if the counter is 0, exit
19  j loop             # keep looping
20 exit:
21  li $v0, 2          # the syscall to print a float to the console
22  syscall             # print to screen
23  li $v0, 10         # load syscall to exit program
24  syscall             # exit program
25
```

Line: 8 Column: 45 ☒ Show Line Numbers

Mars Messages Run I/O

13667.07  
-- program is finished running --

Clear

Registers		
	Coproc 1	Coproc 0
Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	10
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	268501008
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194384
hi		0
lo		0



File Edit Run Settings Tools Help

Run speed at max (no interaction)

Edit Execute

summation.asm

```

1 .data
2   arr: .float      82.0116, 84.0116, 83.0116
3   counter: word    3
4   const: .float    82.0116
5   sum: .float      0.0
6 .text
7   la $t0, arr      # Initialize pointer
8   lw $t1, counter  # Load the counter into $t1
9   lwc1 $f12, sum    # Initialize the sum (which starts at 0.0) into $f12 (also the register printing a double reads from in future syscall)
10  lwc1 $f1, const   # Initialize the constant we're going to subtract by
11
12  loop:
13    lwc1 $f2, 0($t0) # load the value from the array
14    sub.s $f3, $f2, $f1 # subtract the value from array by the constant, place in $f3
15    mul.s $f3, $f3, $f3 # (x-82.0116)^2
16    add.s $f12, $f12, $f3 # add to the running total
17    addi $t0, $t0, 4    # increment the address for the array index
18    addi $t1, $t1, -1   # decrement the counter
19    beq $t1, $0, exit  # if the counter is 0, exit
20    j loop             # keep looping
21
22  exit:
23    li $v0, 2          # the syscall to print a float to the console
24    syscall            # print to screen
25    li $v0, 10         # load syscall to exit program
26    syscall            # exit program

```

Line: 15 Column: 50 ☒ Show Line Numbers

Mars Messages

Run I/O

5.0  
-- program is finished running --

Clear

Registers Coproc 1 Coproc 0

Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	10
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	268501064
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$s0	16	0
\$s1	17	0
\$s2	18	0
\$s3	19	0
\$s4	20	0
\$s5	21	0
\$s6	22	0
\$s7	23	0
\$t8	24	0
\$t9	25	0
\$k0	26	0
\$k1	27	0
\$gp	28	268468224
\$sp	29	2147479548
\$fp	30	0
\$ra	31	0
pc		4194384
hi		0
lo		0

1) Hand calculated result:

$$\sum_{i=1}^6 (x_i - 82.0116)^2 \quad \text{Array} = \{82.6, -1.0, -6.7348483, 0.0, 100.9, 82.0116\}$$

$$(82.6 - 82.0116)^2 = 0.34621456$$

$$(-1.0 - 82.0116)^2 = 6890.925735$$

$$(-6.7348483 - 82.0116)^2 = 7875.932086$$

$$(0.0 - 82.0116)^2 = 6725.902535$$

$$(100.9 - 82.0116)^2 = 356.7716546$$

$$(82.0116 - 82.0116)^2 = 0.0$$

$$\Sigma = 21,849.87823 \approx 21,849.877$$

Easy Experiment:

$$\sum_{i=1}^4 (x_i - 82.0116)^2 \quad \text{Array} = \{82.6, -1.3, 0.0, 82.0116\}$$

$$(82.6 - 82.0116)^2 = 0.34621456$$

$$(-1.3 - 82.0116)^2 = 6940.822695$$

$$(0.0 - 82.0116)^2 = 6725.902535$$

$$(82.0116 - 82.0116)^2 = 0$$

$$\Sigma = 13,667.07144 \approx 13,667.07$$

I think my calculator has higher precision (or is just doing 64-bit operations)

Super-Easy Experiment the array was  $\{82.0116, 84.0116, 83.0116\}$   
which comes out to  $0^2 + 2^2 + 1^2 = 5$  and the program got the exact same.

#2

ldc1, Constant  $\rightarrow$  lui \$1, 4097  
ldc1 \$f0, 40(\$1)

lui \$1, 0x00001001  $\leftarrow$

001111	000000	00001	0x1001
opcode	rs	rt	imm

0x3C011001

ldc1 \$f0, 0x00000028(\$1) \*

110101	00001	00000	0x0028
opcode	rs	rt	imm

0xD4200028

lwc1 \$f2, 100(\$t1)

110001	01001	00010	0x0064
opcode	rs	rt	imm

0xC5220064

sub.d \$f20, \$f22, \$24

010001	10001	11000	10110	10100	000001
opcode	fmt	ft	fs	fd	funct

0x4638B501

lui \$1, 0x1001  $\rightarrow$  0x3C011001

001111	00000	00001	0x1001
opcode	rs	rt	imm

sdc1 \$f0, 0x00000030(\$1)  $\rightarrow$  0xf4020030

111101	00001	00000	0x0030
opcode	rs	rt	imm