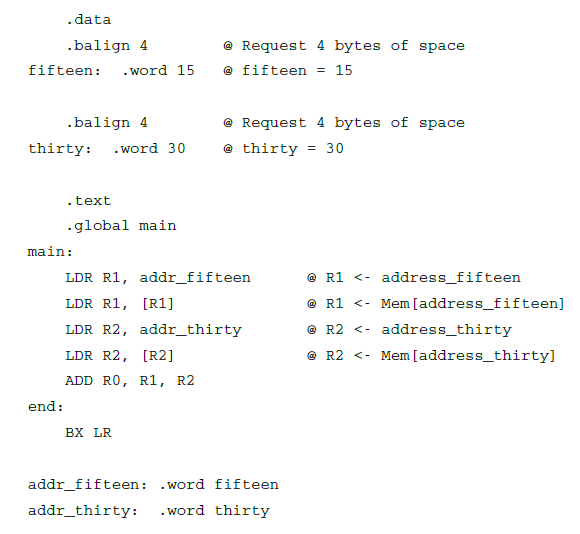
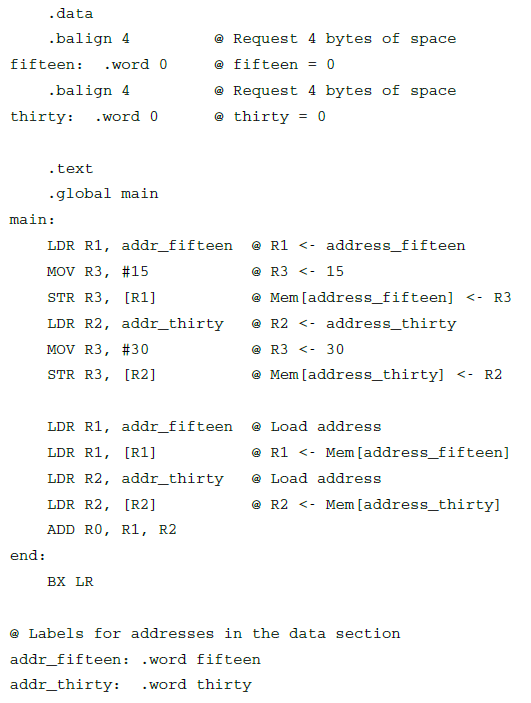
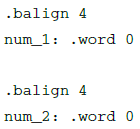
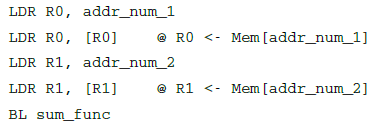
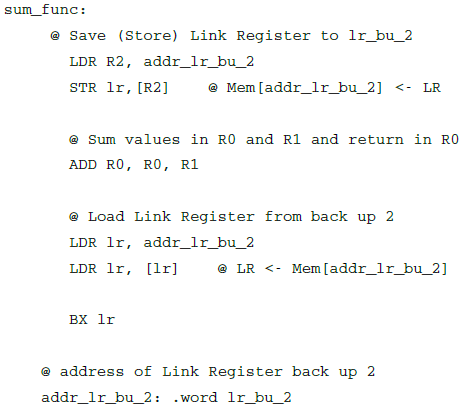
นายธนพล วงศ์อาษา 62010356

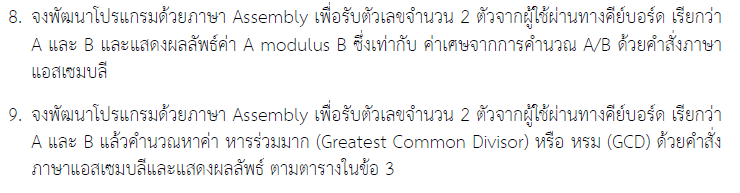
นายสุทธิราช ภูโท 62010966

**การทดลองที่ 7 การเรียกใช้และสร้างฟังค์ชันในโปรแกรมภาษาแอสเซมบลี**

**ส่วนการทดลอง \*โค้ดเพิ่มเติมดูได้จากไฟล์ที่แนบมา**

1. **การโหลดค่าตัวแปรจากหน่วยความจำมาพักในรีจิสเตอร์**
   1. บันทึกผลและอธิบายผลที่เกิดขึ้น ข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง SWI (Software Interrupt) โปรแกรมนี้จะทำการบวกเลข 15 และ 30 เข้าด้วยกันและนำไปเก็บไว้ใน register R0
   2. บันทึกผลและอธิบายผลที่เกิดขึ้นเพื่อเปรียบเทียบกับข้อที่แล้ว****โปรแกรมนี้จะทำการกำหนดค่าเริ่มต้นให้ตัวแปร fifteen และ thirty เป็น 0 ก่อนและจึงนำค่าคงที่ 15,30 ที่กำหนดไว้ใน register ไป store ไว้ที่ตัวแปรดังกล่าว แล้วจึงค่อยทำการคำนวณแบบข้อที่แล้ว จุดที่แตกต่างคือโปรแกรมนี้ค่าเริ่มต้นตัวแปรเริ่มที่ 0 แล้วทำการ store ค่าใหม่เข้าไปอีกทีแต่โปรแกรมแรกตัวแปรแต่ละตัวจะมีค่าเริ่มต้นเลย
2. **การเรียกใช้ฟังค์ชันและตัวแปรชนิดประโยค**
   1. คำสั่ง echo $? มีไว้เพื่ออะไร**:** แสดงค่าที่เก็บภายใน register R0 ออกมาทาง terminal
3. **การสร้างฟังค์ชันเสริมด้วยภาษาแอสเซมบลี**
   1. ระบุซอร์สโค้ดใน Lab7\_6.s ว่าตรงกับประโยคภาษา C ต่อไปนี้ **int num1, num2**
   2. ระบุซอร์สโค้ดใน Lab7\_6.s ว่าตรงกับประโยคภาษา C ต่อไปนี้ **sum = num1 + num2**โดยใช้ฟังก์ชัน sum\_func ซึ่งมีเนื้อหาภายในคือ****
   3. เหตุใดจึงผู้อ่านจึงไม่ต้องใช้คำสั่ง echo $? แล้ว: มีการใช้คำสั่ง printf แล้วซึ่งจะทำการแสดงผลออกทาง terminal ทำให้ไม่จำเป็นต้องเรียกคำสั่ง echo $? เพื่ออ่านค่าใน register R0 อีก

**คำถามท้ายการทดลอง**



**ตอบ**

.data

.balign 4

get\_num\_1: .asciz "Number 1 :\n"

.balign 4

get\_num\_2: .asciz "Number 2 :\n"

.balign 4

pattern: .asciz "%d"

.balign 4

num\_1: .word 0

.balign 4

num\_2: .word 0

.balign 4

output: .asciz "%d mod %d = %d\n"

.balign 4

lr\_bu: .word 0

.balign 4

lr\_bu\_2: .word 0

.text

mod\_func:

LDR R2, =lr\_bu\_2

STR lr,[R2]

mod\_loop:

CMP R0,R1

BLT end\_mod

SUB R0, R0, R1

BL mod\_loop

end\_mod:

LDR lr,=lr\_bu\_2

LDR lr,[lr]

BX lr

.global main

main:

LDR R1, =lr\_bu

STR lr,[R1]

LDR R0,=get\_num\_1

BL printf

LDR R0, =pattern

LDR R1, =num\_1

BL scanf

LDR R0, =get\_num\_2

BL printf

LDR R0, =pattern

LDR R1, =num\_2

BL scanf

LDR R0, =num\_1

LDR R0, [R0]

LDR R1, =num\_2

LDR R1,[R1]

BL mod\_func

MOV R3, R0

LDR R0, =output

LDR R1, =num\_1

LDR R1, [R1]

LDR R2, =num\_2

LDR R2, [R2]

BL printf

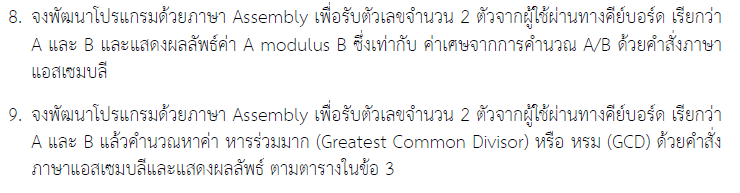
LDR lr, =lr\_bu

LDR lr,[lr]

BX lr

.global printf

.global scanf



**ตอบ**

.data

.balign 4

message: .asciz "GCD : Greatest Common Divisor\n"

.balign 4

get\_num\_1: .asciz "Number 1 : "

.balign 4

get\_num\_2: .asciz "Number 2 : "

.balign 4

pattern: .asciz "%d"

.balign 4

test: .asciz "Test %d\n"

.balign 4

num\_1: .word 0

.balign 4

num\_2: .word 0

.balign 4

output: .asciz "GCD of %d and %d is %d.\n"

.balign 4

lr\_bu: .word 0

.balign 4

lr\_bu\_2: .word 0

.balign 4

lr\_bu\_3: .word 0

.text

mod\_func:

ldr r2, =lr\_bu\_3

str lr, [r2]

mod\_loop:

cmp r0, r1

blt end\_mod

sub r0, r0, r1

bl mod\_loop

end\_mod:

ldr lr, =lr\_bu\_3

ldr lr, [lr]

bx lr

gcd\_func:

ldr r2, =lr\_bu\_2

str lr, [r2]

mov r4, #0

gcd\_loop:

cmp r1, r4

beq end\_gcd

bl mod\_func

mov r2, r0

mov r0, r1

mov r1, r2

bl gcd\_loop

end\_gcd:

ldr lr, =lr\_bu\_2

ldr lr, [lr]

bx lr

.global main

main:

ldr r1, =lr\_bu

str lr, [r1]

ldr r0, =message

bl printf

ldr r0, =get\_num\_1

bl printf

ldr r0, =pattern

ldr r1, =num\_1

bl scanf

ldr r0, =get\_num\_2

bl printf

ldr r0, =pattern

ldr r1, =num\_2

bl scanf

ldr r0, =num\_1

ldr r0, [r0]

ldr r1, =num\_2

ldr r1, [r1]

bl gcd\_func

mov r3, r0

ldr r0, =output

ldr r1, =num\_1

ldr r1, [r1]

ldr r2, =num\_2

ldr r2, [r2]

bl printf

ldr lr, =lr\_bu

ldr lr, [lr]

bx lr

.global printf

.global scanf