

CS270 Programming Assignment 5

“Recursive implementation of GCD”

Program due Friday, April 12 via RamCT by 1 PM.

Goals

Write an LC3 program that implements the Euclidean algorithm that computes the greatest common divisor (GCD). This widely used algorithm was originally given by Euclid of Alexandria, the father of geometry, who lived around 300 BC. The algorithm is based on the facts that for two integers a and b , $\text{gcd}(a, b) = \text{gcd}(b, a \bmod b)$ where b is non-zero, and $\text{gcd}(a, 0) = a$. For example $\text{GCD}(54,24) = 6$.

In the recursive version the algorithm may be stated as

```
int gcd(int a, int b) {
    if (b == 0)
        return a;
    return gcd(b, a % b);
}
```

Where $a\%b$ is the modulo operation.

You will also learn how to use activation records.

Program Structure Specification

Your program should use a main function with a subroutine GCD that takes two integers as arguments and returns the greatest common divisor value. You must implement the subroutine so that it calls itself recursively.

You must use the activation record exactly as specified in the textbook. Use the code used in Recitation 9 as a guide.

You must copy argument A into local variable locA and B into local variable locB. You need

locations for two local variables, dynamic link (caller's frame pointer) return address, return value and the two parameters A and B, in the given sequence (from lower to higher addresses). The new frame pointer should point to locB.

The main function should get two values from locations labeled Value1 and Value2. The final result should be placed in location GCDValue. The program should work correctly if one of the two values is zero.

The program must start with this code.

```
;File: PA5.asm
;Description: --> fill this in (single line only)
;Author: --> fill this in
;Date: --> fill this in
; R1: ->fill if used      R2: ->fill if used
; R3: ->fill if used      R4: ->fill if used
; ***** BEGIN RESERVED SECTION *****
; You may initialize values here (i.e., change the value of a .FILL
; statement), but you may *not* add or remove instructions or perform
; any edit which changes the address of a label in this section
        .ORIG x3000      ; DO NOT change any code before label GCDvalue
        BR BEGIN_CODE   ; jump to beginning of code
Value1   .FILL 54
Value2   .FILL 24
GCDValue .BLKW 1        ;end of specified locations
; ***** END RESERVED SECTION *****
;-----you may add more variables or code below this line -----
BEGIN_CODE      ; your code begins here
```

At the end of the code include result of a test run as seen on the console in form of a comment by recording the contents of the three locations Value1, Value2 and GCDValue.

Integration and testing:

Write a small program that will calculate $A\%B$. You can do that by subtracting B from A a number of times until result is just less than B, or 0. Test it carefully. Then write the main program with the subroutines. Test them using a combination of values before submitting.

Submission Instructions

- Submit the pa5.asm file will be submitted directly to RamCT.

Reminders:

1. Adds comments appropriately.
2. Illegal inputs (when any one of the two values is negative) should generate message "Illegal Input".

Grading Criteria

To grade the assignment, we will examine and run program (30 points), and we will verify that your computation gets the right answers on the example data and our own test data (35 points). In addition points will be given for coding style and comments (15 points), following assignment directions (20 points). No points if you do not use recursion. The grading factors we consider for coding style include having clear and concise comments, consistent indentation, and the minimal amount of code to solve the problem.

Late Policy

There is a 25% penalty for late submission. They cannot be more than 24 hours late.