

Source File: lab28.asm
Input: Standard Input
Output: Standard Output
Value: 2

Write an assembly language function that receives a signed 32-bit integer and displays its internal octal representation. Use the `bt` instruction to access the bits. Using any shift or division instruction is prohibited. A description of the function as well as client code for testing your implementation is shown in Figure 1, and a sample execution sequence is shown in Figure 2. To use the Makefile as distributed in class, add a target of `lab28` to `targets2AsmFiles`.

```

1  [list -]
2  %INCLUDE "Along32.inc"
3  %INCLUDE "Macros_Along.inc"
4  [list +]
5
6  ;-----
7  extern PrintOctal
8  ; HLL prototype: void PrintOctal(int n);
9  ; Prints the internal octal representation of n
10 ; Receives: EAX = signed 32-bit integer
11 ; Returns:  nothing
12 ;-----
13
14 SECTION .data
15 hrule   times 30 db ('-')
16         db    10,0
17 spacer1 times 2 db ' '
18         db    0
19 spacer2 times 4 db ' '
20         db    0
21 header  times 2 db ' '
22         db    ' Decimal '
23         times 7 db ' '
24         db    'Octal',10,0
25 ten     dd    10
26
27 SECTION .bss
28 h       resd 1
29 num     resd 1
30 width  resd 1
31
32 SECTION .text
33         global _start
34 _start:
35         call  ReadDec           ; read an unsigned integer
36         mov   [h],eax          ; move the integer to n
37
38         mov   edx,hrule        ; print the table header
39         call  WriteString

```

Figure 1. `/usr/local/3304/src/lab28main.asm` (Part 1 of 2)

```

40     mov     edx,header
41     call   WriteString
42     mov     edx,hrule
43     call   WriteString
44 .L0:
45     cmp     dword [h],0           ; while h >= 0 do
46     je     .L5
47     call   ReadInt              ; read a signed 32-bit integer
48     mov     dword [num], eax     ; save a copy in num
49     mov     edx,spacer1
50     call   WriteString
51                                     ; determine the width of the input num
52     mov     dword [width],1
53     mov     eax,[num]
54 .L1:
55     cdq
56     idiv   dword [ten]          ; convert from dword to qword
57     cmp     eax,0              ; signed division by 10
58     je     .L2                ; if the quotient is 0, we're done
59     inc     dword [width]      ; else increment the width
60     jmp    .L1
61 .L2:
62     mov     ecx,10
63     sub     ecx,[width]        ; insert enough spaces to right-justify
64                                     ; num
65 .L3:
66     cmp     ecx,0
67     je     .L4
68     mov     al,' '
69     call   WriteChar
70     dec     ecx
71     jmp    .L3
72 .L4:
73     mov     eax,[num]
74     call   WriteInt
75     mov     edx,spacer2
76     call   WriteString
77     call   PrintOctal
78     mov     al,10
79     call   WriteChar
80     dec     dword [h]
81     jmp    .L0                ; end while
82 .L5:
83     mov     edx,hrule
84     call   WriteString        ; print the table footer
85     Exit   {0}

```

Figure 1. /usr/local/3304/src/lab28main.asm (Part 2 of 2)

```

1  newuser@csunix ~/3304/28> cp /usr/local/3304/data/28/* .
2  newuser@csunix ~/3304/28> cp /usr/local/3304/src/Makefile .
3  newuser@csunix ~/3304/28> cp /usr/local/3304/src/lab28main.asm .
4  newuser@csunix ~/3304/28> touch lab28.asm
5  newuser@csunix ~/3304/28> make lab28
6  nasm -f elf32 -l lab28main.lst -o lab28main.o lab28main.asm -I/usr/local/3304/include/ -I.
7  nasm -f elf32 -l lab28.lst -o lab28.o lab28.asm -I/usr/local/3304/include/ -I.
8  ld -m elf_i386 --dynamic-linker /lib/ld-linux.so.2 -o lab28 lab28main.o lab28.o \
9  /usr/local/3304/src/Along32.o -lc
10 newuser@csunix ~/3304/28> ../irvine_test.sh lab28 01.dat
11 -----
12          Decimal          Octal
13 -----
14          +0             0000000000
15          +1             0000000001
16          -1             3777777777
17          +2             0000000002
18          -2             3777777776
19          +3             0000000003
20          -3             3777777775
21          +12            0000000014
22          -12            3777777764
23          +123           0000000173
24          -123           3777777605
25          +1234          0000002322
26          -1234          3777775456
27          +12345         0000030071
28          -12345         3777747707
29          +123456        00000361100
30          -123456        37777416700
31          +1234567       00004553207
32          -1234567       37773224571
33          +12345678      00057060516
34          -12345678      37720717262
35          +123456789     00726746425
36          -123456789     37051031353
37          +2147483647    17777777777
38          -2147483647    20000000001
39          -2147483648    20000000000
40 -----
41 newuser@csunix ~/3304/28> ../irvine_test.sh lab28 01.dat > my.out
42 newuser@csunix ~/3304/28> diff 01.out my.out
43 newuser@csunix ~/3304/28>

```

Figure 2. Commands to Assemble, Link, & Run Lab 28