```
; This subroutine computes GCD value of two unsigned 8-bit integers.
; The algorithm implemented is Euclid's subtraction algorithm, and is given
; by the pseudo code below:
;       gcd(a,b):
;           while(b!=0):
;               if a >= b: a = a-b
;               else: swap a and b (will need a temporary variable)
;           return a
;
; The values of a and b are 8-bit values stored in memory in the variables a and b
; respectively. At the end of the program, the GCD of (a,b) is stored in the variable
; a as an unsigned 8 bit integer and the value of variable b will always be 0.
; The value of a is then returned to the accumulator. For error handling,
; faulty inputs (e.g. gcd(0,0)) will return 0.
;
; Revision History
;   10 Feb 23 Steven Lei       Created pseudo code with comments
;   10 Feb 23 Steven Lei       Converted pseudo code to assembly
;   11 Feb 23 Steven Lei       Hand compliled .asm to .obj
;   11 Feb 23 Steven Lei       Debugged program and ran test cases
;   11 Feb 23 Steven Lei       Successfully tested program

;0000           EuclidLoop:             ;loop until b is 0
0000  8001;         LDD   b             ;load the value of b
0001  3033;         CMPI  0             ;compare b to 0
0002  9F14;         JZ    Done          ;if b is now 0, we're done
0003  1F80;         NOP
                   ;JNZ   EuclidCheck   ;otherwise compute the gcd by decrementing a

;0004           EuclidCheck:            ;first check if a>=b
0004  8000;         LDD   a             ;load the value of a in the accumulator
0005  3001;         CMP   b             ;compare a to b
0006  8C04;         JAE   Sub           ;if a>=b then go to subtract branch
0007  1F80;         NOP
0008  8F06;         JB    Swap          ;else a<b, so go to swap branch
0009  1F80;         NOP

;000A           Sub:                    ;subtract b from a
000A  8000;         LDD   a             ;load the value of a in accumulator
000B  1001;         SUB   b             ;subtract the value of b from accumulator
000C  A000;         STD   a             ;and store it to a
000D  C000;         JMP   EuclidLoop    ;jump back to the start of EuclidLoop
000E  1F80;         NOP

;000F           Swap:                   ;swap a and b using register X as a temporary variable
000F  8000;         LDD   a             ;load the value of a in the accumulator
0010  0780;         TAX                 ;store it in register X
0011  8001;         LDD   b             ;load the value of b in the accumulator
0012  A000;         STD   a             ;store it in a
0013  6701;         TXA                 ;load the value of register X in the accumulator
0014  A001;         STD   b             ;store it in b
0015  C000;         JMP   EuclidLoop    ;go back to the start of the EuclidLoop
0016  1F80;

;0017            Done:                  ;done with the calculation
0017  8000;         LDD   a             ;load the value of a (the gcd value) into the accumulator
```

```
0018  1F00;         RTS                ;and return

;Variables
;00  ??  a         DB    ?          ;the first number for GCD comparison
;01  ??  b         DB    ?          ;the second number for GCD comparison
```

```
Caltech10 Simlulator v1.5.6
L
filename: gcd.obj
Done.
00/ 00 0
01/ 00 0
S

RETURN  A:00 F:01 X:00 S:00 P:0000
00/ 00 32
01/ 00 0
S

RETURN  A:32 F:01 X:00 S:00 P:0000
00/ 32 0
01/ 00 32
S

RETURN  A:32 F:01 X:00 S:00 P:0000
```

Edge case 0 and 0:

(should return 0 as defined in the comments)

GCD(0,0) = 0

Edge case: 0 and random number

GCD (32,0) = 32

Edge case: random number and 0

GCD(0,32) = 32

```
00/ 32 FF
01/ 00 FF
S

RETURN  A:FF F:01 X:00 S:00 P:0000
00/ FF FF
01/ 00 0
S

RETURN  A:FF F:01 X:00 S:00 P:0000
00/ FF 0
01/ 00 FF
S

RETURN  A:FF F:01 X:00 S:00 P:0000
```

Edge case largest number (FF)

GCD(FF,FF) = FF

Edge case: largest number and 0

GCD(FF,0) = FF

Edge case: 0 and largest number

GCD(0,FF) = FF

```
00/ FF 5
01/ 00 FF
S

RETURN  A:05 F:01 X:00 S:00 P:0000

00/ 05 FF
01/ 00 5
S

RETURN  A:05 F:01 X:00 S:00 P:0000
```

Edge case: random number and largest number

GCD(5,FF) = 5

Edge case: largest number and random number

GCD(FF,5) = 5

```
00/ 09
01/ 00 3
S

RETURN  A:03 F:01 X:00 S:00 P:0000

00/ 03 3
?
01/ 00 9
S

RETURN  A:03 F:01 X:00 S:00 P:0000
```

General case: GCD(a,b) = GCD(b,a)

GCD(9,3) = 3

GCD(3,9) = 3

```
00/ 02 25
01/ 00 3
S

RETURN  A:01 F:01 X:00 S:00 P:0000

00/ 01 25
01/ 00 9
S

RETURN  A:01 F:01 X:00 S:00 P:0000
```

General case: prime numbers

25 is 37 in hex

GCD(25,3) = 1

General case: prime and composite number

25 is 37 in hex

GCD(25,9) = 1

```
00/ 00 1
01/ 00 1
S

RETURN  A:01 F:01 X:00 S:00 P:0000

00/ 01 30
01/ 00 90
S

RETURN  A:30 F:01 X:00 S:00 P:0000
```

General cases

GCD(1,1) = 1

30 is 48 in hex, 90 is 144

GCD(30,90) = 30

```
00/ 01 8
01/ 00 10
S

RETURN  A:08 F:01 X:00 S:00 P:0000


00/ 08 2
01/ 00 6
S

RETURN  A:02 F:01 X:00 S:00 P:0000

00/ 02 F
01/ 00 3
S

RETURN  A:03 F:01 X:00 S:00 P:0000
```

10 is 16 in hex

GCD(8,10) = 8

GCD(2,6) = 2

F is 15 in hex

GCD(15, 3) = 3