

Source File: lab14.asm
Input: Standard Input
Output: Standard Output
Value: 1

Reserve three doublewords in the `.bss` section to represent the variables `val1`, `val2`, and `val3`. Create a `.bss` section like the following:

```
1 SECTION .bss
2 val1    resd    1
3 val2    resd    1
4 val3    resd    1
```

Accept values for the variables through standard input. Use the following code segment as a guide to display the hexadecimal contents of an area of memory. Use this code segment to display the contents of the three variables after they have been initialized.

```
1 mov     esi, val1           ; Place address of variable in esi
2 mov     ecx, 3             ; Place # of units to display in ecx
3 mov     ebx, 4             ; Unit type (1=byte, 2=word, 4=dword)
4 call    DumpMem
```

Do not alter the contents of the locations symbolically referred to as `val1`, `val2`, and `val3` after they have been read in. Use only the `eax` register to compute

$$-val1 - val2 - val3$$

Show the contents of the registers after each step in the calculation. Your program should adhere to ordinary rules of algebraic precedence and associativity. Display the contents of the three variables after the calculation is completed. Add an identification section as shown below to the top of your source file.

```
; Your name
; CS 3304
; Lab 14
```

A sample execution sequence is shown in Figure 1. To use the `Makefile` as distributed in class, add a target of `lab14` to `targetsAsmLanguage`.

```
1 newuser@csunix ~/3304/14> cp /usr/local/3304/src/Makefile .
2 newuser@csunix ~/3304/14> touch lab14.asm
3 newuser@csunix ~/3304/14> make
4 nasm -f elf32 -l lab14.lst -o lab14.o lab14.asm -I/usr/local/3304/include/ -I.
5 ld -m elf_i386 --dynamic-linker /lib/ld-linux.so.2 -o lab14 lab14.o \
6 /usr/local/3304/src/Along32.o -lc
7 newuser@csunix ~/3304/14> # Execute ./lab14 with interactive input
8 newuser@csunix ~/3304/14> ./lab14
9 1
10 2
11 3
12
13 Dump of offset 0804A15C
14 -----
15 00000001 00000002 00000003
16
17 EAX=00000003 EBX=00000004 ECX=00000003 EDX=F7739480
18 ESI=0804A15C EDI=080481B0 EBP=00000000 ESP=FFA492C0
19 EIP=080481E7 EFL=00000282 CF=0 SF=1 ZF=0 OF=0 AF=0 PF=0
20
21
22 EAX=00000001 EBX=00000004 ECX=00000003 EDX=F7739480
23 ESI=0804A15C EDI=080481B0 EBP=00000000 ESP=FFA492C0
24 EIP=080481F1 EFL=00000282 CF=0 SF=1 ZF=0 OF=0 AF=0 PF=0
25
26
27 EAX=FFFFFFFF EBX=00000004 ECX=00000003 EDX=F7739480
28 ESI=0804A15C EDI=080481B0 EBP=00000000 ESP=FFA492C0
29 EIP=080481F8 EFL=00000297 CF=1 SF=1 ZF=0 OF=0 AF=1 PF=1
30
31
32 EAX=FFFFFFFD EBX=00000004 ECX=00000003 EDX=F7739480
33 ESI=0804A15C EDI=080481B0 EBP=00000000 ESP=FFA492C0
34 EIP=08048203 EFL=00000282 CF=0 SF=1 ZF=0 OF=0 AF=0 PF=0
35
36
37 EAX=FFFFFFFA EBX=00000004 ECX=00000003 EDX=F7739480
38 ESI=0804A15C EDI=080481B0 EBP=00000000 ESP=FFA492C0
39 EIP=0804820E EFL=00000286 CF=0 SF=1 ZF=0 OF=0 AF=0 PF=1
40
41
42 Dump of offset 0804A15C
43 -----
44 00000001 00000002 00000003
```

Figure 1. Commands to Assemble, Link, & Run Lab 14 (Part 1 of 2)

```

45 newuser@csunix ~/3304/14> # Create an input file 01.dat
46 newuser@csunix ~/3304/14> # Enter the data items -1, -2, -3, one item per line
47 newuser@csunix ~/3304/14> cat 01.dat
48 -1
49 -2
50 -3
51 newuser@csunix ~/3304/14> # Execute ./lab14 using ../irvine_test.sh
52 newuser@csunix ~/3304/14> ../irvine_test.sh lab14 01.dat
53
54 Dump of offset 0804A15C
55 -----
56 FFFFFFFF FFFFFFFE FFFFFFFD
57
58 EAX=FFFFFFFD EBX=00000004 ECX=00000003 EDX=F779C480
59 ESI=0804A15C EDI=080481B0 EBP=00000000 ESP=FFD66930
60 EIP=080481E7 EFL=00000286 CF=0 SF=1 ZF=0 OF=0 AF=0 PF=1
61
62
63 EAX=FFFFFFF EBX=00000004 ECX=00000003 EDX=F779C480
64 ESI=0804A15C EDI=080481B0 EBP=00000000 ESP=FFD66930
65 EIP=080481F1 EFL=00000286 CF=0 SF=1 ZF=0 OF=0 AF=0 PF=1
66
67
68 EAX=00000001 EBX=00000004 ECX=00000003 EDX=F779C480
69 ESI=0804A15C EDI=080481B0 EBP=00000000 ESP=FFD66930
70 EIP=080481F8 EFL=00000213 CF=1 SF=0 ZF=0 OF=0 AF=1 PF=0
71
72
73 EAX=00000003 EBX=00000004 ECX=00000003 EDX=F779C480
74 ESI=0804A15C EDI=080481B0 EBP=00000000 ESP=FFD66930
75 EIP=08048203 EFL=00000217 CF=1 SF=0 ZF=0 OF=0 AF=1 PF=1
76
77
78 EAX=00000006 EBX=00000004 ECX=00000003 EDX=F779C480
79 ESI=0804A15C EDI=080481B0 EBP=00000000 ESP=FFD66930
80 EIP=0804820E EFL=00000217 CF=1 SF=0 ZF=0 OF=0 AF=1 PF=1
81
82
83 Dump of offset 0804A15C
84 -----
85 FFFFFFFF FFFFFFFE FFFFFFFD
86 newuser@csunix ~/3304/14>

```

Figure 1. Commands to Assemble, Link, & Run Lab 14 (Part 2 of 2)