

CS 211

LAB 6 : Dynamic Memory Allocation

Name – Gautam Kumar Mahar

Roll No. – 2103114

Branch – Computer Science Engineering

PART B

Write Mips assembly programs for the following exercises. Execute in single stepping mode and check the updates in data memory and other registers during dynamic memory allocation process.

Exercise:

- 1) Dynamically allocate memory to store a structure ‘Date’, which has three variables of integer type, namely date, month and year. The values for the same have to be entered by the user. Place the next date in next memory location and also display it on the screen.**

Output -

A screenshot of a debugger interface, likely QSpim, showing assembly code and registers for a C++ program. The assembly code is in x86-64 format, with comments explaining the purpose of various instructions. The registers window shows integer and floating-point register values. The memory window shows the cleared memory. The status bar at the bottom provides copyright information and license details.

Exercise 1 Is Here .
PLEASE GIVE INPUT .-->
Please Enter the date .--> 10
Please Enter the month .--> 11
Please Enter the year .--> 2021

OUTPUT .
Here is next date .--> 20
Here is Next date month .--> 11
Here is Next date year .--> 2021

QtSpim
Console
Registers
Text Segment
Data Segment
Window Help

FP Regs nt Regs [16] Data Text

```

Int Regs [16]
PC = 4001dc
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000ff10
HI = 0
LO = 0
$10 ($d0) = 0
$11 ($a1) = 10010000
$12 ($v0) = a
$13 ($v1) = 0
$14 ($a0) = 7e5
$15 ($t0) = 2
$16 ($a1) = b
$17 ($a3) = 7e5
$18 ($t0) = 10040000
$19 ($t1) = 0
$1a ($t2) = 0
$1b ($t3) = 0
$1c ($t4) = 0
$1d ($t5) = 14
$1e ($t6) = 14
$1f ($t7) = 0
$20 ($t8) = 0
$21 ($t9) = 0
$22 ($t10) = 0
$23 ($t11) = 0
$24 ($t12) = 0
$25 ($t13) = 0
$26 ($t14) = 0
$27 ($t15) = 0

```

```

[00400180] 08100061 3c0x400184 [DONE] ; 163: j DONE # Jump to the "DONE" label
[00400184] 3c011001 lui $1, 4097 [OUTPUT_OF_DATE]; 167: la $a0, OUTPUT_OF_DATE # Load the address of
the "OUTPUT_OF_DATE" string into $a0
[00400188] 3c240080 ori $2, $0, 4 ; 168: li $v0, 4 # Load the value 4 into $v0
[00400190] 00000000 syscall ; 169: syscall # Execute the syscall to print the string
[00400194] 3c240000 ori $2, $0, 4 ; 170: li $v0, 0($15) # Load the value at memory address
$171: lw $a0, 0($15)
[00400198] 34200001 ori $2, $0, 1 ; 172: li $v0, 1 # Load the value 1 into $v0
[0040019c] 00000000 syscall ; 173: syscall # Execute the syscall to print the value
[004001a0] 3c011001 lui $1, 4097 [OUTPUT_OF_MONTH]; 175: la $a0, OUTPUT_OF_MONTH # Load the address
of the "OUTPUT_OF_MONTH" string into $a0
[004001a4] 3c2400a1 ori $2, $1, 161 [OUTPUT_OF_MONTH]
[004001a8] 3c240000 ori $2, $0, 4 ; 176: li $v0, 4 # Load the value 4 into $v0
[004001ac] 00000000 syscall ; 177: syscall # Execute the syscall to print the string
[004001b0] 8da40000 lw $4, 4($13) ; 178: li $v0, 4($15) # Load the value at memory address
$179: lw $a0, 4($15)
[004001b4] 3c240001 ori $2, $0, 1 ; 180: li $v0, 1 # Load the value 1 into $v0, which
corresponds to the "print integer" syscall
[004001b8] 00000000 syscall ; 181: syscall # Execute the syscall to print the value
[004001bc] 3c011001 lui $1, 4097 [OUTPUT_OF_YEAR]; 183: la $a0, OUTPUT_OF_YEAR # Load the address of
the "OUTPUT_OF_YEAR" string into $a0
[004001c4] 3c2400b0 ori $2, $1, 191 [OUTPUT_OF_YEAR]
[004001c8] 3c240000 ori $2, $0, 4 ; 184: li $v0, 4 # Load the value 4 into $v0, which
corresponds to the "print integer" syscall
[004001cc] 8da40000 lw $4, 8($13) ; 185: syscall # Execute the syscall to print the string
[004001d0] 00000000 syscall ; 186: syscall # Execute the syscall to print the value
[004001d4] 3c240001 ori $2, $0, 1 ; 188: li $v0, 1 # Load the value 1 into $v0, which
corresponds to the "print integer" syscall
[004001d8] 00000000 syscall ; 189: syscall # Execute the syscall to print the value
[004001dc] 3c240000 ori $2, $0, 10 ; 191: li $v0, 10 # Load the value 10 into $v0, which
corresponds to the "exit" syscall
[004001e0] 00000000 syscall ; 192: syscall # Execute the syscall to exit the program

```

Memory and registers cleared

SPIM Version 0.1.23 of December 4, 2021
Copyright 1990-2021 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.

Exercise 1 Is Here .
PLEASE GIVE INPUT .-->
Please Enter the date .--> 11
Please Enter the month .--> 12
Please Enter the year .--> 2009

OUTPUT .
Here is next date .--> 1
Here is Next date month .--> 2
Here is Next date year .--> 2009

QtSpim
Console
Registers
Text Segment
Data Segment
Window Help

FP Regs nt Regs [16] Data Text

```

Int Regs [16]
PC = 4001dc
EPC = 0
Cause = 0
BadVAddr = 0
Status = 3000ff10
HI = 0
LO = 0
$10 ($d0) = 0
$11 ($a1) = 10010000
$12 ($v0) = a
$13 ($v1) = 0
$14 ($a0) = 7d9
$15 ($t0) = 1f
$16 ($a1) = 7d9
$17 ($t1) = 0
$18 ($t2) = 0
$19 ($t3) = 0
$20 ($t4) = 0
$21 ($t5) = 10040000
$22 ($t6) = 0
$23 ($t7) = 0
$24 ($t8) = 20
$25 ($t9) = 0
$26 ($t10) = 0
$27 ($t11) = 0

```

```

[00400180] 08100061 3c0x400184 [DONE] ; 163: j DONE # Jump to the "DONE" label
[00400184] 3c011001 lui $1, 4097 [OUTPUT_OF_DATE]; 167: la $a0, OUTPUT_OF_DATE # Load the address of
the "OUTPUT_OF_DATE" string into $a0
[00400188] 3c240080 ori $2, $0, 4 ; 168: li $v0, 4 # Load the value 4 into $v0
[00400190] 00000000 syscall ; 169: syscall # Execute the syscall to print the string
[00400194] 3c240000 ori $2, $0, 4 ; 170: li $v0, 0($15) # Load the value at memory address
$171: lw $a0, 0($15)
[00400198] 34200001 ori $2, $0, 1 ; 172: li $v0, 1 # Load the value 1 into $v0
[0040019c] 00000000 syscall ; 173: syscall # Execute the syscall to print the value
[004001a0] 3c011001 lui $1, 4097 [OUTPUT_OF_MONTH]; 175: la $a0, OUTPUT_OF_MONTH # Load the address
of the "OUTPUT_OF_MONTH" string into $a0
[004001a4] 3c2400a1 ori $2, $1, 161 [OUTPUT_OF_MONTH]
[004001a8] 3c240000 ori $2, $0, 4 ; 176: li $v0, 4 # Load the value 4 into $v0
[004001ac] 00000000 syscall ; 177: syscall # Execute the syscall to print the string
[004001b0] 8da40000 lw $4, 4($13) ; 178: li $v0, 4($15) # Load the value at memory address
$179: lw $a0, 4($15)
[004001b4] 3c240001 ori $2, $0, 1 ; 180: li $v0, 1 # Load the value 1 into $v0, which
corresponds to the "print integer" syscall
[004001b8] 00000000 syscall ; 181: syscall # Execute the syscall to print the value
[004001bc] 3c011001 lui $1, 4097 [OUTPUT_OF_YEAR]; 183: la $a0, OUTPUT_OF_YEAR # Load the address of
the "OUTPUT_OF_YEAR" string into $a0
[004001c4] 3c2400b0 ori $2, $1, 191 [OUTPUT_OF_YEAR]
[004001c8] 3c240000 ori $2, $0, 4 ; 184: li $v0, 4 # Load the value 4 into $v0, which
corresponds to the "print integer" syscall
[004001cc] 8da40000 lw $4, 8($13) ; 185: syscall # Execute the syscall to print the string
[004001d0] 00000000 syscall ; 186: syscall # Execute the syscall to print the value
[004001d4] 3c240001 ori $2, $0, 1 ; 188: li $v0, 1 # Load the value 1 into $v0, which
corresponds to the "print integer" syscall
[004001d8] 00000000 syscall ; 189: syscall # Execute the syscall to print the value
[004001dc] 3c240000 ori $2, $0, 10 ; 191: li $v0, 10 # Load the value 10 into $v0, which
corresponds to the "exit" syscall
[004001e0] 00000000 syscall ; 192: syscall # Execute the syscall to exit the program

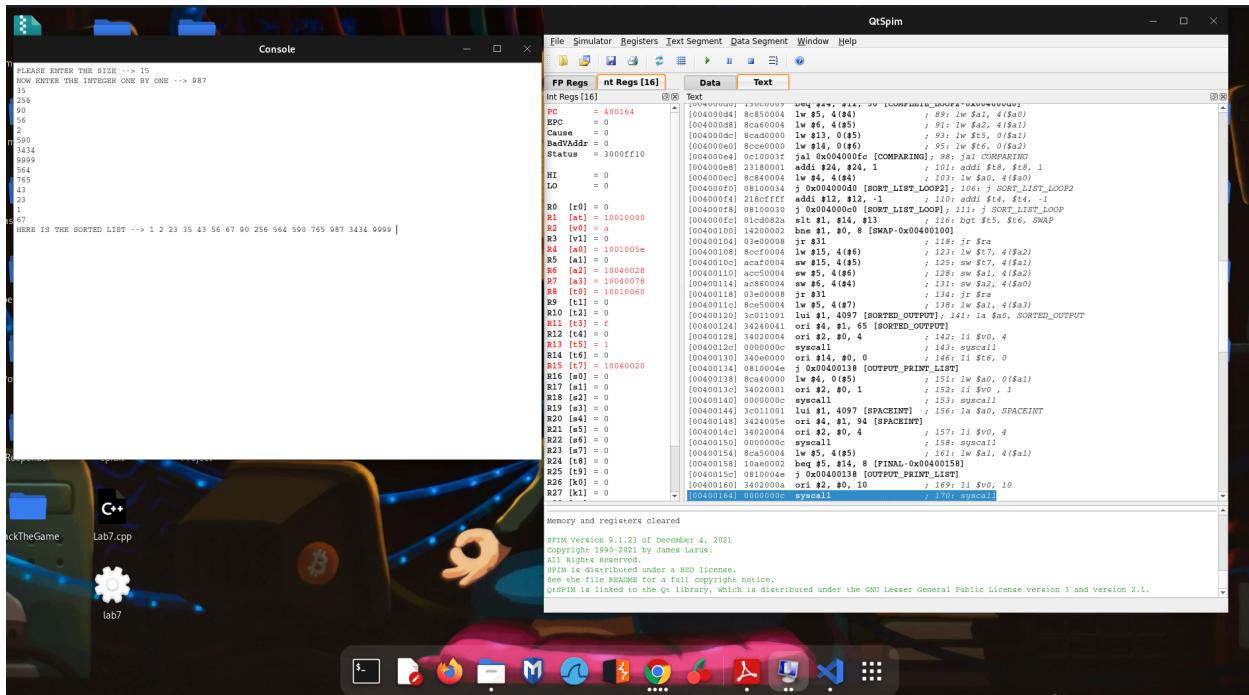
```

Memory and registers cleared

SPIM Version 0.1.23 of December 4, 2021
Copyright 1990-2021 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.

2) Create a linked list of integers entered by user. Sort the list and print the same. Take user input initially for the total number of nodes.

Output -



The screenshot shows the QtSpim debugger interface. The assembly code window displays the assembly code for a sorting program. The code uses a bubble sort algorithm to sort a linked list of integers. The user input size is 15, and the list contains integers from 1 to 987. The sorted output is printed at the end. The assembly code includes comments for each instruction, such as [COMPARING], [SWAP], and [SORTED_OUTPUT]. The registers window shows the state of various寄存器 (Registers) and memory locations. The text window shows the assembly code and its corresponding comments. The status bar at the bottom provides copyright information for QtSpim version 0.1.23.

```

PLEASE ENTER THE SIZE --> 15
NOW ENTER THE INTEGER ONE BY ONE --> 987
15
256
90
56
2
590
3434
9393
564
765
43
23
1
97
HERE IS THE SORTED LIST --> 1 2 23 35 43 56 67 90 256 564 590 765 987 3434 9999 |

```

FP Regs	Int Regs [16]	Data	Text
PC = 400164	Text		lwc000000 12VUVUVUVW LMW \$t0, \$t1, \$t2, 20 [SWAPPING_SORTED_VALUES] VUVUVUVUV
BC = 0	[00400004] 8c850004 lw \$t5, 4(\$t0)		; \$t1; lw \$t2, 4(\$t0)
Cause = 0	[00400005] 8c860004 lw \$t6, 4(\$t5)		; \$t3; lw \$t2, 4(\$t5)
RsxDVAdr = 0	[00400006] 8c870004 lw \$t7, 4(\$t6)		; \$t4; lw \$t2, 4(\$t6)
Status = 3000ff10	[00400004] 0c100003 jal 0x004000f6c [COMPARISON]; 98; jal COMPARISON		; \$t5; lw \$t2, 4(\$t7)
HI = 0	[00400008] 23180004 addi \$t24, \$t24, 1		; 101; addi \$t8, \$t8, 1
LO = 0	[00400009] 8c880004 lw \$t8, 4(\$t1)		; 102; lw \$t2, 4(\$t1)
R0 [t0] = 0	[00400004] 218cfffc addi \$t12, \$t12, -1		; 110; addi \$t4, \$t4, -1
R1 [t1] = 10010000	[00400008] 08100030 j 0x0040000 [BORT_LIST_LOOP]; 111; j SORT_LIST_LOOP		; 111; addi \$t5, \$t5, 8 [SWAP]
R2 [t2] = a	[0040000c] 01200030 swt \$t1, \$t4, 4(\$t2)		; 112; swt \$t5, 8 [SWAP]
R3 [t3] = 10010006	[0040000d] 02000030 lwt \$t1, \$t4, 4(\$t3)		; 113; lwt \$t5, 8 [SWAP]
R4 [t4] = 10010056	[00400008] 08e00004 lr \$t31		; 118; lr \$t5
R5 [t5] = 10010056	[00400008] ac8c0004 lw \$t15, 4(\$t6)		; 122; lw \$t7, 4(\$t2)
R6 [t6] = 10040028	[00400010] ac8d0004 lw \$t16, 4(\$t5)		; 123; lw \$t8, 4(\$t1)
R7 [t7] = 10040078	[00400004] 8c850004 sw \$t5, 4(\$t6)		; 128; sw \$t5, 4(\$t2)
R8 [t8] = 10010060	[00400014] ac860004 sw \$t6, 4(\$t4)		; 131; sw \$t2, 4(\$t0)
R9 [t9] = 10010060	[00400018] 03e00008 jr \$t3		; 134; jr \$t5
R10 [t10] = 0	[0040001c] 8c870004 lw \$t1, 4(\$t7)		; 138; lw \$t1, 4(\$t3)
R11 [t11] = f	[00400024] 34240091 lui \$t1, 4097 [SORTED_OUTPUT]; 141; la \$t5, SORTED_OUTPUT		; 142; la \$t5, SORTED_OUTPUT
R12 [t12] = 14	[00400128] 34200004 ori \$t2, \$t0, 4		; 142; li \$t0, 4
R13 [t13] = 1	[0040012c] 08000004 syscall		; 147; syscall
R14 [t14] = 0	[00400130] 08000000 sysret		; 147; li \$t6, 0
R15 [t15] = 10040020	[00400134] 08100040 j 0x00400138 [OUTPUT_PRINT_LIST]		; 151; lw \$t0, 0 (\$t1)
R16 [t16] = 0	[00400138] 8c840000 lw \$t4, 0(\$t5)		; 151; lw \$t0, 0 (\$t1)
R17 [t17] = 0	[0040013c] 34200001 ori \$t2, \$t0, 1		; 152; li \$t0, 1
R18 [t18] = 0	[00400140] 34200000 lwt \$t1, \$t4, 4(\$t2)		; 152; lwt \$t0, 1
R19 [t19] = 0	[00400144] 3c011001 lui \$t1, 4097 [SPACEINT]; 156; la \$t0, SPACEINT		; 156; la \$t0, SPACEINT
R20 [t20] = 4	[00400148] 34240094 ori \$t2, \$t0, 4		; 157; li \$t0, 4
R21 [t21] = 0	[0040014c] 34200000 ori \$t2, \$t0, 4		; 158; syscall
R22 [t22] = 0	[00400150] 08000004 sysret		; 161; lw \$t1, 4(\$t1)
R23 [t23] = 0	[00400154] 8c850004 lw \$t5, 4(\$t5)		; 161; lw \$t1, 4(\$t1)
R24 [t24] = 0	[00400158] 10ae0002 beq \$t5, \$t4, 8 [FINAL-0x00400158]		; 165; li \$v0, 10
R25 [t25] = 0	[0040015c] 08100040 j 0x00400138 [OUTPUT_PRINT_LIST]		; 165; li \$v0, 10
R26 [t26] = 0	[00400160] 34200004 ori \$t2, \$t0, 10		; 165; li \$v0, 10
R27 [t27] = 0	[00400164] 08000004 sysret		; 165; li \$v0, 10

Memory and registers cleared
QtSpim version 0.1.23 of December 4, 2021
Copyright 1990-2021 by James Larus.
All Rights Reserved.
QtSpim 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.

A screenshot of the QtSpim debugger interface. The assembly code window displays the following assembly code for a sorting algorithm:PLEASE ENTER THE SIZE --> 5
NOW ENTER THE INTEGER ONE BY ONE --> 55
76
56
34
HERE IS THE SORTED LIST --> 34 55 65 76 99

FP Regs nt Regs [16]
Int Regs [16]
PC = 400164
BC = 0
Cause = 0
BaseVAaddr = 0
Status = 3800ff10

Data Text
Text
004000d1 14200000 lw \$t0, 4(\$4) ; 1-89: lw \$t0, 4(\$4)
004000d8 8ca00004 lw \$t0, 4(\$5) ; 9: 1JW \$t0, 4(\$5)
004000dc 8cad0000 lw \$t1, 0(\$5) ; 93: lw \$t1, 0(\$5)
004000e0 8cc00004 lw \$t1, 0(\$6) ; 95: lw \$t1, 0(\$6)
004000f0 8c000004 lw \$t1, 0(\$7) ; 97: lw \$t1, 0(\$7)
004000f1 8c000004 lw \$t1, 0(\$8) ; 99: lw \$t1, 0(\$8)
004000f8 23180001 addi \$t2, \$t4, 24, 1 ; 101: addi \$t2, \$t4, 24, 1
004000fc 8c840001 lw \$t4, 4(\$4) ; 103: lw \$t4, 4(\$4)
004000f0 08100034 0x00000000 ; [SORT_LIST_LOOP2], 106: j SORT_LIST_LOOP2
004000f1 01cd082a alr \$t1, \$t4, -1 ; 107: alr \$t1, \$t4, -1
004000f8 08100030 j 0x004000c0 ; [SORT_LIST_LOOP], 111: j SORT_LIST_LOOP
004000fc 01cd082a alr \$t1, \$t4, #13 ; 116: bgt \$t5, \$t6, SWAP
00400100 8c000004 lw \$t1, 4(\$4) ; 117: lw \$t1, 4(\$4)
00400101 8c000004 lw \$t1, 4(\$5) ; 118: lw \$t1, 4(\$5)
00400102 14200000 bne \$t1, \$t0, 8 ; [SWAP-0x00400100]
00400103 8c000004 lw \$t1, 4(\$6) ; 119: lw \$t1, 4(\$6)
00400104 8c000004 lw \$t1, 4(\$7) ; 120: lw \$t1, 4(\$7)
00400105 8c000004 lw \$t1, 4(\$8) ; 121: lw \$t1, 4(\$8)
00400106 8c000004 lw \$t1, 4(\$9) ; 122: lw \$t1, 4(\$9)
00400107 8c000004 lw \$t1, 4(\$10) ; 123: lw \$t1, 4(\$10)
00400108 8c000004 lw \$t1, 4(\$11) ; 124: lw \$t1, 4(\$11)
00400109 8c000004 lw \$t1, 4(\$12) ; 125: lw \$t1, 4(\$12)
00400110 acf00000 sw \$t5, 4(\$6) ; 126: sw \$t5, 4(\$6)
00400111 acf00000 sw \$t5, 4(\$7) ; 127: sw \$t5, 4(\$7)
00400112 acf00000 sw \$t5, 4(\$8) ; 128: sw \$t5, 4(\$8)
00400113 acf00000 sw \$t5, 4(\$9) ; 129: sw \$t5, 4(\$9)
00400114 acf00000 sw \$t5, 4(\$10) ; 130: sw \$t5, 4(\$10)
00400115 acf00000 sw \$t5, 4(\$11) ; 131: sw \$t5, 4(\$11)
00400116 acf00000 sw \$t5, 4(\$12) ; 132: sw \$t5, 4(\$12)
00400117 acf00000 sw \$t5, 4(\$13) ; 133: sw \$t5, 4(\$13)
00400118 acf00000 sw \$t5, 4(\$14) ; 134: sw \$t5, 4(\$14)
00400119 acf00000 sw \$t5, 4(\$15) ; 135: sw \$t5, 4(\$15)
0040011a acf00000 sw \$t5, 4(\$16) ; 136: sw \$t5, 4(\$16)
0040011b acf00000 sw \$t5, 4(\$17) ; 137: sw \$t5, 4(\$17)
0040011c acf00000 sw \$t5, 4(\$18) ; 138: sw \$t5, 4(\$18)
0040011d acf00000 sw \$t5, 4(\$19) ; 139: sw \$t5, 4(\$19)
0040011e acf00000 sw \$t5, 4(\$20) ; 140: sw \$t5, 4(\$20)
0040011f acf00000 sw \$t5, 4(\$21) ; 141: sw \$t5, 4(\$21)
00400120 3c011001 lui \$t1, 4097 ; [SORTED_OUTPUT], 141: la \$a0, SORTED_OUTPUT
00400121 34240004 addi \$t1, \$t0, 65 ; [SORTED_OUTPUT]
00400122 34240004 addi \$t1, \$t0, 66 ; 142: li \$t0, 6
00400123 00000000 syscall ; 143: syscall
00400124 340e0000 ori \$t4, \$t0, 0 ; 146: li \$t0, 4
00400125 340e0000 ori \$t4, \$t0, 1 ; 147: li \$t0, 1
00400126 340e0000 ori \$t4, \$t0, 2 ; 148: li \$t0, 2
00400127 340e0000 ori \$t4, \$t0, 3 ; 149: li \$t0, 3
00400128 340e0000 ori \$t4, \$t0, 4 ; 150: li \$t0, 4
00400129 340e0000 ori \$t4, \$t0, 5 ; 151: li \$t0, 5
0040012a 340e0000 ori \$t4, \$t0, 6 ; 152: li \$t0, 6
0040012b 340e0000 ori \$t4, \$t0, 7 ; 153: li \$t0, 7
0040012c 340e0000 ori \$t4, \$t0, 8 ; 154: li \$t0, 8
0040012d 340e0000 ori \$t4, \$t0, 9 ; 155: li \$t0, 9
0040012e 340e0000 ori \$t4, \$t0, 10 ; 156: li \$t0, 10
0040012f 340e0000 ori \$t4, \$t0, 11 ; 157: li \$t0, 11
00400130 340e0000 ori \$t4, \$t0, 12 ; 158: li \$t0, 12
00400131 8c000004 lw \$t5, 4(\$1) ; 159: lw \$t5, 4(\$1)
00400132 8c000004 lw \$t5, 4(\$2) ; 160: lw \$t5, 4(\$2)
00400133 8c000004 lw \$t5, 4(\$3) ; 161: lw \$t5, 4(\$3)
00400134 8c000004 lw \$t5, 4(\$4) ; 162: lw \$t5, 4(\$4)
00400135 8c000004 lw \$t5, 4(\$5) ; 163: lw \$t5, 4(\$5)
00400136 8c000004 lw \$t5, 4(\$6) ; 164: lw \$t5, 4(\$6)
00400137 8c000004 lw \$t5, 4(\$7) ; 165: lw \$t5, 4(\$7)
00400138 8c000004 lw \$t5, 4(\$8) ; 166: lw \$t5, 4(\$8)
00400139 8c000004 lw \$t5, 4(\$9) ; 167: lw \$t5, 4(\$9)
0040013a 8c000004 lw \$t5, 4(\$10) ; 168: lw \$t5, 4(\$10)
0040013b 8c000004 lw \$t5, 4(\$11) ; 169: lw \$t5, 4(\$11)
0040013c 8c000004 lw \$t5, 4(\$12) ; 170: lw \$t5, 4(\$12)
0040013d 8c000004 lw \$t5, 4(\$13) ; 171: lw \$t5, 4(\$13)
0040013e 8c000004 lw \$t5, 4(\$14) ; 172: lw \$t5, 4(\$14)
0040013f 8c000004 lw \$t5, 4(\$15) ; 173: lw \$t5, 4(\$15)
00400140 8c000004 lw \$t5, 4(\$16) ; 174: lw \$t5, 4(\$16)
00400141 3c011001 lui \$t1, 6097 ; [SPACEINT], 175: la \$a0, SPACEINT
00400142 340e0000 ori \$t4, \$t1, 94 ; 176: li \$t1, 94
00400143 340e0000 ori \$t4, \$t1, 95 ; 177: li \$t1, 95
00400144 340e0000 ori \$t4, \$t1, 96 ; 178: li \$t1, 96
00400145 340e0000 ori \$t4, \$t1, 97 ; 179: li \$t1, 97
00400146 340e0000 ori \$t4, \$t1, 98 ; 180: li \$t1, 98
00400147 340e0000 ori \$t4, \$t1, 99 ; 181: li \$t1, 99
00400148 340e0000 ori \$t4, \$t1, 100 ; 182: li \$t1, 100
00400149 340e0000 ori \$t4, \$t1, 101 ; 183: li \$t1, 101
0040014a 340e0000 ori \$t4, \$t1, 102 ; 184: li \$t1, 102
0040014b 340e0000 ori \$t4, \$t1, 103 ; 185: li \$t1, 103
0040014c 340e0000 ori \$t4, \$t1, 104 ; 186: li \$t1, 104
0040014d 340e0000 ori \$t4, \$t1, 105 ; 187: li \$t1, 105
0040014e 340e0000 ori \$t4, \$t1, 106 ; 188: li \$t1, 106
0040014f 340e0000 ori \$t4, \$t1, 107 ; 189: li \$t1, 107
00400150 340e0000 ori \$t4, \$t1, 108 ; 190: li \$t1, 108
00400151 8c000004 lw \$t5, 4(\$1) ; 191: lw \$t5, 4(\$1)
00400152 8c000004 lw \$t5, 4(\$2) ; 192: lw \$t5, 4(\$2)
00400153 8c000004 lw \$t5, 4(\$3) ; 193: lw \$t5, 4(\$3)
00400154 8c000004 lw \$t5, 4(\$4) ; 194: lw \$t5, 4(\$4)
00400155 8c000004 lw \$t5, 4(\$5) ; 195: lw \$t5, 4(\$5)
00400156 8c000004 lw \$t5, 4(\$6) ; 196: lw \$t5, 4(\$6)
00400157 8c000004 lw \$t5, 4(\$7) ; 197: lw \$t5, 4(\$7)
00400158 8c000004 lw \$t5, 4(\$8) ; 198: lw \$t5, 4(\$8)
00400159 8c000004 lw \$t5, 4(\$9) ; 199: lw \$t5, 4(\$9)
0040015a 8c000004 lw \$t5, 4(\$10) ; 200: lw \$t5, 4(\$10)
0040015b 8c000004 lw \$t5, 4(\$11) ; 201: lw \$t5, 4(\$11)
0040015c 8c000004 lw \$t5, 4(\$12) ; 202: lw \$t5, 4(\$12)
0040015d 8c000004 lw \$t5, 4(\$13) ; 203: lw \$t5, 4(\$13)
0040015e 8c000004 lw \$t5, 4(\$14) ; 204: lw \$t5, 4(\$14)
0040015f 8c000004 lw \$t5, 4(\$15) ; 205: lw \$t5, 4(\$15)
00400160 34020000 ori \$t2, \$t0, 10 ; 206: li \$t0, 10
00400161 34020000 ori \$t2, \$t0, 11 ; 207: li \$t0, 11
00400162 34020000 ori \$t2, \$t0, 12 ; 208: li \$t0, 12
00400163 34020000 ori \$t2, \$t0, 13 ; 209: li \$t0, 13
00400164 34020000 ori \$t2, \$t0, 14 ; 210: li \$t0, 14
00400165 34020000 ori \$t2, \$t0, 15 ; 211: li \$t0, 15
00400166 34020000 ori \$t2, \$t0, 16 ; 212: li \$t0, 16
00400167 34020000 ori \$t2, \$t0, 17 ; 213: li \$t0, 17
00400168 34020000 ori \$t2, \$t0, 18 ; 214: li \$t0, 18
00400169 34020000 ori \$t2, \$t0, 19 ; 215: li \$t0, 19
0040016a 34020000 ori \$t2, \$t0, 20 ; 216: li \$t0, 20
0040016b 34020000 ori \$t2, \$t0, 21 ; 217: li \$t0, 21
0040016c 34020000 ori \$t2, \$t0, 22 ; 218: li \$t0, 22
0040016d 34020000 ori \$t2, \$t0, 23 ; 219: li \$t0, 23
0040016e 34020000 ori \$t2, \$t0, 24 ; 220: li \$t0, 24
0040016f 34020000 ori \$t2, \$t0, 25 ; 221: li \$t0, 25
00400170 34020000 ori \$t2, \$t0, 26 ; 222: li \$t0, 26
00400171 34020000 ori \$t2, \$t0, 27 ; 223: li \$t0, 27
00400172 34020000 ori \$t2, \$t0, 28 ; 224: li \$t0, 28
00400173 34020000 ori \$t2, \$t0, 29 ; 225: li \$t0, 29
00400174 34020000 ori \$t2, \$t0, 30 ; 226: li \$t0, 30
00400175 34020000 ori \$t2, \$t0, 31 ; 227: li \$t0, 31
00400176 34020000 ori \$t2, \$t0, 32 ; 228: li \$t0, 32
00400177 34020000 ori \$t2, \$t0, 33 ; 229: li \$t0, 33
00400178 34020000 ori \$t2, \$t0, 34 ; 230: li \$t0, 34
00400179 34020000 ori \$t2, \$t0, 35 ; 231: li \$t0, 35
0040017a 34020000 ori \$t2, \$t0, 36 ; 232: li \$t0, 36
0040017b 34020000 ori \$t2, \$t0, 37 ; 233: li \$t0, 37
0040017c 34020000 ori \$t2, \$t0, 38 ; 234: li \$t0, 38
0040017d 34020000 ori \$t2, \$t0, 39 ; 235: li \$t0, 39
0040017e 34020000 ori \$t2, \$t0, 40 ; 236: li \$t0, 40
0040017f 34020000 ori \$t2, \$t0, 41 ; 237: li \$t0, 41
00400180 34020000 ori \$t2, \$t0, 42 ; 238: li \$t0, 42
00400181 34020000 ori \$t2, \$t0, 43 ; 239: li \$t0, 43
00400182 34020000 ori \$t2, \$t0, 44 ; 240: li \$t0, 44
00400183 34020000 ori \$t2, \$t0, 45 ; 241: li \$t0, 45
00400184 34020000 ori \$t2, \$t0, 46 ; 242: li \$t0, 46
00400185 34020000 ori \$t2, \$t0, 47 ; 243: li \$t0, 47
00400186 34020000 ori \$t2, \$t0, 48 ; 244: li \$t0, 48
00400187 34020000 ori \$t2, \$t0, 49 ; 245: li \$t0, 49
00400188 34020000 ori \$t2, \$t0, 50 ; 246: li \$t0, 50
00400189 34020000 ori \$t2, \$t0, 51 ; 247: li \$t0, 51
0040018a 34020000 ori \$t2, \$t0, 52 ; 248: li \$t0, 52
0040018b 34020000 ori \$t2, \$t0, 53 ; 249: li \$t0, 53
0040018c 34020000 ori \$t2, \$t0, 54 ; 250: li \$t0, 54
0040018d 34020000 ori \$t2, \$t0, 55 ; 251: li \$t0, 55
0040018e 34020000 ori \$t2, \$t0, 56 ; 252: li \$t0, 56
0040018f 34020000 ori \$t2, \$t0, 57 ; 253: li \$t0, 57
00400190 34020000 ori \$t2, \$t0, 58 ; 254: li \$t0, 58
00400191 34020000 ori \$t2, \$t0, 59 ; 255: li \$t0, 59
00400192 34020000 ori \$t2, \$t0, 60 ; 256: li \$t0, 60
00400193 34020000 ori \$t2, \$t0, 61 ; 257: li \$t0, 61
00400194 34020000 ori \$t2, \$t0, 62 ; 258: li \$t0, 62
00400195 34020000 ori \$t2, \$t0, 63 ; 259: li \$t0, 63
00400196 34020000 ori \$t2, \$t0, 64 ; 260: li \$t0, 64
00400197 34020000 ori \$t2, \$t0, 65 ; 261: li \$t0, 65
00400198 34020000 ori \$t2, \$t0, 66 ; 262: li \$t0, 66
00400199 34020000 ori \$t2, \$t0, 67 ; 263: li \$t0, 67
0040019a 34020000 ori \$t2, \$t0, 68 ; 264: li \$t0, 68
0040019b 34020000 ori \$t2, \$t0, 69 ; 265: li \$t0, 69
0040019c 34020000 ori \$t2, \$t0, 70 ; 266: li \$t0, 70
0040019d 34020000 ori \$t2, \$t0, 71 ; 267: li \$t0, 71
0040019e 34020000 ori \$t2, \$t0, 72 ; 268: li \$t0, 72
0040019f 34020000 ori \$t2, \$t0, 73 ; 269: li \$t0, 73
004001a0 34020000 ori \$t2, \$t0, 74 ; 270: li \$t0, 74
004001a1 34020000 ori \$t2, \$t0, 75 ; 271: li \$t0, 75
004001a2 34020000 ori \$t2, \$t0, 76 ; 272: li \$t0, 76
004001a3 34020000 ori \$t2, \$t0, 77 ; 273: li \$t0, 77
004001a4 34020000 ori \$t2, \$t0, 78 ; 274: li \$t0, 78
004001a5 34020000 ori \$t2, \$t0, 79 ; 275: li \$t0, 79
004001a6 34020000 ori \$t2, \$t0, 80 ; 276: li \$t0, 80
004001a7 34020000 ori \$t2, \$t0, 81 ; 277: li \$t0, 81
004001a8 34020000 ori \$t2, \$t0, 82 ; 278: li \$t0, 82
004001a9 34020000 ori \$t2, \$t0, 83 ; 279: li \$t0, 83
004001aa 34020000 ori \$t2, \$t0, 84 ; 280: li \$t0, 84
004001ab 34020000 ori \$t2, \$t0, 85 ; 281: li \$t0, 85
004001ac 34020000 ori \$t2, \$t0, 86 ; 282: li \$t0, 86
004001ad 34020000 ori \$t2, \$t0, 87 ; 283: li \$t0, 87
004001ae 34020000 ori \$t2, \$t0, 88 ; 284: li \$t0, 88
004001af 34020000 ori \$t2, \$t0, 89 ; 285: li \$t0, 89
004001b0 34020000 ori \$t2, \$t0, 90 ; 286: li \$t0, 90
004001b1 34020000 ori \$t2, \$t0, 91 ; 287: li \$t0, 91
004001b2 34020000 ori \$t2, \$t0, 92 ; 288: li \$t0, 92
004001b3 34020000 ori \$t2, \$t0, 93 ; 289: li \$t0, 93
004001b4 34020000 ori \$t2, \$t0, 94 ; 290: li \$t0, 94
004001b5 34020000 ori \$t2, \$t0, 95 ; 291: li \$t0, 95
004001b6 34020000 ori \$t2, \$t0, 96 ; 292: li \$t0, 96
004001b7 34020000 ori \$t2, \$t0, 97 ; 293: li \$t0, 97
004001b8 34020000 ori \$t2, \$t0, 98 ; 294: li \$t0, 98
004001b9 34020000 ori \$t2, \$t0, 99 ; 295: li \$t0, 99
004001ba 34020000 ori \$t2, \$t0, 100 ; 296: li \$t0, 100
004001bb 34020000 ori \$t2, \$t0, 101 ; 297: li \$t0, 101
004001bc 34020000 ori \$t2, \$t0, 102 ; 298: li \$t0, 102
004001bd 34020000 ori \$t2, \$t0, 103 ; 299: li \$t0, 103
004001be 34020000 ori \$t2, \$t0, 104 ; 300: li \$t0, 104
004001bf 34020000 ori \$t2, \$t0, 105 ; 301: li \$t0, 105
004001c0 34020000 ori \$t2, \$t0, 106 ; 302: li \$t0, 106
004001c1 34020000 ori \$t2, \$t0, 107 ; 303: li \$t0, 107
004001c2 34020000 ori \$t2, \$t0, 108 ; 304: li \$t0, 108
004001c3 34020000 ori \$t2, \$t0, 109 ; 305: li \$t0, 109
004001c4 34020000 ori \$t2, \$t0, 110 ; 306: li \$t0, 110
004001c5 34020000 ori \$t2, \$t0, 111 ; 307: li \$t0, 111
004001c6 34020000 ori \$t2, \$t0, 112 ; 308: li \$t0, 112
004001c7 34020000 ori \$t2, \$t0, 113 ; 309: li \$t0, 113
004001c8 34020000 ori \$t2, \$t0, 114 ; 310: li \$t0, 114
004001c9 34020000 ori \$t2, \$t0, 115 ; 311: li \$t0, 115
004001ca 34020000 ori \$t2, \$t0, 116 ; 312: li \$t0, 116
004001cb 34020000 ori \$t2, \$t0, 117 ; 313: li \$t0, 117
004001cc 34020000 ori \$t2, \$t0, 118 ; 314: li \$t0, 118
004001cd 34020000 ori \$t2, \$t0, 119 ; 315: li \$t0, 119
004001ce 34020000 ori \$t2, \$t0, 120 ; 316: li \$t0, 120
004001cf 34020000 ori \$t2, \$t0, 121 ; 317: li \$t0, 121
004001d0 34020000 ori \$t2, \$t0, 122 ; 318: li \$t0, 122
004001d1 34020000 ori \$t2, \$t0, 123 ; 319: li \$t0, 123
004001d2 34020000 ori \$t2, \$t0, 124 ; 320: li \$t0, 124
004001d3 34020000 ori \$t2, \$t0, 125 ; 321: li \$t0, 125
004001d4 34020000 ori \$t2, \$t0, 126 ; 322: li \$t0, 126
004001d5 34020000 ori \$t2, \$t0, 127 ; 323: li \$t0, 127
004001d6 34020000 ori \$t2, \$t0, 128 ; 324: li \$t0, 128
004001d7 34020000 ori \$t2, \$t0, 129 ; 325: li \$t0, 129
004001d8 34020000 ori \$t2, \$t0, 130 ; 326: li \$t0, 130
004001d9 34020000 ori \$t2, \$t0, 131 ; 327: li \$t0, 131
004001da 34020000 ori \$t2, \$t0, 132 ; 328: li \$t0, 132
004001db 34020000 ori \$t2, \$t0, 133 ; 329: li \$t0, 133
004001dc 34020000 ori \$t2, \$t0, 134 ; 330: li \$t0, 134
004001dd 34020000 ori \$t2, \$t0, 135 ; 331: li \$t0, 135
004001de 34020000 ori \$t2, \$t0, 136 ; 332: li \$t0, 136
004001df 34020000 ori \$t2, \$t0, 137 ; 333: li \$t0, 137
004001e0 34020000 ori \$t2, \$t0, 138 ; 334: li \$t0, 138
004001e1 34020000 ori \$t2, \$t0, 139 ; 335: li \$t0, 139
004001e2 34020000 ori \$t2, \$t0, 140 ; 336: li \$t0, 140
004001e3 34020000 ori \$t2, \$t0, 141 ; 337: li \$t0, 141
004001e4 34020000 ori \$t2, \$t0, 142 ; 338: li \$t0, 142
004001e5 34020000 ori \$t2, \$t0, 143 ; 339: li \$t0, 143
004001e6 34020000 ori \$t2, \$t0, 144 ; 340: li \$t0, 144
004001e7 34020000 ori \$t2, \$t0, 145 ; 341: li \$t0, 145
004001e8 34020000 ori \$t2, \$t0, 146 ; 342: li \$t0, 146
004001e9 34020000 ori \$t2, \$t0, 147 ; 343: li \$t0, 147
004001ea 34020000 ori \$t2, \$t0, 148 ; 344: li \$t0, 148
004001eb 34020000 ori \$t2, \$t0, 149 ; 345: li \$t0, 149
004001ec 34020000 ori \$t2, \$t0, 150 ; 346: li \$t0, 150
004001ed 34020000 ori \$t2, \$t0, 151 ; 347: li \$t0, 151
004001ef 34020000 ori \$t2, \$t0, 152 ; 348: li \$t0, 152
004001f0 34020000 ori \$t2, \$t0, 153 ; 349: li \$t0, 153
004001f1 34020000 ori \$t2, \$t0, 154 ; 350: li \$t0, 154
004001f2 34020000 ori \$t2, \$t0, 155 ; 351: li \$t0, 155
004001f3 34020000 ori \$t2, \$t0, 156 ; 352: li \$t0, 156
004001f4 34020000 ori \$t2, \$t0, 157 ; 353: li \$t0, 157
004001f5 34020000 ori \$t2, \$t0, 158 ; 354: li \$t0, 158
004001f6 34020000 ori \$t2, \$t0, 159 ; 355: li \$t0, 159
004001f7 34020000 ori \$t2, \$t0, 160 ; 356: li \$t0, 160
004001f8 34020000 ori \$t2, \$t0, 161 ; 357: li \$t0, 161
004001f9 34020000 ori \$t2, \$t0, 162 ; 358: li \$t0, 162
004001fa 34020000 ori \$t2, \$t0, 163 ; 359: li \$t0, 163
004001fb 34020000 ori \$t2, \$t0, 164 ; 360: li \$t0, 164
004001fc 34020000 ori \$t2, \$t0, 165 ; 361: li \$t0, 165
004001fd 34020000 ori \$t2, \$t0, 166 ; 362: li \$t0, 166
004001fe 34020000 ori \$t2, \$t0, 167 ; 363: li \$t0, 167
004001ff 34020000 ori \$t2, \$t0, 168 ; 364: li \$t0, 168
00400100 34020000 ori \$t2, \$t0, 169 ; 365: li \$t0, 169
00400101 34020000 ori \$t2, \$t0, 170 ; 366: li \$t0, 170
00400102 34020000 ori \$t2, \$t0, 171 ; 367: li \$t0, 171
00400103 34020000 ori \$t2, \$t0, 172 ; 368: li \$t0, 172
00400104 34020000 ori \$t2, \$t0, 173 ; 369: li \$t0, 173
00400105 34020000 ori \$t2, \$t0, 174 ; 370: li \$t0, 174
00400106 34020000 ori \$t2, \$t0, 175

A screenshot of the QtSpim debugger interface. The assembly code window displays the following assembly code for a sorted list insertion algorithm:PLEASE ENTER THE SIZE --> 6
NOW ENTER THE INTEGER ONE BY ONE --> 10
56
3
99
8
78
HERE IS THE SORTED LIST --> 3 8 10 56 78 99

File Simulator Registers Text Segment Data Segment Window Help
FP Regs nt Regs [16]
Int Regs [16]
Text
00000000 12345678 lw \$4, -8(\$4) ;W4 \$4, -8(\$4), 34 _COMPARING..._DOOVER--VVVVVVVV
00000041 8c950004 lw \$5, 4(\$4) ;R5 \$5, 4(\$4), 35 _COMPARING..._DOOVER--VVVVVVVV
00000042 8c950005 lw \$6, 8(\$4) ;R6 \$6, 8(\$4), 36 _COMPARING..._DOOVER--VVVVVVVV
000000dc 8ce00000 lw \$13, 0(\$45) ;R13 \$13, 0(\$45), 37 _COMPARING..._DOOVER--VVVVVVVV
000000e0 8cce00000 lw \$14, 0(\$45) ;R14 \$14, 0(\$45), 38 _COMPARING..._DOOVER--VVVVVVVV
000000e1 300ff10 status = 3000ff10
000000e4 0c100037 jal 0x0040000f ;[COMPARING]; 98: jal COMPARING
000000e5 21810000 addi \$24, \$24, 1 ;addi \$24, \$24, 1
000000e6 0c100004 lw \$4, 4(\$4) ;R4 \$4, 4(\$4), 101 _SWAP--VVVVVVVV
000000f0 08100034 j 0x00400030 ;[SORT_LIST_LOOP]; 106: j SORT_LIST_LOOP
000000f4 218cffef addi \$12, \$12, -1 ;addi \$12, \$12, -1
000000f8 08100030 j 0x00400030 ;[SORT_LIST_LOOP]; 111: j SORT_LIST_LOOP
000000f9 08100030 j 0x00400030 ;[SORT_LIST_LOOP]; 111: j SORT_LIST_LOOP
00000100 14200002 bne \$1, \$0, 8 (\$NAP-0x00400100)
00000104 0e000008 jr \$31 ;jr \$31
00000108 8ccf0004 lw \$10, 4(\$4) ;R10 \$10, 4(\$4), 118: jr \$24
00000109 8c950004 lw \$11, 4(\$45) ;R11 \$11, 4(\$45), 119: jr \$24
00000110 acc000004 sw \$5, 4(\$6) ;sw \$5, 4(\$6), 123: lw \$7, 4(\$42)
00000114 ac860004 sw \$6, 4(\$4) ;sw \$6, 4(\$4), 125: lw \$8, 4(\$42)
00000118 0e000000 jr \$31 ;jr \$31, 128: sw \$4, 4(\$42)
00000120 3c011001 lui \$1, 4097 ;lui \$1, 4097
00000124 08100041 lui \$1, 4097 ;lui \$1, 4097 [SORTED_OUTPUT]; 141: lw \$29, SORTED_OUTPUT
00000124 34240001 ori \$4, \$1, 65 ;[SORTED_OUTPUT]
00000128 34240004 ori \$2, \$0, 4 ;r 142: li \$0, 4
00000130 34240004 ori \$2, \$0, 4 ;r 143: li \$0, 4
00000131 34240004 ori \$2, \$0, 4 ;r 144: li \$0, 4
00000132 34240004 ori \$2, \$0, 4 ;r 145: li \$0, 4
00000133 34240004 ori \$2, \$0, 4 ;r 146: li \$0, 0
00000134 08100040 j 0x00400138 ;[OUTPUT_PRINT_LIST]
00000138 8ca40000 lw \$4, 0(\$5) ;R4 \$4, 0(\$5), 151: lw \$0, 0(\$45)
00000139 8ca40000 lw \$4, 0(\$5) ;R4 \$4, 0(\$5), 151: lw \$0, 0(\$45)
00000140 08000000 syscall ;r 152: syscall
00000144 3c011001 lui \$1, 4097 ;lui \$1, 4097 [SPACEINT]; 156: la \$29, SPACEINT
00000148 34240004 ori \$4, \$1, 9 ;[SPACEINT]
00000150 34240004 ori \$2, \$0, 4 ;r 157: li \$0, 4
00000154 08ca50004 lw \$5, 4(\$5) ;R5 \$5, 4(\$5), 161: lw \$4, 4(\$42)
00000158 18ae0002 beq \$5, \$14, 8 (\$FINAL-0x00400158)
00000160 08100038 j 0x00400138 ;[OUTPUT_PRINT_LIST]
00000160 34230008 srl \$2, \$0, 10 ;r 169: li \$0, 10
00000164 08000000 syscall ;r 170: syscall

Memory and registers cleared

QtSpim version 9.1.23 of December 4, 2021
copyright 1990-2021 by James Larus.
All Rights Reserved.

This software is licensed under a BSL 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.

Thank You