Source File: `lab37.asm`
Input: Standard Input
Output: Standard Output
Value: 2

If u and v are integers, not both zero, we say that their greatest common divisor, $\gcd(u, v)$, is the largest nonnegative integer that evenly divides both u and v. When u and v are both zero, every integer evenly divides zero, so it is convenient to set $\gcd(0, 0) = 0$. When either u or v is zero, define $\gcd(u, 0) = |u|$ and $\gcd(0, v) = |v|$. Use Euclid's algorithm to determine the greatest common divisor.

Write an assembly function that will implement the algorithm described above for determining the greatest common divisor of two signed 32-bit integers. 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 `lab37` to `targets2AsmFiles`.

```
1  [list -]
2  %INCLUDE "Along32.inc"
3  %INCLUDE "Macros_Along.inc"
4  [list +]
5
6  ;-----------------------------------------------------------------------
7  extern  fflush
8  ; HLL prototype: int fflush(FILE *stream);
9  ; For output streams (and for update streams on which the last
10 ; operation was output), writes any unwritten data from the stream's
11 ; buffer to the associated output device.
12 ;
13 ; For input streams (and for update streams on which the last
14 ; operation was input), the behavior is undefined.
15 ;
16 ; If stream is a null pointer, all open output streams are flushed,
17 ; including the ones manipulated within library packages or otherwise
18 ; not directly accessible to the program.
19 ; Receives: stream - the file stream to write out
20 ; Returns: Returns zero on success. Otherwise EOF is returned and the
21 ;          error indicator of the file stream is set.
22 ; Source: https://en.cppreference.com/w/c/io/fflush
23 ;-----------------------------------------------------------------------
24
25 ;-----------------------------------------------------------------------
26 extern  printf
27 ; HLL prototype: int printf(const char *format, ...);
28 ; Loads the data from the given locations, converts them to character
29 ; string equivalents and writes the results to the output stream
30 ; stdout.
31 ; Receives: format - pointer to a null-terminated byte string
32 ;                    specifying how to interpret the data
33 ;           ...    - arguments specifying data to print. If any
34 ;                    argument after default argument promotions is
35 ;                    not the type expected by the corresponding
36 ;                    conversion specifier, or if there are fewer
```

Figure 1. `/usr/local/3304/src/lab37main.asm` (Part 1 of 4)

```
37 ;                    arguments than required by format, the behavior
38 ;                    is undefined. If there are more arguments than
39 ;                    required by format, the extraneous arguments are
40 ;                    evaluated and ignored.
41 ; Returns: number of characters transmitted to the output stream or
42 ;          negative value if an output error or an encoding error
43 ;          (for string and character conversion specifiers) occurred
44 ; Source: https://en.cppreference.com/w/c/io/fprintf
45 ;-----------------------------------------------------------------------
46
47 ;-----------------------------------------------------------------------
48 extern  GCD
49 ; HLL prototype: int GCD(int x, int y);
50 ; Returns the greatest common divisor (gcd) of two signed 32-bit
51 ; integers.
52 ; Implements the following pseudocode:
53 ;      int GCD(int x, int y)
54 ;      {
55 ;        x = abs(x);
56 ;        y = abs(y);
57 ;        if (x == 0)
58 ;          return y;
59 ;        else if (y == 0)
60 ;          return x;
61 ;        do
62 ;        {
63 ;          int n = x % y;
64 ;          x = y;
65 ;          y = n;
66 ;        } while (y > 0);
67 ;        return x;
68 ;      }
69 ; Receives: two signed 32-bit integers via the system stack
70 ; Returns:  a 32-bit integer in EAX
71 ;-----------------------------------------------------------------------
72
73 SECTION .data
74 fmt     db      `%s%11d%s%11d%s%10d\n`,0
75 fmt2    db      `%s`,0
76 hrule   times   44 db ('-')
77         db      10,0
78 spacer2 times   2 db ' '
79         db      0
80 spacer4 times   4 db ' '
81         db      0
82 header  db      `       X              Y          GCD(X,Y)\n`,0
83
```

Figure 1. `/usr/local/3304/src/lab37main.asm` (Part 2 of 4)

```
84  SECTION .bss
85  h       resd    1
86  x       resd    1
87  y       resd    1
88
89  SECTION .text
90          global  _start
91  _start:
92          call    ReadDec                 ; read an unsigned integer
93          mov     [h],eax                 ; move the integer to h
94
95          mov     edx,hrule               ; write hrule
96          call    WriteString
97          mov     edx,header              ; write headings
98          call    WriteString
99          mov     edx,hrule               ; write hrule
100         call    WriteString
101
102 .L0:
103         cmp     dword [h],0             ; while h >= 0 do
104         je      .L1
105         call    ReadInt                 ;   read a signed 32-bit integer
106         mov     [x],eax                 ;   save a copy in x
107         call    ReadInt                 ;   read a second signed 32-bit integer
108         mov     [y],eax                 ;   save a copy in y
109
110         push    dword [y]               ;   call GCD(x, y)
111         push    dword [x]
112         call    GCD
113         add     esp,8                   ; clean up the system stack
114
115 ;  Make a call to the C function printf
116 ;    printf("%s%11d%s%11d%s%10d\n", spacer2, x, spacer4, y, spacer4, gcd);
117         push    eax                     ;   the gcd is in EAX
118         push    spacer4                 ;   pointer to string spacer4
119         push    dword [y]               ;   push the value of y
120         push    spacer4                 ;   pointer to string spacer4
121         push    dword [x]               ;   push the value of x
122         push    spacer2                 ;   pointer to string spacer2
123         push    fmt                     ;   pointer to the format string
124         call    printf
125         add     esp,28                  ;   clean up the system stack
126
127         dec     dword [h]               ;   decrement h
128         jmp     .L0                     ; end while
129 .L1:
```

Figure 1. `/usr/local/3304/src/lab37main.asm` (Part 3 of 4)

```
130 ;  Make a call to the C function fflush; if a null pointer is passed, all
131 ;  open output streams are flushed
132 ;    fflush(0);
133          push    0                       ; push NULL (0) pointer on stack
134          call    fflush
135          add     esp,4                   ; clean up the system stack
136
137          mov     edx,hrule               ; write hrule
138          call    WriteString
139
140          Exit    {0}
```

Figure 1. `/usr/local/3304/src/lab37main.asm` (Part 4 of 4)

```
newuser@csunix ~/3304/37> cp /usr/local/3304/data/37/* .
newuser@csunix ~/3304/37> cp /usr/local/3304/src/Makefile .
newuser@csunix ~/3304/37> cp /usr/local/3304/src/lab37main.asm .
newuser@csunix ~/3304/37> touch lab37.asm
newuser@csunix ~/3304/37> make
nasm -f elf32  -l lab37main.lst -o lab37main.o lab37main.asm -I/usr/local/3304/include/ -I.
nasm -f elf32  -l lab37.lst -o lab37.o lab37.asm -I/usr/local/3304/include/ -I.
ld -m elf_i386 --dynamic-linker /lib/ld-linux.so.2 -o lab37 lab37main.o lab37.o \
/usr/local/3304/src/Along32.o -lc
newuser@csunix ~/3304/37> ../irvine_test.sh lab37 01.dat
-----------------------------------------
       X              Y          GCD(X,Y)
-----------------------------------------
            -3              4             1
             3              4             1
             3             -4             1
            -3             -4             1
            25             45             5
             8             99             1
           129           6579           129
          1935            249             3
          1331           1651             1
          2301           1079            13
             3           1260             3
             6            198             6
            43           1935            43
           207           6579             9
             5              7             1
           -25            -35             5
           -83           1651             1
           127          -1079             1
          1079           1651            13
          1651           1079            13
           361            551            19
           361           -551            19
          -361            551            19
          -361           -551            19
             0              0             0
          3304              0          3304
             0           3304          3304
         -3304              0          3304
             0          -3304          3304
  -2147483647     2147483647    2147483647
-----------------------------------------
newuser@csunix ~/3304/37> ../irvine_test.sh lab37 01.dat > my.out
newuser@csunix ~/3304/37> diff 01.out my.out
newuser@csunix ~/3304/37>
```

Figure 2. Commands to Assemble, Link, & Run Lab 37