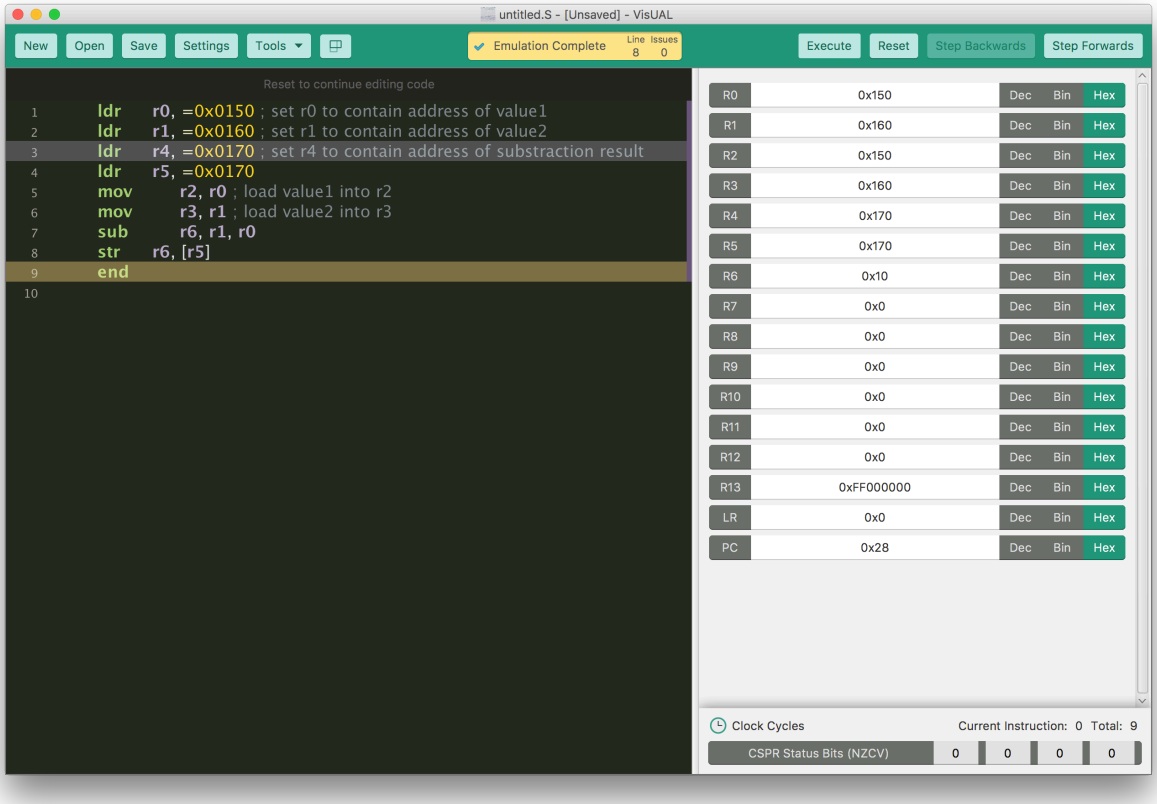
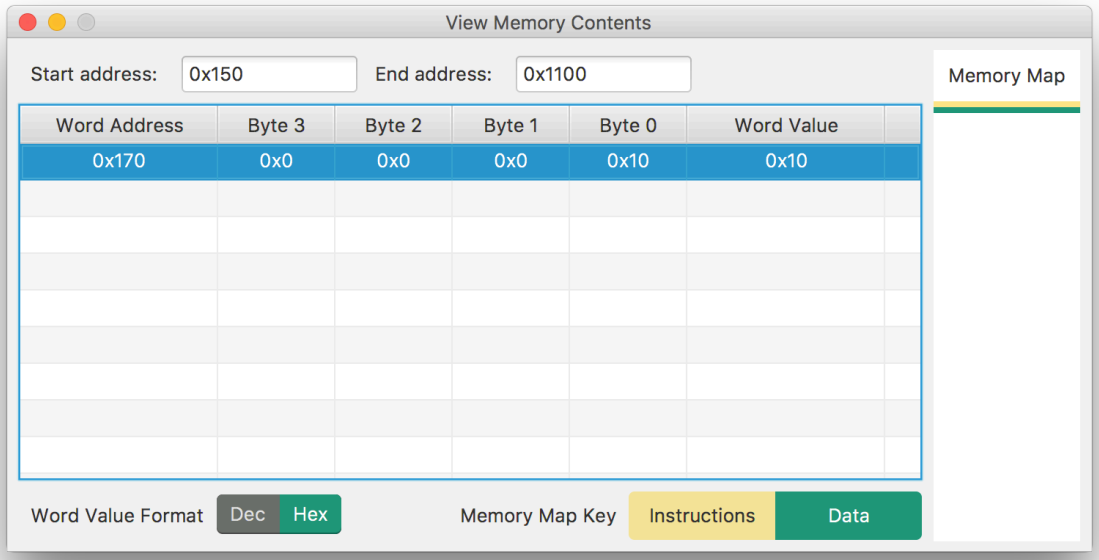
Willis Allstead
4/16/18
CS 219

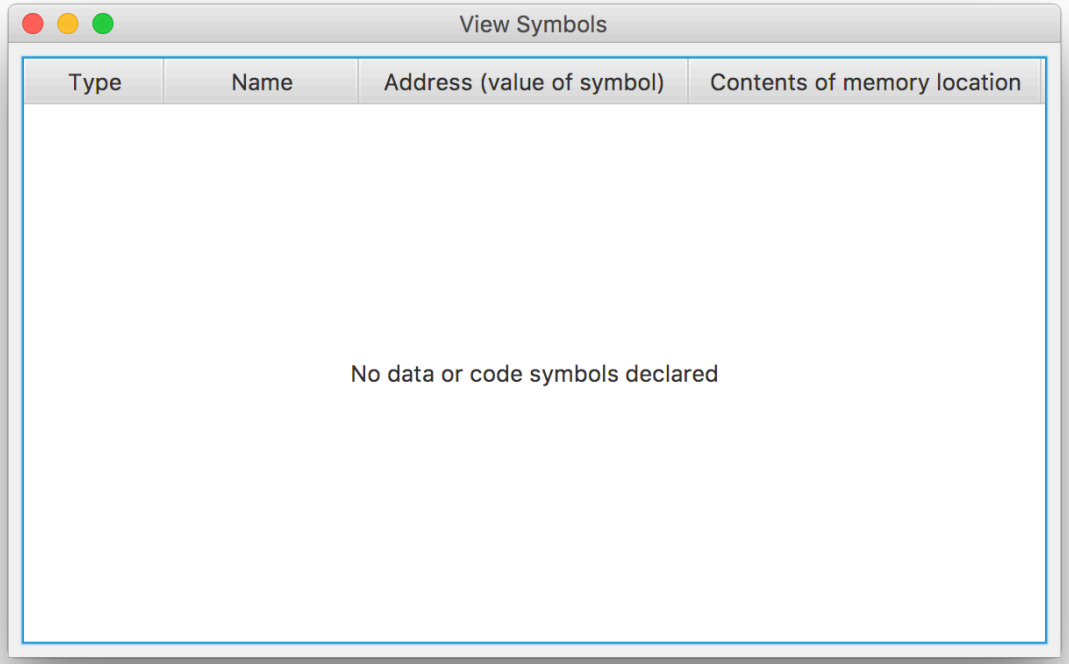
HW9

1)

- I'm trying to learn how to subtract two 64-bit integers in ARM assembly and save it to a memory location.

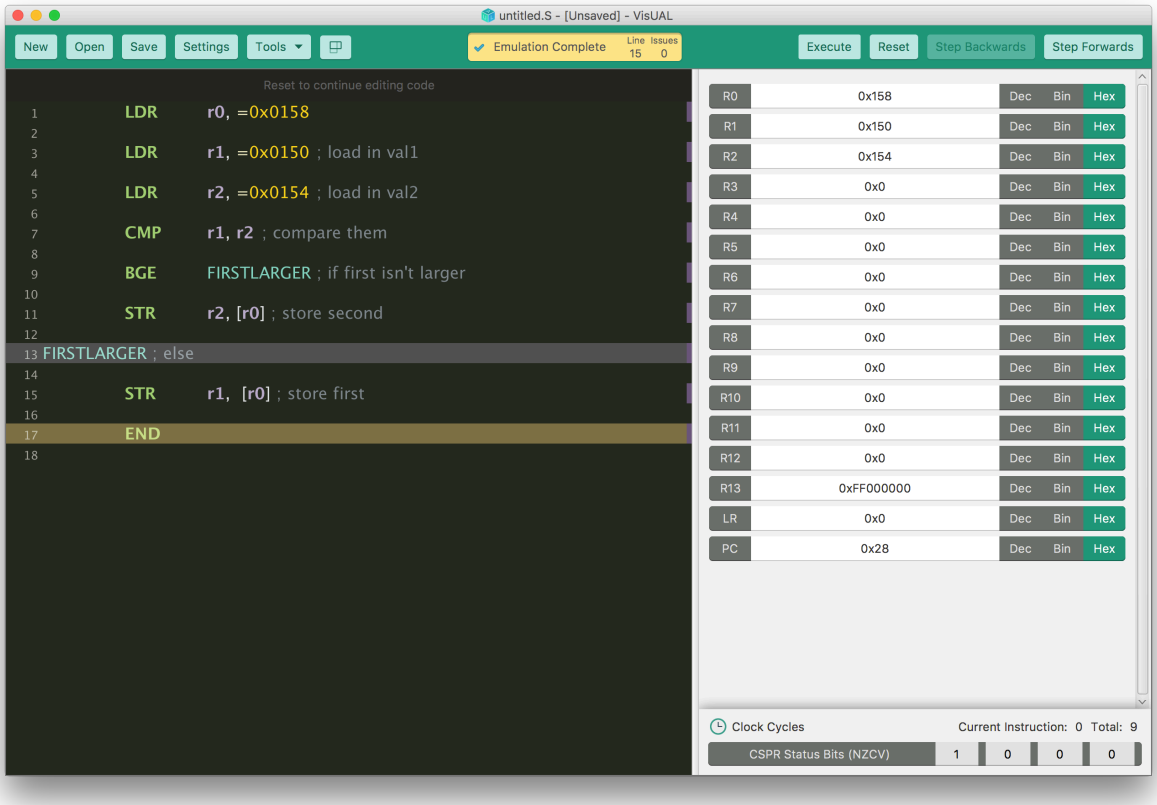


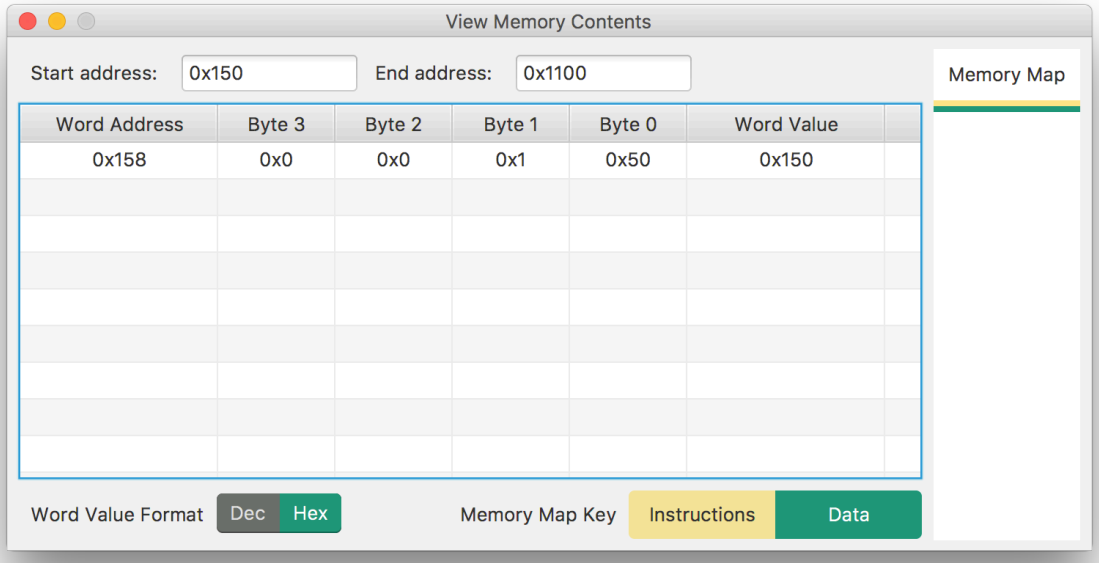

View Memory Contents
Start address:
0x150
End address:
0x1100
Memory Map
Word Address
Byte 3
Byte 2
Byte 1
Byte 0
Word Value
0x170
0x0
0x0
0x0
0x10
0x10
Word Value Format
Dec
Hex
Memory Map Key
Instructions
Data

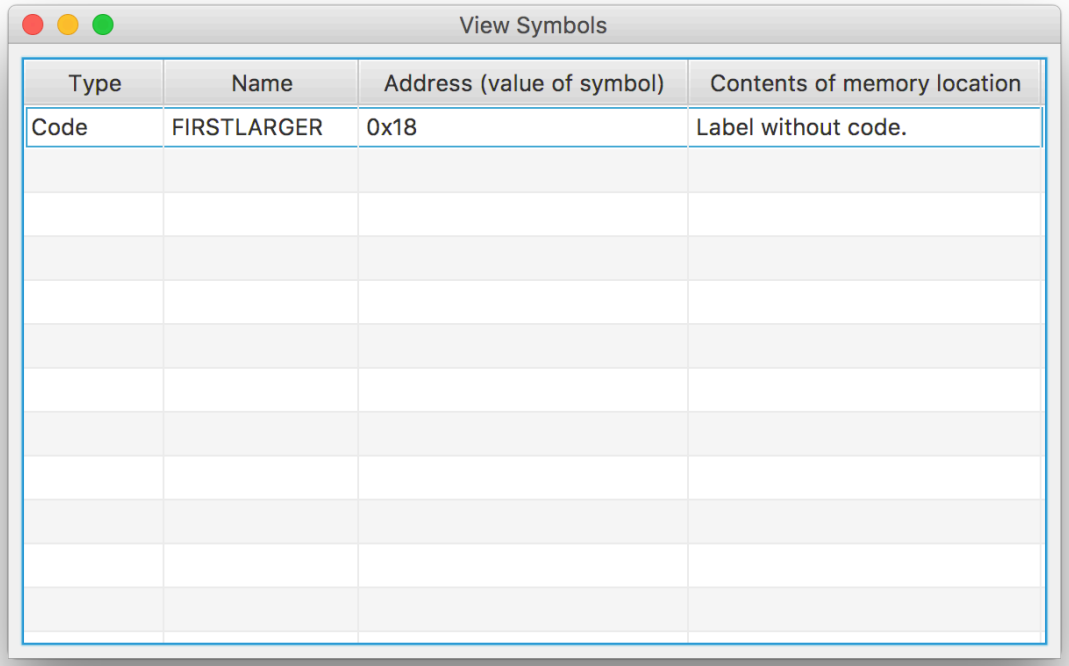

View Symbols
Type
Name
Address (value of symbol)
Contents of memory location
No data or code symbols declared

3)

- I'm trying to learn how to compare two numbers in ARM assembly.

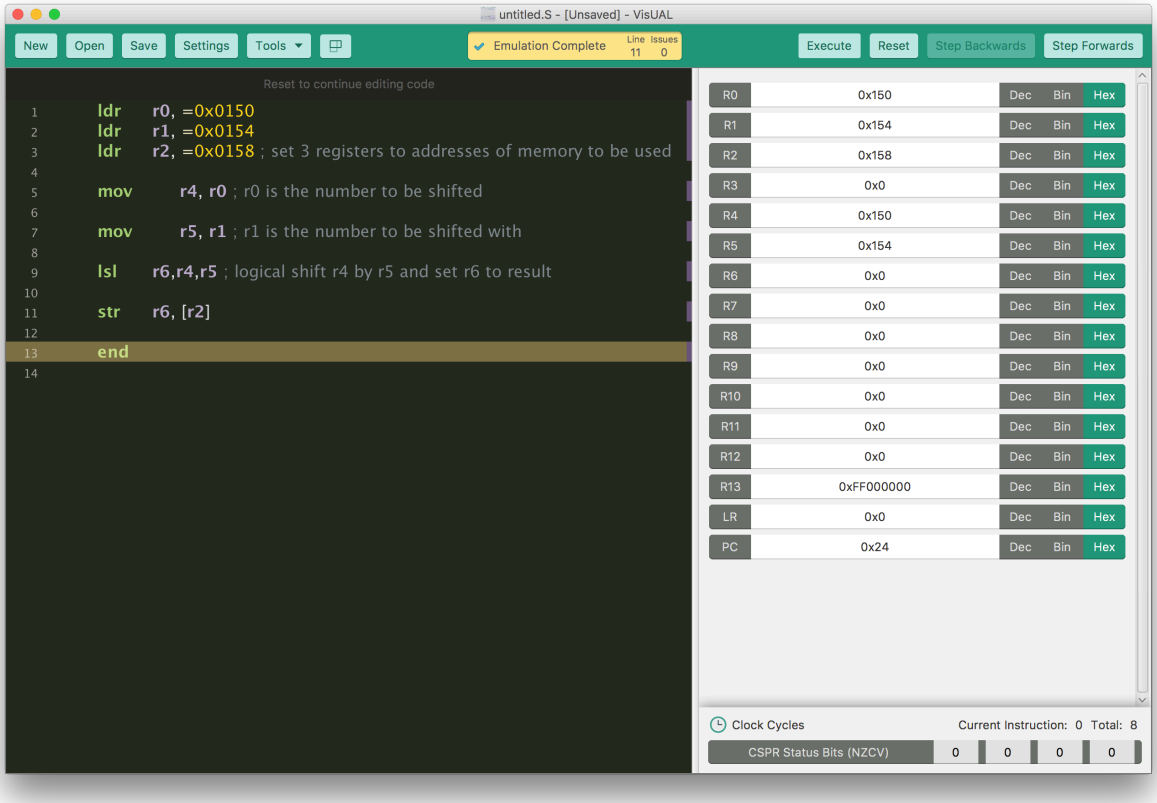


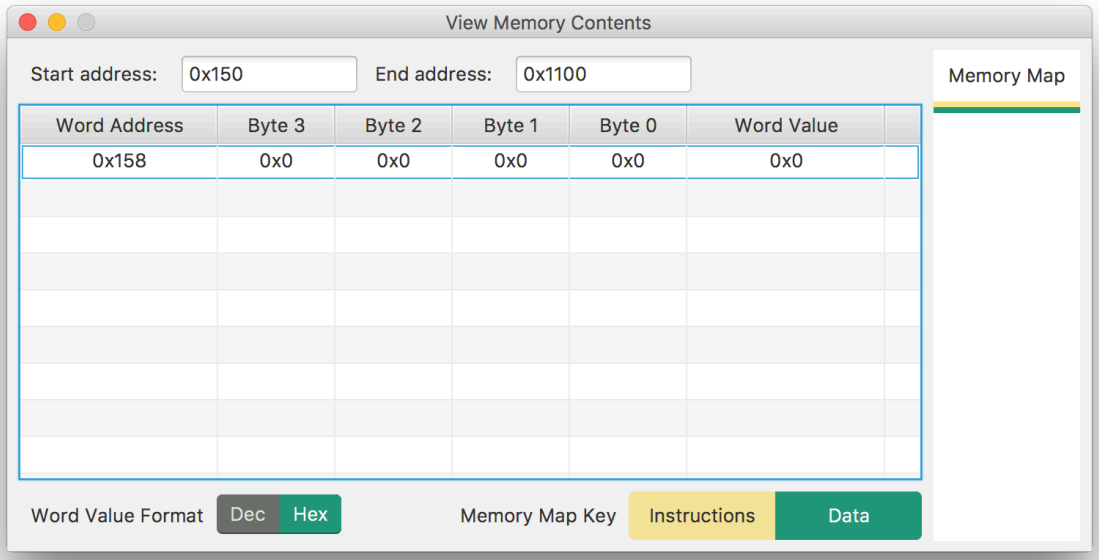


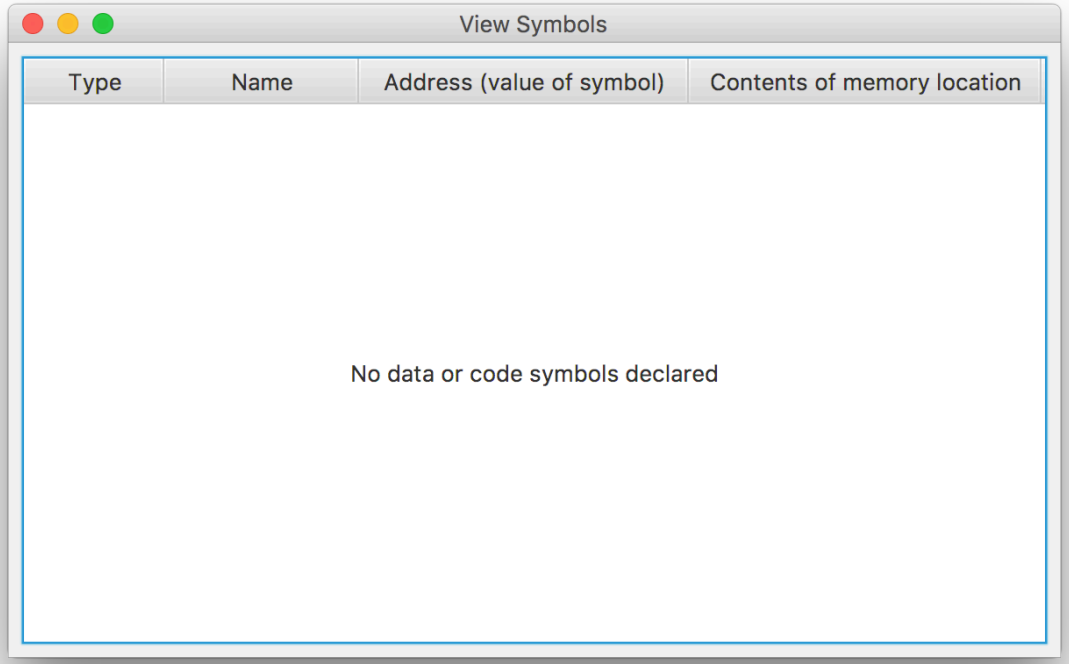

View Symbols
Type
Name
Address (value of symbol)
Contents of memory location
Code
FIRSTLARGER
0x18
Label without code.

4)

- I'm trying to learn how to logically left shift a number in ARM assembly and save it to a memory location

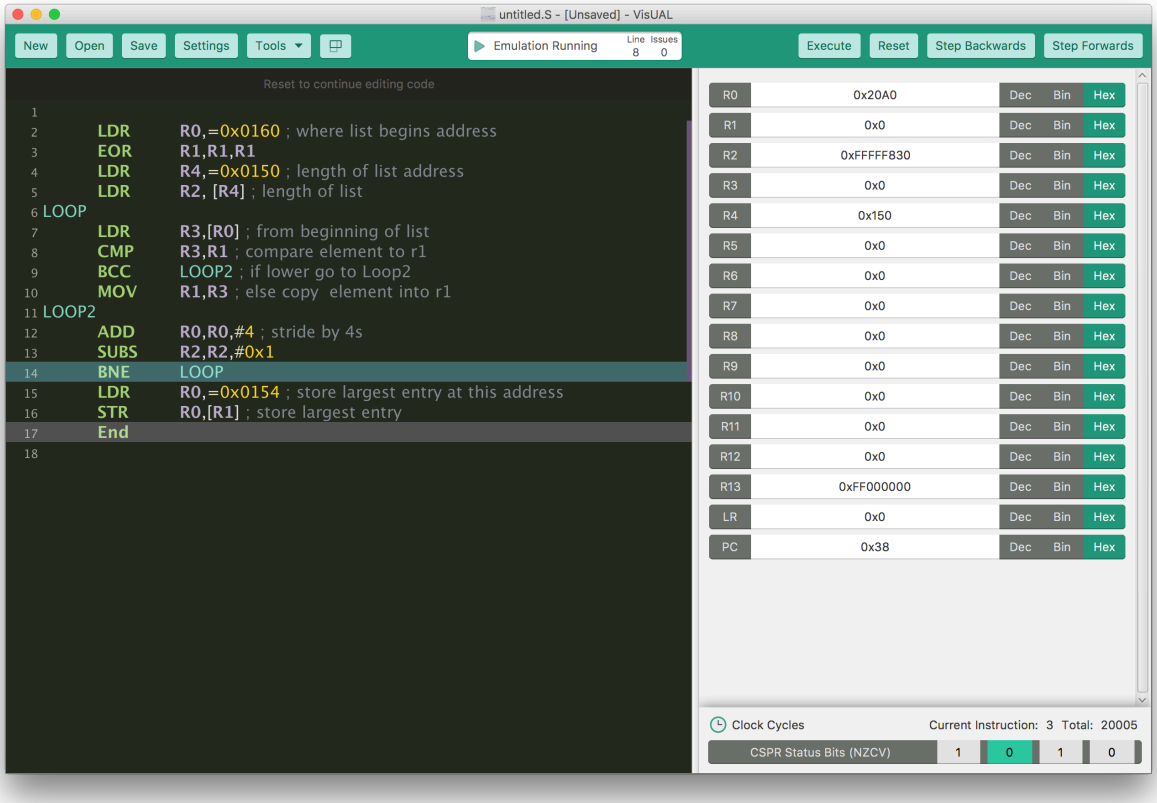


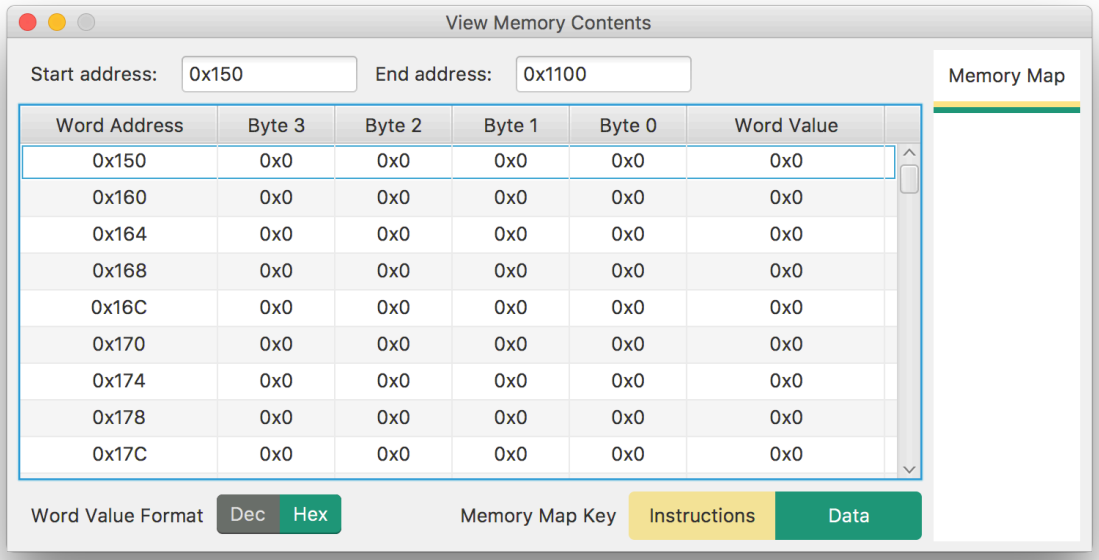

View Memory Contents
Start address:
0x150
End address:
0x1100
Memory Map
Word Address
Byte 3
Byte 2
Byte 1
Byte 0
Word Value
0x158
0x0
0x0
0x0
0x0
0x0
Word Value Format
Dec
Hex
Memory Map Key
Instructions
Data

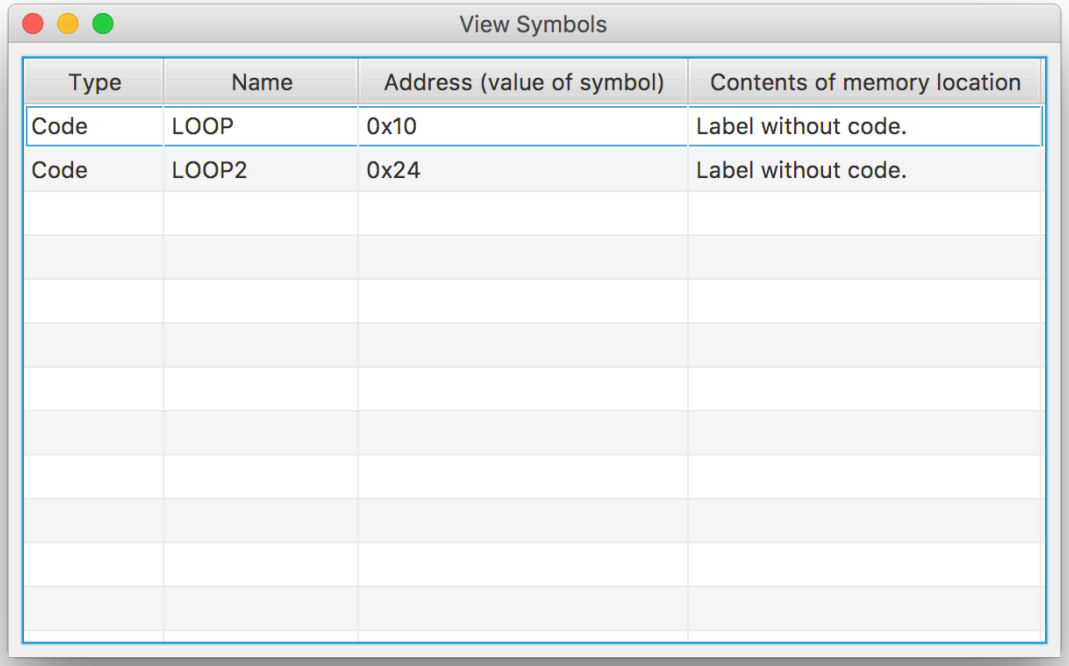

View Symbols
Type
Name
Address (value of symbol)
Contents of memory location
No data or code symbols declared

5)

- I'm trying to learn how to find the largest 32-bit word in a list, and store the largest word in a memory location.

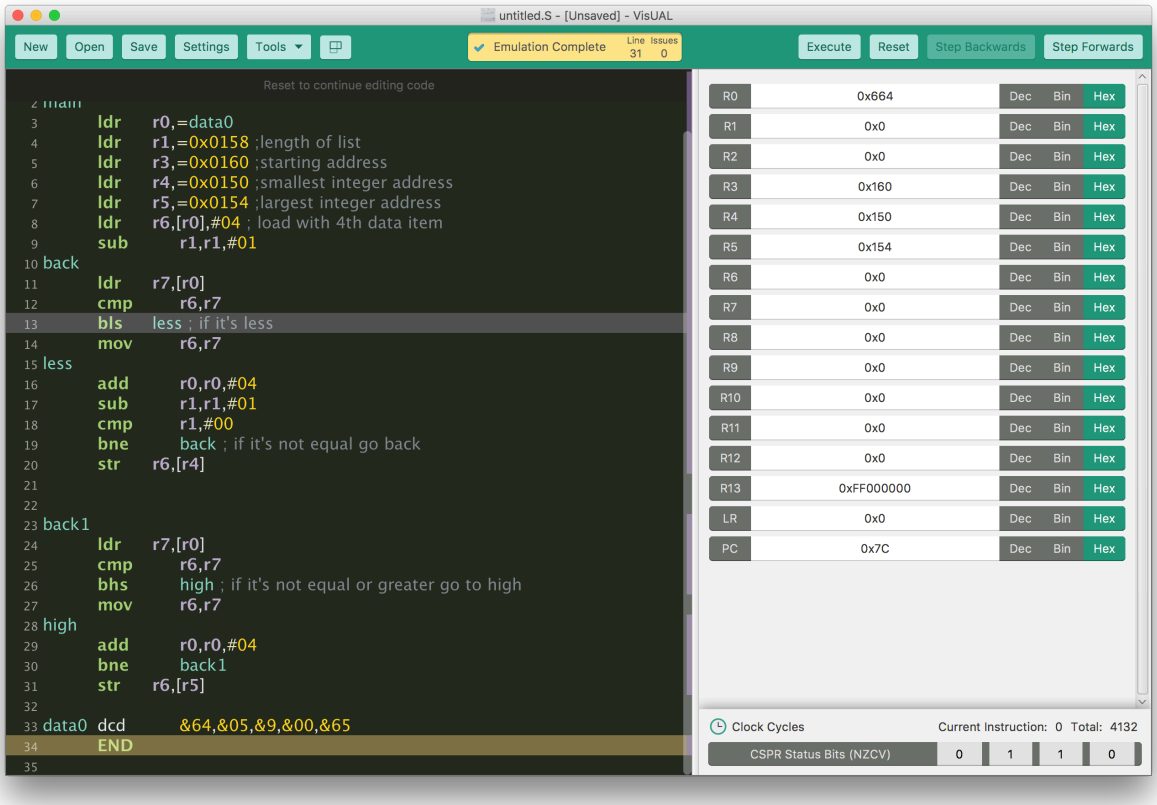


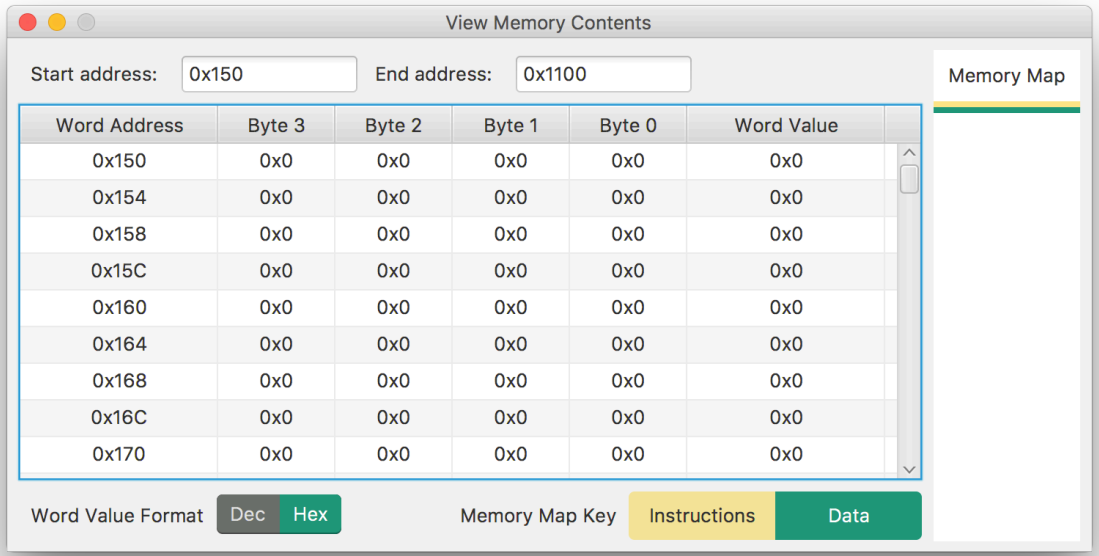


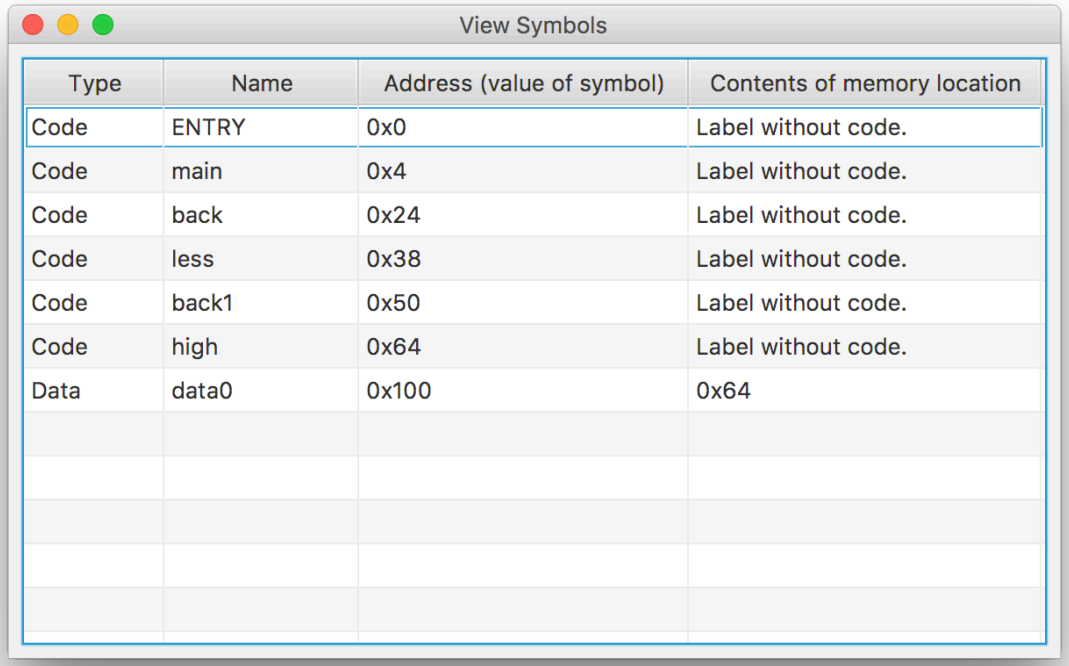


6)

- I'm trying to learn how to scan a list of unsigned bytes to find the smallest and largest numbers in the list.


untitled.S - [Unsaved] - VisUAL
New
Open
Save
Settings
Tools
Emulation Complete
Line 31 Issues 0
Execute
Reset
Step Backwards
Step Forwards
Reset to continue editing code
main
ldr r0,=data0
ldr r1,=0x0158 ;length of list
ldr r3,=0x0160 ;starting address
ldr r4,=0x0150 ;smallest integer address
ldr r5,=0x0154 ;largest integer address
ldr r6,[r0],#04 ; load with 4th data item
sub r1,r1,#01
back
ldr r7,[r0]
cmp r6,r7
bls less ; if it's less
mov r6,r7
less
add r0,r0,#04
sub r1,r1,#01
cmp r1,#00
bne back ; if it's not equal go back
str r6,[r4]
back1
ldr r7,[r0]
cmp r6,r7
bhs high ; if it's not equal or greater go to high
mov r6,r7
high
add r0,r0,#04
bne back1
str r6,[r5]
data0 dcd &64,&05,&9,&00,&65
END
R0 0x664 Dec Bin Hex
R1 0x0 Dec Bin Hex
R2 0x0 Dec Bin Hex
R3 0x160 Dec Bin Hex
R4 0x150 Dec Bin Hex
R5 0x154 Dec Bin Hex
R6 0x0 Dec Bin Hex
R7 0x0 Dec Bin Hex
R8 0x0 Dec Bin Hex
R9 0x0 Dec Bin Hex
R10 0x0 Dec Bin Hex
R11 0x0 Dec Bin Hex
R12 0x0 Dec Bin Hex
R13 0xFF000000 Dec Bin Hex
LR 0x0 Dec Bin Hex
PC 0x7C Dec Bin Hex
Clock Cycles
Current Instruction: 0 Total: 4132
CSPR Status Bits (NZCV) 0 1 1 0


View Memory Contents
Start address: 0x150
End address: 0x1100
Memory Map
Word Address Byte 3 Byte 2 Byte 1 Byte 0 Word Value
0x150 0x0 0x0 0x0 0x0 0x0
0x154 0x0 0x0 0x0 0x0 0x0
0x158 0x0 0x0 0x0 0x0 0x0
0x15C 0x0 0x0 0x0 0x0 0x0
0x160 0x0 0x0 0x0 0x0 0x0
0x164 0x0 0x0 0x0 0x0 0x0
0x168 0x0 0x0 0x0 0x0 0x0
0x16C 0x0 0x0 0x0 0x0 0x0
0x170 0x0 0x0 0x0 0x0 0x0
Word Value Format
Dec
Hex
Memory Map Key
Instructions
Data


View Symbols
Type
Name
Address (value of symbol)
Contents of memory location
Code
ENTRY
0x0
Label without code.
Code
main
0x4
Label without code.
Code
back
0x24
Label without code.
Code
less
0x38
Label without code.
Code
back1
0x50
Label without code.
Code
high
0x64
Label without code.
Data
data0
0x100
0x64