

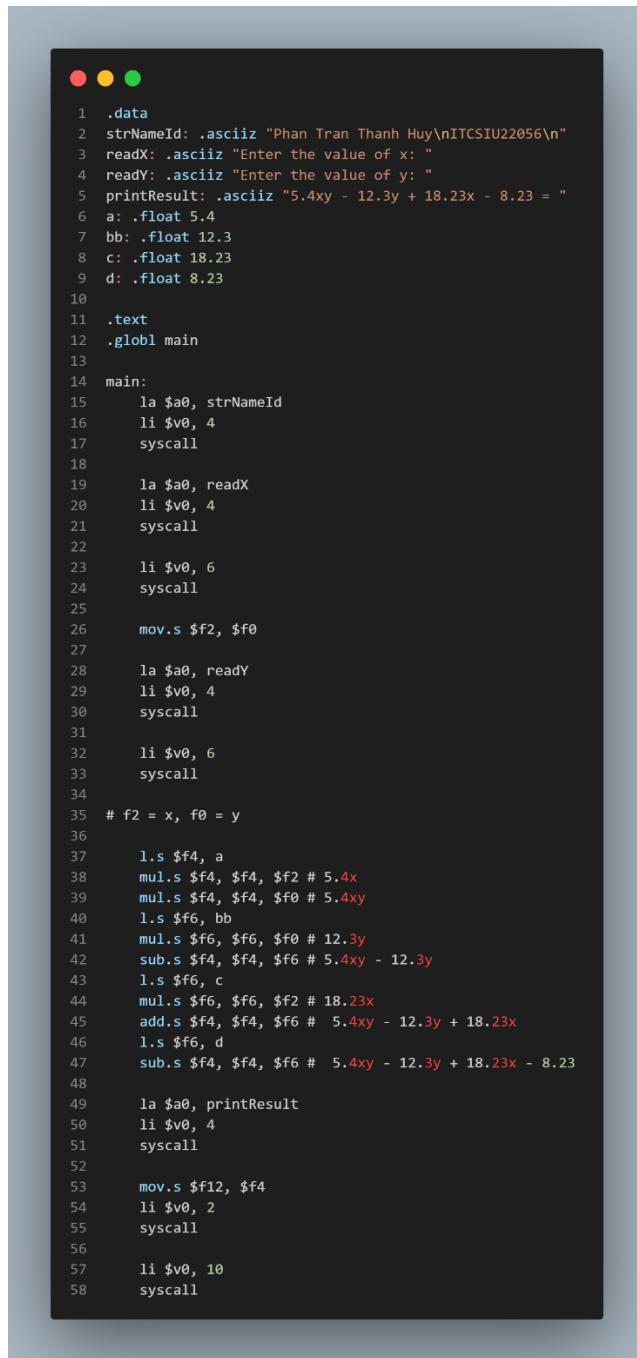
Name: Phan Trần Thanh Huy

ID: ITCSIU22056

Lab 8

Exercise 1 – Arithmetic Expression

Code:



```
1 .data
2 strNameId: .asciiz "Phan Tran Thanh Huy\nITCSIU22056\n"
3 readX: .asciiz "Enter the value of x: "
4 ready: .asciiz "Enter the value of y: "
5 printResult: .asciiz "5.4xy - 12.3y + 18.23x - 8.23 = "
6 a: .float 5.4
7 bb: .float 12.3
8 c: .float 18.23
9 d: .float 8.23
10
11 .text
12 .globl main
13
14 main:
15     la $a0, strNameId
16     li $v0, 4
17     syscall
18
19     la $a0, readX
20     li $v0, 4
21     syscall
22
23     li $v0, 6
24     syscall
25
26     mov.s $f2, $f0
27
28     la $a0, ready
29     li $v0, 4
30     syscall
31
32     li $v0, 6
33     syscall
34
35 # f2 = x, f0 = y
36
37     l.s $f4, a
38     mul.s $f4, $f4, $f2 # 5.4x
39     mul.s $f4, $f4, $f0 # 5.4xy
40     l.s $f6, bb
41     mul.s $f6, $f6, $f0 # 12.3y
42     sub.s $f4, $f4, $f6 # 5.4xy - 12.3y
43     l.s $f6, c
44     mul.s $f6, $f6, $f2 # 18.23x
45     add.s $f4, $f4, $f6 # 5.4xy - 12.3y + 18.23x
46     l.s $f6, d
47     sub.s $f4, $f4, $f6 # 5.4xy - 12.3y + 18.23x - 8.23
48
49     la $a0, printResult
50     li $v0, 4
51     syscall
52
53     mov.s $f12, $f4
54     li $v0, 2
55     syscall
56
57     li $v0, 10
58     syscall
```

Test case:

IP Regs	Int Regs [16]	Data	Text
Int Regs [16]			
Int Regs [16]			
PC	= 40000c0	[004000048] 46000086 mov.s \$f2, \$f0 ; 26: mov.	.data
EPC	= 0	[0040004c] 3c011001 lui \$1, 4097 [readyY] ; 28: la \$	strNameId: .asciiz "Phan Tran Thanh Huy\nITCSIU22056\n"
Cause	= 0	[00400050] 34240038 ori \$4, \$1, 56 [readyY]	readx: .asciiz "Enter the value of x: "
BadAddr	= 0	[00400054] 34020004 ori \$2, \$0, 4 ; 29: li \$	readyy: .asciiz "Enter the value of y: "
Status	= 3000ff10	[00400058] 0000000c	printResult: .asciiz "5.4xy - 12.3y + 18.23x - 8.23 = "
HI	= 0	[00400060] 34000000	a: float 5.4
LO	= 0	[00400064] 3c011001 ITCSIU2056	bb: float 12.3
R0 [r0] = 0		[00400068] 42400070 Enter the value of x: 5	c: float 18.23
R1 [at] = 10010000		[00400070] 46022102 Enter the value of y: 6	d: float 8.23
R2 [v0] = a		[00400074] 3c011001	
R3 [v1] =		[00400078] 4260074	
R4 [a0] = 1001004f		[0040007c] 46003182	
R5 [a1] = 7fffff1a		[00400080] 46003184	
R6 [a2] = 7fffffb0		[00400084] 3c011001	
R7 [a3] = 0		[00400088] 4260078	
R8 [t0] = 0		[0040008c] 46023182	
R9 [t1] = 0		[00400090] 46062100	
R10 [t2] = 0		[00400094] 3c011001	
R11 [t3] = 0		[00400098] 426007c	
R12 [t4] = 0		[0040009c] 46062101	
R13 [t5] = 0		[004000a0] 3c011001	
R14 [t6] = 0		[004000a4] 342404f2	
R15 [t7] = 0		[004000a8] 342404f4	
R16 [s0] = 0		[004000ac] 0000000c	
R17 [s1] = 0		[004000b0] 46002306	
R18 [s2] = 0		[004000b4] 34020002	
R19 [s3] = 0		[004000b8] 0000000c	
R20 [s4] = 0		[004000bc] 3402000a	
R21 [s5] = 0		[004000c0] 0000000c	
...	...		
Memory and registers cleared			
SPIM Version 0.1.24 of August 1, 2023 (final)			
Copyright 1990-2023 by James Larus.			
All Rights Reserved			
SPIM is distributed under a BSD license.			
See the file README for a full copyright notice.			
QsSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and ver			
Memory and registers cleared			
SPIM Version 0.1.24 of August 1, 2023 (final)			
Copyright 1990-2023 by James Larus.			
All Rights Reserved			
SPIM is distributed under a BSD license.			
See the file README for a full copyright notice.			
QsSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and ver			

File Edit View

.data
strNameId: .asciiz "Phan Tran Thanh Huy\nITCSIU22056\n"
readx: .asciiz "Enter the value of x: "
ready: .asciiz "Enter the value of y: "
printResult: .asciiz "5.4xy - 12.3y + 18.23x - 8.23 = "
a: .float 5.4
bb: .float 12.3
c: .float 18.23
d: .float 8.23

.text
.globl main

main:
 la \$a0, strNameId
 li \$v0, 4
 syscall

la \$a0, readx
 li \$v0, 4
 syscall

li \$v0, 6
 syscall

mov.s \$f2, \$f0

la \$a0, ready
 li \$v0, 4
 syscall

li \$v0, 6
 syscall

f2 = x, f0 = y

l.s \$f4, a
 mul.s \$f4, \$f4, \$f2 # 5.4x
 mul.s \$f4, \$f4, \$f0 # 5.4xy
 l.s \$f4, b

Ln 17, Col 9 880 characters 100% Windows (CRLF) UTF-8

Single Step:

FP Regs

FIR	= 9800
FCSR	= 0
Single Precision	
FG0	= 40a00000
FG1	= 0
FG2	= 0
FG3	= 0
FG4	= 0
FG5	= 0
FG6	= 0
FG7	= 0
FG8	= 0
FG9	= 0
FG10	= 0
FG11	= 0
FG12	= 0
FG13	= 0
FG14	= 0
FG15	= 0
FG16	= 0
FG17	= 0
FG18	= 0
FG19	= 0
FG20	= 0
FG21	= 0
FG22	= 0
FG23	= 0
FG24	= 0
FG25	= 0
more	= ^

Text

```
[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4 # envp
[00400010] 00c23021
[00400014] 0c100009 Phan Tran Thanh Huy
[00400018] 00000000 ITCS1122056
[0040001c] 3402000a Enter the value of x: 5
[00400020] 0000000c
[00400024] 3c041001
[00400028] 34020004
[0040002c] 0000000c
[00400030] 3c011001
[00400034] 34240021
[00400038] 34020004
[0040003c] 0000000c
[00400040] 34020006
[00400044] 0000000c
[00400048] 46000086
[0040004c] 3c011001
[00400050] 34240038
[00400054] 34020004
[00400058] 0000000c
[0040005c] 34020006
[00400060] 0000000c
[00400064] 3c011001
[00400068] c4240070
[0040006c] 46022102
[00400070] 46002102
[00400074] 3c011001
[00400078] 3c011001
```

User Text Segment [00400000]..[00440000]

Console

Memory and registers cleared

SPIM Version 9.1.24 of August 1, 2023 (final)
Copyright 1980-2023 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

Memory and registers cleared

SPIM Version 9.1.24 of August 1, 2023 (final)
Copyright 1980-2023 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

User Text Segment [00400000]..[00440000]

Console

Memory and registers cleared

SPIM Version 9.1.24 of August 1, 2023 (final)
Copyright 1980-2023 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

FP Regs Text

```

FP21 = 0
FP22 = 0
FP23 = 0
FP24 = 0
FP25 = 0
FP26 = 0
FP27 = 0
FP28 = 0
FP29 = 0
FP30 = 0
FP31 = 0

Double Precision
FP0 = 40c00000
FP1 = 40a00000
FP2 = 40a00000
FP3 = 0
FP4 = 0
FP5 = 0
FP6 = 0
FP7 = 0
FP8 = 0
FP9 = 0
FP10 = 0
FP11 = 0
FP12 = 0
FP13 = 0
FP14 = 0
FP15 = 0
FP16 = 0
FP17 = 0
FP18 = 0
FP19 = 0
FP20 = 0
FP21 = 0
FP22 = 0
FP23 = 0
FP24 = 0
FP25 = 0
FP26 = 0
FP27 = 0
FP28 = 0
FP29 = 0
FP30 = 0
FP31 = 0

[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp) # argo
[00400001] 8fa40004 addiu $5, $29, 4 ; 184: addiu $a2 $sp 4 # argv
[00400002] 34a60000 addiu $5, $5, 4 ; 185: addiu $a2 $a2 4 # envp
[00400003] 00041080 sll $2, $4, 2 ; 186: sll $a2 $a2 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v0
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 34020000 nop ; 189: nop
[0040001c] 34020004 ori $2, $0, 10 ; 190: li $v0 10
[00400020] 00000000 syscall ; 191: li $v0 10
[00400024] 34020004 lw $4, 4097 [$strNameId] ; 192: syscall # syscall 10 (ex)
[00400028] 34020004 ori $2, $0, 4 ; 193: li $a0, strNameId
[0040002c] 00000000 syscall ; 194: syscall
[00400030] 3c011001 lui $1, 4097 [readX] ; 195: la $a0, readX
[00400034] 34240021 ori $4, $1, 33 [readX] ; 196: li $v0, 4
[00400038] 34020004 ori $2, $0, 4 ; 197: syscall
[0040003c] 00000000 syscall ; 198: syscall
[00400040] 34020006 ori $2, $0, 6 ; 199: li $v0, 6
[00400044] 00000000 syscall ; 200: syscall
[00400048] 34020004 ori $2, $0, 4 ; 201: li $v0, 4
[0040004c] 3c011001 lui $1, 4097 [readY] ; 202: la $a0, readY
[00400050] 34240038 ori $4, $1, 56 [readY] ; 203: li $v0, 6
[00400054] 34020004 ori $2, $0, 4 ; 204: syscall
[00400058] 00000000 syscall ; 205: syscall
[0040005c] 34020006 ori $2, $0, 6 ; 206: li $v0, 6
[00400060] 00000000 syscall ; 207: syscall
[00400064] 3c011001 lui $1, 4097 ; 208: li $a0, strNameId
[00400068] 34240070 lwcl $4, 112($1) ; 209: lwcl $f4, $f4, $f2 # 5.4
[00400072] 34020002 mul.s $f4, $f4, $f2 ; 210: mul.s $f4, $f4, $f0
[00400076] 46002102 mul.s $f4, $f4, $f0 ; 211: mul.s $f4, $f4, $f0 # 5.4
[0040007a] 3c011001 lui $1, 4097 ; 212: la $a0, bb
[00400084] 34020004 lwcl $4, 120($1) ; 213: lwcl $f4, $f6, $f2 # 18.
[00400088] 46023182 mul.s $f6, $f6, $f2 ; 214: mul.s $f6, $f6, $f0 # 18.
[0040008c] 34020004 lwcl $4, 116($1) ; 215: lwcl $f4, $f4, $f6 # 12.
[00400090] 46002102 mul.s $f6, $f6, $f0 ; 216: mul.s $f6, $f6, $f0 # 12.
[00400094] 34020004 sub.s $f4, $f4, $f6 ; 217: sub.s $f4, $f4, $f6 # 5.4
[00400098] 3c011001 lui $1, 4097 ; 218: la $a0, c
[0040009c] 34020004 lwcl $4, 112($1) ; 219: lwcl $f4, $f4, $f2 # 5.4
[004000a0] 46002102 mul.s $f4, $f4, $f2 ; 220: mul.s $f4, $f4, $f0 # 5.4
[004000a4] 34020004 ori $2, $0, 4 ; 221: syscall
[004000a8] 00000000 syscall ; 222: syscall
[004000ac] 34020004 ori $2, $0, 6 ; 223: li $v0, 6
[004000b0] 00000000 syscall ; 224: syscall
[004000b4] 46000096 mov.s $f2, $f0 ; 225: mov.s $f2, $f0
[004000b8] 3c011001 lui $1, 4097 [readY] ; 226: la $a0, readyY
[004000bc] 34240038 ori $4, $1, 56 [readY] ; 227: li $v0, 4
[004000c0] 34020004 ori $2, $0, 4 ; 228: syscall
[004000c4] 00000000 syscall ; 229: syscall
[004000c8] 34020006 ori $2, $0, 6 ; 230: li $v0, 6
[004000cc] 00000000 syscall ; 231: syscall
[004000d0] 34020004 ori $2, $0, 4 ; 232: li $v0, 4
[004000d4] 3c011001 lui $1, 4097 ; 233: la $a0, a
[004000d8] 34240070 lwcl $4, 112($1) ; 234: lwcl $f4, $f4, $f6 # 5.4
[004000e2] 46002102 mul.s $f4, $f4, $f2 ; 235: mul.s $f4, $f4, $f0 # 5.4
[004000e6] 34020004 ori $2, $0, 4 ; 236: syscall
[004000ea] 00000000 syscall ; 237: syscall
[004000f4] 34020004 ori $2, $0, 6 ; 238: li $v0, 6
[004000f8] 00000000 syscall ; 239: syscall
[004000fc] 34020004 ori $2, $0, 4 ; 240: li $v0, 4
[00400100] 3c011001 lui $1, 4097 ; 241: la $a0, bb
[00400104] 34020004 lwcl $4, 120($1) ; 242: lwcl $f4, $f6, $f2 # 18.
[00400108] 46023182 mul.s $f6, $f6, $f2 ; 243: mul.s $f6, $f6, $f0 # 18.
[00400112] 34020004 lwcl $4, 116($1) ; 244: lwcl $f4, $f4, $f6 # 12.
[00400116] 46002102 mul.s $f6, $f6, $f0 ; 245: mul.s $f6, $f6, $f0 # 12.
[00400120] 34020004 sub.s $f4, $f4, $f6 ; 246: sub.s $f4, $f4, $f6 # 5.4
[00400124] 3c011001 lui $1, 4097 ; 247: la $a0, c
[00400128] 34020004 lwcl $4, 112($1) ; 248: lwcl $f4, $f4, $f2 # 5.4
[00400132] 46002102 mul.s $f4, $f4, $f2 ; 249: mul.s $f4, $f4, $f0 # 5.4
[00400136] 34020004 ori $2, $0, 4 ; 250: syscall
[00400140] 00000000 syscall ; 251: syscall
[00400144] 34020004 ori $2, $0, 6 ; 252: li $v0, 6
[00400148] 00000000 syscall ; 253: syscall
[00400152] 34020004 ori $2, $0, 4 ; 254: li $v0, 4
[00400156] 3c011001 lui $1, 4097 ; 255: la $a0, a
[00400160] 34240038 ori $4, $1, 56 [readY] ; 256: li $v0, 4
[00400164] 34020004 ori $2, $0, 4 ; 257: syscall
[00400168] 00000000 syscall ; 258: syscall
[00400172] 34020004 ori $2, $0, 6 ; 259: li $v0, 6
[00400176] 00000000 syscall ; 260: syscall
[00400180] 34020004 ori $2, $0, 4 ; 261: li $v0, 4
[00400184] 3c011001 lui $1, 4097 ; 262: la $a0, bb
[00400188] 34020004 lwcl $4, 120($1) ; 263: lwcl $f4, $f6, $f2 # 18.
[00400192] 46023182 mul.s $f6, $f6, $f2 ; 264: mul.s $f6, $f6, $f0 # 18.
[00400196] 34020004 lwcl $4, 116($1) ; 265: lwcl $f4, $f4, $f6 # 12.
[00400200] 46002102 mul.s $f6, $f6, $f0 ; 266: mul.s $f6, $f6, $f0 # 12.
[00400204] 34020004 sub.s $f4, $f4, $f6 ; 267: sub.s $f4, $f4, $f6 # 5.4
[00400208] 3c011001 lui $1, 4097 ; 268: la $a0, c
[00400212] 34020004 lwcl $4, 112($1) ; 269: lwcl $f4, $f4, $f2 # 5.4
[00400216] 46002102 mul.s $f4, $f4, $f2 ; 270: mul.s $f4, $f4, $f0 # 5.4
[00400220] 34020004 ori $2, $0, 4 ; 271: syscall
[00400224] 00000000 syscall ; 272: syscall
[00400228] 34020004 ori $2, $0, 6 ; 273: li $v0, 6
[00400232] 00000000 syscall ; 274: syscall
[00400236] 34020004 ori $2, $0, 4 ; 275: li $v0, 4
[00400240] 3c011001 lui $1, 4097 ; 276: la $a0, a
[00400244] 34240038 ori $4, $1, 56 [readY] ; 277: li $v0, 4
[00400248] 34020004 ori $2, $0, 4 ; 278: syscall
[00400252] 00000000 syscall ; 279: syscall
[00400256] 34020004 ori $2, $0, 6 ; 280: li $v0, 6
[00400260] 00000000 syscall ; 281: syscall
[00400264] 34020004 ori $2, $0, 4 ; 282: li $v0, 4
[00400268] 3c011001 lui $1, 4097 ; 283: la $a0, bb
[00400272] 34020004 lwcl $4, 120($1) ; 284: lwcl $f4, $f6, $f2 # 18.
[00400276] 46023182 mul.s $f6, $f6, $f2 ; 285: mul.s $f6, $f6, $f0 # 18.
[00400280] 34020004 lwcl $4, 116($1) ; 286: lwcl $f4, $f4, $f6 # 12.
[00400284] 46002102 mul.s $f6, $f6, $f0 ; 287: mul.s $f6, $f6, $f0 # 12.
[00400288] 34020004 sub.s $f4, $f4, $f6 ; 288: sub.s $f4, $f4, $f6 # 5.4
[00400292] 3c011001 lui $1, 4097 ; 289: la $a0, c
[00400296] 34020004 lwcl $4, 112($1) ; 290: lwcl $f4, $f4, $f2 # 5.4
[00400300] 46002102 mul.s $f4, $f4, $f2 ; 291: mul.s $f4, $f4, $f0 # 5.4
[00400304] 34020004 ori $2, $0, 4 ; 292: syscall
[00400308] 00000000 syscall ; 293: syscall
[00400312] 34020004 ori $2, $0, 6 ; 294: li $v0, 6
[00400316] 00000000 syscall ; 295: syscall
[00400320] 34020004 ori $2, $0, 4 ; 296: li $v0, 4
[00400324] 3c011001 lui $1, 4097 ; 297: la $a0, a
[00400328] 34240038 ori $4, $1, 56 [readY] ; 298: li $v0, 4
[00400332] 34020004 ori $2, $0, 4 ; 299: syscall
[00400336] 00000000 syscall ; 300: syscall
[00400340] 34020004 ori $2, $0, 6 ; 301: li $v0, 6
[00400344] 00000000 syscall ; 302: syscall
[00400348] 34020004 ori $2, $0, 4 ; 303: li $v0, 4
[00400352] 3c011001 lui $1, 4097 ; 304: la $a0, bb
[00400356] 34020004 lwcl $4, 120($1) ; 305: lwcl $f4, $f6, $f2 # 18.
[00400360] 46023182 mul.s $f6, $f6, $f2 ; 306: mul.s $f6, $f6, $f0 # 18.
[00400364] 34020004 lwcl $4, 116($1) ; 307: lwcl $f4, $f4, $f6 # 12.
[00400368] 46002102 mul.s $f6, $f6, $f0 ; 308: mul.s $f6, $f6, $f0 # 12.
[00400372] 34020004 sub.s $f4, $f4, $f6 ; 309: sub.s $f4, $f4, $f6 # 5.4
[00400376] 3c011001 lui $1, 4097 ; 310: la $a0, c
[00400380] 34020004 lwcl $4, 112($1) ; 311: lwcl $f4, $f4, $f2 # 5.4
[00400384] 46002102 mul.s $f4, $f4, $f2 ; 312: mul.s $f4, $f4, $f0 # 5.4
[00400388] 34020004 ori $2, $0, 4 ; 313: syscall
[00400392] 00000000 syscall ; 314: syscall
[00400396] 34020004 ori $2, $0, 6 ; 315: li $v0, 6
[00400400] 00000000 syscall ; 316: syscall
[00400404] 34020004 ori $2, $0, 4 ; 317: li $v0, 4
[00400408] 3c011001 lui $1, 4097 ; 318: la $a0, a
[00400412] 34240038 ori $4, $1, 56 [readY] ; 319: li $v0, 4
[00400416] 34020004 ori $2, $0, 4 ; 320: syscall
[00400420] 00000000 syscall ; 321: syscall
[00400424] 34020004 ori $2, $0, 6 ; 322: li $v0, 6
[00400428] 00000000 syscall ; 323: syscall
[00400432] 34020004 ori $2, $0, 4 ; 324: li $v0, 4
[00400436] 3c011001 lui $1, 4097 ; 325: la $a0, bb
[00400440] 34020004 lwcl $4, 120($1) ; 326: lwcl $f4, $f6, $f2 # 18.
[00400444] 46023182 mul.s $f6, $f6, $f2 ; 327: mul.s $f6, $f6, $f0 # 18.
[00400448] 34020004 lwcl $4, 116($1) ; 328: lwcl $f4, $f4, $f6 # 12.
[00400452] 46002102 mul.s $f6, $f6, $f0 ; 329: mul.s $f6, $f6, $f0 # 12.
[00400456] 34020004 sub.s $f4, $f4, $f6 ; 330: sub.s $f4, $f4, $f6 # 5.4
[00400460] 3c011001 lui $1, 4097 ; 331: la $a0, c
[00400464] 34020004 lwcl $4, 112($1) ; 332: lwcl $f4, $f4, $f2 # 5.4
[00400468] 46002102 mul.s $f4, $f4, $f2 ; 333: mul.s $f4, $f4, $f0 # 5.4
[00400472] 34020004 ori $2, $0, 4 ; 334: syscall
[00400476] 00000000 syscall ; 335: syscall
[00400480] 34020004 ori $2, $0, 6 ; 336: li $v0, 6
[00400484] 00000000 syscall ; 337: syscall
[00400488] 34020004 ori $2, $0, 4 ; 338: li $v0, 4
[00400492] 3c011001 lui $1, 4097 ; 339: la $a0, a
[00400496] 34240038 ori $4, $1, 56 [readY] ; 340: li $v0, 4
[00400500] 34020004 ori $2, $0, 4 ; 341: syscall
[00400504] 00000000 syscall ; 342: syscall
[00400508] 34020004 ori $2, $0, 6 ; 343: li $v0, 6
[00400512] 00000000 syscall ; 344: syscall
[00400516] 34020004 ori $2, $0, 4 ; 345: li $v0, 4
[00400520] 3c011001 lui $1, 4097 ; 346: la $a0, bb
[00400524] 34020004 lwcl $4, 120($1) ; 347: lwcl $f4, $f6, $f2 # 18.
[00400528] 46023182 mul.s $f6, $f6, $f2 ; 348: mul.s $f6, $f6, $f0 # 18.
[00400532] 34020004 lwcl $4, 116($1) ; 349: lwcl $f4, $f4, $f6 # 12.
[00400536] 46002102 mul.s $f6, $f6, $f0 ; 350: mul.s $f6, $f6, $f0 # 12.
[00400540] 34020004 sub.s $f4, $f4, $f6 ; 351: sub.s $f4, $f4, $f6 # 5.4
[00400544] 3c011001 lui $1, 4097 ; 352: la $a0, c
[00400548] 34020004 lwcl $4, 112($1) ; 353: lwcl $f4, $f4, $f2 # 5.4
[00400552] 46002102 mul.s $f4, $f4, $f2 ; 354: mul.s $f4, $f4, $f0 # 5.4
[00400556] 34020004 ori $2, $0, 4 ; 355: syscall
[00400560] 00000000 syscall ; 356: syscall
[00400564] 34020004 ori $2, $0, 6 ; 357: li $v0, 6
[00400568] 00000000 syscall ; 358: syscall
[00400572] 34020004 ori $2, $0, 4 ; 359: li $v0, 4
[00400576] 3c011001 lui $1, 4097 ; 360: la $a0, a
[00400580] 34240038 ori $4, $1, 56 [readY] ; 361: li $v0, 4
[00400584] 34020004 ori $2, $0, 4 ; 362: syscall
[00400588] 00000000 syscall ; 363: syscall
[00400592] 34020004 ori $2, $0, 6 ; 364: li $v0, 6
[00400596] 00000000 syscall ; 365: syscall
[00400600] 34020004 ori $2, $0, 4 ; 366: li $v0, 4
[00400604] 3c011001 lui $1, 4097 ; 367: la $a0, bb
[00400608] 34020004 lwcl $4, 120($1) ; 368: lwcl $f4, $f6, $f2 # 18.
[00400612] 46023182 mul.s $f6, $f6, $f2 ; 369: mul.s $f6, $f6, $f0 # 18.
[00400616] 34020004 lwcl $4, 116($1) ; 370: lwcl $f4, $f4, $f6 # 12.
[00400620] 46002102 mul.s $f6, $f6, $f0 ; 371: mul.s $f6, $f6, $f0 # 12.
[00400624] 34020004 sub.s $f4, $f4, $f6 ; 372: sub.s $f4, $f4, $f6 # 5.4
[00400628] 3c011001 lui $1, 4097 ; 373: la $a0, c
[00400632] 34020004 lwcl $4, 112($1) ; 374: lwcl $f4, $f4, $f2 # 5.4
[00400636] 46002102 mul.s $f4, $f4, $f2 ; 375: mul.s $f4, $f4, $f0 # 5.4
[00400640] 34020004 ori $2, $0, 4 ; 376: syscall
[00400644] 00000000 syscall ; 377: syscall
[00400648] 34020004 ori $2, $0, 6 ; 378: li $v0, 6
[00400652] 00000000 syscall ; 379: syscall
[00400656] 34020004 ori $2, $0, 4 ; 380: li $v0, 4
[00400660] 3c011001 lui $1, 4097 ; 381: la $a0, a
[00400664] 34240038 ori $4, $1, 56 [readY] ; 382: li $v0, 4
[00400668] 34020004 ori $2, $0, 4 ; 383: syscall
[00400672] 00000000 syscall ; 384: syscall
[00400676] 34020004 ori $2, $0, 6 ; 385: li $v0, 6
[00400680] 00000000 syscall ; 386: syscall
[00400684] 34020004 ori $2, $0, 4 ; 387: li $v0, 4
[00400688] 3c011001 lui $1, 4097 ; 389: la $a0, bb
[00400692] 34020004 lwcl $4, 120($1) ; 390: lwcl $f4, $f6, $f2 # 18.
[00400696] 46023182 mul.s $f6, $f6, $f2 ; 391: mul.s $f6, $f6, $f0 # 18.
[00400700] 34020004 lwcl $4, 116($1) ; 392: lwcl $f4, $f4, $f6 # 12.
[00400704] 46002102 mul.s $f6, $f6, $f0 ; 393: mul.s $f6, $f6, $f0 # 12.
[00400708] 34020004 sub.s $f4, $f4, $f6 ; 394: sub.s $f4, $f4, $f6 # 5.4
[00400712] 3c011001 lui $1, 4097 ; 395: la $a0, c
[00400716] 34020004 lwcl $4, 112($1) ; 396: lwcl $f4, $f4, $f2 # 5.4
[00400720] 46002102 mul.s $f4, $f4, $f2 ; 397: mul.s $f4, $f4, $f0 # 5.4
[00400724] 34020004 ori $2, $0, 4 ; 398: syscall
[00400728] 00000000 syscall ; 399: syscall
[00400732] 34020004 ori $2, $0, 6 ; 400: li $v0, 6
[00400736] 00000000 syscall ; 401: syscall
[00400740] 34020004 ori $2, $0, 4 ; 402: li $v0, 4
[00400744] 3c011001 lui $1, 4097 ; 403: la $a0, a
[00400748] 34240038 ori $4, $1, 56 [readY] ; 404: li $v0, 4
[00400752] 34020004 ori $2, $0, 4 ; 405: syscall
[00400756] 00000000 syscall ; 406: syscall
[00400760] 34020004 ori $2, $0, 6 ; 407: li $v0, 6
[00400764] 00000000 syscall ; 408: syscall
[00400768] 34020004 ori $2, $0, 4 ; 409: li $v0, 4
[00400772] 3c011001 lui $1, 4097 ; 410: la $a0, bb
[00400776] 34020004 lwcl $4, 120($1) ; 411: lwcl $f4, $f6, $f2 # 18.
[00400780] 46023182 mul.s $f6, $f6, $f2 ; 412: mul.s $f6, $f6, $f0 # 18.
[00400784] 34020004 lwcl $4, 116($1) ; 413: lwcl $f4, $f4, $f6 # 12.
[00400788] 46002102 mul.s $f6, $f6, $f0 ; 414: mul.s $f6, $f6, $f0 # 12.
[00400792] 34020004 sub.s $f4, $f4, $f6 ; 415: sub.s $f4, $f4, $f6 # 5.4
[00400796] 3c011001 lui $1, 4097 ; 416: la $a0, c
[00400800] 34020004 lwcl $4, 112($1) ; 417: lwcl $f4, $f4, $f2 # 5.4
[00400804] 46002102 mul.s $f4, $f4, $f2 ; 418: mul.s $f4, $f4, $f0 # 5.4
[00400808] 34020004 ori $2, $0, 4 ; 419: syscall
[00400812] 00000000 syscall ; 420: syscall
[00400816] 34020004 ori $2, $0, 6 ; 421: li $v0, 6
[00400820] 00000000 syscall ; 422: syscall
[00400824] 34020004 ori $2, $0, 4 ; 423: li $v0, 4
[00400828] 3c011001 lui $1, 4097 ; 424: la $a0, a
[00400832] 34240038 ori $4, $1, 56 [readY] ; 425: li $v0, 4
[00400836] 34020004 ori $2, $0, 4 ; 426: syscall
[00400840] 00000000 syscall ; 427: syscall
[00400844] 34020004 ori $2, $0, 6 ; 428: li $v0, 6
[00400848] 00000000 syscall ; 429: syscall
[00400852] 34020004 ori $2, $0, 4 ; 430: li $v0, 4
[00400856] 3c011001 lui $1, 4097 ; 431: la $a0, bb
[00400860] 34020004 lwcl $4, 120($1) ; 432: lwcl $f4, $f6, $f2 # 18.
[00400864] 46023182 mul.s $f6, $f6, $f2 ; 433: mul.s $f6, $f6, $f0 # 18.
[00400868] 34020004 lwcl $4, 116($1) ; 434: lwcl $f4, $f4, $f6 # 12.
[00400872] 46002102 mul.s $f6, $f6, $f0 ; 435: mul.s $f6, $f6, $f0 # 12.
[00400876] 34020004 sub.s $f4, $f4, $f6 ; 436: sub.s $f4, $f4, $f6 # 5.4
[00400880] 3c011001 lui $1, 4097 ; 437: la $a0
```

FP Regs ▾ Text

FC21 = 0	[00400014] Oc010009 jal 0x00400024 [main] ; 138: jal main
FC22 = 0	[00400018] 00000000 nop ; 189: nop
FC23 = 0	[0040001c] 3402000a ori \$2, \$0, 10 ; 191: li \$v0, 10
FC24 = 0	[00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit+)
FC25 = 0	[00400024] 3c041001 lui \$4, 4097 [strNameId] ; 15: la \$a0, strNameId
FC26 = 0	[00400028] 34020004 ori \$2, \$0, 4 ; 16: li \$v0, 4
FC27 = 0	[0040002c] 0000000c syscall ; 17: syscall
FC28 = 0	[00400030] 34020001 lui \$4, 4097 [readX] ; 19: la \$a0, readX
FC29 = 0	[00400034] 34020021 ori \$4, \$1, 33 [readX] ; 20: li \$v0, 4
FC30 = 0	[00400038] 34020004 ori \$2, \$0, 4 ; 21: syscall
FC31 = 0	[0040003c] 0000000c syscall ; 23: li \$v0, 6
Double Precision	[00400040] 34020006 ori \$2, \$0, 6 ; 24: syscall
FP0 = 40a00000	[00400044] 0000000c syscall ; 26: mov.s \$f2, \$f0
FP2 = 40a00000	[00400048] 46000086 mov.s \$t2, \$f0 ; 28: la \$a0, readY
FP4 = 43220000	[0040004c] 3c011001 lui \$1, 4097 [readY] ; 29: li \$v0, 4
FP6 = 0	[00400050] 34240038 ori \$4, \$1, 56 [readY] ; 30: syscall
FP8 = 0	[00400054] 34020004 ori \$2, \$0, 4 ; 32: li \$v0, 6
FP10 = 0	[00400058] 0000000c syscall ; 33: syscall
FP12 = 0	[00400060] 3c011001 lui \$1, 4097 ; 37: l.s \$f4, a
FP14 = 0	[00400068] c4240070 lwl.s \$f4, 112(\$1) ; 38: mul.s \$f4, \$f4, \$f2 # 5.4
FP16 = 0	[0040006c] 46022102 mul.s \$f4, \$f4, \$f2 ; 39: mul.s \$f4, \$f4, \$f0 # 5.4
FP18 = 0	[00400070] 46020102 mul.s \$f4, \$f4, \$f0 ; 40: l.s \$f6, b
FP20 = 0	[00400074] 3c011001 lui \$1, 4097 ; 41: mul.s \$f6, \$f6, \$f0 # 12.
FP22 = 0	[00400078] c4260078 lwl.s \$f6, 116(\$1) ; 42: sub.s \$f4, \$f4, \$f6 # 5.4
FP24 = 0	[0040007c] 46003182 mul.s \$f6, \$f6, \$f0 ; 43: l.s \$f6, c
FP26 = 0	[00400080] 46062101 sub.s \$f4, \$f4, \$f6 ; 44: mul.s \$f6, \$f6, \$f2 # 18.
FP28 = 0	[00400084] 3c011001 lui \$1, 4097 ; 45: add.s \$f4, \$f4, \$f6 # 5.4
FP30 = 0	[00400088] c4260079 lwl.s \$f6, 120(\$1) ; 46: add.s \$f4, \$f4, \$f6 # 5.4

FP Regs	Op	Text
FC21 = 0	[00400014] 0c100009	jal 0x00400024 [main] ; 188: jal main
FC22 = 0	[00400018] 00000000	nop ; 189: nop
FC23 = 0	[0040001c] 3402000a	ori \$2, \$0, 10 ; 191: li \$v0 10
FC24 = 0	[00400020] 00000000	syscall ; 192: syscall # syscall 10 (exit)
FC25 = 0	[00400024] 3c014001	lui \$4, 4097 [strNameId] ; 15: la \$a0, strNameId
FC26 = 0	[00400028] 34020004	ori \$2, \$0, 4 ; 16: li \$v0, 4
FG27 = 0	[00400030] 00000000	syscall ; 17: syscall
FG28 = 0	[00400030] 3c011001	lui \$4, 4097 [readX] ; 19: la \$a0, readX
FG29 = 0	[00400034] 34020021	ori \$4, \$1, 33 [readX] ; 20: li \$v0, 4
FG30 = 0	[00400038] 34020004	ori \$2, \$0, 4 ; 21: syscall
FG31 = 0	[0040003c] 00000000	syscall ; 23: li \$v0, 6
Double Precision		
FP0 = 0	[00400040] 34020006	ori \$2, \$0, 6 ; 24: syscall
FP0 = 0	[00400044] 00000000	syscall ; 26: mov.s \$f2, \$f0
FP0 = 0	[00400048] 46000086	mov.s \$f2, \$f0 ; 28: la \$a0, ready
FP0 = 0	[0040004c] 3c011001	lui \$1, 4097 [ready] ; 29: li \$v0, 4
FP2 = 0	[00400050] 34020030	ori \$4, \$1, 56 [readyY] ; 30: syscall
FP2 = 0	[00400054] 00000000	00000004 ; 32: li \$v0, 6
FP8 = 0	[00400058] 00000000	syscall ; 33: syscall
FP8 = 0	[0040005c] 34020006	ori \$2, \$0, 6 ; 37: la \$f4, a
FP10 = 0	[00400060] 00000000	syscall ; 38: mul.s \$f4, \$f4, \$f2 # 5.4x
FP12 = 0	[00400064] 3c011001	lui \$1, 4097 ; 39: mul.s \$f4, \$f4, \$f0 # 5.4xy
FP14 = 0	[00400068] c4240070	lwl \$1, \$4, 112(\$1) ; 40: 1.s \$f6, bb
FP16 = 0	[0040006c] 46022102	mul.s \$f4, \$f4, \$f2
FP18 = 0	[00400070] 46002102	mul.s \$f4, \$f4, \$f0
FP20 = 0	[00400074] 3c011001	lui \$1, 4097 ; 41: mul.s \$f6, \$f6, \$f0 # 12.3y
FP22 = 0	[00400078] c4260074	lwl \$1, \$6, 116(\$1) ; 42: sub.s \$f4, \$f4, \$f6 # 12.3y
FP24 = 0	[0040007c] 46003182	mul.s \$f6, \$f6, \$f0 # 12.3y
FP26 = 0	[00400080] 46022101	sub.s \$f4, \$f4, \$f6 ; 43: 1.s \$f6, c
FP28 = 0	[00400084] 3c011001	lui \$1, 4097
FP30 = 0	[00400088] c4260078	lwl \$1, \$6, 120(\$1) ; 44: mul.s \$f6, \$f6, \$f2 # 18.23x
	[0040008c] 46023182	mul.s \$f6, \$f6, \$f2 # 18.23x


```
Memory and registers cleared
ARM64 Version 0.1.24 of August 1. 2023 (final)
# Regs   F  Text
[0] PC = 0
[1] R0 = 0
[2] R1 = 0
[3] R2 = 0
[4] R3 = 0
[5] R4 = 0
[6] R5 = 0
[7] R6 = 0
[8] R7 = 0
[9] R8 = 0
[10] R9 = 0
[11] R10 = 0
[12] R11 = 0
[13] R12 = 0
[14] R13 = 0
[15] R14 = 0
[16] R15 = 0
[17] R16 = 0
[18] R17 = 0
[19] R18 = 0
[20] R19 = 0
[21] R20 = 0
[22] R21 = 0
[23] R22 = 0
[24] R23 = 0
[25] R24 = 0
[26] R25 = 0
[27] R26 = 0
[28] R27 = 0
[29] R28 = 0
[30] R29 = 0
[31] R30 = 0
[32] R31 = 0

Double Precision
[33] FP0 = 40c00000
[34] FP1 = 40a00000
[35] FP2 = 40a00000
[36] FP3 = 43220000
[37] FP4 = 43220000
[38] FP5 = 43220000
[39] FP6 = 43220000
[40] FP7 = 43220000
[41] FP8 = 0
[42] FP10 = 0
[43] FP11 = 0
[44] FP12 = 0
[45] FP13 = 0
[46] FP14 = 0
[47] FP15 = 0
[48] FP16 = 0
[49] FP18 = 0
[50] FP20 = 0
[51] FP22 = 0
[52] FP24 = 0
[53] FP26 = 0
[54] FP28 = 0
[55] FP30 = 0

[0] [00000014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[1] [00000018] 00000000 nop ; 189: nop
[2] [0000001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[3] [00000020] 00000000 syscall ; 192: syscall # syscall 10 (exit)
[4] [00000024] 3c041001 lui $4, 4097 [strNameId] ; 15: la $a0, strNameId
[5] [00000028] 34020004 ori $2, $0, 4 ; 16: li $v0, 4
[6] [0000002c] 00000000 syscall ; 17: syscall
[7] [00000030] 3c011001 lui $1, 4097 [readX] ; 19: la $a0, readX
[8] [00000034] 34020021 ori $2, $1, 33 [readX] ; 20: li $v0, 4
[9] [00000038] 34020024 ori $2, $0, 4 ; 21: syscall
[10] [0000003c] 00000000 syscall ; 22: syscall
[11] [00000040] 34020006 ori $2, $0, 6 ; 23: li $v0, 6
[12] [00000044] 00000000 syscall ; 24: syscall
[13] [00000048] 46000086 mov.s $f2, $f0 ; 26: mov.s $f2, $f0
[14] [0000004c] 3c011001 lui $1, 4097 [readY] ; 28: la $a0, ready
[15] [00000050] 34020024 ori $2, $1, 56 [readY] ; 29: li $v0, 4
[16] [00000054] 34020004 ori $2, $0, 4 ; 30: syscall
[17] [00000058] 00000000 syscall ; 31: syscall
[18] [0000005c] 34020006 ori $2, $0, 6 ; 32: li $v0, 6
[19] [00000060] 00000000 syscall ; 33: syscall
[20] [00000064] 3c011001 lui $1, 4097 ; 37: l.s $f4, a
[21] [00000068] c4240070 lws1 $f4, 112($1) ; 38: mul.s $f4, $f4, $f2 # 5.4x
[22] [0000006c] 46022102 mul.s $f4, $f4, $f2 ; 39: mul.s $f4, $f4, $f2 # 5.4xy
[23] [00000070] 46022102 mul.s $f4, $f4, $f0 ; 40: l.s $f6, bb
[24] [00000074] 46022102 lui $1, 4097 ; 41: mul.s $f6, $f6, $f0 # 12.3y
[25] [00000078] 46022102 lws1 $f6, 116($1) ; 42: sub.s $f4, $f4, $f6 # 5.4xy - 12.3y
[26] [0000007c] 46003182 mul.s $f6, $f6, $f0 ; 43: l.s $f6, c
[27] [00000080] 46062101 sub.s $f4, $f4, $f6 ; 44: mul.s $f6, $f6, $f2 # 18.2x
[28] [00000084] 3c011001 lui $1, 4097 ; 45: add.s $f4, $f4, $f6 # 18.2y - 19.2x
[29] [00000088] c4260078 lws1 $f6, 120($1) ; 46: add.s $f4, $f4, $f6 # 18.2xy
[30] [0000008c] 46023182 mul.s $f6, $f6, $f2 ; 47: add.s $f4, $f4, $f6 # 18.2z
[31] [00000090] 46003100 add.s $f4, $f4, $f6 ; 48: add.s $f4, $f4, $f6 # 18.2xy - 19.2z

Memory and registers cleared
```

```

FG21 = 0
FG22 = 0
FG23 = 0
FG24 = 0
FG25 = 0
FG26 = 0
FG27 = 0
FG28 = 0
FG29 = 0
FG30 = 0
FG31 = 0

Double Precision
FP0 = 40c00000
FP1 = 40a00000
FP2 = 42b06666
FP3 = 4253999a
FP4 = 4253999a
FP5 = 4253999a
FP6 = 4253999a
FP7 = 4253999a
FP8 = 4253999a
FP9 = 4253999a
FP10 = 0
FP11 = 0
FP12 = 0
FP13 = 0
FP14 = 0
FP15 = 0
FP16 = 0
FP17 = 0
FP18 = 0
FP19 = 0
FP20 = 0
FP21 = 0
FP22 = 0
FP23 = 0
FP24 = 0
FP25 = 0
FP26 = 0
FP27 = 0
FP28 = 0
FP29 = 0
FP30 = 0

[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0, 10
[00400020] 0000000a syscall ; 192: syscall # syscall 10 (exit)
[00400024] 3c41001 lui $4, 4097 [strNameId] ; 15: la $a0, strNameId
[00400028] 34020004 ori $2, $0, 4 ; 16: li $v0, 4
[0040002c] 3402000c syscall ; 17: syscall
[00400030] 3c011001 lui $1, 4097 [readX] ; 19: la $a0, readX
[00400034] 34240201 ori $4, $1, 33 [readX] ; 20: li $v0, 4
[00400038] 34020004 ori $2, $0, 4 ; 21: syscall
[0040003c] 0000000c syscall ; 22: syscall
[00400040] 34020006 ori $2, $0, 6 ; 23: li $v0, 6
[00400044] 00000006 syscall ; 24: syscall
[00400048] 46000086 mov.s $f2, $f0 ; 26: mov.s $f2, $f0
[0040004c] 3c011001 lui $1, 4097 [readY] ; 28: la $a0, ready
[00400050] 34020004 ori $2, $0, 4 ; 29: li $v0, 4
[00400054] 0000000c syscall ; 30: syscall
[0040005c] 34020006 ori $2, $0, 6 ; 32: li $v0, 6
[00400060] 0000000c syscall ; 33: syscall
[00400064] 3c011001 lui $1, 4097 ; 37: 1.s $f4, a
[00400068] 42420070 lwl $4, 112($1) ; 38: mul.s $f4, $f4, $f2 # 5.4x
[0040006c] 46022102 mul.s $f4, $f4, $f2 ; 39: mul.s $f4, $f4, $f0 # 5.4xy
[00400070] 46002102 mul.s $f4, $f4, $f0 ; 40: 1.s $f6, bb
[00400074] 3c011001 lui $1, 4097 ; 41: mul.s $f6, $f6, $f0 # 12.3y
[00400078] 42600774 lwl $6, 116($1) ; 42: sub.s $f6, $f4, $f6 # 5.4xy - 12.3y
[0040007c] 46002101 sub.s $f4, $f4, $f6 ; 43: 1.s $f6, c
[00400084] 3c011001 lui $1, 4097 ; 44: mul.s $f6, $f6, $f2 # 18.23x
[00400088] 42600782 lwl $6, 120($1) ; 45: add.s $f6, $f4, $f6 # 5.4xy + 18.23x
[0040008c] 46022102 mul.s $f4, $f4, $f2 ; 46: 1.s $f6, d
[00400090] 46022100 add.s $f4, $f4, $f6 ; 47: sub.s $f6, $f4, $f6 # 5.4xy - 12.3y + 18.23x
[00400094] 3c011001 lui $1, 4097 ; 48: la $a0, printResult
[00400098] 42600778 lwl $6, 124($1) ; 49: 1a $a0, printResult
[0040009c] 46002101 sub.s $f4, $f4, $f6 ; 50: li $v0, 4
[004000a0] 3c011001 lui $1, 4097 ; 51: syscall
[004000a4] 34020004 ori $4, $1, 79 [printResult] ; 52: mov.s $f12, $f4
[004000a8] 3402000c syscall ; 53: mov.s $f12, $f4
[004000b0] 46002306 mov.s $f12, $f4 ; 54: li $v0, 2

Memory and registers cleared

```

SFM Version 9.1.24 of August 1, 2023 (final)
Copyright 1990-2023 by James Larus.
All Rights Reserved.
SFM is distributed under a BSD license.
See the file LICENSE for full copyright notice.
QCSFIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.

Memory and registers cleared

```

FP Regs   Text
FG21 = 0
FG22 = 0
FG23 = 0
FG24 = 0
FG25 = 0
FG26 = 0
FG27 = 0
FG28 = 0
FG29 = 0
FG30 = 0
FG31 = 0

Double Precision
FP0 = 40c00000
FP1 = 40a00000
FP2 = 42b06666
FP3 = 4253999a
FP4 = 4253999a
FP5 = 4253999a
FP6 = 4253999a
FP7 = 4253999a
FP8 = 4253999a
FP9 = 4253999a
FP10 = 0
FP11 = 0
FP12 = 0
FP13 = 0
FP14 = 0
FP15 = 0
FP16 = 0
FP17 = 0
FP18 = 0
FP19 = 0
FP20 = 0
FP21 = 0
FP22 = 0
FP23 = 0
FP24 = 0
FP25 = 0
FP26 = 0
FP27 = 0
FP28 = 0
FP29 = 0
FP30 = 0

[0040003c] 0000000c syscall ; 21: syscall
[00400040] 34020006 ori $2, $0, 6 ; 23: li $v0, 6
[00400044] 0000000c syscall ; 24: syscall
[00400048] 46000086 mov.s $f2, $f0 ; 26: mov.s $f2, $f0
[0040004c] 3c011001 lui $1, 4097 [readY] ; 28: la $a0, ready
[00400050] 34020004 ori $2, $0, 4 ; 29: li $v0, 4
[00400054] 0000000c syscall ; 30: syscall
[0040005c] 34020006 ori $2, $0, 6 ; 32: li $v0, 6
[00400060] 0000000c syscall ; 33: syscall
[00400064] 3c011001 lui $1, 4097 ; 37: 1.s $f4, a
[00400068] 42420070 lwl $4, 112($1) ; 38: mul.s $f4, $f4, $f2 # 5.4x
[0040006c] 46022102 mul.s $f4, $f4, $f2 ; 39: mul.s $f4, $f4, $f0 # 5.4xy
[00400070] 46002102 mul.s $f4, $f4, $f0 ; 40: 1.s $f6, bb
[00400074] 3c011001 lui $1, 4097 ; 41: mul.s $f6, $f6, $f0 # 12.3y
[00400078] 42600774 lwl $6, 116($1) ; 42: sub.s $f6, $f4, $f6 # 5.4xy - 12.3y
[0040007c] 46002101 sub.s $f4, $f4, $f6 ; 43: 1.s $f6, c
[00400084] 3c011001 lui $1, 4097 ; 44: mul.s $f6, $f6, $f2 # 18.23x
[00400088] 42600782 lwl $6, 120($1) ; 45: add.s $f6, $f4, $f6 # 5.4xy + 18.23x
[0040008c] 46022102 mul.s $f4, $f4, $f2 ; 46: 1.s $f6, d
[00400090] 46022100 add.s $f4, $f4, $f6 ; 47: sub.s $f6, $f4, $f6 # 5.4xy - 12.3y + 18.23x
[00400094] 3c011001 lui $1, 4097 ; 48: la $a0, printResult
[00400098] 42600778 lwl $6, 124($1) ; 49: 1a $a0, printResult
[0040009c] 46002101 sub.s $f4, $f4, $f6 ; 50: li $v0, 4
[004000a0] 3c011001 lui $1, 4097 ; 51: syscall
[004000a4] 34020004 ori $4, $1, 79 [printResult] ; 52: mov.s $f12, $f4
[004000a8] 3402000c syscall ; 53: mov.s $f12, $f4
[004000b0] 46002306 mov.s $f12, $f4 ; 54: li $v0, 2

Memory and registers cleared

```

SFM Version 9.1.24 of August 1, 2023 (final)
Copyright 1990-2023 by James Larus.
All Rights Reserved.

```

FP Regs   Text
FG21 = 0
FG22 = 0
FG23 = 0
FG24 = 0
FG25 = 0
FG26 = 0
FG27 = 0
FG28 = 0
FG29 = 0
FG30 = 0
FG31 = 0

Double Precision
FP0 = 40c00000
FP1 = 40a00000
FP2 = 42b06666
FP3 = 4253999a
FP4 = 4253999a
FP5 = 4253999a
FP6 = 4253999a
FP7 = 4253999a
FP8 = 4253999a
FP9 = 4253999a
FP10 = 0
FP11 = 0
FP12 = 0
FP13 = 0
FP14 = 0
FP15 = 0
FP16 = 0
FP17 = 0
FP18 = 0
FP19 = 0
FP20 = 0
FP21 = 0
FP22 = 0
FP23 = 0
FP24 = 0
FP25 = 0
FP26 = 0
FP27 = 0
FP28 = 0
FP29 = 0
FP30 = 0

[0040003c] 0000000c syscall ; 21: syscall
[00400040] 34020006 ori $2, $0, 6 ; 23: li $v0, 6
[00400044] 0000000c syscall ; 24: syscall
[00400048] 46000086 mov.s $f2, $f0 ; 26: mov.s $f2, $f0
[0040004c] 3c011001 lui $1, 4097 [readY] ; 28: la $a0, ready
[00400050] 34020004 ori $2, $0, 4 ; 29: li $v0, 4
[00400054] 0000000c syscall ; 30: syscall
[0040005c] 34020006 ori $2, $0, 6 ; 32: li $v0, 6
[00400060] 0000000c syscall ; 33: syscall
[00400064] 3c011001 lui $1, 4097 ; 37: 1.s $f4, a
[00400068] 42420070 lwl $4, 112($1) ; 38: mul.s $f4, $f4, $f2 # 5.4x
[0040006c] 46022102 mul.s $f4, $f4, $f2 ; 39: mul.s $f4, $f4, $f0 # 5.4xy
[00400070] 46002102 mul.s $f4, $f4, $f0 ; 40: 1.s $f6, bb
[00400074] 3c011001 lui $1, 4097 ; 41: mul.s $f6, $f6, $f0 # 12.3y
[00400078] 42600774 lwl $6, 116($1) ; 42: sub.s $f6, $f4, $f6 # 5.4xy - 12.3y
[0040007c] 46002101 sub.s $f4, $f4, $f6 ; 43: 1.s $f6, c
[00400084] 3c011001 lui $1, 4097 ; 44: mul.s $f6, $f6, $f2 # 18.23x
[00400088] 42600782 lwl $6, 120($1) ; 45: add.s $f6, $f4, $f6 # 5.4xy + 18.23x
[0040008c] 46022102 mul.s $f4, $f4, $f2 ; 46: 1.s $f6, d
[00400090] 46022100 add.s $f4, $f4, $f6 ; 47: sub.s $f6, $f4, $f6 # 5.4xy - 12.3y + 18.23x
[00400094] 3c011001 lui $1, 4097 ; 48: la $a0, printResult
[00400098] 42600778 lwl $6, 124($1) ; 49: 1a $a0, printResult
[0040009c] 46002101 sub.s $f4, $f4, $f6 ; 50: li $v0, 4
[004000a0] 3c011001 lui $1, 4097 ; 51: syscall
[004000a4] 34020004 ori $4, $1, 79 [printResult] ; 52: mov.s $f12, $f4
[004000a8] 3402000c syscall ; 53: mov.s $f12, $f4
[004000b0] 46002306 mov.s $f12, $f4 ; 54: li $v0, 2

Memory and registers cleared

```

SFM Version 9.1.24 of August 1, 2023 (final)
Copyright 1990-2023 by James Larus.
All Rights Reserved.

FP Regs ⌂ Text

```

FP21 = 0          [0040003c] 0000000c syscall    ; 21: syscall
FP22 = 0          [0040003e] 00000006 ori $2, $0, 6   ; 23: li $v0, 6
FP23 = 0          [00400040] 34020000 syscall    ; 24: syscall
FP24 = 0          [00400048] 46000086 mov.s $F2, $F0   ; 26: mov.s $f2, $f0
FP25 = 0          [0040004c] 3c011001 lui $1, 4097 [readY] ; 28: la $a0, readY
FP26 = 0          [00400050] 34240038 ori $4, $1, 56 [readY]
FP27 = 0          [00400054] 34020004 ori $2, $0, 4   ; 29: li $v0, 4
FP28 = 0          [00400058] 00000000 syscall    ; 30: syscall
FP29 = 0          [0040005c] 34020006 ori $2, $0, 6   ; 32: li $v0, 6
FP30 = 0          [00400060] 00000000 syscall    ; 33: syscall
FP31 = 0          [00400068] -4240070 lwl $4, 112($1) ; 37: 1.s $f4, a
FP0 = 40c00000    [0040006c] 46022102 mul.s $F4, $F4, $F2 ; 38: mul.s $f4, $f4, $f2 # 5.4x
FP2 = 40a00000    [00400070] 46002102 mul.s $F4, $F4, $F0 ; 39: mul.s $f4, $f4, $f0 # 5.4xy
FP4 = 43335999    [00400074] 3c011001 lui $1, 4097 ; 40: 1.s $f6, bb
FP6 = 42b64ccc    [00400078] -4260074 lwc $6, 116($1)
FP8 = 0          [0040007c] 46003182 mul.s $F6, $F6, $F0 ; 41: mul.s $f6, $f6, $f0 # 12.3y
FP10 = 0          [00400080] 46062101 sub.s $F4, $F4, $F6 ; 42: sub.s $f4, $f4, $f6 # 5.4xy - 12.3y
FP12 = 0          [00400084] 3c011001 lui $1, 4097 ; 43: 1.s $f6, c
FP14 = 0          [00400088] -426007c lwc $6, 124($1)
FP16 = 0          [0040008c] 46023182 mul.s $F6, $F6, $F2 ; 44: mul.s $f6, $f6, $f2 # 18.23x
FP18 = 0          [00400090] 46062100 add.s $F4, $F4, $F6 ; 45: add.s $f4, $f4, $f6 # 5.4xy - 12.3y + 18.23x
FP20 = 0          [00400094] 3c011001 lui $1, 4097 ; 46: 1.s $f6, d
FP22 = 0          [00400098] 46002100 sub.s $F4, $F4, $F6 ; 47: sub.s $f4, $f4, $f6 # 5.4xy - 12.3y + 18.23x - 8.23
FP24 = 0          [004000a0] 3c011001 lui $1, 4097 [printResult] ; 49: la $a0, printResult
FP26 = 0          [004000a4] 34240048 ori $4, $1, 79 [printResult]
FP28 = 0          [004000a8] 34020004 ori $2, $0, 4   ; 50: li $v0, 4
FP30 = 0          [004000b0] 46002306 mov.s $F12, $F4 ; 53: mov.s $f12, $f4
FP31 = 0          [004000b4] 34020002 ori $2, $0, 2   ; 54: li $v0, 2

Memory and registers cleared
SPIM Version 9.1.24 of August 1, 2023 (final)
Copyright 1990-2023 by James Larus.
All Rights Reserved.
See the file License for a full copyright notice.

FP Regs      ⌂    Text
```

```

FP21 = 0          [0040003c] 0000000c syscall    ; 21: syscall
FP22 = 0          [0040003e] 00000006 ori $2, $0, 6   ; 23: li $v0, 6
FP23 = 0          [00400040] 34020000 syscall    ; 24: syscall
FP24 = 0          [00400048] 46000086 mov.s $F2, $F0   ; 26: mov.s $f2, $f0
FP25 = 0          [0040004c] 3c011001 lui $1, 4097 [readY] ; 28: la $a0, readY
FP26 = 0          [00400050] 34240038 ori $4, $1, 56 [readY]
FP27 = 0          [00400054] 34020004 ori $2, $0, 4   ; 29: li $v0, 4
FP28 = 0          [00400058] 00000000 syscall    ; 30: syscall
FP29 = 0          [0040005c] 34020006 ori $2, $0, 6   ; 32: li $v0, 6
FP30 = 0          [00400060] 00000000 syscall    ; 33: syscall
FP31 = 0          [00400068] -4240070 lwl $4, 112($1) ; 37: 1.s $f4, a
FP0 = 40c00000    [0040006c] 46022102 mul.s $F4, $F4, $F2 ; 38: mul.s $f4, $f4, $f2 # 5.4x
FP2 = 40a00000    [00400070] 46002102 mul.s $F4, $F4, $F0 ; 39: mul.s $f4, $f4, $f0 # 5.4xy
FP4 = 43335999    [00400074] 3c011001 lui $1, 4097 ; 40: 1.s $f6, bb
FP6 = 4103a614    [00400078] -4260074 lwc $6, 116($1)
FP8 = 0          [0040007c] 46003182 mul.s $F6, $F6, $F0 ; 41: mul.s $f6, $f6, $f0 # 12.3y
FP10 = 0          [00400080] 46062101 add.s $F4, $F4, $F6 ; 42: sub.s $f4, $f4, $f6 # 5.4xy - 12.3y
FP12 = 0          [00400084] 3c011001 lui $1, 4097 ; 43: 1.s $f6, c
FP14 = 0          [00400088] -4260078 lwc $6, 120($1)
FP16 = 0          [0040008c] 46023182 mul.s $F6, $F6, $F2 ; 44: mul.s $f6, $f6, $f2 # 18.23x
FP18 = 0          [00400090] 46062100 add.s $F4, $F4, $F6 ; 45: add.s $f4, $f4, $f6 # 5.4xy - 12.3y + 18.23x
FP20 = 0          [00400094] 3c011001 lui $1, 4097 ; 46: 1.s $f6, d
FP22 = 0          [00400098] -426007c lwc $6, 124($1)
FP24 = 0          [0040009c] 46062101 sub.s $F4, $F4, $F6 ; 47: sub.s $f4, $f4, $f6 # 5.4xy - 12.3y + 18.23x - 8.23
FP26 = 0          [004000a0] 3c011001 lui $1, 4097 [printResult] ; 49: la $a0, printResult
FP28 = 0          [004000a4] 34020004 ori $2, $0, 4   ; 50: li $v0, 4
FP30 = 0          [004000b0] 46002306 mov.s $F12, $F4 ; 53: mov.s $f12, $f4
FP31 = 0          [004000b4] 34020002 ori $2, $0, 2   ; 54: li $v0, 2

Memory and registers cleared
SPIM Version 9.1.24 of August 1, 2023 (final)
Copyright 1990-2023 by James Larus.
All Rights Reserved.
See the file License for a full copyright notice.

```

FP Regs

```

FP21 = 0
FP22 = 0
FP23 = 0
FP24 = 0
FP25 = 0
FP26 = 0
FP27 = 0
FP28 = 0
FP29 = 0
FP30 = 0
FP31 = 0

Double Precision
FP0 = 40c00000
FP2 = 40a00000
FP12 = 432b1eb8
FP6 = 4103ae14
FP8 = 0
FP14 = 0
FP16 = 0
FP18 = 0
FP20 = 0
FP22 = 0
FP24 = 0
FP26 = 0
FP28 = 0
FP30 = 0

```

Text

```

[00400003c] 0000000c syscall ; 21: syscall
[00400040] 34020006 ori $2, $0, 6 ; 23: li $v0, 6
[00400044] 0000000c syscall ; 24: syscall
[00400048] 3c011001 lui $1, 4097 [readY] ; 26: mov.s $f2, $f0
[00400050] 34240038 ori $4, $1, 56 [readY] ; 28: la $a0, readY
[00400054] 34020004 ori $2, $0, 4 ; 29: li $v0, 4
[00400058] 0000000c syscall ; 30: syscall
[0040005c] 34020006 ori $2, $0, 6 ; 32: li $v0, 6
[00400060] 0000000c syscall ; 33: syscall
[00400064] 3c011001 lui $1, 4097 ; 37: 1.s $f4, a
[00400066] 46022102 lui $1, 112($1) ; 38: mul.s $f4, $f4, $f2 # 5.4x
[00400070] 46002102 mul.s $f4, $f4, $f0 ; 39: mul.s $f4, $f4, $f0 # 5.4x
[00400074] 3c011001 lui $1, 4097 ; 40: 1.s $f6, bb
[00400078] c4260074 lwl $f6, 116($1) ; 41: mul.s $f6, $f6, $f0 # 12.3y
[0040007c] 46002102 sub.s $f4, $f4, $f6 ; 42: sub.s $f4, $f4, $f6 # 5.4xy - 12.3y
[00400084] 3c011001 lui $1, 4097 ; 43: 1.s $f6, c
[00400088] 46022102 lui $1, 120($1) ; 44: mul.s $f6, $f6, $f2 # 18.23x
[00400090] 46022100 add.s $f4, $f4, $f6 ; 45: add.s $f4, $f4, $f6 # 5.4xy - 12.3y + 18.23x
[00400094] 3c011001 lui $1, 4097 ; 46: 1.s $f6, d
[00400098] c426007c lwl $f6, 124($1) ; 47: sub.s $f4, $f4, $f6 # 5.4xy - 12.3y + 18.23x - 8.23
[0040009c] 46002101 sub.s $f4, $f4, $f6 ; 48: la $a0, printResult
[004000a0] 3c011001 lui $1, 4097 [printResult]; 49: la $a0, printResult
[004000a4] 3424004f ori $4, $1, 79 [printResult]
[004000a8] 34020004 ori $2, $0, 4 ; 50: li $v0, 4
[004000ac] 0000000c syscall ; 51: syscall
[004000b0] 46002306 mov.s $f12, $f4 ; 53: mov.s $f12, $f4
[004000b4] 34020002 ori $2, $0, 2 ; 54: li $v0, 2

```

Memory and registers cleared

SFRM Version 1.1-24 of August 1, 2023 (final)
Copyright 1990-2023 by James Larus.
All Rights Reserved.

FP Regs

```

FP21 = 0
FP22 = 0
FP23 = 0
FP24 = 0
FP25 = 0
FP26 = 0
FP27 = 0
FP28 = 0
FP29 = 0
FP30 = 0
FP31 = 0

Double Precision
FP0 = 40c00000
FP2 = 40a00000
FP12 = 432b1eb8
FP6 = 4103ae14
FP8 = 0
FP14 = 0
FP16 = 0
FP18 = 0
FP20 = 0
FP22 = 0
FP24 = 0
FP26 = 0
FP28 = 0
FP30 = 0

```

Text

```

[004000039] 0000000c syscall ; 20: syscall
[0040005c] 34020006 ori $2, $0, 6 ; 32: li $v0, 6
[00400060] 0000000c syscall ; 33: syscall
[00400064] 3c011001 lui $1, 4097 ; 37: 1.s $f4, a
[00400066] 46022102 lui $1, 112($1) ; 38: mul.s $f4, $f4, $f2 # 5.4x
[00400070] 46002102 mul.s $f4, $f4, $f0 ; 39: mul.s $f4, $f4, $f0 # 5.4x
[00400074] 3c011001 lui $1, 4097 ; 40: 1.s $f6, bb
[00400078] c4260074 lwl $f6, 116($1) ; 41: mul.s $f6, $f6, $f0 # 12.3y
[0040007c] 46002102 sub.s $f4, $f4, $f6 ; 42: sub.s $f4, $f4, $f6 # 5.4xy - 12.3y
[00400084] 3c011001 lui $1, 4097 ; 43: 1.s $f6, c
[00400088] 46022102 lui $1, 120($1) ; 44: mul.s $f6, $f6, $f2 # 18.23x
[00400090] 46022100 add.s $f4, $f4, $f6 ; 45: add.s $f4, $f4, $f6 # 5.4xy - 12.3y + 18.23x
[00400094] 3c011001 lui $1, 4097 ; 46: 1.s $f6, d
[00400098] c426007c lwl $f6, 124($1) ; 47: sub.s $f4, $f4, $f6 # 5.4xy - 12.3y + 18.23x - 8.23
[0040009c] 46002101 sub.s $f4, $f4, $f6 ; 48: la $a0, printResult
[004000a0] 3c011001 lui $1, 4097 [printResult]; 49: la $a0, printResult
[004000a4] 3424004f ori $4, $1, 79 [printResult]
[004000a8] 34020004 ori $2, $0, 4 ; 50: li $v0, 4
[004000ac] 0000000c syscall ; 51: syscall
[004000b0] 46002306 mov.s $f12, $f4 ; 53: mov.s $f12, $f4
[004000b4] 34022002 ori $2, $0, 2 ; 54: li $v0, 2
[004000b8] 0000000c syscall ; 55: syscall
[004000bc] 3402000a ori $2, $0, 10 ; 57: li $v0, 10
[004000c0] 0000000c syscall ; 58: syscall

```

Kernel Text Segment [80000000]..[80010000]

```

[80000180] 0001d821 addu $27, $0, $1 ; 90: move $k1 $at # Save $at
[80000184] 3c019000 lui $1, -8672 ; 92: sw $v0 $1 # Not re-entrant and we can't trust $sp
[80000188] ac220200 sw $2, 512($1) ; 92: sw $v0 $1 # Not re-entrant and we can't trust $sp

```

Memory and registers cleared

Exercise 2 – Harmonic Series

Code:

```
● ● ●

1  .data
2  strNameId: .asciiz "Phan Tran Thanh Huy\nITCSIU22056\n"
3  readN: .asciiz "Enter the nth term: "
4  printResult: .asciiz "The result of Harmoni Series is: "
5
6  .text
7  .globl main
8
9  main:
10     la $a0, strNameId
11     li $v0, 4
12     syscall
13
14     la $a0, readN
15     li $v0, 4
16     syscall
17
18     li $v0, 5
19     syscall
20
21     move $t0, $v0
22
23     li.d $f0, 0.0
24     li.d $f2, 1.0
25     LOOP:
26     mtc1 $t0, $f4
27     cvt.d.w $f4, $f4
28     div.d $f6, $f2, $f4
29     add.d $f0, $f0, $f6
30     addi $t0, $t0, -1
31     bne $t0, $0, LOOP
32     EXIT:
33     la $a0, printResult
34     li $v0, 4
35     syscall
36
37     mov.d $f12, $f0
38     li $v0, 3
39     syscall
40
41     li $v0, 10
42     syscall
```

Test case:

```
Console
Phan Tran Thanh Huy
ITCSIU22056
Enter the nth term: 2
The result of Harmoni Series is: 1.5
```

```
Console
Phan Tran Thanh Huy
ITCSIU22056
Enter the nth term: 22
The result of Harmoni Series is: 3.6908132502172748
```

```
Console
Phan Tran Thanh Huy
ITCSIU22056
Enter the nth term: 220
The result of Harmoni Series is: 5.9731142167670912
```

```
Console
Phan Tran Thanh Huy
ITCSIU22056
Enter the nth term: 2205
The result of Harmoni Series is: 8.2759251930124584
```

```
Console
Phan Tran Thanh Huy
ITCSIU22056
Enter the nth term: 22056
The result of Harmoni Series is: 10.578578287002221
```

Single Step:

FP Regs	Int Regs [16]	Data	Text
FP21 = 0 FP22 = 0 FP23 = 0 FP24 = 0 FP25 = 0 FP26 = 0 FP27 = 0 FP28 = 0 FP29 = 0 FP30 = 0 FP31 = 0			<pre>[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main [00400018] 00000000 nop ; 189: nop [0040001c] 3402000a ori \$2, \$0, 10 ; 191: li \$v0 10 [00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit) [00400024] 3c041001 lui \$4, 4097 [strNameId] ; 10: la \$a0, strNameId [00400028] 34020004 ori \$2, \$0, 4 ; 11: li \$v0, 4 [0040002c] 0000000c syscall ; 12: syscall [00400030] 3c011001 lui \$1, 4097 [readN] ; 14: la \$a0, readN [00400034] 34240021 ori \$4, \$1, 33 [readN] [00400038] 34020004 ori \$2, \$0, 4 ; 15: li \$v0, 4 [0040003c] 0000000c syscall ; 16: syscall [00400040] 34020005 ori \$2, \$0, 5 ; 18: li \$v0, 5 [00400044] 0000000c syscall ; 19: syscall [00400048] 00024021 addu \$8, \$0, \$2 ; 21: move \$t0, \$v0 [0040004c] 34010000 ori \$1, \$0, 0 ; 23: li.d \$f0, 0.0 [00400050] 44810000 mtc1 \$1, \$f0 [00400054] 34010000 ori \$1, \$0, 0 [00400058] 44810800 mtc1 \$1, \$f1 [0040005c] 34010000 ori \$1, \$0, 0 ; 24: li.d \$f2, 1.0 [00400060] 44811000 mtc1 \$1, \$f2 [00400064] 3c013ff0 lui \$1, 16368 [00400068] 44811800 mtc1 \$1, \$f3 [0040006c] 44882000 mtc1 \$8, \$f4 ; 26: mtc1 \$t0, \$f4 [00400070] 46202121 cvt.d.w \$f4, \$f4 ; 27: cvt.d.w \$f4, \$f4 [00400074] 46241183 div.d \$f6, \$f2, \$f4 ; 28: div.d \$f6, \$f2, \$f4 [00400078] 46260000 add.d \$f0, \$f0, \$f6 ; 29: add.d \$f0, \$f0, \$f6 [0040007c] 2108ffff addi \$8, \$8, -1 ; 30: addi \$t0, \$t0, -1 [00400080] 1500ffff bne \$8, \$0, -20 [LOOP-0x00400080] [00400084] 3c011001 lui \$1, 4097 [printResult]; 33: la \$a0, printResult [00400088] 34240036 ori \$4, \$1, 54 [printResult] [0040008c] 34020004 ori \$2, \$0, 4 ; 34: li \$v0, 4 [00400090] 0000000c syscall ; 35: syscall </pre>

Calculating step:

FP Regs	Int Regs [16]	Data	Text
FP21 = 0 FP22 = 0 FP23 = 0 FP24 = 0 FP25 = 0 FP26 = 0 FP27 = 0 FP28 = 0 FP29 = 0 FP30 = 0 FP31 = 0			<pre>[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main [00400018] 00000000 nop ; 189: nop [0040001c] 3402000a ori \$2, \$0, 10 ; 191: li \$v0 10 [00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit) [00400024] 3c041001 lui \$4, 4097 [strNameId] ; 10: la \$a0, strNameId [00400028] 34020004 ori \$2, \$0, 4 ; 11: li \$v0, 4 [0040002c] 0000000c syscall ; 12: syscall [00400030] 3c011001 lui \$1, 4097 [readN] ; 14: la \$a0, readN [00400034] 34240021 ori \$4, \$1, 33 [readN] [00400038] 34020004 ori \$2, \$0, 4 ; 15: li \$v0, 4 [0040003c] 0000000c syscall ; 16: syscall [00400040] 34020005 ori \$2, \$0, 5 ; 18: li \$v0, 5 [00400044] 0000000c syscall ; 19: syscall [00400048] 00024021 addu \$8, \$0, \$2 ; 21: move \$t0, \$v0 [0040004c] 34010000 ori \$1, \$0, 0 ; 23: li.d \$f0, 0.0 [00400050] 44810000 mtc1 \$1, \$f0 [00400054] 34010000 ori \$1, \$0, 0 [00400058] 44810800 mtc1 \$1, \$f1 [0040005c] 34010000 ori \$1, \$0, 0 ; 24: li.d \$f2, 1.0 [00400060] 44811000 mtc1 \$1, \$f2 [00400064] 3c013ff0 lui \$1, 16368 [00400068] 44811800 mtc1 \$1, \$f3 [0040006c] 44882000 mtc1 \$8, \$f4 ; 26: mtc1 \$t0, \$f4 [00400070] 46202121 cvt.d.w \$f4, \$f4 ; 27: cvt.d.w \$f4, \$f4 [00400074] 46241183 div.d \$f6, \$f2, \$f4 ; 28: div.d \$f6, \$f2, \$f4 [00400078] 46260000 add.d \$f0, \$f0, \$f6 ; 29: add.d \$f0, \$f0, \$f6 [0040007c] 2108ffff addi \$8, \$8, -1 ; 30: addi \$t0, \$t0, -1 [00400080] 1500ffff bne \$8, \$0, -20 [LOOP-0x00400080] [00400084] 3c011001 lui \$1, 4097 [printResult]; 33: la \$a0, printResult [00400088] 34240036 ori \$4, \$1, 54 [printResult] [0040008c] 34020004 ori \$2, \$0, 4 ; 34: li \$v0, 4 [00400090] 0000000c syscall ; 35: syscall </pre>

FP Regs	Int Regs [16]	Data	Text
FP21 = 0 FP22 = 0 FP23 = 0 FP24 = 0 FP25 = 0 FP26 = 0 FP27 = 0 FP28 = 0 FP29 = 0 FP30 = 0 FP31 = 0			<pre> [00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main [00400018] 00000000 nop ; 189: nop [0040001c] 3402000a ori \$2, \$0, 10 ; 191: li \$v0, 10 [00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit) [00400024] 3c041001 lui \$4, 4097 [strNameId] ; 10: la \$a0, strNameId [00400028] 34020004 ori \$2, \$0, 4 ; 11: li \$v0, 4 [0040002c] 0000000c syscall ; 12: syscall [00400030] 3c011001 lui \$1, 4097 [readN] ; 14: la \$a0, readN [00400034] 34240021 ori \$4, \$1, 33 [readN] [00400038] 34020004 ori \$2, \$0, 4 ; 15: li \$v0, 4 [0040003c] 0000000c syscall ; 16: syscall [00400040] 34020005 ori \$2, \$0, 5 ; 18: li \$v0, 5 [00400044] 0000000c syscall ; 19: syscall [00400048] 00024021 addu \$8, \$0, \$2 ; 21: move \$t0, \$v0 [0040004c] 34010000 ori \$1, \$0, 0 ; 23: li.d \$f0, 0.0 [00400050] 44810000 mtc1 \$1, \$f0 [00400054] 34010000 ori \$1, \$0, 0 [00400058] 44810800 mtc1 \$1, \$f1 [0040005c] 34010000 ori \$1, \$0, 0 ; 24: li.d \$f2, 1.0 [00400060] 44811000 mtc1 \$1, \$f2 [00400064] 3c013fff lui \$1, 16368 [00400068] 44811800 mtc1 \$1, \$f3 [0040006c] 44882000 mtc1 \$8, \$f4 ; 26: mtc1 \$t0, \$f4 [00400070] 46202121 cvt.d.w \$f4, \$f4 ; 27: cvt.d.w \$f4, \$f4 [00400074] 46241183 div.d \$f6, \$f2, \$f4 ; 28: div.d \$f6, \$f2, \$f4 [00400078] 46260000 add.d \$f0, \$f0, \$f6 ; 29: add.d \$f0, \$f0, \$f6 [0040007c] 2108ffff addi \$8, \$8, -1 ; 30: addi \$t0, \$t0, -1 [00400080] 1500ffffb bne \$8, \$0, -20 [LOOP-0x00400080] [00400084] 3c011001 lui \$1, 4097 [printResult]; 33: la \$a0, printResult [00400088] 34240036 ori \$4, \$1, 54 [printResult] [0040008c] 34020004 ori \$2, \$0, 4 ; 34: li \$v0, 4 [00400090] 0000000c syscall ; 35: syscall </pre>

FP Regs	Int Regs [16]	Data	Text
FP21 = 0 FP22 = 0 FP23 = 0 FP24 = 0 FP25 = 0 FP26 = 0 FP27 = 0 FP28 = 0 FP29 = 0 FP30 = 0 FP31 = 0			<pre> [00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main [00400018] 00000000 nop ; 189: nop [0040001c] 3402000a ori \$2, \$0, 10 ; 191: li \$v0, 10 [00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit) [00400024] 3c041001 lui \$4, 4097 [strNameId] ; 10: la \$a0, strNameId [00400028] 34020004 ori \$2, \$0, 4 ; 11: li \$v0, 4 [0040002c] 0000000c syscall ; 12: syscall [00400030] 3c011001 lui \$1, 4097 [readN] ; 14: la \$a0, readN [00400034] 34240021 ori \$4, \$1, 33 [readN] [00400038] 34020004 ori \$2, \$0, 4 ; 15: li \$v0, 4 [0040003c] 0000000c syscall ; 16: syscall [00400040] 34020005 ori \$2, \$0, 5 ; 18: li \$v0, 5 [00400044] 0000000c syscall ; 19: syscall [00400048] 00024021 addu \$8, \$0, \$2 ; 21: move \$t0, \$v0 [0040004c] 34010000 ori \$1, \$0, 0 ; 23: li.d \$f0, 0.0 [00400050] 44810000 mtc1 \$1, \$f0 [00400054] 34010000 ori \$1, \$0, 0 [00400058] 44810800 mtc1 \$1, \$f1 [0040005c] 34010000 ori \$1, \$0, 0 ; 24: li.d \$f2, 1.0 [00400060] 44811000 mtc1 \$1, \$f2 [00400064] 3c013fff lui \$1, 16368 [00400068] 44811800 mtc1 \$1, \$f3 [0040006c] 44882000 mtc1 \$8, \$f4 ; 26: mtc1 \$t0, \$f4 [00400070] 46202121 cvt.d.w \$f4, \$f4 ; 27: cvt.d.w \$f4, \$f4 [00400074] 46241183 div.d \$f6, \$f2, \$f4 ; 28: div.d \$f6, \$f2, \$f4 [00400078] 46260000 add.d \$f0, \$f0, \$f6 ; 29: add.d \$f0, \$f0, \$f6 [0040007c] 2108ffff addi \$8, \$8, -1 ; 30: addi \$t0, \$t0, -1 [00400080] 1500ffffb bne \$8, \$0, -20 [LOOP-0x00400080] [00400084] 3c011001 lui \$1, 4097 [printResult]; 33: la \$a0, printResult [00400088] 34240036 ori \$4, \$1, 54 [printResult] [0040008c] 34020004 ori \$2, \$0, 4 ; 34: li \$v0, 4 [00400090] 0000000c syscall ; 35: syscall </pre>

FP Regs	Int Regs [16]	Data	Text
FP21 = 0			[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
FP22 = 0			[00400018] 00000000 nop ; 189: nop
FP23 = 0			[0040001c] 3402000a ori \$2, \$0, 10 ; 191: li \$v0 10
FP24 = 0			[00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit)
FP25 = 0			[00400024] 3c041001 lui \$4, 4097 [strNameId] ; 10: la \$a0, strNameId
FP26 = 0			[00400028] 34020004 ori \$2, \$0, 4 ; 11: li \$v0, 4
FP27 = 0			[0040002c] 0000000c syscall ; 12: syscall
FP28 = 0			[00400030] 3c011001 lui \$1, 4097 [readN] ; 14: la \$a0, readN
FP29 = 0			[00400034] 34240021 ori \$4, \$1, 33 [readN]
FP30 = 0			[00400038] 34020004 ori \$2, \$0, 4 ; 15: li \$v0, 4
FP31 = 0			[0040003c] 0000000c syscall ; 16: syscall
Double Precision			[00400040] 34020005 ori \$2, \$0, 5 ; 18: li \$v0, 5
FP0 = 3fa745d1745d1746			[00400044] 0000000c syscall ; 19: syscall
FP2 = 3ff00000000000000			[00400048] 00024021 addu \$8, \$0, \$2 ; 21: move \$t0, \$v0
FP4 = 4036000000000000			[0040004c] 34010000 ori \$1, \$0, 0 ; 23: li.d \$f0, 0.0
FP6 = 3fa745d1745d1746			[00400050] 44810000 mtc1 \$1, \$f0
FP8 = 0			[00400054] 34010000 ori \$1, \$0, 0
FP10 = 0			[00400058] 44810800 mtc1 \$1, \$f1
FP12 = 0			[0040005c] 34010000 ori \$1, \$0, 0 ; 24: li.d \$f2, 1.0
FP14 = 0			[00400060] 44811000 mtc1 \$1, \$f2
FP16 = 0			[00400064] 3c013ff0 lui \$1, 16368
FP18 = 0			[00400068] 44811800 mtc1 \$1, \$f3
FP20 = 0			[0040006c] 44882000 mtc1 \$8, \$f4 ; 26: mtc1 \$t0, \$f4
FP22 = 0			[00400070] 46202121 cvt.d.w \$f4, \$f4 ; 27: cvt.d.w \$f4, \$f4
FP24 = 0			[00400074] 46241183 div.d \$f6, \$f2, \$f4 ; 28: div.d \$f6, \$f2, \$f4
FP26 = 0			[00400078] 46260000 add.d \$f0, \$f0, \$f6 ; 29: add.d \$f0, \$f0, \$f6
FP28 = 0			[0040007c] 2108ffff addi \$8, \$8, -1 ; 30: addi \$t0, \$t0, -1
FP30 = 0			[00400080] 1500ffffb bne \$8, \$0, -20 [LOOP-0x00400080]
			[00400084] 3c011001 lui \$1, 4097 [printResult]; 33: la \$a0, printResult
			[00400088] 34240036 ori \$4, \$1, 54 [printResult]
			[0040008c] 34020004 ori \$2, \$0, 4 ; 34: li \$v0, 4

Int Regs [16]	Text
PC = 400080	[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
EPC = 0	[00400018] 00000000 nop ; 189: nop
Cause = 0	[0040001c] 3402000a ori \$2, \$0, 10 ; 191: li \$v0 10
BadVAddr = 0	[00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit)
Status = 3000ff10	[00400024] 3c041001 lui \$4, 4097 [strNameId] ; 10: la \$a0, strNameId
HI = 0	[00400028] 34020004 ori \$2, \$0, 4 ; 11: li \$v0, 4
LO = 0	[0040002c] 0000000c syscall ; 12: syscall
R0 [r0] = 0	[00400030] 3c011001 lui \$1, 4097 [readN] ; 14: la \$a0, readN
R1 [at] = 3ff00000	[00400034] 34240021 ori \$4, \$1, 33 [readN]
R2 [v0] = 16	[00400038] 34020004 ori \$2, \$0, 4 ; 15: li \$v0, 4
R3 [v1] = 0	[0040003c] 0000000c syscall ; 16: syscall
R4 [a0] = 10010021	[00400040] 34020005 ori \$2, \$0, 5 ; 18: li \$v0, 5
R5 [a1] = 7ffff1a8	[00400044] 0000000c syscall ; 19: syscall
R6 [a2] = 7ffff1b0	[00400048] 00024021 addu \$8, \$0, \$2 ; 21: move \$t0, \$v0
R7 [a3] = 0	[0040004c] 34010000 ori \$1, \$0, 0 ; 23: li.d \$f0, 0.0
R8 [t0] = 15	[00400050] 44810000 mtc1 \$1, \$f0
R9 [t1] = 0	[00400054] 44810800 mtc1 \$1, \$f1 ; 24: li.d \$f2, 1.0
R10 [t2] = 0	[00400058] 34010000 ori \$1, \$0, 0
R11 [t3] = 0	[00400060] 44811000 mtc1 \$1, \$f2
R12 [t4] = 0	[00400064] 3c013ff0 lui \$1, 16368
R13 [t5] = 0	[00400068] 44811800 mtc1 \$1, \$f3
R14 [t6] = 0	[0040006c] 44882000 mtc1 \$8, \$f4 ; 26: mtc1 \$t0, \$f4
R15 [t7] = 0	[00400070] 46202121 cvt.d.w \$f4, \$f4 ; 27: cvt.d.w \$f4, \$f4
R16 [s0] = 0	[00400074] 46241183 div.d \$f6, \$f2, \$f4 ; 28: div.d \$f6, \$f2, \$f4
R17 [s1] = 0	[00400078] 46260000 add.d \$f0, \$f0, \$f6 ; 29: add.d \$f0, \$f0, \$f6
R18 [s2] = 0	[0040007c] 2108ffff addi \$8, \$8, -1 ; 30: addi \$t0, \$t0, -1
R19 [s3] = 0	[00400080] 1500ffffb bne \$8, \$0, -20 [LOOP-0x00400080]
R20 [s4] = 0	[00400084] 3c011001 lui \$1, 4097 [printResult]; 33: la \$a0, printResult
R21 [s5] = 0	[00400088] 34240036 ori \$4, \$1, 54 [printResult]
...	[0040008c] 34020004 ori \$2, \$0, 4 ; 34: li \$v0, 4

Memory and registers altered

FP Regs	Text
FG21 = 0	[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
FG22 = 0	[00400018] 00000000 nop ; 189: nop
FG23 = 0	[0040001c] 3402000a ori \$2, \$0, 10 ; 191: li \$v0 10
FG24 = 0	[00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit)
FG25 = 0	[00400024] 3c041001 lui \$4, 4097 [strNameId] ; 10: la \$a0, strNameId
FG26 = 0	[00400028] 34020004 ori \$2, \$0, 4 ; 11: li \$v0, 4
FG27 = 0	[00400030] 3c011001 syscall ; 12: syscall
FG28 = 0	[00400034] 34240021 ori \$4, \$1, 33 [readN] ; 14: la \$a0, readN
FG29 = 0	[00400038] 34020004 ori \$2, \$0, 4 ; 15: li \$v0, 4
FG30 = 0	[0040003c] 0000000c syscall ; 16: syscall
FG31 = 0	[00400040] 34020005 ori \$2, \$0, 5 ; 18: li \$v0, 5
Double Precision	[00400044] 0000000c syscall ; 19: syscall
FP0 = 3fa745d1745d1746	[00400048] 00024021 addu \$8, \$0, \$2 ; 21: move \$t0, \$v0
FP2 = 3ff00000000000000	[0040004c] 34010000 ori \$1, \$0, 0 ; 23: li.d \$f0, 0.0
FP4 = 4036000000000015	[00400050] 44810000 mtcl \$1, \$f0
FP6 = 3fa745d1745d1746	[00400054] 34010000 ori \$1, \$0, 0
FP8 = 0	[00400058] 44810800 mtcl \$1, \$f1
FP10 = 0	[0040005c] 34010000 ori \$1, \$0, 0
FP12 = 0	[00400060] 44811000 mtcl \$1, \$f2
FP14 = 0	[00400064] 3c013fff0 lui \$1, 16368
FP16 = 0	[00400068] 44811800 mtcl \$1, \$f3
FP18 = 0	[0040006c] 44882000 mtcl \$8, \$f4 ; 26: mtcl \$t0, \$f4
FP20 = 0	[00400070] 46202121 cvt.d.w \$f4, \$f4 ; 27: cvt.d.w \$f4, \$f4
FP22 = 0	[00400074] 46241183 div.d \$f6, \$f2, \$f4 ; 28: div.d \$f6, \$f2, \$f4
FP24 = 0	[00400078] 46260000 add.d \$f0, \$f0, \$f6 ; 29: add.d \$f0, \$f0, \$f6
FP26 = 0	[0040007c] 2108ffff addi \$8, \$8, -1 ; 30: addi \$t0, \$t0, -1
FP28 = 0	[00400080] 1500ffffb bne \$8, \$0, -20 [LOOP-0x00400080]
FP30 = 0	[00400084] 3c011001 lui \$1, 4097 [printResult]; 33: la \$a0, printResult
	[00400088] 34240036 ori \$4, \$1, 54 [printResult]
	[0040008c] 34020004 ori \$2, \$0, 4 ; 34: li \$v0, 4
	syscall ; 35: syscall
	syscall ; 36: syscall

It will repeat the same step until the t0 = 0

Print the result:

FP Regs	Int Regs [16]	Data	Text
FG21 = 0			[00400044] 0000000c syscall ; 19: syscall
FG22 = 0			[00400048] 00024021 addu \$8, \$0, \$2 ; 21: move \$t0, \$v0
FG23 = 0			[0040004c] 34010000 ori \$1, \$0, 0 ; 23: li.d \$f0, 0.0
FG24 = 0			[00400050] 44810000 mtcl \$1, \$f0
FG25 = 0			[00400054] 34010000 ori \$1, \$0, 0
FG26 = 0			[00400058] 44810800 mtcl \$1, \$f1
FG27 = 0			[0040005c] 34010000 ori \$1, \$0, 0 ; 24: li.d \$f2, 1.0
FG28 = 0			[00400060] 44811000 mtcl \$1, \$f2
FG29 = 0			[00400064] 3c013fff0 lui \$1, 16368
FG30 = 0			[00400068] 44811800 mtcl \$1, \$f3
FG31 = 0			[0040006c] 44882000 mtcl \$8, \$f4 ; 26: mtcl \$t0, \$f4
Double Precision			[00400070] 46202121 cvt.d.w \$f4, \$f4 ; 27: cvt.d.w \$f4, \$f4
FP0 = 400d86c918ea9d00			[00400074] 46241183 div.d \$f6, \$f2, \$f4 ; 28: div.d \$f6, \$f2, \$f4
FP2 = 3ff00000000000000			[00400078] 46260000 add.d \$f0, \$f0, \$f6 ; 29: add.d \$f0, \$f0, \$f6
FP4 = 3ff00000000000000			[00400080] 1500ffffb bne \$8, \$0, -20 [LOOP-0x00400080]
FP6 = 3ff00000000000000			[00400084] 3c011001 lui \$1, 4097 [printResult]; 33: la \$a0, printResult
FP8 = 0			[00400088] 34240036 ori \$4, \$1, 54 [printResult]
FP10 = 0			[0040008c] 34020004 ori \$2, \$0, 4 ; 34: li \$v0, 4
FP12 = 400d86c918ea9d00			[00400090] 0000000c syscall ; 35: syscall
			[00400094] 46200306 mov.d \$f12, \$f0 ; 37: mov.d \$f12, \$f0
			[00400098] 34020003 ori \$2, \$0, 3 ; 38: li \$v0, 3
			[0040009c] 0000000c syscall ; 39: syscall
			[004000a0] 34020004 ori \$2, \$0, 10 ; 41: li \$v0, 10
			[004000a4] 0000000c syscall ; 42: syscall
			[80000180] 0001d821 addu \$27, \$0, \$1 ; 90: move \$k1 Sat # Save Sat
			[80000184] 3c019000 lui \$1, -28672 ; 92: sw \$v0 \$1 # Not re-entrant and we can't trust \$sp
			[80000188] ac220200 sw \$2, 512(\$1)
			[8000018c] 3c019000 lui \$1, -28672 ; 93: sw \$a0 \$2 # But we need to use these registers
			[80000190] sc200204 sw \$4, 516(\$1)

Exercise 3 – Web Page RGB Colors

Code:



```
1 .data
2 strNameId: .asciiz "Phan Tran Thanh Huy\nITCSIU22056\n"
3 printResult: .asciiz "The result is: "
4 color: .word 0x00FF0000
5 strSpace: .asciiz " "
6
7 .text
8 .globl main
9
10 main:
11     la $a0, strNameId
12     li $v0, 4
13     syscall
14
15     la $a0, printResult
16     li $v0, 4
17     syscall
18
19     lw $s0, color
20     li $s1, 2      # for checking
21     li $s2, 16
22     li $t0, 5      # amount of elements
23     li $t1, 0      # for sum
24     move $t2, $t0
25     li $t3, 1
26
27 LOOP1:           # Use to calculate the 16^nth
28     beq $t2, $0, LOOP2
29     mul $t3, $t3, $s2
30     addi $t2, $t2, -1
31     j LOOP1
32
33 LOOP2:           # Find the value of the hex in the nth position
34     div $s0, $t3
35     mflo $t4
36     mul $t3, $t3, $t4
37     sub $s0, $s0, $t3
38     div $t0, $s1
39     mfhi $t5
40     beq $t5, $0, LABEL1    # If it is not the second RGB, we will sum = sum + 16*value
41     mul $t4, $t4, $s2
42     add $t1, $t1, $t4
43     addi $t0, $t0, -1      # Increase counter
44     move $t2, $t0
45     li $t3, 1
46     bne $t0, $0, EXIT
47     j LOOP1
48
49 LABEL1:
50     add $t1, $t1, $t4      # If it is the second RGB, we will sum = sum + 1*value
51     mtc1 $t1, $f0          # Convert int to float
52     cvt.s.w $f0, $f0
53     li.s $f2, 255.0
54     div.s $f4, $f0, $f2    # sum(float)/255
55
56     mov.s $f12, $f4        # Print the result
57     li $v0, 2
58     syscall
59
60     la $a0, strSpace
61     li $v0, 4
62     syscall
63
64     addi $t0, $t0, -1      # Increase counter
65     li $t1, 0              # Reset the sum
66     move $t2, $t0
67     li $t3, 1
68     bne $t0, $0, EXIT
69     j LOOP1
70
71 EXIT:
72     li $v0, 10
73     syscall
```

Test case:

Data	Text
Text	
[00400088] 11a00009 beq \$13, \$0, 36 [LABEL1-0x00400088] [0040008c] 71926002 mul \$12, \$12, \$18 ; 41: mul \$t4 [00400090] 012c4820 add \$9, \$9, \$12 ; 42: add \$t5 [00400094] 2108ff 4 Console [00400098] 00085c Phan Tran Thanh Huy [0040009c] 340b00 ITCSIU22056 [004000a0] 010000 The result is: 1.00000000 0.00000000 0.00000000 [004000a4] 142000 [004000a8] 081000 [004000ac] 012c48 [004000b0] 448900 [004000b4] 468000 [004000b8] 3c0143 [004000bc] 448110 [004000c0] 460203 [004000c4] 460023 [004000c8] 340200 [004000cc] 000000 [004000d0] 3c0110 [004000d4] 342400 [004000d8] 340200 [004000dc] 000000 [004000e0] 2108ff [004000e4] 340900 [004000e8] 000850 [004000ec] 340b00 [004000f0] 010000 [004000f4] 142000 [004000f8] 081000 [004000fc] 340200 [00400100] 000000	File Ctrl View .data strNameId: .asciiz "Phan Tran Thanh Huy\nITCSIU22056\n" printResult: .asciiz "The result is: " color: .word 0x00FF0000 strSpace: .asciiz " " .text .globl main main: la \$a0, strNameId li \$v0, 4 syscall la \$a0, printResult li \$v0, 4 syscall lw \$s0, color li \$s1, 2 # for checking li \$s2, 16 li \$t0, 5 # amount of elements li \$t1, 0 # for sum move \$t2, \$t0 li \$t3, 1 LOOP1: # Use to calculate the 16 ^{nth} beq \$t2, \$0, LOOP2 mul \$t3, \$t3, \$s2 addi \$t2, \$t2, -1 j LOOP1 LOOP2: # Find the value of the hex in the nth position div \$s0, \$t3 mflo \$t4 mul \$t3, \$t3, \$t4 sub \$s0, \$s0, \$t3

Single Step:

```

P Regs  Text
FG21 = 0 [0040003c] 0000000c syscall ; 17: syscall
FG22 = 0 [00400040] 3c011001 lui $1, 4097 ; 19: lw $s0, color
FG23 = 0 [00400044] 8c300034 lw $16, $2($1)
FG24 = 0 [00400048] 34110002 ori $17, $0, 2 ; 20: li $s1, 2 # for checking
FG25 = 0 [0040004c] 34120010 ori $18, $0, 16 ; 21: li $s2, 16
FG26 = 0 [00400050] 34080005 ori $8, $0, 5 ; 22: li $t0, 5 # amount of elements
FG27 = 0 [00400054] 34090000 ori $9, $0, 0 ; 23: li $t1, 0 # for sum
FG28 = 0 [00400058] 00085021 addu $10, $0, $8 ; 24: move $t2, $t0
FG29 = 0 [0040005c] 340b0001 ori $11, $0, 1 ; 25: li $t3, 1
FG30 = 0 [00400060] 11400004 beq $10, $0, 16 [LOOP2-0x00400060]
FG31 = 0 [00400064] 71725802 mul $11, $11, $18 ; 29: mul $t3, $t3, $s2
[00400068] 214fffff addi $10, $10, -1 ; 30: addi $t2, $t2, -1
[0040006c] 00810000 0x00400066 [LOOP1] ; 31: j LOOP1
[00400070] 020b001a div $16, $11 ; 34: div $s0, $t3
[00400074] 00006012 mflo $12 ; 35: mflo $t4
FP2 = 0 [00400078] 716c5802 mul $11, $11, $12 ; 36: mul $t3, $t3, $t4
FP4 = 0 [0040007c] 020b0022 sub $16, $16, $11 ; 37: sub $s0, $s0, $t3
FP6 = 0 [00400080] 0111001a div $8, $17 ; 38: div $t0, $s1
FP8 = 0 [00400084] 00096810 mfhi $13 ; 39: mfhi $t5
FP10 = 0 [00400088] 11a00009 beq $13, $0, 36 [LABEL1-0x00400088]
FP12 = 0 [0040008c] 71926002 mul $12, $12, $18 ; 41: mul $t4, $t4, $s2
FP14 = 0 [00400090] 012c4820 add $9, $9, $12 ; 42: add $t1, $t1, $t4
FP16 = 0 [00400094] 2108ffff addi $8, $8, -1 ; 43: addi $t0, $t0, -1 # Increase counter
FP18 = 0 [00400098] 00085021 addu $10, $0, $8 ; 44: move $t2, $t0
FP20 = 0 [0040009c] 340b0000 ori $11, $0, 1 ; 45: li $t3, 1
FP22 = 0 [004000a0] 01000082 slt $1, $8, $0 ; 46: bit $t0, $0, EXIT
FP24 = 0 [004000a4] 14200016 bne $1, $0, 88 [EXIT-0x004000a4]
FP26 = 0 [004000a8] 08100018 j 0x00400060 [LOOP1] ; 47: j LOOP1
FP28 = 0 [004000ac] 012c4820 add $9, $9, $12 ; 50: add $t1, $t1, $t4 # If it is the second RGB, we will sum = sum + 1*value
FP30 = 0 [004000b0] 44590000 mtcl $9, $f0 ; 51: mtcl $t1, $f0 # Convert int to float
[004000b4] 46800020 cvt.s.w $f0, $f0 ; 52: cvt.s.w $f0, $f0

```

FP Regs	Int Regs [16]		Data	Text
FP Regs	Int Regs [16]			
FP21 = 0			[00400064] 71725802	mul \$11, \$11, \$18 ; 29: mul \$t3, \$t3, \$s2
FP22 = 0			[00400068] 214affff	addi \$10, \$10, -1 ; 30: addi \$t2, \$t2, -1
FP23 = 0			[0040006c] 08100018	j 0x00400060 [LOOP1] ; 31: j LOOP1
FP24 = 0			[00400070] 020b0001	div \$16, \$11 ; 34: div \$s0, \$t3
FP25 = 0			[00400074] 00006012	mflo \$12 ; 35: mflo \$t4
FP26 = 0			[00400078] 716c5802	mul \$11, \$11, \$12 ; 36: mul \$t3, \$t3, \$t4
FP27 = 0			[0040007c] 020b0022	sub \$16, \$16, \$11 ; 37: sub \$s0, \$s0, \$s0, \$t3
FP28 = 0			[00400080] 0111001a	div \$8, \$17 ; 38: div \$t0, \$s1
FP29 = 0			[00400084] 00006810	mfhi \$13 ; 39: mfhi \$t5
FP30 = 0			[00400088] 11a00009	beq \$13, \$0, 36 [LABEL1-0x04000088]
FP31 = 0			[0040008c] 71926002	mul \$12, \$12, \$18 ; 41: mul \$t4, \$t4, \$s2
Double Precision			[00400090] 012c4820	add \$9, \$9, \$12 ; 42: add \$t1, \$t1, \$t4
FP0 = 437f0000			[00400094] 2108ffff	addi \$8, \$8, -1 ; 43: addi \$t0, \$t0, -1 # Increase counter
FP2 = 0			[00400098] 00085021	addu \$10, \$0, \$8 ; 44: move \$t2, \$t0
FP4 = 0			[0040009c] 340b0001	ori \$11, \$0, 1 ; 45: li \$t3, 1
FP6 = 0			[004000a0] 14200001	slt \$1, \$8, \$0 ; 46: blt \$t0, \$0, EXIT
FP8 = 0			[004000a4] 14200016	bne \$1, \$0, 88 [EXIT-0x04000044]
FP10 = 0			[004000a8] 08100018	j 0x00400060 [LOOP1] ; 47: j LOOP1
FP12 = 0			[004000ac] 012c4820	add \$9, \$9, \$12 ; 50: add \$t1, \$t1, \$t4 # If it is the second RGB, we will sum = sum + 1*value
FP14 = 0			[004000b0] 44890000	mtc1 \$9, \$f0 ; 51: mtc1 \$t1, \$f0 # Convert int to float
FP16 = 0			[004000b4] 46890020	cvt.s.w \$f0, \$f0 ; 52: cvt.s.w \$f0, \$f0
FP18 = 0			[004000b8] 3c01437e	lui \$1, 17279 ; 53: li \$t9, 255.0
FP20 = 0			[004000bc] 44811000	mtc1 \$1, \$f2
FP22 = 0			[004000c0] 46020103	div.s \$f4, \$f0, \$f2 ; 54: div.s \$t4, \$t0, \$f2 # sum(float)/255
FP24 = 0			[004000c4] 46020002	mov.s \$f12, \$f4 ; 56: mov.s \$t12, \$f4 # Print the result
FP26 = 0			[004000c8] 34020002	ori \$2, \$0, 2 ; 57: li \$v0, 2
FP28 = 0			[004000cc] 00000000	syscall ; 58: syscall
FP30 = 0			[004000d0] 3c010010	lui \$1, 4097 [strSpace] ; 60: la \$a0, strSpace
			[004000d4] 34240038	ori \$4, \$1, 56 [strSpace]
			[004000d8] 34020004	ori \$2, \$0, 4 ; 61: li \$v0, 4
			[004000dc] 00000000	syscall ; 62: syscall
			[004000e0] 01086666	add.s \$f0, \$f0, \$f0 ; 64: add.s \$t0, \$t0, \$t0 # Increase counter

FP Regs	Int Regs [16]		Data	Text
FP21 = 0			[00400064] 71725802	mul \$11, \$11, \$18 ; 29: mul \$t3, \$t3, \$s2
FP22 = 0			[00400068] 214affff	addi \$10, \$10, -1 ; 30: addi \$t2, \$t2, -1
FP23 = 0			[0040006c] 0810018	j 0x04000060 [LOOP1] ; 31: j LOOP1
FP24 = 0			[00400070] 0200b001a	div \$16, \$11 ; 34: div \$s0, \$t3
FP25 = 0			[00400074] 000006012	mflo \$12 ; 35: mflo \$t4
FP26 = 0			[00400078] 716c5802	mul \$11, \$11, \$12 ; 36: mul \$t3, \$t3, \$t4
FP27 = 0			[0040007c] 020b8022	sub \$16, \$16, \$11 ; 37: sub \$s0, \$s0, \$t3
FP28 = 0			[00400080] 0111001a	div \$8, \$17 ; 38: div \$t0, \$s1
FP29 = 0			[00400084] 00006810	mfhi \$13 ; 39: mfhi \$t5
FP30 = 0			[00400088] 11a0009	beq \$13, \$0, 36 [LABEL1-0x04000088]
FP31 = 0			[0040008c] 71926002	mul \$12, \$12, \$18 ; 41: mul \$t4, \$t4, \$s2
Double Precision	[00400090] 012c4820		add \$9, \$9, \$12	; 42: add \$t1, \$t1, \$t4
	[00400094] 2108ffff		add \$8, \$8, -1	; 43: addi \$t0, \$t0, -1 # Increase counter
FP0 = 437f0000	[00400098] 00085021		addi \$10, \$0, \$8	; 44: move \$t2, \$t0
FP2 = 437f0000	[0040009c] 3400b001		ori \$11, \$0, 1	; 45: li \$t3, 1
FP4 = 0	[004000a0] 14200016		slt \$1, \$8, \$0	; 46: bit \$t0, \$0, EXIT
FP6 = 0	[004000a4] 14200016		bne \$1, \$0, 88 [EXIT-0x0400004]	
FP8 = 0	[004000a8] 0810018		j 0x04000060 [LOOP1]	; 47: j LOOP1
FP10 = 0	[004000ac] 012c4820		add \$9, \$9, \$12	; 50: add \$t1, \$t1, \$t4 # If it is the second RGB, we will sum = sum + 1*value
FP12 = 0	[004000b0] 44890000		mtcl \$9, \$f0	; 51: mtcl \$t1, \$f0 # Convert int to float
FP14 = 0	[004000b4] 46800020		cvt.s.w \$f0, \$f0	; 52: cvt.s.w \$f0, \$f0
FP16 = 0	[004000b8] 3c01437e		lui \$1, 17279	; 53: li \$t2, 255.0
FP18 = 0	[004000bc] 44811000		mtcl \$1, \$f2	
FP20 = 0	[004000c0] 46020013		div.s \$f4, \$f0, \$f2	; 54: div.s \$f4, \$f0, \$f2 # sum(float)/255
FP22 = 0	[004000c4] 46002306		mov.s \$f12, \$f4	; 56: mov.s \$f12, \$f4 # Print the result
FP24 = 0	[004000c8] 34020002		ori \$2, \$0, 2	; 57: li \$v0, 2
FP26 = 0	[004000cc] 00000000		syscall	; 58: syscall
FP28 = 0	[004000d0] 3c010010		lui \$1, 4097 [strSpace]	; 60: la \$a0, strSpace
FP30 = 0	[004000d4] 34240038		ori \$4, \$1, 56 [strSpace]	
	[004000d8] 34020004		ori \$2, \$0, 4	; 61: li \$v0, 4
	[004000dc] 00000000		syscall	; 62: syscall
	[004000e0] 03166666		add.s \$f0, \$f0, -1	; 64: add.s \$f0, \$f0, -1 # Increase counter

FP Regs	Int Regs [16]	Data	Text
FP21 = 0			[0040004c] 34120010 ori \$18, \$0, 16 ; 21: li \$s2, 16
FG22 = 0			[00400050] 34080005 ori \$8, \$0, 5 ; 22: li \$t0, 5 # amount of elements
FG23 = 0			[00400054] 34090000 ori \$9, \$0, 0 ; 23: li \$t1, 0 # for sum
FG24 = 0			[00400058] 00085021 addu \$10, \$0, \$8 ; 24: move \$t2, \$t0
FG25 = 0			[0040005c] 340b0001 ori \$11, \$0, 1 ; 25: li \$t3, 1
FG26 = 0			[00400060] 11400000 beq \$10, \$0, 16 [LOOP2-0x00400060]
FG27 = 0			[00400064] 71725802 mul \$11, \$11, \$18 ; 29: mul \$t3, \$t3, \$s2
FG28 = 0			[00400068] 214a0fff addi \$10, \$10, -1 ; 30: addi \$t2, \$t2, -1
FG29 = 0			[0040006c] 08100018 j 0x00400060 [LOOP1] ; 31: j LOOP1
FG30 = 0			[00400070] 020b001a div \$16, \$11 ; 34: div \$s0, \$t3
FG31 = 0			[00400074] 00060012 mflo \$t2 ; 35: mflo \$t4
			[00400078] 716c5800 mul \$11, \$11, \$12 ; 36: mul \$t3, \$t3, \$t4
Double Precision			[0040007c] 020b0022 sub \$16, \$16, \$11 ; 37: sub \$s0, \$s0, \$t3
FP0 = 437f0000			[00400080] 0111001a div \$8, \$17 ; 38: div \$t0, \$s1
FP2 = 437f0000			[00400084] 00068100 mfhi \$t3 ; 39: mfhi \$t5
FP4 = 3f800000			[00400088] 11a00000 beq \$13, \$0, 36 [LABEL1-0x00400088]
			[0040008c] 71926002 mul \$12, \$12, \$18 ; 41: mul \$t4, \$t4, \$s2
FP6 = 0			[00400090] 012c4820 add \$9, \$9, \$12 ; 42: add \$t1, \$t1, \$t4
FP8 = 0			[00400094] 2108ffff addi \$8, \$8, -1 ; 43: addi \$t0, \$t0, -1 # Increase counter
FP10 = 0			[00400098] 00085021 addu \$10, \$0, \$8 ; 44: move \$t2, \$t0
FP12 = 0			[0040009c] 340b0001 ori \$11, \$0, 1 ; 45: li \$t3, 1
FP14 = 0			[004000a0] 01000082a slt \$1, \$8, \$0 ; 46: bit \$t0, \$0, EXIT
FP16 = 0			[004000a4] 14200016 bne \$1, \$0, 88 [EXIT-0x004000a4]
FP18 = 0			[004000a8] 08100000 00085021 j 0x00400060 [LOOP1] ; 47: j LOOP1
FP20 = 0			[004000ac] 012c4820 add \$9, \$9, \$12 ; 50: add \$t1, \$t1, \$t4 # If it is the second RGB, we will sum = sum + 1*value
FP22 = 0			[004000b0] 44900000 mtc1 \$9, \$t0 ; 51: mtc1 \$t1, \$f0 # Convert int to float
FP24 = 0			[004000b4] 46800020 cvt.s.w \$f0, \$f0 ; 52: cvt.s.w \$f0, \$f0
FP26 = 0			[004000b8] 3c01437f lui \$1, 17279 ; 53: li. \$t2, 255.0
FP28 = 0			[004000bc] 44811000 mtc1 \$1, \$t2 ; 54: mtc1 \$t2, \$t2
FP30 = 0			[004000c4] 46020103 div.s \$f4, \$f0, \$t2 ; 55: div.s \$t4, \$f0, \$t2 # sum(float)/255
			[004000c8] 46002300 mov.s \$t1, \$f4 ; 56: mov.s \$f12, \$f4 # Print the result

Print the result:

FP Regs	Int Regs [16]	Data	Text
FP21 = 0			Text
FP22 = 0		[00400050] 34090005 ori \$8, \$0, 5 ; 22: li \$t0, 5 # amount of elements	
FP23 = 0		[00400054] 34090009 ori \$9, \$0, 0 ; 23: li \$t1, 0 # for sum	
FP24 = 0		[00400058] 00085021 addu \$10, \$0, \$8 ; 24: move \$t2, \$t0	
FP25 = 0		[0040005c] 34000001 ori \$11, \$0, 1 ; 25: li \$t3, 1	
FP26 = 0		[00400060] 11400004 beq \$10, \$0, 16 [LOOP2-0x00400060]	
FP27 = 0		[00400064] 71725002 mul \$11, \$11, \$18 ; 29: mul \$t3, \$t3, \$s2	
FP28 = 0		[00400068] 214affff addi \$10, \$10, -1 ; 30: addi \$t2, \$t2, -1	
FP29 = 0		[0040006c] 08100018 j 0x00400060 [LOOP1] ; 31: j LOOP1	
FP30 = 0		[00400070] 020b0001a div \$16, \$11 ; 34: div \$s0, \$t3	
FP31 = 0		[00400074] 00006012 mflo \$12 ; 35: mflo \$t4	
Double Precision		[00400078] 716c5802 mul \$11, \$11, \$12 ; 36: mul \$t3, \$t3, \$t4	
FP0 = 437F0000		[0040007c] 020b0822 sub \$16, \$16, \$11 ; 37: sub \$s0, \$s0, \$t3	
FP1 = 437F0000		[00400080] 0111001a div \$9, \$17 ; 38: div \$t0, \$s1	
FP2 = 3F800000		[00400084] 00006810 mfhi \$13 ; 39: mfhi \$t5	
FP3 = 3F800000		[00400088] 11a00009 beq \$13, \$0, 36 [LABEL1-0x00400088]	
FP4 = 3F800000		[0040008c] 71926002 mul \$12, \$12, \$18 ; 41: mul \$t4, \$t4, \$s2	
FP5 = 0		[00400090] 012c4820 add \$9, \$9, \$12 ; 42: add \$t1, \$t1, \$t4	
FP6 = 0		[00400094] 2108ffff add \$8, \$8, -1 ; 43: addi \$t0, \$t0, -1 # Increase counter	
FP8 = 0		[00400098] 00085021 addu \$10, \$0, \$8 ; 44: move \$t2, \$t0	
FP10 = 0		[0040009c] 34000001 ori \$11, \$0, 1 ; 45: li \$t3, 1	
FP12 = 3F800000		[004000a0] 0100082a srl \$18, \$8, \$0 ; 46: bit \$t0, \$0, EXIT	
FP14 = 0		[004000a4] 14200016 bne \$1, \$0, 88 [EXIT-0x004000a4]	
FP16 = 0		[004000a8] 08100018 j 0x00400064 [LOOP1] ; 47: j LOOP1	
FP18 = 0		[004000ac] 012c4820 add \$9, \$9, \$12 ; 50: add \$t1, \$t1, \$t4 # If it is the second RGB, we will sum = sum + l*value	
FP20 = 0		[004000b0] 44890000 mtc1 \$9, \$f0 ; 51: mtc1 \$t1, \$f0 # Convert int to float	
FP22 = 0		[004000b4] 46800020 cvt.s.w \$f0, \$f0 ; 52: cvt.s.w \$f0, \$f0	
FP24 = 0		[004000b8] 3c01437f lui \$1, 17279 ; 53: l.s \$f2, 255.0	
FP26 = 0		[004000bc] 44811000 mtc1 \$1, \$f2	
FP28 = 0		[004000c0] 44811000 div.s \$f4, \$f0, \$f2 ; 54: div.s \$f4, \$f0, \$f2 # sum(float)/255	
FP30 = 0		[004000c4] 46020004 mov.s \$t12, \$f4 ; 56: mov.s \$t12, \$f4 # Print the result	
		[004000c8] 34020002 ori \$2, \$0, 2 ; 57: j1 \$v0, 2	

FP Regs	Int Regs [16]	Data	Text
FP Regs	Int Regs [16]	FP Regs	Text
FP21 = 0		[00400084] 00006810	mfhi \$13 ; 39: mfhi St5
FP22 = 0		[00400088] 11a00009	beq \$13, \$0, 36 [LABEL1-0x00400088]
FP23 = 0		[0040008c] 71926002	mul \$12, \$12, \$18 ; 41: mul St4, St4, St2
FP24 = 0		[00400090] 012c4820	add \$9, \$9, \$12 ; 42: add St1, St1, St4
FP25 = 0		[00400094] 2108ffff	addi \$8, \$8, -1 ; 43: addi St0, St0, -1 # Increase counter
FP26 = 0		[00400098] 00858021	addu \$10, \$0, \$8 ; 44: move St2, St0
FP27 = 0		[0040009c] 340b0001	ori \$11, \$0, 1 ; 45: li St3, 1
FP28 = 0		[004000a0] 0100082a	slt \$1, \$8, \$0 ; 46: bit St0, St0, EXIT
FP29 = 0		[004000a4] 14200016	bne \$1, \$0, 88 [EXIT-0x04000041]
FP30 = 0		[004000a8] 08100018	j 0x00400060 [LLOOP1] ; 47: j LLOOP1
FP31 = 0		[004000ac] 012c4820	add \$9, \$9, \$12 ; 50: add St1, St1, St4 # If it is the second
Double Precision		[004000b0] 44890000	mtc1 \$9, \$F0 ; 51: mtc1 St1, St0 # Convert int to float
FP0 = 437f20000		[004000b4] 46800020	cvt.s.w \$0, \$F0 ; 52: cvt.s.w St0, St0
FP2 = 437fe0000		[004000b8] 3c01437f	lui \$1, 17279 ; 53: li St2, 255.0
FP4 = 3f8000000		[004000bc] 44511000	mtc1 \$1, \$F2
FP6 = 0		[004000c0] 46020103	div.s \$F4, \$F0, \$F2 ; 54: div.s St4, St0, St0 # sum(float)/255
FP8 = 0		[004000c4] 46020306	mov.s \$F12, \$F4 ; 55: mov.s St12, St4 # Print the result
FP10 = 0		[004000c8] 340b0001	ori \$2, \$0, 2 ; 57: li Sv0, 2
FP12 = 3f8000000		[004000cc] 0000000c	syscall ; 58: syscall
FP14 = 0		[004000d0] 3c011001	ori \$1, \$0, 4097 [strSpace] ; 60: la \$a0, strSpace
FP16 = 0		[004000d4] 34240038	ori \$4, \$1, 56 [strSpace]
FP18 = 0		[004000d8] 34020004	ori \$2, \$0, 4 ; 61: li Sv0, 4
FP20 = 0		[004000dc] 0000000c	syscall ; 62: syscall
FP22 = 0		[004000e0] 2108ffff	addi \$8, \$8, -1 ; 64: addi St0, St0, -1 # Increase counter
FP24 = 0		[004000e4] 34090000	ori \$9, \$0, 0 ; 65: li St1, 0 # Reset the sum
FP26 = 0		[004000e8] 00858021	addu \$10, \$0, \$8 ; 66: move St2, St0
FP28 = 0		[004000ec] 340b0001	slt \$1, \$8, \$0 ; 67: li St3, 1
FP30 = 0		[004000f0] 0100082a	bne \$1, \$0, 8 [EXIT-0x04000041] ; 69: j LLOOP1
		[004000f4] 14200002	lui \$1, \$0, 8 ; 72: li Sv0, 10
		[004000f8] 08100018	3.0x00400060 [LLOOP1] ; 73:
		[004000fc] 3402000a	ori \$2, \$0, 10
		[00400100] 00000001	syscall ; 74:

FP Regs	Int Regs [16]	Data	Text
nt Regs [16]			Text
PC = 4000e4			
EPC = 0			
Cause = 0			
BadAddr = 0			
Status = 3000ff10			
HI = 0			
LO = 2			
R0 [r0] = 0			
R1 [at] = 10010000			
R2 [v0] = 4			
R3 [v1] = 0			
R4 [a0] = 10010038			
R5 [a1] = 7fffffa8			
R6 [a2] = 7fffffb0			
R7 [a3] = 0			
RR [t0] = 2			
R9 [t1] = ff			
R10 [t2] = 0			
R11 [t3] = 000000			
R12 [t4] = f			
R13 [t5] = 0			
R14 [t6] = 0			
R15 [t7] = 0			
R16 [s0] = 0			
R17 [s1] = 2			
R18 [s2] = 10			
R19 [s3] = 0			
R20 [s4] = 0			
R21 [s5] = 0			
			mfhi \$13 ; 39: mfhi \$t5
			[00400084] 00006810 beq \$13, \$0, 36 [LABEL1-0x0400088]
			[0040008c] 71926002 mul \$12, \$12, \$18 ; 41: mul \$t4, \$t4, \$s2
			[00400090] 012c4820 add \$9, \$9, \$12 ; 42: add \$t1, \$t1, \$t4
			[00400094] 2108ffff addi \$8, \$8, -1 ; 43: addi \$t0, \$t0, -1 # Increase counter
			[00400098] 00085021 addi \$10, \$0, \$8 ; 44: move \$t2, \$t0
			[0040009c] 3400b001 ori \$11, \$0, 1 ; 45: li \$t3, 1
			[004000a0] 0100082a slt \$1, \$8, \$0 ; 46: bit \$t0, \$0, EXIT
			[004000a4] 14200016 bne \$1, \$0, 88 [EXIT-0x004000a4]
			j 0x00400060 [LOOP1] ; 47: j LOOP1
			[004000a8] 08100018 add \$9, \$9, \$12 ; 50: add \$t1, \$t1, \$t4 # If it is the second RGB, we will sum = sum + 1*value
			[004000ac] 012c4820 mtc1 \$9, \$f0 ; 51: mtc1 \$t1, \$f0 # Convert int to float
			[004000b0] 44800000 cvt.s.w \$f0, \$f0 ; 52: cvt.s.w \$f0, \$f0
			[004000b8] 3c01437f lui \$1, 17279 ; 53: li \$s2, 255.0
			[004000bc] 44811000 mtc1 \$1, \$f2
			[004000c0] 46020103 div.s \$f4, \$f0, \$f2 ; 54: div.s \$t4, \$f0, \$f2 # sum(float)/255
			[004000c4] 46002306 mov.s \$f12, \$f4 ; 56: mov.s \$f12, \$f4 # Print the result
			[004000c8] 34002002 ori \$2, \$0, 2 ; 57: li \$v0, 2
			[004000cc] 00000000 syscall ; 58: syscall
			[004000d0] 3c011001 lui \$1, 4097 [strSpace] ; 60: la \$a0, strSpace
			[004000d4] 34240038 ori \$4, \$1, 56 [strSpace]
			[004000d8] 34020004 ori \$2, \$0, 4 ; 61: li \$v0, 4
			[004000dc] 00000000 syscall ; 62: syscall
			[004000e0] 2108ffff addi \$8, \$8, -1 ; 64: addi \$t0, \$t0, -1 # Increase counter
			[004000e4] 34090000 ori \$9, \$0, 0 ; 65: li \$t1, 0 # Reset the sum
			[004000e8] 00085021 addi \$10, \$0, \$8 ; 66: move \$t2, \$t0
			[004000ec] 3400b001 ori \$11, \$0, 1 ; 67: li \$t3, 1
			[004000f0] 0100082a slt \$1, \$8, \$0 ; 68: bit \$t0, \$0, EXIT
			[004000f4] 14200020 bne \$1, \$0, 8 [EXIT-0x004000f4]
			j 0x00400060 [LOOP1] ; 69: j LOOP1
			[004000f8] 08100018 ori \$2, \$0, 10 ; 72: li \$v0, 10
			[004000fc] 34020004 syscall ; 73: syscall
			[00400100] 09000000

It will repeat the same step until the $t_0 = 0$

Extra (Input the RGB code):

Code:

```
● ● ●
1 .data
2 strNameId: .asciiz "Phan Tran Thanh Huy\nITCSIU22056\n"
3 enterTheRGB: .asciiz "Enter the RGB code: "
4 printResult: .asciiz "The result is: "
5 arr: .space 100
6 strSpace: .asciiz " "
7
8 .text
9 .globl main
10
11 main:
12    la $a0, strNameId
13    li $v0, 4
14    syscall
15
16    la $a0, enterTheRGB
17    li $v0, 4
18    syscall
19
20    la $a0, arr
21    li $a1, 100
22    li $v0, 8
23    syscall
24
25    la $a0, printResult
26    li $v0, 4
27    syscall
28
29    la $s0, arr
30    li $s1, 2      # for checking
31    li $s2, 16
32    li $t0, 6      # amount of elements
33    li $t1, 0      # for sum
34    move $t2, $t0
35
36 LOOP1:
37    lb $t4, 0($s0)
38    bge $t4, 97, ELSE1
39    bge $t4, 65, ELSE2
40    bge $t4, 48, ELSE3
41
42 ELSE1:
43    addi $t4, $t4, -87
44    j LOOP2
45 ELSE2:
46    addi $t4, $t4, -55
47    j LOOP2
48 ELSE3:
49    addi $t4, $t4, -48
50    j LOOP2
51
52 LOOP2:
53    div $t0, $s1
54    mfhi $t3
55    bne $t3, $0, LABEL1      # If it is not the second RGB, we will sum = sum + 16*value
56    mui $t4, $t4, $s2
57    add $t1, $t1, $t4
58    addi $s0, $s0, 1
59    addi $t0, $t0, -1      # Increase counter
60    move $t2, $t0
61    li $t3, 1
62    beq $t0, $0, EXIT
63    j LOOP1
64
65 LABEL1:
66    add $t1, $t1, $t4      # If it is the second RGB, we will sum = sum + 1*value
67    mtc1 $t1, $f0           # Convert int to float
68    cvt.s.w $f0, $f0
69    li.s $f2, 255.0
70    div.s $f4, $f0, $f2      # sum(float)/255
71
72    mov.s $f12, $f4          # Print the result
73    li $v0, 2
74    syscall
75
76    la $a0, strSpace
77    li $v0, 4
78    syscall
79
80    addi $s0, $s0, 1
81    addi $t0, $t0, -1      # Increase counter
82    li $t1, 0                # Reset the sum
83    move $t2, $t0
84    li $t3, 1
85    beq $t0, $0, EXIT
86    j LOOP1
87
88 EXIT:
89    li $v0, 10
90    syscall
```

Test case:

Pure Red:

```
Console
Phan Tran Thanh Huy
ITCSIU22056
Enter the RGB code: ff0000
The result is: 1.00000000 0.00000000 0.00000000
```

Pure Green:

```
Console
Phan Tran Thanh Huy
ITCSIU22056
Enter the RGB code: 00ff00
The result is: 0.00000000 1.00000000 0.00000000
```

Pure Blue:

```
Console
Phan Tran Thanh Huy
ITCSIU22056
Enter the RGB code: 0000ff
The result is: 0.00000000 0.00000000 1.00000000
```

Single Step:

Convert to float to divide 255

FP Regs	Int Regs [16]	Text
FIR = 9800 FCSR = 0 Single Precision FG0 = c4a4e000 FG1 = 0 FG2 = 0 FG3 = 0 FG4 = 0 FG5 = 0 FG6 = 0 FG7 = 0 FG8 = 0 FG9 = 0 FG10 = 0 FG11 = 0 FG12 = 0 FG13 = 0 FG14 = 0 FG15 = 0 FG16 = 0 FG17 = 0 FG18 = 0 FG19 = 0 FG20 = 0 FG21 = 0 FG22 = 0 FG23 = 0 FG24 = 0 FG25 = 0		[00400074] 34080006 ori \$8, \$0, 6 ; 32: li \$t0, 6 # amount of elements [00400078] 34090000 ori \$9, \$0, 0 ; 33: li \$t1, 0 # for sum [0040007c] 00085021 addu \$10, \$0, \$8 ; 34: move \$t2, \$t0 [00400080] 820c0000 lb \$12, 0(\$16) ; 37: lb \$t4, 0(\$s0) [00400088] 10200005 beq \$1, \$0, 20 [ELSE1-0x0400088] [00400084] 29810061 slti \$1, \$12, 97 ; 38: bge \$t4, 97, ELSE1 [0040008c] 29810041 slti \$1, \$12, 65 ; 39: bge \$t4, 65, ELSE2 [00400094] 29810030 slti \$1, \$12, 48 ; 40: bge \$t4, 48, ELSE3 [00400098] 10200005 beq \$1, \$0, 20 [ELSE2-0x0400090] [0040009c] 218cffa9 addi \$12, \$12, -87 ; 43: addi \$t4, \$t4, -87 [004000a0] 08100024 j 0x04000ba [LOOP2] ; 44: j LOOP2 [004000a4] 218cffc9 addi \$12, \$12, -55 ; 46: addi \$t4, \$t4, -55 [004000a8] 08100024 j 0x04000ba [LOOP2] ; 47: j LOOP2 [004000ac] 218cffd0 addi \$12, \$12, -48 ; 49: addi \$t4, \$t4, -48 [004000b0] 08100024 j 0x04000ba [LOOP2] ; 50: j LOOP2 [004000b8] 00005810 mfhi \$11 ; 54: mfhi \$t3 [004000bc] 15600009 bne \$11, \$0, 36 [LABEL1-0x04000bc] [004000c0] 71926002 mul \$12, \$12, \$18 ; 56: mul \$t4, \$t4, \$s2 [004000c4] 012c4820 add \$9, \$9, \$12 ; 57: add \$t1, \$t1, \$t4 [004000c8] 22100001 addi \$16, \$16, 1 ; 58: addi \$s0, \$s0, 1 [004000cc] 2108ffff addi \$8, \$8, -1 ; 59: addi \$t0, \$t0, -1 # Increase counter [004000d0] 00085021 addu \$10, \$0, \$8 ; 60: move \$t2, \$t0 [004000d4] 340b0001 ori \$11, \$0, 1 ; 61: li \$t3, 1 [004000d8] 11000016 beq \$8, \$0, 88 [EXIT-0x04000d8] [004000dc] 08100024 j 0x0400080 [LOOP1] ; 63: j LOOP1 [004000e0] 012c4820 add \$9, \$9, \$12 ; 66: add \$t1, \$t1, \$t4 # If it is the second RGB, we will sum = sum + 1*value [004000e4] 44890000 mtcl \$9, \$f0 ; 67: mtcl \$t1, \$f0 # Convert int to float [004000e8] 48800020 cvt.s.w \$f0, \$f0 ; 68: cvt.s.w \$f0, \$f0 [004000ec] 3c01437f lui \$1, 17279 ; 69: li.s \$t2, 255.0

FP Regs	Int Regs [16]	Text
FIR = 9800 FCSR = 0 Single Precision FG0 = c4a4e000 FG1 = 0 FG2 = 0 FG3 = 0 FG4 = 0 FG5 = 0 FG6 = 0 FG7 = 0 FG8 = 0 FG9 = 0 FG10 = 0 FG11 = 0 FG12 = 0 FG13 = 0 FG14 = 0 FG15 = 0 FG16 = 0 FG17 = 0 FG18 = 0 FG19 = 0 FG20 = 0 FG21 = 0 FG22 = 0 FG23 = 0 FG24 = 0 FG25 = 0		[00400070] 34120010 ori \$18, \$0, 16 ; 31: li \$s2, 16 [00400074] 34080006 ori \$8, \$0, 6 ; 32: li \$t0, 6 # amount of elements [00400078] 34090000 ori \$9, \$0, 0 ; 33: li \$t1, 0 # for sum [0040007c] 00085021 addu \$10, \$0, \$8 ; 34: move \$t2, \$t0 [00400084] 29810061 slti \$1, \$12, 97 ; 38: bge \$t4, 97, ELSE1 [00400088] 10200005 beq \$1, \$0, 20 [ELSE1-0x0400088] [0040008c] 29810041 slti \$1, \$12, 65 ; 39: bge \$t4, 65, ELSE2 [00400090] 10200005 beq \$1, \$0, 20 [ELSE2-0x0400090] [00400094] 29810030 slti \$1, \$12, 48 ; 40: bge \$t4, 48, ELSE3 [00400098] 10200005 beq \$1, \$0, 20 [ELSE3-0x0400098] [0040009c] 218cffa9 addi \$12, \$12, -87 ; 43: addi \$t4, \$t4, -87 [004000a0] 08100024 j 0x04000b4 [LOOP2] ; 44: j LOOP2 [004000a4] 218cffc9 addi \$12, \$12, -55 ; 46: addi \$t4, \$t4, -55 [004000a8] 08100024 j 0x04000b4 [LOOP2] ; 47: j LOOP2 [004000ac] 218cffd0 addi \$12, \$12, -48 ; 49: addi \$t4, \$t4, -48 [004000b0] 08100024 j 0x04000b4 [LOOP2] ; 50: j LOOP2 [004000b4] 0110001a div \$8, \$17 ; 53: div \$t0, \$s1 [004000b8] 00005810 mfhi \$11 ; 54: mfhi \$t3 [004000bc] 15600009 bne \$11, \$0, 36 [LABEL1-0x04000bc] [004000c0] 71926002 mul \$12, \$12, \$18 ; 56: mul \$t4, \$t4, \$s2 [004000c4] 012c4820 add \$9, \$9, \$12 ; 57: add \$t1, \$t1, \$t4 [004000c8] 22100001 addi \$16, \$16, 1 ; 58: addi \$s0, \$s0, 1 [004000cc] 2108ffff addi \$8, \$8, -1 ; 59: addi \$t0, \$t0, -1 # Increase counter [004000d0] 00085021 addu \$10, \$0, \$8 ; 60: move \$t2, \$t0 [004000d4] 340b0001 ori \$11, \$0, 1 ; 61: li \$t3, 1 [004000d8] 11000016 beq \$8, \$0, 88 [EXIT-0x04000d8] [004000dc] 08100024 j 0x0400080 [LOOP1] ; 63: j LOOP1 [004000e0] 012c4820 add \$9, \$9, \$12 ; 66: add \$t1, \$t1, \$t4 # If it is the second RGB, we will sum = sum + 1*value [004000e4] 44890000 mtcl \$9, \$f0 ; 67: mtcl \$t1, \$f0 # Convert int to float [004000e8] 46800020 cvt.s.w \$f0, \$f0 ; 68: cvt.s.w \$f0, \$f0 [004000ec] 3c01437f lui \$1, 17279 ; 69: li.s \$t2, 255.0

memory and registers cleared

Divide the number to 255 step:

FP Regs	Int Regs [16]	Text
FIR = 9800 FCSR = 0 Single Precision FG0 = c4a4e000 FG1 = 0 FG2 = 43f00000 FG3 = 0 FG4 = 0 FG5 = 0 FG6 = 0 FG7 = 0 FG8 = 0 FG9 = 0 FG10 = 0 FG11 = 0 FG12 = 0 FG13 = 0 FG14 = 0 FG15 = 0 FG16 = 0 FG17 = 0 FG18 = 0 FG19 = 0 FG20 = 0 FG21 = 0 FG22 = 0 FG23 = 0 FG24 = 0 FG25 = 0		[0040007c] 00085021 addu \$10, \$0, \$8 ; 34: move \$t2, \$t0 [00400080] 820c0000 lb \$12, 0(\$16) ; 37: lb \$t4, 0(\$s0) [00400084] 29810061 slti \$1, \$12, 97 ; 38: bge \$t4, 97, ELSE1 [00400088] 10200005 beq \$1, \$0, 20 [ELSE1-0x0400088] [0040008c] 29810041 slti \$1, \$12, 65 ; 39: bge \$t4, 65, ELSE2 [00400090] 10200005 beq \$1, \$0, 20 [ELSE2-0x0400090] [00400094] 29810030 slti \$1, \$12, 48 ; 40: bge \$t4, 48, ELSE3 [00400098] 10200005 beq \$1, \$0, 20 [ELSE3-0x0400098] [0040009c] 218cffa9 addi \$12, \$12, -87 ; 43: addi \$t4, \$t4, -87 [004000a0] 08100024 j 0x04000b4 [LOOP2] ; 44: j LOOP2 [004000a4] 218cffc9 addi \$12, \$12, -55 ; 46: addi \$t4, \$t4, -55 [004000a8] 08100024 j 0x04000b4 [LOOP2] ; 47: j LOOP2 [004000ac] 218cffd0 addi \$12, \$12, -48 ; 49: addi \$t4, \$t4, -48 [004000b0] 08100024 j 0x04000b4 [LOOP2] ; 50: j LOOP2 [004000b4] 0110001a div \$8, \$17 ; 53: div \$t0, \$s1 [004000b8] 00005810 mfhi \$11 ; 54: mfhi \$t3 [004000bc] 15600009 bne \$11, \$0, 36 [LABEL1-0x04000bc] [004000c0] 71926002 mul \$12, \$12, \$18 ; 56: mul \$t4, \$t4, \$s2 [004000c4] 012c4820 add \$9, \$9, \$12 ; 57: add \$t1, \$t1, \$t4 [004000c8] 22100001 addi \$16, \$16, 1 ; 58: addi \$s0, \$s0, 1 [004000cc] 2108ffff addi \$8, \$8, -1 ; 59: addi \$t0, \$t0, -1 # Increase counter [004000d0] 00085021 addu \$10, \$0, \$8 ; 60: move \$t2, \$t0 [004000d4] 340b0001 ori \$11, \$0, 1 ; 61: li \$t3, 1 [004000d8] 11000016 beq \$8, \$0, 88 [EXIT-0x04000d8] [004000dc] 08100024 j 0x0400080 [LOOP1] ; 63: j LOOP1 [004000e0] 012c4820 add \$9, \$9, \$12 ; 66: add \$t1, \$t1, \$t4 # If it is the second RGB, we will sum = sum + 1*value [004000e4] 44890000 mtcl \$9, \$f0 ; 67: mtcl \$t1, \$f0 # Convert int to float [004000e8] 46800020 cvt.s.w \$f0, \$f0 ; 68: cvt.s.w \$f0, \$f0 [004000ec] 3c01437f lui \$1, 17279 ; 69: li.s \$t2, 255.0

FP Regs

FIR = 9800
FCSR = 0

Single Precision
FG0 = c44e000
FG1 = 0
FG2 = 437f0000
FG3 = 0
FG4 = 00a50506

FG5 = 0
FG6 = 0
FG7 = 0
FG8 = 0
FG9 = 0
FG10 = 0
FG11 = 0
FG12 = 0
FG13 = 0
FG14 = 0
FG15 = 0
FG16 = 0
FG17 = 0
FG18 = 0
FG19 = 0
FG20 = 0
FG21 = 0
FG22 = 0
FG23 = 0
FG24 = 0
FG25 = 0

Text

```
[00400098] 10200005 beg $1, $0, 20 [ELSE3-0x00400098]
[0040009c] 218cffa9 addi $12, $12, -87 ; 43: addi $t4, $t4, -87
[004000a0] 0810002d j 0x004000b4 [LOOP2]
[004000a4] 218cffc9 addi $12, $12, -55 ; 46: addi $t4, $t4, -55
[004000a8] 0810002d j 0x004000b4 [LOOP2]
[004000a0] 218cffd0 addi $12, $12, -48 ; 49: addi $t4, $t4, -48
[004000b0] 0810002d j 0x004000b4 [LOOP2]
[004000b4] 0111001a div $8, $17 ; 53: div $t0, $s1
[004000b8] 00005810 mfhi $11 ; 54: mfhi $t3
[004000bc] 15600009 bne $11, $0, 36 [LABEL1-0x004000bc]
[004000c0] 71926002 mul $12, $12, $18 ; 56: mul $t4, $t4, $s2
[004000c4] 012c4820 add $9, $9, $12 ; 57: add $t1, $t1, $t4
[004000c8] 22100001 addi $16, $16, 1 ; 58: addi $s0, $s0, 1
[004000cc] 2108ffff addi $8, $8, -1 ; 59: addi $t0, $t0, -1 # Increase counter
[004000d0] 00085021 addu $10, $0, $8 ; 60: move $t2, $t0
[004000d4] 340b0001 ori $11, $0, 1 ; 61: li $t3, 1
[004000d8] 11000016 beq $8, $0, 88 [EXIT-0x004000d8]
[004000dc] 08100020 j 0x00400080 [LOOP1]
[004000e0] 012c4820 add $9, $9, $12 ; 66: add $t1, $t1, $t4 # If it is the second RGB, we will sum = sum + 1*value
[004000e4] 44890000 mtcl $9, $f0 ; 67: mtcl $t1, $f0 # Convert int to float
[004000e8] 46800020 cvt.s.w $f0, $f0 ; 68: cvt.s.w $f0, $f0
[004000ec] 3c01437f lui $1, 17279 ; 69: li $s2, 255.0
[004000f0] 44811000 mtcl $1, $f2
[004000f4] 46020103 div.s $f4, $f0, $f2 ; 70: div.s $f4, $f0, $f2 # sum(float)/255
[004000f8] 46020236 mov.s $f12, $f4 ; 72: mov.s $f12, $f4 # Print the result
[004000fc] 34020002 ori $2, $0, 2 ; 73: li $v0, 2
[00400100] 0000000c syscall ; 74: syscall
[00400104] 3c011001 lui $1, 4097 [strSpace] ; 76: la $a0, strSpace
[00400108] 342400aa ori $4, $1, 170 [strSpace]
[0040010c] 34020004 ori $2, $0, 4 ; 77: li $v0, 4
[00400110] 0000000c syscall ; 78: syscall
```

Print the result:

FP Regs

FIR = 9800
FCSR = 0

Single Precision
FG0 = c44e000
FG1 = 0
FG2 = 437f0000
FG3 = 0
FG4 = 00a50506

FG5 = 0
FG6 = 0
FG7 = 0
FG8 = 0
FG9 = 0
FG10 = 0
FG11 = 0
FG12 = 0
FG13 = 0
FG14 = 0
FG15 = 0
FG16 = 0
FG17 = 0
FG18 = 0
FG19 = 0
FG20 = 0
FG21 = 0
FG22 = 0
FG23 = 0
FG24 = 0
FG25 = 0

Text

```
[00400098] 10200005 beg $1, $0, 20 [ELSE3-0x00400098]
[0040009c] 218cffa9 addi $12, $12, -87 ; 43: addi $t4, $t4, -87
[004000a0] 0810002d j 0x004000b4 [LOOP2]
[004000a4] 218cffc9 addi $12, $12, -55 ; 46: addi $t4, $t4, -55
[004000a8] 0810002d j 0x004000b4 [LOOP2]
[004000a0] 218cffd0 addi $12, $12, -48 ; 49: addi $t4, $t4, -48
[004000b0] 0810002d j 0x004000b4 [LOOP2]
[004000b4] 0111001a div $8, $17 ; 53: div $t0, $s1
[004000b8] 00005810 mfhi $11 ; 54: mfhi $t3
[004000bc] 15600009 bne $11, $0, 36 [LABEL1-0x004000bc]
[004000c0] 71926002 mul $12, $12, $18 ; 56: mul $t4, $t4, $s2
[004000c4] 012c4820 add $9, $9, $12 ; 57: add $t1, $t1, $t4
[004000c8] 22100001 addi $16, $16, 1 ; 58: addi $s0, $s0, 1
[004000cc] 2108ffff addi $8, $8, -1 ; 59: addi $t0, $t0, -1 # Increase counter
[004000d0] 00085021 addu $10, $0, $8 ; 60: move $t2, $t0
[004000d4] 340b0001 ori $11, $0, 1 ; 61: li $t3, 1
[004000d8] 11000016 beq $8, $0, 88 [EXIT-0x004000d8]
[004000dc] 08100020 j 0x00400080 [LOOP1]
[004000e0] 012c4820 add $9, $9, $12 ; 66: add $t1, $t1, $t4 # If it is the second RGB, we will sum = sum + 1*value
[004000e4] 44890000 mtcl $9, $f0 ; 67: mtcl $t1, $f0 # Convert int to float
[004000e8] 46800020 cvt.s.w $f0, $f0 ; 68: cvt.s.w $f0, $f0
[004000ec] 3c01437f lui $1, 17279 ; 69: li $s2, 255.0
[004000f0] 44811000 mtcl $1, $f2
[004000f4] 46020103 div.s $f4, $f0, $f2 ; 70: div.s $f4, $f0, $f2 # sum(float)/255
[004000f8] 46020236 mov.s $f12, $f4 ; 72: mov.s $f12, $f4 # Print the result
[004000fc] 34020002 ori $2, $0, 2 ; 73: li $v0, 2
[00400100] 0000000c syscall ; 74: syscall
[00400104] 3c011001 lui $1, 4097 [strSpace] ; 76: la $a0, strSpace
[00400108] 342400aa ori $4, $1, 170 [strSpace]
[00400110] 0000000c syscall ; 78: syscall
```

FP Regs

Int Regs [16]

Int Regs [16]

PC = 40011c
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000ff10

HI = 1
LO = 2

R0 [r0] = 0
R1 [at] = 10010000
R2 [v0] = 4
R3 [v1] = 0
R4 [a0] = 100100aa
R5 [a1] = 64
R6 [a2] = 7ffff1a8
R7 [a3] = 0
R8 [t1] = 4
R9 [t2] = ffffffad9
R10 [t3] = 5
R11 [t4] = 1
R12 [t4] = ffffffa9
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 10010048
R17 [s1] = 2
R18 [s2] = 10
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0

Data

Text

```
[004000a4] 218cffc9 addi $12, $12, -55 ; 46: addi $t4, $t4, -55
[004000a8] 0810002d j 0x004000b4 [LOOP2]
[004000ac] 218cffd0 addi $12, $12, -48 ; 49: addi $t4, $t4, -48
[004000b0] 0810002d j 0x004000b4 [LOOP2]
[004000b4] 0111001a div $8, $17 ; 53: div $t0, $s1
[004000b8] 00005810 mfhi $11 ; 54: mfhi $t3
[004000bc] 15600009 bne $11, $0, 36 [LABEL1-0x004000bc]
[004000c0] 71926002 mul $12, $12, $18 ; 56: mul $t4, $t4, $s2
[004000c4] 012c4820 add $9, $9, $12 ; 57: add $t1, $t1, $t4
[004000c8] 22100001 addi $16, $16, 1 ; 58: addi $s0, $s0, 1
[004000cc] 2108ffff addi $8, $8, -1 ; 59: addi $t0, $t0, -1 # Increase counter
[004000d0] 00085021 addu $10, $0, $8 ; 60: move $t2, $t0
[004000d4] 340b0001 ori $11, $0, 1 ; 61: li $t3, 1
[004000d8] 11000016 beq $8, $0, 88 [EXIT-0x004000d8]
[004000dc] 08100020 j 0x00400080 [LOOP1]
[004000e0] 012c4820 add $9, $9, $12 ; 66: add $t1, $t1, $t4 # If it is the second RGB, we will sum = sum + 1*value
[004000e4] 44890000 mtcl $9, $f0 ; 67: mtcl $t1, $f0 # Convert int to float
[004000e8] 46800020 cvt.s.w $f0, $f0 ; 68: cvt.s.w $f0, $f0
[004000ec] 3c01437f lui $1, 17279 ; 69: li $s2, 255.0
[004000f0] 44811000 mtcl $1, $f2
[004000f4] 46020103 div.s $f4, $f0, $f2 ; 70: div.s $f4, $f0, $f2 # sum(float)/255
[004000f8] 46020236 mov.s $f12, $f4 ; 72: mov.s $f12, $f4 # Print the result
[004000fc] 34020002 ori $2, $0, 2 ; 73: li $v0, 2
[00400100] 0000000c syscall ; 74: syscall
[00400104] 3c011001 lui $1, 4097 [strSpace] ; 76: la $a0, strSpace
[00400108] 342400aa ori $4, $1, 170 [strSpace]
[00400110] 0000000c syscall ; 78: syscall
[00400114] 22100001 addi $16, $16, 1 ; 80: addi $s0, $s0, 1
[00400118] 2108ffff addi $8, $8, -1 ; 81: addi $t0, $t0, -1 # Increase counter
[0040011c] 34090000 ori $9, $0, 0 ; 82: li $t1, 0 # Reset the sum
```

It will repeat until t0 = 0

Exercise 4 – Polynomial Evaluation

Code:

```
 1  .data
 2  strNameId: .asciiz "Phan Tran Thanh Huy\nITCSIU22056\n"
 3  readX: .asciiz "Enter the value of x: "
 4  printResult: .asciiz "The result is: "
 5  n: .word 5
 6  a: .float 4.3, -12.4, 6.8, -0.45, 3.6
 7
 8  .text
 9  .globl main
10
11 main:
12     la $a0, strNameId
13     li $v0, 4
14     syscall
15
16     la $a0, readX
17     li $v0, 4
18     syscall
19
20     la $s0, a
21     lw $t0, n
22     li.s $f1, 0.0    #sum
23
24     li $v0, 6    # Enter x
25     syscall
26     mov.s $f2, $f0    # f2 = f0 = x
27
28 LOOP:
29     lwc1 $f3, 0($s0)      # f3 = a[i]
30     mul.s $f1, $f1, $f2    # sum = sum*x
31     add.s $f1, $f1, $f3    # sum = sum + a[i] = sum*x + a[i]
32     addi $s0, $s0, 4
33     addi $t0, $t0, -1      # Increase counter
34     beq $t0, $0, EXIT      # If t0 = 0, we stop the loop
35     j LOOP
36
37 EXIT:
38     la $a0, printResult
39     li $v0, 4
40     syscall
41
42     mov.s $f12, $f1        # Print the result
43     li $v0, 2
44     syscall
45
46     li $v0, 10
47     syscall
```

Test case:

FP Regs Int Regs [16]

FP Regs

Int Regs [16]

Text

File Edit View

.text

.globl main

main:

 la \$a0, strNameId

 li \$v0, 4

 syscall

 la \$a0, readX

 li \$v0, 4

 syscall

 la \$s0, a

 lw \$t0, n

 li.s \$f1, 0.0 #sum

 li \$v0, 6 # Enter x

 syscall

 mov.s \$f2, \$f0 # f2 = f0 = x

LOOP:

 lwcl \$f3, 0(\$s0) # f3 = a[i]

 mul.s \$f1, \$f1, \$f2 # sum = sum*x

 add.s \$f1, \$f1, \$f3 # sum = sum + a[i] = sum*x + a[i]

 addi \$s0, \$s0, 4

 addi \$t0, \$t0, -1 # Increase counter

 beq \$t0, \$0, EXIT # If t0 = 0, we stop the loop

 j LOOP

EXIT:

 la \$a0, printResult

 li \$v0, 4

 syscall

 mov.s \$f12, \$f1 # Print the result

 li \$v0, 2

 syscall

 li \$v0, 10

 syscall

Memory and registers cleared

SPIM Version 5.1.24 of August 1, 2023 (final)

Copyright 1990-2023 by James Larus.

All Rights Reserved.

SPIM is distributed under a BSD license.

See the file README for a full copyright notice.

QsSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version

Memory and registers cleared

SPIM Version 5.1.24 of August 1, 2023 (final)

Copyright 1990-2023 by James Larus.

All Rights Reserved.

SPIM is distributed under a BSD license.

See the file README for a full copyright notice.

QsSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version

Ln 47, Col 10 771 of 771 characters 100% Windows (CRLF) UTF-8

Single Step:

FP Regs Int Regs [16]

FP Regs

Int Regs [16]

Data Text

Text

File Edit View

[00400000] 8fa40000 lw \$4, 0(\$29) ; 183: lw \$a0, 0(\$sp) # argc

[00400004] 27a50004 addiu \$5, \$29, 4 ; 184: addiu \$a1 \$a0 4 # argv

[00400008] 24a60004 addiu \$6, \$5, 4 ; 185: addiu \$a2 \$a1 4 # envp

[0040000c] 00041080 sll \$2, \$4, 2 ; 186: sll \$v0 \$a0 2

[00400010] 00c23021 addu \$6, \$6, \$2 ; 187: addu \$a2 \$a2 \$v0

[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main

[00400018] 00000000 nop ; 189: nop

[0040001c] 3402000a ori \$2, \$0, 10 ; 191: li \$v0 10

[00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit)

[00400024] 3c041001 lui \$4, 4097 [strNameId] ; 12: la \$a0, strNameId

[00400028] 34020004 ori \$2, \$0, 4 ; 13: li \$v0, 4

[0040002c] 0000000c syscall ; 14: syscall

[00400030] 3c011001 lui \$1, 4097 [readX] ; 16: la \$a0, readX

[00400034] 34240021 ori \$4, \$1, 33 [readX]

[00400038] 34020004 ori \$2, \$0, 4 ; 17: li \$v0, 4

[0040003c] 0000000c syscall ; 18: syscall

[00400040] 3c011001 lui \$1, 4097 [a] ; 20: la \$s0, a

[00400044] 3430004c ori \$16, \$1, 76 [a]

[00400048] 3c011001 lui \$1, 4097 ; 21: lw \$t0, n

[0040004c] 8c280048 lw \$8, 72(\$1)

[00400050] 34010000 ori \$1, \$0, 0 ; 22: li.s \$f1, 0.0 #sum

[00400054] 44810800 mtcl \$1, \$f1

[00400058] 34020006 ori \$2, \$0, 6 ; 24: li \$v0, 6 # Enter x

[0040005c] 0000000c syscall ; 25: syscall

[00400060] 46000086 mov.s \$f2, \$f0 ; 26: mov.s \$f2, \$f0 # f2 = f0 = x

[00400064] c6030000 lwcl \$f3, 0(\$16) ; 29: lwcl \$f3, 0(\$s0) # f3 = a[i]

[00400068] 46020842 mul.s \$f1, \$f1, \$f2 ; 30: mul.s \$f1, \$f1, \$f2 # sum = sum*x

[0040006c] 46030840 add.s \$f1, \$f1, \$f3 ; 31: add.s \$f1, \$f1, \$f3 # sum = sum + a[i] = sum*x + a[i]

[00400070] 22100004 addi \$16, \$16, 4 ; 32: addi \$s0, \$s0, 4

[00400074] 2108ffff addi \$8, \$8, -1 ; 33: addi \$t0, \$t0, -1 # Increase counter

FP Regs Text

```

FIR = 9800
FCSR = 0

Single Precision
FG0 = 40400000
FG1 = 0
FG2 = 40400000
FG3 = 0
FG4 = 0
FG5 = 0
FG6 = 0
FG7 = 0
FG8 = 0
FG9 = 0
FG10 = 0
FG11 = 0
FG12 = 0
FG13 = 0
FG14 = 0
FG15 = 0
FG16 = 0
FG17 = 0
FG18 = 0
FG19 = 0
FG20 = 0
FG21 = 0
FG22 = 0
FG23 = 0
FG24 = 0
FG25 = 0

```

User Text Segment [00400000]..[00440000]

```

[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v0
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 34020004 ori $2, $0, 10 ; 191: li $v0 10
[00400020] 00000000 syscall ; 192: syscall # syscall 10 (exit)
[00400024] 3c041001 lui $4, 4097 [strNameId] ; 12: la $a0, strNameId
[00400028] 34020004 ori $2, $0, 4 ; 13: li $v0, 4
[0040002c] 00000000 syscall ; 14: syscall
[00400030] 3c011001 lui $1, 4097 [readX] ; 16: la $a0, readX
[00400034] 34240021 ori $4, $1, 33 [readX] ; 21: lw $t0, n
[00400038] 34020004 ori $2, $0, 4 ; 17: li $v0, 4
[0040003c] 00000000 syscall ; 18: syscall
[00400040] 3c011001 lui $1, 4097 [a] ; 20: la $s0, a
[00400044] 3430004c ori $16, $1, 76 [a] ; 23: lwcl $f3, 0($s0) # f3 = a[i]
[00400048] 3c011001 lui $1, 4097 ; 26: mov.s $f2, $f0 # f2 = f0 = x
[0040004c] 8c280048 lw $8, 72($1) ; 29: lwcl $f3, 0($s0) # f3 = a[i]
[00400050] 34010000 ori $1, $0, 0 ; 30: mul.s $f1, $f1, $f2 # sum = sum*x
[00400054] 44810800 mtcl $1, $f1 ; 31: add.s $f1, $f1, $f3 # sum = sum + a[i] = sum*x + a[i]
[00400058] 34020006 ori $2, $0, 6 ; 32: addi $s0, $s0, 4
[00400060] 46000086 mov.s $f2, $f0 ; 33: addi $t0, $t0, -1 # Increase counter

```

FP Regs Int Regs [16] Data Text

```

FIR = 9800
FCSR = 0

Single Precision
FG0 = 40400000
FG1 = 0
FG2 = 40400000
FG3 = 4089999a
FG4 = 0
FG5 = 0
FG6 = 0
FG7 = 0
FG8 = 0
FG9 = 0
FG10 = 0
FG11 = 0
FG12 = 0
FG13 = 0
FG14 = 0
FG15 = 0
FG16 = 0
FG17 = 0
FG18 = 0
FG19 = 0
FG20 = 0
FG21 = 0
FG22 = 0
FG23 = 0
FG24 = 0
FG25 = 0

```

User Text Segment [00400000]..[00440000]

```

[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v0
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 34020004 ori $2, $0, 10 ; 191: li $v0 10
[00400020] 00000000 syscall ; 192: syscall # syscall 10 (exit)
[00400024] 3c041001 lui $4, 4097 [strNameId] ; 12: la $a0, strNameId
[00400028] 34020004 ori $2, $0, 4 ; 13: li $v0, 4
[0040002c] 00000000 syscall ; 14: syscall
[00400030] 3c011001 lui $1, 4097 [readX] ; 16: la $a0, readX
[00400034] 34240021 ori $4, $1, 33 [readX] ; 21: lw $t0, n
[00400038] 34020004 ori $2, $0, 4 ; 17: li $v0, 4
[0040003c] 00000000 syscall ; 18: syscall
[00400040] 3c011001 lui $1, 4097 [a] ; 20: la $s0, a
[00400044] 3430004c ori $16, $1, 76 [a] ; 23: lwcl $f3, 0($s0) # f3 = a[i]
[00400048] 3c011001 lui $1, 4097 ; 26: mul.s $f1, $f1, $f2 # sum = sum*x
[0040004c] 8c280048 lw $8, 72($1) ; 29: mul.s $f1, $f1, $f3 # sum = sum + a[i] = sum*x + a[i]
[00400050] 34010000 ori $1, $0, 0 ; 30: add.s $f1, $f1, $f3 # sum = sum + a[i] = sum*x + a[i]
[00400054] 44810800 mtcl $1, $f1 ; 31: addi $s0, $s0, 4
[00400058] 34020006 ori $2, $0, 6 ; 32: addi $t0, $t0, -1 # Increase counter

```

FP Regs Text

```

FIR = 9800
FCSR = 0

Single Precision
FG0 = 40400000
FG1 = 4089999a
FG2 = 4089999a
FG3 = 4089999a
FG4 = 0
FG5 = 0
FG6 = 0
FG7 = 0
FG8 = 0
FG9 = 0
FG10 = 0
FG11 = 0
FG12 = 0
FG13 = 0
FG14 = 0
FG15 = 0
FG16 = 0
FG17 = 0
FG18 = 0
FG19 = 0
FG20 = 0
FG21 = 0
FG22 = 0
FG23 = 0
FG24 = 0
FG25 = 0

```

User Text Segment [00400000]..[00440000]

```

[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v0
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 34020004 ori $2, $0, 10 ; 191: li $v0 10
[00400020] 00000000 syscall ; 192: syscall # syscall 10 (exit)
[00400024] 3c041001 lui $4, 4097 [strNameId] ; 12: la $a0, strNameId
[00400028] 34020004 ori $2, $0, 4 ; 13: li $v0, 4
[0040002c] 00000000 syscall ; 14: syscall
[00400030] 3c011001 lui $1, 4097 [readX] ; 16: la $a0, readX
[00400034] 34240021 ori $4, $1, 33 [readX] ; 21: lw $t0, n
[00400038] 34020004 ori $2, $0, 4 ; 17: li $v0, 4
[0040003c] 00000000 syscall ; 18: syscall
[00400040] 3c011001 lui $1, 4097 [a] ; 20: la $s0, a
[00400044] 3430004c ori $16, $1, 76 [a] ; 23: lwcl $f3, 0($s0) # f3 = a[i]
[00400048] 3c011001 lui $1, 4097 ; 26: mov.s $f2, $f0 # f2 = f0 = x
[0040004c] 8c280048 lw $8, 72($1) ; 29: lwcl $f3, 0($s0) # f3 = a[i]
[00400050] 34010000 ori $1, $0, 0 ; 30: mul.s $f1, $f1, $f2 # sum = sum*x
[00400054] 44810800 mtcl $1, $f1 ; 31: add.s $f1, $f1, $f3 # sum = sum + a[i] = sum*x + a[i]
[00400058] 34020006 ori $2, $0, 6 ; 32: addi $s0, $s0, 4
[00400060] 46000086 mov.s $f2, $f0 ; 33: addi $t0, $t0, -1 # Increase counter

```

FP Regs	Int Regs [16]	Data	Text
PC = 400074			
EPC = 0			
Cause = 0			
BadVAddr = 0			
Status = 3000ff10			
HI = 0			
LO = 0			
R0 [r0] = 0			
R1 [at] = 0			
R2 [v0] = 6			
R3 [v1] = 0			
R4 [a0] = 10010021			
R5 [a1] = 7ffff1a8			
R6 [a2] = 7ffff1b0			
R7 [a3] = 0			
R8 [t0] = 5			
R9 [t1] = 0			
R10 [t2] = 0			
R11 [t3] = 0			
R12 [t4] = 0			
R13 [t5] = 0			
R14 [t6] = 0			
R15 [t7] = 0			
R16 [s0] = 10010050			
R17 [s1] = 0			
R18 [s2] = 0			
R19 [s3] = 0			
R20 [s4] = 0			
R21 [s5] = 0			

```

[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp) # argo
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v0
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit)
[00400024] 3c041001 lui $4, 4097 [strNameId] ; 12: la $a0, strNameId
[00400028] 34020004 ori $2, $0, 4 ; 13: li $v0, 4
[0040002c] 0000000c syscall ; 14: syscall
[00400030] 3c011001 lui $1, 4097 [readX] ; 15: la $a0, readX
[00400034] 34240021 ori $4, $1, 33 [readX] ; 16: la $a0, readX
[00400038] 34020004 ori $2, $0, 4 ; 17: li $v0, 4
[0040003c] 0000000c syscall ; 18: syscall
[00400040] 3c011001 lui $1, 4097 [a] ; 19: la $a0, a
[00400044] 3430004c ori $16, $1, 76 [a] ; 20: la $s0, a
[00400048] 3c011001 lui $1, 4097 ; 21: lw $t0, n
[0040004c] 8c280048 lw $8, 72($1) ; 22: li.s $f1, 0.0 #sum
[00400050] 34010000 ori $1, $0, 0 ; 23: mul.s $f1, $f1, $f2 # sum = sum*x
[00400054] 44810800 mtcl $1, $f1 ; 24: li $v0, 6 # Enter x
[00400058] 34020006 ori $2, $0, 6 ; 25: syscall
[00400060] 46000086 mov.s $f2, $f0 ; 26: mov.s $f2, $f0 # f2 = f0 = x
[00400064] c6030000 lwcl $f3, 0($s0) # f3 = a[i] ; 29: lwcl $f3, 0($s0) # f3 = a[i]
[00400068] 46020842 mul.s $f1, $f1, $f2 ; 30: mul.s $f1, $f1, $f2 # sum = sum*x
[0040006c] 46030840 add.s $f1, $f1, $f3 ; 31: add.s $f1, $f1, $f3 # sum = sum + a[i] = sum*x + a[i]
[00400070] 22100004 addi $16, $16, 4 ; 32: addi $s0, $s0, 4
[00400074] 2108ffff addi $8, $8, -1 ; 33: addi $t0, $t0, -1 # Increase counter
[00400078] 2108ffff addi $8, $8, -1 ; 34: beg $t0, $0, EXIT # If t0 = 0, we stop the loop

```

FP Regs	Int Regs [16]	Data	Text
PC = 400078			
EPC = 0			
Cause = 0			
BadVAddr = 0			
Status = 3000ff10			
HI = 0			
LO = 0			
R0 [r0] = 0			
R1 [at] = 0			
R2 [v0] = 6			
R3 [v1] = 0			
R4 [a0] = 10010021			
R5 [a1] = 7ffff1a8			
R6 [a2] = 7ffff1b0			
R7 [a3] = 0			
R8 [t0] = 4			
R9 [t1] = 0			
R10 [t2] = 0			
R11 [t3] = 0			
R12 [t4] = 0			
R13 [t5] = 0			
R14 [t6] = 0			
R15 [t7] = 0			
R16 [s0] = 10010050			
R17 [s1] = 0			
R18 [s2] = 0			
R19 [s3] = 0			
R20 [s4] = 0			
R21 [s5] = 0			

```

[00400000] 8fa40000 lw $4, 0($29) ; 183: lw $a0 0($sp) # argo
[00400004] 27a50004 addiu $5, $29, 4 ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004 addiu $6, $5, 4 ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080 sll $2, $4, 2 ; 186: sll $v0 $a0 2
[00400010] 00c23021 addu $6, $6, $2 ; 187: addu $a2 $a2 $v0
[00400014] 0c100009 jal 0x00400024 [main] ; 188: jal main
[00400018] 00000000 nop ; 189: nop
[0040001c] 3402000a ori $2, $0, 10 ; 191: li $v0 10
[00400020] 0000000c syscall ; 192: syscall # syscall 10 (exit)
[00400024] 3c041001 lui $4, 4097 [strNameId] ; 12: la $a0, strNameId
[00400028] 34020004 ori $2, $0, 4 ; 13: li $v0, 4
[0040002c] 0000000c syscall ; 14: syscall
[00400030] 3c011001 lui $1, 4097 [readX] ; 15: la $a0, readX
[00400034] 34240021 ori $4, $1, 33 [readX] ; 16: la $a0, readX
[00400038] 34020004 ori $2, $0, 4 ; 17: li $v0, 4
[0040003c] 0000000c syscall ; 18: syscall
[00400040] 3c011001 lui $1, 4097 [a] ; 19: la $a0, a
[00400044] 3430004c ori $16, $1, 76 [a] ; 20: la $s0, a
[00400048] 3c011001 lui $1, 4097 ; 21: lw $t0, n
[0040004c] 8c280048 lw $8, 72($1) ; 22: li.s $f1, 0.0 #sum
[00400050] 34010000 ori $1, $0, 0 ; 23: mul.s $f1, $f1, $f2 # sum = sum*x
[00400054] 44810800 mtcl $1, $f1 ; 24: li $v0, 6 # Enter x
[00400058] 34020006 ori $2, $0, 6 ; 25: syscall
[00400060] 46000086 mov.s $f2, $f0 ; 26: mov.s $f2, $f0 # f2 = f0 = x
[00400064] c6030000 lwcl $f3, 0($s0) # f3 = a[i] ; 29: lwcl $f3, 0($s0) # f3 = a[i]
[00400068] 46020842 mul.s $f1, $f1, $f2 ; 30: mul.s $f1, $f1, $f2 # sum = sum*x
[0040006c] 46030840 add.s $f1, $f1, $f3 ; 31: add.s $f1, $f1, $f3 # sum = sum + a[i] = sum*x + a[i]
[00400070] 22100004 addi $16, $16, 4 ; 32: addi $s0, $s0, 4
[00400074] 2108ffff addi $8, $8, -1 ; 33: addi $t0, $t0, -1 # Increase counter
[00400078] 2108ffff addi $8, $8, -1 ; 34: beg $t0, $0, EXIT # If t0 = 0, we stop the loop

```

It will repeat the same step until the t0 = 0

Print the result:

FP Regs	Int Regs [16]	Data	Text
FIR = 9800			
FCSR = 0			
Single Precision			
FG0 = 40400000			
FG1 = 4299e669			
FG2 = 40400000			
FG3 = 40666666			
FG4 = 0			
FG5 = 0			
FG6 = 0			
FG7 = 0			
FG8 = 0			
FG9 = 0			
FG10 = 0			
FG11 = 0			
FG12 = 4299e669			
FG13 = 0			
FG14 = 0			
FG15 = 0			
FG16 = 0			
FG17 = 0			
FG18 = 0			
FG19 = 0			
FG20 = 0			
FG21 = 0			
FG22 = 0			
FG23 = 0			
FG24 = 0			
FG25 = 0			
... = ...			

```

[0040001c] 3402000a ori $2, $0, 10      ; 191: li $v0, 10
[00400020] 0000000c syscall                ; 192: syscall # syscall 10 (exit)
[00400024] 3c041001 lui $4, 4097 [strNameId]; 12: la $a0, strNameId
[00400028] 34020004 ori $2, $0, 4       ; 13: li $v0, 4
[0040002c] 0000000c syscall                ; 14: syscall
[00400034] 34240021 ori $4, $1, 33 [readX] ; 16: la $a0, readX
[00400038] 34020004 ori $2, $0, 4       ; 17: li $v0, 4
[0040003c] 0000000c syscall                ; 18: syscall
[00400040] 3c011001 lui $1, 4097 [a]      ; 20: la $s0, a
[00400044] 3430004c ori $16, $1, 76 [a]   ; 21: lw $t0, n
[00400048] 3c011001 lui $1, 4097
[0040004c] 8c280048 lw $8, 72($1)
[00400050] 34010000 ori $1, $0, 0       ; 22: li.s $f1, 0.0 #sum
[00400054] 44810800 mtcl $1, $f1
[0040005c] 0000000c syscall                ; 24: li $v0, 6 # Enter x
[00400060] 46000086 mov.s $f2, $f0      ; 26: mov.s $f2, $f0 # f2 = f0 = x
[00400064] c6030000 lwc1 $f3, 0($16)   ; 29: lwc1 $f3, 0($s0) # f3 = a[i]
[00400068] 46020842 mul.s $f1, $f1, $f2 ; 30: mul.s $f1, $f1, $f2 # sum = sum*x
[0040006c] 46030840 add.s $f1, $f1, $f3 ; 31: add.s $f1, $f1, $f3 # sum = sum + a[i] = sum*x + a[i]
[00400070] 22100004 addi $16, $16, 4   ; 32: addi $s0, $s0, 4
[00400074] 2108ffff addi $8, $9, -1   ; 33: addi $t0, $t0, -1 # Increase counter
[00400078] 11000002 beq $8, $0, 8 [EXIT-0x00400078]; 34: beq $t0, $0, EXIT # If t0 = 0, we stop the loop
[0040007c] 08100019 j 0x00400064 [LOOP] ; 35: j LOOP
[00400080] 3c011001 lui $1, 4097 [printResult]; 38: la $a0, printResult
[00400084] 34240038 ori $4, $1, 56 [printResult]
[00400088] 34020004 ori $2, $0, 4       ; 39: li $v0, 4
[0040008c] 0000000c syscall                ; 40: syscall
[00400090] 46000b06 mov.s $f12, $f1   ; 42: mov.s $f12, $f1 # Print the result
[00400094] 34020002 ori $2, $0, 2       ; 43: li $v0, 2

```