

ภาคผนวก G

การทดลองที่ 7 การเรียกใช้และสร้างฟังก์ชันในโปรแกรมภาษาแอสเซมบลี

ผู้อ่านควรจะต้องทำความเข้าใจเนื้อหาของบทที่ 4 หัวข้อ 4.8 และ ทำการทดลองที่ 5 และการทดลองที่ 6 ในภาคผนวกก่อนหน้า โดยการทดลองนี้จะเสริมความเข้าใจของผู้อ่านให้เพิ่มมากขึ้น ตามวัตถุประสงค์เหล่านี้

- เพื่อพัฒนาโปรแกรมภาษาแอสเซมบลีเรียกใช้งานตัวแปรเดี่ยวหรือตัวแปรสเกลาร์ (Scalar)
- เพื่อพัฒนาโปรแกรมแอสเซมบลีเรียกใช้งานตัวแปรชุดหรืออาร์เรย์ (Array)
- เพื่อเรียกใช้ฟังก์ชันจากไลบรารีพื้นฐานด้วยโปรแกรมภาษาแอสเซมบลี ในหัวข้อที่ 4.8
- เพื่อสร้างฟังก์ชันเสริมในโปรแกรมภาษาแอสเซมบลี

G.1 การใช้งานตัวแปรในดาต้าเซ็กเมนต์

ตัวแปรต่าง ๆ ที่ประกาศโดยใช้ชื่อ **เลเบล** ต้องการพื้นที่ในหน่วยความจำสำหรับจัดเก็บค่าตามที่ได้สรุปในตารางที่ 2.1 ตัวแปรจะมีสองชนิดแบ่งตามพื้นที่ในการจัดเก็บค่า คือ

- ตัวแปรชนิด**โกลบอล** (Global Variable) อาศัยพื้นที่สำหรับเก็บค่าของตัวแปรเหล่านี้ เรียกว่า **ดาต้าเซ็กเมนต์** (Data Segment) ซึ่งผู้เขียนได้กล่าวไปแล้วในบทที่ 4 และ
- ตัวแปรชนิด**โลคอล** (Local Variable) อาศัยพื้นที่ภายใน**สแต็กเซ็กเมนต์** (Stack Segment) สำหรับจัดเก็บค่าชั่วคราว เนื่องจากฟังก์ชันคือชุดคำสั่งย่อยที่ฟังก์ชัน main() ในภาษา C หรือ main: ในภาษาแอสเซมบลีเป็นผู้เรียกใช้ และเมื่อทำงานเสร็จสิ้น ฟังก์ชันนั้นจะต้องรีเทิร์นกลับมาหาฟังก์ชัน main() หรือ main: ในที่สุด ดังนั้น ตัวแปรชนิดโลคอลจึงใช้พื้นที่จัดเก็บค่าในสแต็กเฟรมภายในสแต็กเซ็กเมนต์แทน เพราะสแต็กเฟรมจะมีการจองพื้นที่ (PUSH) และคืนพื้นที่ (POP) ในรูปแบบ

แบบ Last In First Out ตามที่อธิบายในหัวข้อที่ 3.3.3 ทำให้ไม่จำเป็นต้องใช้พื้นที่ในบริเวณดาต้าเซ็กเมนต์ ผู้อ่านสามารถทำความเข้าใจหัวข้อนี้เพิ่มเติมในการทดลองที่ 8 ภาคผนวก H

G.1.1 การโหลดค่าตัวแปรเดียวจากหน่วยความจำมาพักในรีจิสเตอร์

1. ย้ายไคเรกทอรีไปยัง `/home/pi/asm` โดยใช้คำสั่ง `$ cd /home/pi/asm`
2. สร้างไคเรกทอรี **Lab7** โดยใช้คำสั่ง `$ mkdir Lab7`
3. ย้ายไคเรกทอรีเข้าไปใน **Lab7**
4. ตรวจสอบว่าไคเรกทอรีปัจจุบันโดยใช้คำสั่ง `pwd`
5. สร้างไฟล์ **Lab7_1.s** ตามซอร์สโค้ดต่อไปนี้ ผู้อ่านสามารถข้ามประโยคคอมเมนต์ได้ เมื่อทำความเข้าใจแต่ละคำสั่งแล้ว

```
.data
.balign 4          @ Request 4 bytes of space
fifteen: .word 15   @ fifteen = 15

.balign 4          @ Request 4 bytes of space
thirty: .word 30    @ thirty = 30

.text
.global main
main:
    LDR R1, addr_fifteen    @ R1 <- address_fifteen
    LDR R1, [R1]            @ R1 <- Mem[address_fifteen]
    LDR R2, addr_thirty     @ R2 <- address_thirty
    LDR R2, [R2]            @ R2 <- Mem[address_thirty]
    ADD R0, R1, R2
end:
    BX LR

addr_fifteen: .word fifteen
addr_thirty: .word thirty
```

6. สร้าง makefile ภายในไดเรกทอรี Lab7 และกรอกคำสั่งดังนี้

```
Lab7_1:
    gcc -o Lab7_1 Lab7_1.s
```

7. ทำการ make และรันโปรแกรมโดยใช้คำสั่ง

```
$ make Lab7_1
$ ./Lab7_1
$ echo $?
```

8. สร้างไฟล์ **Lab7_2.s** ตามโค้ดต่อไปนี้จากไฟล์ **Lab7_1.s** ผู้อ่านสามารถข้ามประโยคคอมเมนต์ได้เมื่อทำความเข้าใจแต่ละคำสั่งแล้ว

```
.data
    .balign 4           @ Request 4 bytes of space
fifteen:  .word 0       @ fifteen = 0
    .balign 4           @ Request 4 bytes of space
thirty:  .word 0       @ thirty = 0

.text
.global main
main:
    LDR R1, addr_fifteen @ R1 <- address_fifteen
    MOV R3, #15          @ R3 <- 15
    STR R3, [R1]         @ Mem[address_fifteen] <- R3
    LDR R2, addr_thirty  @ R2 <- address_thirty
    MOV R3, #30          @ R3 <- 30
    STR R3, [R2]         @ Mem[address_thirty] <- R2

    LDR R1, addr_fifteen @ Load address
    LDR R1, [R1]         @ R1 <- Mem[address_fifteen]
    LDR R2, addr_thirty  @ Load address
    LDR R2, [R2]         @ R2 <- Mem[address_thirty]
    ADD R0, R1, R2

end:
    BX LR
```

```
@ Labels for addresses in the data section
addr_fifteen: .word fifteen
addr_thirty: .word thirty
```

9. เพิ่มประโยคต่อไปนีใน makefile ให้รองรับ Lab7_2

```
Lab7_2:
    gcc -o Lab7_2 Lab7_2.s
```

10. ทำการ make และรันโปรแกรมโดยใช้คำสั่ง

```
$ make Lab7_2
$ ./Lab7_2
$ echo $?
```

บันทึกผลและอธิบายผลที่เกิดขึ้นเพื่อเปรียบเทียบกับข้อที่แล้ว

G.1.2 การใช้งานตัวแปรชุดหรืออาร์เรย์ ชนิด word

ภาษาแอสเซมบลีจะกำหนดชนิดตามหลังชื่อตัวแปร เช่น `.word`, `.hword`, และ `.byte` ใช้กำหนดขนาดของตัวแปรนั้น ๆ ขนาด 32, 16 และ 8 บิตตามลำดับ ยกตัวอย่าง คือ:

```
numbers:    .word 1,2,3,4
```

เป็นการประกาศและตั้งค่าตัวแปรชนิดอาร์เรย์ของ word ซึ่งต้องการพื้นที่ 4 ไบต์ต่อข้อมูลหนึ่งตำแหน่ง ซึ่งจะตรงกับประโยคต่อไปนีในภาษา C

```
int numbers={1,2,3,4}
```

1. สร้างไฟล์ **Lab7_3.s** ตามโค้ดต่อไปนี ผู้อ่านสามารถข้ามประโยคคอมเมนต์ได้ เมื่อทำความเข้าใจแต่ละคำสั่งแล้ว

```
.data
primes:
    .word 2
    .word 3
```

10. ทำการ make และรันโปรแกรมโดยใช้คำสั่ง

```
$ make Lab7_2
```

```
$ ./Lab7_2
```

```
$ echo $?
```

บันทึกผลและอธิบายผลที่เกิดขึ้นเพื่อเปรียบเทียบกับข้อที่แล้ว

ตอบ

ผลลัพธ์ที่ได้จากการรันโปรแกรมคือ 45 เหมือนกัน การรันโปรแกรมครั้งที่ 1 จะคืนค่าตัวแปร loop เป็น 15 และ 30 และ ตัวแปร a จะเก็บไว้ที่ รีจิสเตอร์ในเฟรมเวิร์ก และไว้ที่ 0 รีจิสเตอร์ และเก็บไว้ที่ รีจิสเตอร์ที่ 0 แล้วรีเซ็ตค่า

ส่วนการรันโปรแกรมครั้งที่ 2 จะประมวลผล ตัวแปร loop เป็น 15 และไปเก็บค่าในเฟรมเวิร์ก โดยใช้ตัวชี้ STR เพื่อชี้ไปที่ค่าต่างๆ ตัวแปรนั้นได้คือ 15, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180, 195, 210, 225, 240, 255, 270, 285, 300, 315, 330, 345, 360, 375, 390, 405, 420, 435, 450, 465, 480, 495, 510, 525, 540, 555, 570, 585, 600, 615, 630, 645, 660, 675, 690, 705, 720, 735, 750, 765, 780, 795, 810, 825, 840, 855, 870, 885, 900, 915, 930, 945, 960, 975, 990, 1005, 1020, 1035, 1050, 1065, 1080, 1095, 1110, 1125, 1140, 1155, 1170, 1185, 1200, 1215, 1230, 1245, 1260, 1275, 1290, 1305, 1320, 1335, 1350, 1365, 1380, 1395, 1410, 1425, 1440, 1455, 1470, 1485, 1500, 1515, 1530, 1545, 1560, 1575, 1590, 1605, 1620, 1635, 1650, 1665, 1680, 1695, 1710, 1725, 1740, 1755, 1770, 1785, 1800, 1815, 1830, 1845, 1860, 1875, 1890, 1905, 1920, 1935, 1950, 1965, 1980, 1995, 2010, 2025, 2040, 2055, 2070, 2085, 2100, 2115, 2130, 2145, 2160, 2175, 2190, 2205, 2220, 2235, 2250, 2265, 2280, 2295, 2310, 2325, 2340, 2355, 2370, 2385, 2400, 2415, 2430, 2445, 2460, 2475, 2490, 2505, 2520, 2535, 2550, 2565, 2580, 2595, 2610, 2625, 2640, 2655, 2670, 2685, 2700, 2715, 2730, 2745, 2760, 2775, 2790, 2805, 2820, 2835, 2850, 2865, 2880, 2895, 2910, 2925, 2940, 2955, 2970, 2985, 3000, 3015, 3030, 3045, 3060, 3075, 3090, 3105, 3120, 3135, 3150, 3165, 3180, 3195, 3210, 3225, 3240, 3255, 3270, 3285, 3300, 3315, 3330, 3345, 3360, 3375, 3390, 3405, 3420, 3435, 3450, 3465, 3480, 3495, 3510, 3525, 3540, 3555, 3570, 3585, 3600, 3615, 3630, 3645, 3660, 3675, 3690, 3705, 3720, 3735, 3750, 3765, 3780, 3795, 3810, 3825, 3840, 3855, 3870, 3885, 3900, 3915, 3930, 3945, 3960, 3975, 3990, 4005, 4020, 4035, 4050, 4065, 4080, 4095, 4110, 4125, 4140, 4155, 4170, 4185, 4200, 4215, 4230, 4245, 4260, 4275, 4290, 4305, 4320, 4335, 4350, 4365, 4380, 4395, 4410, 4425, 4440, 4455, 4470, 4485, 4500, 4515, 4530, 4545, 4560, 4575, 4590, 4605, 4620, 4635, 4650, 4665, 4680, 4695, 4710, 4725, 4740, 4755, 4770, 4785, 4800, 4815, 4830, 4845, 4860, 4875, 4890, 4905, 4920, 4935, 4950, 4965, 4980, 4995, 5010, 5025, 5040, 5055, 5070, 5085, 5100, 5115, 5130, 5145, 5160, 5175, 5190, 5205, 5220, 5235, 5250, 5265, 5280, 5295, 5310, 5325, 5340, 5355, 5370, 5385, 5400, 5415, 5430, 5445, 5460, 5475, 5490, 5505, 5520, 5535, 5550, 5565, 5580, 5595, 5610, 5625, 5640, 5655, 5670, 5685, 5700, 5715, 5730, 5745, 5760, 5775, 5790, 5805, 5820, 5835, 5850, 5865, 5880, 5895, 5910, 5925, 5940, 5955, 5970, 5985, 6000, 6015, 6030, 6045, 6060, 6075, 6090, 6105, 6120, 6135, 6150, 6165, 6180, 6195, 6210, 6225, 6240, 6255, 6270, 6285, 6300, 6315, 6330, 6345, 6360, 6375, 6390, 6405, 6420, 6435, 6450, 6465, 6480, 6495, 6510, 6525, 6540, 6555, 6570, 6585, 6600, 6615, 6630, 6645, 6660, 6675, 6690, 6705, 6720, 6735, 6750, 6765, 6780, 6795, 6810, 6825, 6840, 6855, 6870, 6885, 6900, 6915, 6930, 6945, 6960, 6975, 6990, 7005, 7020, 7035, 7050, 7065, 7080, 7095, 7110, 7125, 7140, 7155, 7170, 7185, 7200, 7215, 7230, 7245, 7260, 7275, 7290, 7305, 7320, 7335, 7350, 7365, 7380, 7395, 7410, 7425, 7440, 7455, 7470, 7485, 7500, 7515, 7530, 7545, 7560, 7575, 7590, 7605, 7620, 7635, 7650, 7665, 7680, 7695, 7710, 7725, 7740, 7755, 7770, 7785, 7800, 7815, 7830, 7845, 7860, 7875, 7890, 7905, 7920, 7935, 7950, 7965, 7980, 7995, 8010, 8025, 8040, 8055, 8070, 8085, 8100, 8115, 8130, 8145, 8160, 8175, 8190, 8205, 8220, 8235, 8250, 8265, 8280, 8295, 8310, 8325, 8340, 8355, 8370, 8385, 8400, 8415, 8430, 8445, 8460, 8475, 8490, 8505, 8520, 8535, 8550, 8565, 8580, 8595, 8610, 8625, 8640, 8655, 8670, 8685, 8700, 8715, 8730, 8745, 8760, 8775, 8790, 8805, 8820, 8835, 8850, 8865, 8880, 8895, 8910, 8925, 8940, 8955, 8970, 8985, 9000, 9015, 9030, 9045, 9060, 9075, 9090, 9105, 9120, 9135, 9150, 9165, 9180, 9195, 9210, 9225, 9240, 9255, 9270, 9285, 9300, 9315, 9330, 9345, 9360, 9375, 9390, 9405, 9420, 9435, 9450, 9465, 9480, 9495, 9510, 9525, 9540, 9555, 9570, 9585, 9600, 9615, 9630, 9645, 9660, 9675, 9690, 9705, 9720, 9735, 9750, 9765, 9780, 9795, 9810, 9825, 9840, 9855, 9870, 9885, 9900, 9915, 9930, 9945, 9960, 9975, 9990, 10005, 10020, 10035, 10050, 10065, 10080, 10095, 10110, 10125, 10140, 10155, 10170, 10185, 10200, 10215, 10230, 10245, 10260, 10275, 10290, 10305, 10320, 10335, 10350, 10365, 10380, 10395, 10410, 10425, 10440, 10455, 10470, 10485, 10500, 10515, 10530, 10545, 10560, 10575, 10590, 10605, 10620, 10635, 10650, 10665, 10680, 10695, 10710, 10725, 10740, 10755, 10770, 10785, 10800, 10815, 10830, 10845, 10860, 10875, 10890, 10905, 10920, 10935, 10950, 10965, 10980, 10995, 11010, 11025, 11040, 11055, 11070, 11085, 11100, 11115, 11130, 11145, 11160, 11175, 11190, 11205, 11220, 11235, 11250, 11265, 11280, 11295, 11310, 11325, 11340, 11355, 11370, 11385, 11400, 11415, 11430, 11445, 11460, 11475, 11490, 11505, 11520, 11535, 11550, 11565, 11580, 11595, 11610, 11625, 11640, 11655, 11670, 11685, 11700, 11715, 11730, 11745, 11760, 11775, 11790, 11805, 11820, 11835, 11850, 11865, 11880, 11895, 11910, 11925, 11940, 11955, 11970, 11985, 12000, 12015, 12030, 12045, 12060, 12075, 12090, 12105, 12120, 12135, 12150, 12165, 12180, 12195, 12210, 12225, 12240, 12255, 12270, 12285, 12300, 12315, 12330, 12345, 12360, 12375, 12390, 12405, 12420, 12435, 12450, 12465, 12480, 12495, 12510, 12525, 12540, 12555, 12570, 12585, 12600, 12615, 12630, 12645, 12660, 12675, 12690, 12705, 12720, 12735, 12750, 12765, 12780, 12795, 12810, 12825, 12840, 12855, 12870, 12885, 12900, 12915, 12930, 12945, 12960, 12975, 12990, 13005, 13020, 13035, 13050, 13065, 13080, 13095, 13110, 13125, 13140, 13155, 13170, 13185, 13200, 13215, 13230, 13245, 13260, 13275, 13290, 13305, 13320, 13335, 13350, 13365, 13380, 13395, 13410, 13425, 13440, 13455, 13470, 13485, 13500, 13515, 13530, 13545, 13560, 13575, 13590, 13605, 13620, 13635, 13650, 13665, 13680, 13695, 13710, 13725, 13740, 13755, 13770, 13785, 13800, 13815, 13830, 13845, 13860, 13875, 13890, 13905, 13920, 13935, 13950, 13965, 13980, 13995, 14010, 14025, 14040, 14055, 14070, 14085, 14100, 14115, 14130, 14145, 14160, 14175, 14190, 14205, 14220, 14235, 14250, 14265, 14280, 14295, 14310, 14325, 14340, 14355, 14370, 14385, 14400, 14415, 14430, 14445, 14460, 14475, 14490, 14505, 14520, 14535, 14550, 14565, 14580, 14595, 14610, 14625, 14640, 14655, 14670, 14685, 14700, 14715, 14730, 14745, 14760, 14775, 14790, 14805, 14820, 14835, 14850, 14865, 14880, 14895, 14910, 14925, 14940, 14955, 14970, 14985, 15000, 15015, 15030, 15045, 15060, 15075, 15090, 15105, 15120, 15135, 15150, 15165, 15180, 15195, 15210, 15225, 15240, 15255, 15270, 15285, 15300, 15315, 15330, 15345, 15360, 15375, 15390, 15405, 15420, 15435, 15450, 15465, 15480, 15495, 15510, 15525, 15540, 15555, 15570, 15585, 15600, 15615, 15630, 15645, 15660, 15675, 15690, 15705, 15720, 15735, 15750, 15765, 15780, 15795, 15810, 15825, 15840, 15855, 15870, 15885, 15900, 15915, 15930, 15945, 15960, 15975, 15990, 16005, 16020, 16035, 16050, 16065, 16080, 16095, 16110, 16125, 16140, 16155, 16170, 16185, 16200, 16215, 16230, 16245, 16260, 16275, 16290, 16305, 16320, 16335, 16350, 16365, 16380, 16395, 16410, 16425, 16440, 16455, 16470, 16485, 16500, 16515, 16530, 16545, 16560, 16575, 16590, 16605, 16620, 16635, 16650, 16665, 16680, 16695, 16710, 16725, 16740, 16755, 16770, 16785, 16800, 16815, 16830, 16845, 16860, 16875, 16890, 16905, 16920, 16935, 16950, 16965, 16980, 16995, 17010, 17025, 17040, 17055, 17070, 17085, 17100, 17115, 17130, 17145, 17160, 17175, 17190, 17205, 17220, 17235, 17250, 17265, 17280, 17295, 17310, 17325, 17340, 17355, 17370, 17385, 17400, 17415, 17430, 17445, 17460, 17475, 17490, 17505, 17520, 17535, 17550, 17565, 17580, 17595, 17610, 17625, 17640, 17655, 17670, 17685, 17700, 17715, 17730, 17745, 17760, 17775, 17790, 17805, 17820, 17835, 17850, 17865, 17880, 17895, 17910, 17925, 17940, 17955, 17970, 17985, 18000, 18015, 18030, 18045, 18060, 18075, 18090, 18105, 18120, 18135, 18150, 18165, 18180, 18195, 18210, 18225, 18240, 18255, 18270, 18285, 18300, 18315, 18330, 18345, 18360, 18375, 18390, 18405, 18420, 18435, 18450, 18465, 18480, 18495, 18510, 18525, 18540, 18555, 18570, 18585, 18600, 18615, 18630, 18645, 18660, 18675, 18690, 18705, 18720, 18735, 18750, 18765, 18780, 18795, 18810, 18825, 18840, 18855, 18870, 18885, 18900, 18915, 18930, 18945, 18960, 18975, 18990, 19005, 19020, 19035, 19050, 19065, 19080, 19095, 19110, 19125, 19140, 19155, 19170, 19185, 19200, 19215, 19230, 19245, 19260, 19275, 19290, 19305, 19320, 19335, 19350, 19365, 19380, 19395, 19410, 19425, 19440, 19455, 19470, 19485, 19500, 19515, 19530, 19545, 19560, 19575, 19590, 19605, 19620, 19635, 19650, 19665, 19680, 19695, 19710, 19725, 19740, 19755, 19770, 19785, 19800, 19815, 19830, 19845, 19860, 19875, 19890, 19905, 19920, 19935, 19950, 19965, 19980, 19995, 20010, 20025, 20040, 20055, 20070, 20085, 20100, 20115, 20130, 20145, 20160, 20175, 20190, 20205, 20220, 20235, 20250, 20265, 20280, 20295, 20310, 20325, 20340, 20355, 20370, 20385, 20400, 20415, 20430, 20445, 20460, 20475, 20490, 20505, 20520, 20535, 20550, 20565, 20580, 20595, 20610, 20625, 20640, 20655, 20670, 20685, 20700, 20715, 20730, 20745, 20760, 20775, 20790, 20805, 20820, 20835, 20850, 20865, 20880, 20895, 20910, 20925, 20940, 20955, 20970, 20985, 21000, 21015, 21030, 21045, 21060, 21075, 21090, 21105, 21120, 21135, 21150, 21165, 21180, 21195, 21210, 21225, 21240, 21255, 21270, 21285, 21300, 21315, 21330, 21345, 21360, 21375, 21390, 21405, 21420, 21435, 21450, 21465, 21480, 21495, 21510, 21525, 21540, 21555, 21570, 21585, 21600, 21615, 21630, 21645, 21660, 21675, 21690, 21705, 21720, 21735, 21750, 21765, 21780, 21795, 21810, 21825, 21840, 21855, 21870, 21885, 21900, 21915, 21930, 21945, 21960, 21975, 21990, 22005, 22020, 22035, 22050, 22065, 22080, 22095, 22110, 22125, 22140, 22155, 22170, 22185, 22200, 22215, 22230, 22245, 22260, 22275, 22290, 22305, 22320, 22335, 22350, 22365, 22380, 22395, 22410, 22425, 22440, 22455, 22470, 22485, 22500, 22515, 22530, 22545, 22560, 22575, 22590, 22605, 22620, 22635, 22650, 22665, 22680, 22695, 22710, 22725, 22740, 22755, 22770, 22785, 22800, 22815, 22830, 22845, 22860, 22875, 22890, 22905, 22920, 22935, 22950, 22965, 22980, 22995, 23010, 23025, 23040, 23055, 23070, 23085, 23100, 23115, 23130, 23145, 23160, 23175, 23190, 23205, 23220, 23235, 23250, 23265, 23280, 23295, 23310, 23325, 23340, 23355, 23370, 23385, 23400, 23415, 23430, 23445, 23460, 23475, 23490, 23505, 23520, 23535, 23550, 23565, 23580, 23595, 23610, 23625, 23640, 23655, 23670, 23685, 23700, 23715, 23730, 23745, 23760, 23775, 23790, 23805, 23820, 23835, 23850, 23865, 23880, 23895, 23910, 23925, 23940, 23955, 23970, 23985, 24000, 24015, 24030, 24045, 24060, 24075, 24090, 24105, 24120, 24135, 24150, 24165, 24180, 24195, 24210, 24225, 24240, 24255, 24270, 24285, 24300, 24315, 24330, 24345, 24360, 24375, 24390, 24405, 24420, 24435, 24450, 24465, 24480, 24495, 24510, 24525, 24540, 24555, 24570, 24585, 24600, 24615, 24630, 24645, 24660, 24675, 24690, 24705, 24720, 24735, 24750, 24765, 24780, 24795, 24810, 24825, 24840, 24855, 24870, 24885, 24900, 24915, 24930, 24945, 24960, 24975, 24990, 25005, 25020, 25035, 25050, 25065, 25080, 25095, 25110, 25125, 25140, 25155, 25170, 25185, 25200, 25215, 25230, 25245, 25260, 25275, 25290, 25305, 25320, 25335, 25350, 25365, 25380, 25395, 25410, 25425, 25440, 25455, 25470, 25485, 25500, 25515, 25530, 25545, 25560, 25575, 25590, 25605, 25620, 25635, 25650, 25665, 25680, 25695, 25710, 25725, 25740, 25755, 25770, 25785, 25800, 25815, 25830, 25845, 25860, 25875, 25890, 25905, 25920, 25935, 25950, 25965, 25980, 25995, 26010, 26025, 26040, 26055, 26070, 26085, 26100, 26115, 26130, 26145, 26160, 26175, 26190, 26205, 26220, 26235, 26250, 26265, 26280, 26295, 26310, 26325, 26340, 26355, 26370, 26385, 26400, 26415, 26430, 26445, 26460, 26475, 26490, 26505, 26520, 26535, 26550, 26565, 26580, 26595, 26610, 26625, 26640, 26655, 26670, 26685, 26700, 26715, 26730, 26745, 26760, 26775, 26790, 26805, 26820, 26835, 26850, 26865, 26880, 26895, 26910, 26925, 26940, 26955, 26970, 26985, 27000, 27015, 27030, 27045, 27060, 27075, 27090, 27105, 27120, 27135, 27150, 27165, 27180, 27195, 27210, 27225, 27240, 27255,

```

        .word 5
        .word 7

        .text
        .global main
main:
        LDR R3, =primes    @ Load the address for the data in R3
        LDR R0, [R3, #4]   @ Get the next item in the list
end:
        BX LR

```

2. เพิ่มประโยคต่อไปนี้ใน makefile ให้รองรับ Lab7_3

```

Lab7_3:
        gcc -o Lab7_3 Lab7_3.s

```

3. ทำการ make และรันโปรแกรมโดยใช้คำสั่ง

```

$ make Lab7_3
$ ./Lab7_3
$ echo $?

```

G.1.3 การใช้งานตัวแปรอาร์เรย์ชนิด byte

คำสั่ง **LDRB** ทำงานคล้ายกับคำสั่ง **LDR** แต่เป็นการอ่านค่าของตัวแปรอาร์เรย์ชนิด byte

1. สร้างไฟล์ **Lab7_4.s** ตามโค้ดต่อไปนี้ ผู้อ่านสามารถข้ามประโยคคอมเมนต์ได้ เมื่อทำความเข้าใจแต่ละคำสั่งแล้ว

```

        .data
numbers:        .byte 1, 2, 3, 4, 5

        .text
        .global main
main:
        LDR R3, =numbers    @ Get address
        LDRB R0, [R3, #2]   @ Get next two bytes

```

```
end:
    BX LR
```

2. เพิ่มประโยคต่อไปนี้ใน makefile ให้รองรับ Lab7_4

```
Lab7_4:
    gcc -o Lab7_4 Lab7_4.s
```

3. ทำการ make และรันโปรแกรมโดยใช้คำสั่ง

```
$ make Lab7_4
$ ./Lab7_4
$ echo $?
```

G.1.4 การเรียกใช้ฟังก์ชันและตัวแปรชนิดประโยครหัส ASCII

ฟังก์ชันสำเร็จรูปที่เข้าใจง่ายและใช้สำหรับเรียนรู้การพัฒนาโปรแกรมภาษา C เบื้องต้น คือ ฟังก์ชัน printf ซึ่งถูกกำหนดอยู่ในไฟล์เฮดเดอร์ stdio.h ตามตัวอย่างซอร์สโค้ด ในรูปที่ 3.9 และการทดลองที่ 5 ภาคผนวก E ในการทดลองต่อไปนี้ ผู้อ่านจะสังเกตเห็นว่าการเรียกใช้ฟังก์ชัน printf ในภาษาแอสเซมบลี โดยอาศัยตัวแปรชนิดประโยค (String) ในรูปที่ 2.11 โดยใช้คำสำคัญ (Key Word) เหล่านี้ คือ **.ascii** และ **.asciz** ตัวแปรชนิด **asciz** จะมีตัวอักษรพิเศษ เรียกว่า อักขร-null **NULL** หรือ **/0** ปิดท้ายประโยคเสมอ และอักขร **NULL** จะมีรหัส ASCII เท่ากับ **00₁₆** ตามตารางรหัสแอสกี ในรูปที่ 2.12

1. กรอกคำสั่งต่อไปนี้ลงในไฟล์ใหม่ชื่อ **Lab7_5.s** และทำความเข้าใจประโยคคอมเมนต์แต่ละบรรทัด

```
.data
.balign 4
question: .asciz "What is your favorite number?"

.balign 4
message: .asciz "%d is a great number \n"

.balign 4
pattern: .asciz "%d"

.balign 4
number: .word 0
```

```
.balign 4
lr_bu: .word 0

.text    @ Text segment begins here

@ Used by the compiler to tell libc where main is located
.global main
.func main

main:
    @ Backup the value inside Link Register
    LDR R1, addr_lr_bu
    STR lr, [R1]    @ Mem[addr_lr_bu] <- LR

    @ Load and print question
    LDR R0, addr_question
    BL printf

    @ Define pattern to scanf and where to store number
    LDR R0, addr_pattern
    LDR R1, addr_number
    BL scanf

    @ Print the message with number
    LDR R0, addr_message
    LDR R1, addr_number
    LDR R1, [R1]
    BL printf

    @ Load the value of lr_bu to LR
    LDR lr, addr_lr_bu
    LDR lr, [lr]    @ LR <- Mem[addr_lr_bu]
    BX lr

@ Define addresses of variables
```



```

addr_question:  .word question
addr_message:  .word message
addr_pattern:  .word pattern
addr_number:  .word number
addr_lr_bu:  .word lr_bu

@ Declare printf and scanf functions to be linked with
.global printf
.global scanf

```

2. เพิ่มประโยคใน makefile ให้รองรับ Lab7_5

```

Lab7_5:
    gcc -o Lab7_5 Lab7_5.s

```

3. ทำการ make และรันโปรแกรมโดยใช้คำสั่ง

```

$ make Lab7_5
$ ./Lab7_5

```

G.2 การสร้างฟังก์ชันเสริมด้วยภาษาแอสเซมบลี

หัวข้อที่ 4.8 อธิบายโพลีการทำงานของฟังก์ชัน โดยใช้งานรีจิสเตอร์ R0 - R12 ดังนี้

- รีจิสเตอร์ R0, R1, R2, และ R3 การส่งผ่านพารามิเตอร์ผ่านทางรีจิสเตอร์ R0 ถึง R3 ตามลำดับไปยังฟังก์ชันที่ถูกเรียก (Callee Function) ฟังก์ชันบางตัวต้องการจำนวนพารามิเตอร์มากกว่า 4 ค่า โปรแกรมเมอร์สามารถส่งพารามิเตอร์ผ่านทางสแต็กโดยคำสั่ง PUSH หรือคำสั่งที่ใกล้เคียง
- รีจิสเตอร์ R0 สำหรับรีเทิร์นหรือส่งค่ากลับไปหาฟังก์ชันผู้เรียก (Caller Function)
- R4 - R12 สำหรับการใช้งานทั่วไป การใช้งานรีจิสเตอร์เหล่านี้ ควรตั้งค่าเริ่มต้นก่อนแล้วจึงสามารถนำค่าไปคำนวณต่อได้
- รีจิสเตอร์เฉพาะ ได้แก่ Stack Pointer (SP หรือ R13) Link Register (LR หรือ R14) และ Program Counter (PC หรือ R15) โปรแกรมเมอร์จะต้องเก็บค่าของรีจิสเตอร์เหล่านี้เก็บไว้ (Back up) ในสแต็กโดยเฉพาะรีจิสเตอร์ LR ก่อนเรียกใช้ฟังก์ชัน LR ตามที่อธิบายในหัวข้อที่ 4.8.2

ผู้อ่านสามารถสำเนาซอร์สโค้ดในการทดลองที่แล้วมาปรับแก้เป็นการทดลองนี้ได้

1. ปรับแก้ Lab7_5.s ที่มีให้เป็นไฟล์ใหม่ชื่อ Lab7_6.s ดังต่อไปนี้

```
.data
@ Define all the strings and variables
.balign 4
get_num_1: .asciz "Number 1 : \n"

.balign 4
get_num_2: .asciz "Number 2 : \n"

@ printf and scanf use %d in decimal numbers
.balign 4
pattern: .asciz "%d"

@ Declare and initialize variables: num_1 and num_2
.balign 4
num_1: .word 0

.balign 4
num_2: .word 0

@ Output message pattern
.balign 4
output: .asciz "Resulf of %d + %d = %d\n"

@ Variables to backup link register
.balign 4
lr_bu: .word 0

.balign 4
lr_bu_2: .word 0

.text
sum_func:
    @ Save (Store) Link Register to lr_bu_2
    LDR R2, addr_lr_bu_2
```

```

STR lr, [R2]      @ Mem[addr_lr_bu_2] <- LR

@ Sum values in R0 and R1 and return in R0
ADD R0, R0, R1

@ Load Link Register from back up 2
LDR lr, addr_lr_bu_2
LDR lr, [lr]      @ LR <- Mem[addr_lr_bu_2]

BX lr

@ address of Link Register back up 2
addr_lr_bu_2: .word lr_bu_2

@ main function
.global main

main:
    @ Store (back up) Link Register
    LDR R1, addr_lr_bu
    STR lr, [R1]      @ Mem[addr_lr_bu] <- LR

    @ Print Number 1 :
    LDR R0, addr_get_num_1
    BL printf

    @ Get num_1 from user via keyboard
    LDR R0, addr_pattern
    LDR R1, addr_num_1
    BL scanf

    @ Print Number 2 :
    LDR R0, addr_get_num_2
    BL printf

```

```

    @ Get num_2 from user via keyboard
    LDR R0, addr_pattern
    LDR R1, addr_num_2
    BL scanf

    @ Pass values of num_1 and num_2 to sum_func
    LDR R0, addr_num_1
    LDR R0, [R0]      @ R0 <- Mem[addr_num_1]
    LDR R1, addr_num_2
    LDR R1, [R1]      @ R1 <- Mem[addr_num_2]
    BL sum_func

    @ Copy returned value from sum_func to R3
    MOV R3, R0      @ to printf

    @ Print the output message, num_1, num_2 and result
    LDR R0, addr_output
    LDR R1, addr_num_1
    LDR R1, [R1]
    LDR R2, addr_num_2
    LDR R2, [R2]
    BL printf

    @ Restore Link Register to return
    LDR lr, addr_lr_bu
    LDR lr, [lr]      @ LR <- Mem[addr_lr_bu]
    BX lr

    @ Define pointer variables
    addr_get_num_1: .word get_num_1
    addr_get_num_2: .word get_num_2
    addr_pattern:   .word pattern
    addr_num_1:     .word num_1
    addr_num_2:     .word num_2
    addr_output:    .word output

```

```
addr_lr_bu:      .word lr_bu
```

```
@ Declare printf and scanf functions to be linked with
.global printf
.global scanf
```

2. เพิ่มประโยคใน makefile ให้รองรับ Lab7_6 ดังนี้

```
Lab7_6:
    gcc -o Lab7_6 Lab7_6.s
```

3. ทำการ make และรันโปรแกรมโดยใช้คำสั่ง

```
$ make Lab7_6
$ ./Lab7_6
```

4. ระบุซอร์สโค้ดใน Lab7_6.s ว่าตรงกับประโยคภาษา C ต่อไปนี้

```
int num1, num2
```

5. ระบุซอร์สโค้ดใน Lab7_6.s ว่าตรงกับประโยคภาษา C ต่อไปนี้ `sum = num1 + num2`

6. มีการแบ็กอัปค่าของ LR ลงในสแต็กหรือไม่ หากไม่มีแล้วในการทดลองเก็บค่าของ LR ไว้ที่ใด เพราะเหตุใด

7. วิธีการแบ็กอัปค่า LR ในการทดลองสามารถใช้กับฟังก์ชันเรียกซ้ำ (Recursive Function) ได้หรือไม่ เพราะเหตุใด

G.3 กิจกรรมท้ายการทดลอง

1. เปรียบเทียบการเรียกใช้ฟังก์ชัน printf และ scanf ในภาษา C จากการทดลองที่ 5 ภาคผนวก E กับการทดลองนี้ด้านการส่งพารามิเตอร์

4. ระบุซอร์สโค้ดใน Lab7_6.s ว่าตรงกับประโยคภาษา C ต่อไปนี้

```
int num1, num2
```

ตอบ ตรงส่วน num_1: .word 0 และ num_2: .word 0

5. ระบุซอร์สโค้ดใน Lab7_6.s ว่าตรงกับประโยคภาษา C ต่อไปนี้ $sum = num1 + num2$

ตอบ

```
LDR R0, addr_num_1   ในลวดต่ำแอดเรสตรงของ addr_num_1 มาไว้ใน R0
LDR R0, [R0]          ตั้งค่าที่ R0 ให้ไปมีค่าที่ R0 (หมายถึงแอดเรสตัวที่ 1 num1 ของบิตชีน sum_func)
LDR R1, addr_num_2   ในลวดต่ำแอดเรสตรงของ addr_num_2 มาไว้ใน R1
LDR R1, [R1]          ตั้งค่าที่ R1 ให้ไปมีค่าที่ R1 (หมายถึงแอดเรสตัวที่ 2 num2 ของบิตชีน sum_func)
BL sum_func           ทำการเรียกลังฟังก์ชัน sum_func ด้วยพารามิเตอร์คือค่า
MOV R3, R0            นำค่าที่ฟังก์ชัน sum_func คืนค่ากลับมาผ่าน R0 มาใส่ใน R3 (sum)
```

6. มีการแบ็กอัปค่าของ LR ลงในสแต็กหรือไม่ หากไม่มีแล้วในการทดลองเก็บค่าของ LR ไว้ที่ได้เพราะเหตุใด

ตอบ

ไม่มีการแบ็กอัปค่า LR ลงในสแต็ก เพราะว่ามันก็ค่า LR เก็บไว้ที่ ตัวแปร global แทน
ถ้าไม่มีการแบ็กอัปค่าของ LR ก่อนที่จะเรียกโปรแกรมย่อย จะทำให้ค่า LR ถูกเปลี่ยน และเมื่อต้องการกลับมาจุด
ที่โปรแกรมย่อยดูเรียก จะทำให้กลับมาจุดเดิมไม่ได้ เพราะค่า LR ize ได้ถูกเปลี่ยนไป ทำให้เกิด error ส่วนหนึ่ง
เพราะอาจทำให้โปรแกรมไปทำ instruction อื่น ที่ไม่ถูกต้อง จนทำให้โปรแกรมผิดพลาด หรือ หยุดทำงาน

7. วิธีการแบ็กอัปค่า LR ในการทดลองสามารถใช้กับฟังก์ชันเรียกซ้ำ (Recursive Function) ได้หรือไม่ เพราะเหตุใด

ตอบ

ไม่ได้ เพราะการแบ็กอัปค่า LR ผ่านฟังก์ชันเรียกซ้ำ จะต้องบันทึกค่าแบบ recursive

2. จงบอกความแตกต่างระหว่างการส่งค่าพารามิเตอร์แบบ Pass by Values และ Pass by Reference
3. จงยกตัวอย่างการเรียกใช้ฟังก์ชัน printf ด้วยการส่งค่าพารามิเตอร์แบบ Pass by Values
4. จงยกตัวอย่างการเรียกใช้ฟังก์ชัน scanf ด้วยการส่งค่าพารามิเตอร์แบบ Pass by Reference
5. จงพัฒนาโปรแกรมด้วยภาษา C เพื่อรับตัวเลขจำนวน 2 ตัวจากผู้ใช้ผ่านทางคีย์บอร์ด เรียกว่า A และ B แล้วคำนวณและแสดงผลลัพธ์ ตามตารางต่อไปนี้ "A % B = <Result>".

Input	Output
5 2	5 % 2 = 1
18 6	18 % 6 = 0
5 10	5 % 10 = 5
10 5	10 % 5 = 0

6. จงเปรียบเทียบฟังก์ชัน scanf และ printf ในการทดลองนี้กับการทดลองที่ 5 ภาคผนวก E
7. จงพัฒนาโปรแกรมด้วยภาษา C เพื่อรับตัวเลขจำนวน 2 ตัวจากผู้ใช้ผ่านทางคีย์บอร์ด เรียกว่า A และ B แล้วคำนวณหาค่า หกร่วมมาก (Greatest Common Divisor) หรือ หรม (GCD) และแสดงผลลัพธ์ตามตัวอย่างในตารางต่อไปนี้

Input	Output
5 2	1
18 6	6
49 42	7
81 18	9

8. จงพัฒนาโปรแกรมด้วยภาษา Assembly เพื่อรับตัวเลขจำนวน 2 ตัวจากผู้ใช้ผ่านทางคีย์บอร์ด เรียกว่า A และ B และแสดงผลลัพธ์ A หรือ B ที่มีค่ามากกว่าด้วยคำสั่งภาษาแอสเซมบลี
9. จงพัฒนาโปรแกรมด้วยภาษา Assembly เพื่อรับตัวเลขจำนวน 2 ตัวจากผู้ใช้ผ่านทางคีย์บอร์ด เรียกว่า A และ B และแสดงผลลัพธ์ค่า A modulus B ซึ่งเท่ากับ ค่าเศษจากการคำนวณ A/B ด้วยคำสั่งภาษาแอสเซมบลี
10. จงพัฒนาโปรแกรมด้วยภาษา Assembly เพื่อรับตัวเลขจำนวน 2 ตัวจากผู้ใช้ผ่านทางคีย์บอร์ด เรียกว่า A และ B แล้วคำนวณหาค่า หกร่วมมาก (Greatest Common Divisor) หรือ หรม (GCD) ด้วยคำสั่งภาษาแอสเซมบลีและแสดงผลลัพธ์ ตามตารางในข้อ 6

G.3 กิจกรรมท้ายการทดลอง

1. จงเปรียบเทียบการเรียกใช้ฟังก์ชัน printf และ scanf ในภาษา C จากการทดลองที่ 5 ภาคผนวก E
กับการทดลองนี้ด้านการส่งพารามิเตอร์

ตอบ ในภาษา C printf พารามิเตอร์ ตัวแรก จะเป็น Format ตัวที่ 2 จะเป็นตัวแปร ที่ใช้ในการแสดงค่า
scanf พารามิเตอร์ ตัวแรก จะเป็น Format และตัวที่ 2 จะเป็น pointer ของตัวแปร ที่ scanf อ่านได้
และจะเอาค่าไปใส่
ใน Assembly Lab5 ผู้ใช้งานจะต้องบอกค่า พารามิเตอร์ ตัวแรก ไว้ในรีจิสเตอร์ R0, R1, R2
ด้วยตนเอง จากนั้นค่อยทำ การเรียกฟังก์ชัน

2. จงบอกความแตกต่างระหว่างการส่งค่าพารามิเตอร์แบบ Pass by Values และ Pass by Reference

ตอบ การส่งแบบ Pass by Values จะเป็นการส่งค่าจริงๆไปเลย โดยจะคัดลอก พารามิเตอร์ ไปไว้ใน รีจิสเตอร์
เพื่อส่งค่าไปยัง ฟังก์ชันย่อย และ ฟังก์ชันย่อยนั้น ก็จะใช้ค่าของรีจิสเตอร์ ที่ส่งไปต่อทันที โดยจะไม่ส่งกลับ นอกฟังก์ชัน
ถ้าเป็นกรส่งแบบ Pass by reference จะเป็นการส่งที่อยู่ของตัวแปร เมื่อเรียก ฟังก์ชันย่อย ฟังก์ชันย่อย ก็จะได้อำนาจไปของตัวแปร
ที่อยู่ในรีจิสเตอร์ ตัวนั้นแหละ ที่การแก้ไขค่าที่ รีจิสเตอร์ ก็จะมีผลทันที และเมื่อจบ ฟังก์ชันด้วย

6. จงเปรียบเทียบฟังก์ชัน scanf และ printf ในการทดลองนี้กับการทดลองที่ 5 ภาคผนวก E

ตอบ การทำงานของ scanf และ printf ใน Lab5 และ Lab7 เป็นการทำงาน ที่เหมือนกัน
เพราะใช้ ฟังก์ชันตัวเดียวกัน

