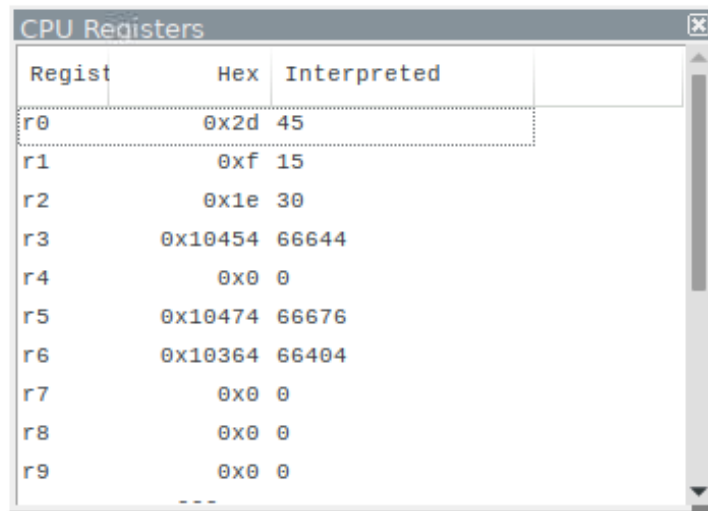
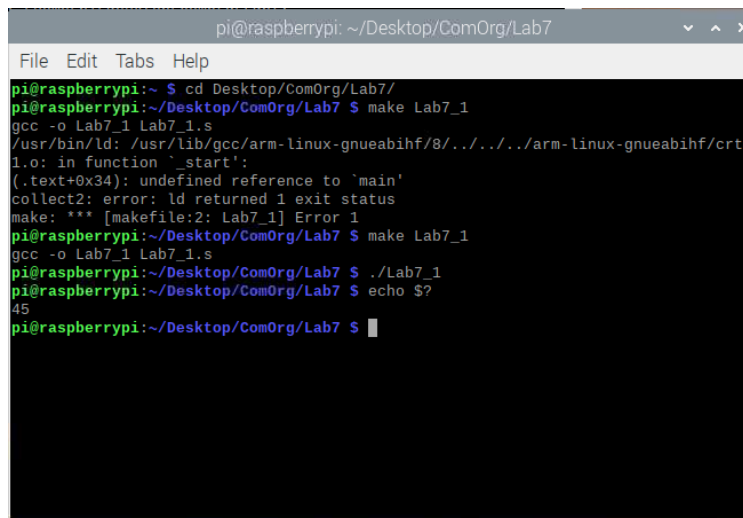


LAB 7 : 62010694 & 62010718

Lab7_1



Regist	Hex	Interpreted
r0	0x2d	45
r1	0xf	15
r2	0x1e	30
r3	0x10454	66644
r4	0x0	0
r5	0x10474	66676
r6	0x10364	66404
r7	0x0	0
r8	0x0	0
r9	0x0	0



```
pi@raspberrypi: ~/Desktop/ComOrg/Lab7
File Edit Tabs Help
pi@raspberrypi:~ $ cd Desktop/ComOrg/Lab7/
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ make Lab7_1
gcc -o Lab7_1 Lab7_1.s
/usr/bin/ld: /usr/lib/gcc/arm-linux-gnueabi/hf/8/../../../../arm-linux-gnueabi/hf/crt
1.o: in function `_start':
(.text+0x34): undefined reference to `main'
collect2: error: ld returned 1 exit status
make: *** [makefile:2: Lab7_1] Error 1
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ make Lab7_1
gcc -o Lab7_1 Lab7_1.s
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ ./Lab7_1
45
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ echo $?
45
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $
```

ตัวแปร fifteen เก็บค่า 15 , ตัวแปร thirty เก็บค่า 30 , เก็บค่าใน R1 , R2 ตามลำดับ แล้ว R0
= R1 + R2 และ return

ข้อมูลเพิ่มเติมเกี่ยวกับคำสั่ง SWI (Software Interrupt) : การอินเทอร์รัพท์จากซอฟต์แวร์ (Software interrupts) การอินเทอร์รัพท์จากซอฟต์แวร์เริ่มต้นโดยการปฏิบัติตามคำสั่งที่ถูกเรียกไปยังโปรแกรมย่อย ซึ่งถูกกำหนดโดยโปรแกรมที่ถูกเขียนไว้

SWI ใช้เพื่อเข้าถึงโหมด Supervisor (svc) สำหรับระบบปฏิบัติการ เนื่องจากในโหมดนี้จะไม่มีการจำกัดในการเข้าถึงระบบ เราสามารถเข้าถึงโหมดนี้ได้โดยทางซอฟต์แวร์อินเทอร์รัพท์ (SWI)

Lab7_2

CPU Registers		
Regist	Hex	Interpreted
r0	0x2d	45
r1	0xf	15
r2	0x1e	30
r3	0x1e	30
r4	0x0	0
r5	0x1048c	66700
r6	0x10364	66404
r7	0x0	0
r8	0x0	0
r9	0x0	0

```
pi@raspberrypi: ~/Desktop/ComOrg/Lab7
File Edit Tabs Help
pi@raspberrypi:~$ cd Desktop/ComOrg/Lab7/
pi@raspberrypi:~/Desktop/ComOrg/Lab7$ make Lab7_1
gcc -o Lab7_1 Lab7_1.s
/usr/bin/ld: /usr/lib/gcc/arm-linux-gnueabi/8/../../../../arm-linux-gnueabi/crt
1.o: in function `_start':
(.text+0x34): undefined reference to `main'
collect2: error: ld returned 1 exit status
make: *** [makefile:2: Lab7_1] Error 1
pi@raspberrypi:~/Desktop/ComOrg/Lab7$ make Lab7_1
gcc -o Lab7_1 Lab7_1.s
pi@raspberrypi:~/Desktop/ComOrg/Lab7$ ./Lab7_1
pi@raspberrypi:~/Desktop/ComOrg/Lab7$ echo $?
45
pi@raspberrypi:~/Desktop/ComOrg/Lab7$ make Lab7_2
gcc -o Lab7_2 Lab7_2.s
pi@raspberrypi:~/Desktop/ComOrg/Lab7$ ./Lab7_2
pi@raspberrypi:~/Desktop/ComOrg/Lab7$ echo $?
45
pi@raspberrypi:~/Desktop/ComOrg/Lab7$
```

ตัวแปร fifteen เก็บค่า 0 , ตัวแปร thirty เก็บค่า 0 , เก็บค่าใน R1 , R2 ตามลำดับ แล้ว R3 = 15 , แล้วใส่ไปในที่ address_fifteen , R3 = 30 , แล้วใส่ไปในที่ address_thirty , R0 = R1

+ R2 แล้ว return

ผลลัพธ์เหมือนกัน

Lab7_3

```
pi@raspberrypi: ~/Desktop/ComOrg/Lab7
File Edit Tabs Help
pi@raspberrypi:~$ cd Desktop/ComOrg/Lab7/
pi@raspberrypi:~/Desktop/ComOrg/Lab7$ make Lab7_1
gcc -o Lab7_1 Lab7_1.s
/usr/bin/ld: /usr/lib/gcc/arm-linux-gnueabi/8/../../../../arm-linux-gnueabi/crt
1.o: in function `_start':
(.text+0x34): undefined reference to `main'
collect2: error: ld returned 1 exit status
make: *** [makefile:2: Lab7_1] Error 1
pi@raspberrypi:~/Desktop/ComOrg/Lab7$ make Lab7_1
gcc -o Lab7_1 Lab7_1.s
pi@raspberrypi:~/Desktop/ComOrg/Lab7$ ./Lab7_1
45
pi@raspberrypi:~/Desktop/ComOrg/Lab7$ make Lab7_2
gcc -o Lab7_2 Lab7_2.s
pi@raspberrypi:~/Desktop/ComOrg/Lab7$ ./Lab7_2
45
pi@raspberrypi:~/Desktop/ComOrg/Lab7$ make Lab7_3
gcc -o Lab7_3 Lab7_3.s
pi@raspberrypi:~/Desktop/ComOrg/Lab7$ ./Lab7_3
3
pi@raspberrypi:~/Desktop/ComOrg/Lab7$
```

prime เก็บเลข 2 3 5 7 , R3 โหลดค่า address ของ primes มา , R0 โหลดค่า R3 มา โดย
เลื่อนตำแหน่งไป 4 bytes (เอาตัวถัดไป) แล้ว return

Lab7_4

```
pi@raspberrypi: ~/Desktop/ComOrg/Lab7
File Edit Tabs Help
(.text+0x34): undefined reference to `main'
collect2: error: ld returned 1 exit status
make: *** [makefile:2: Lab7_1] Error 1
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ make Lab7_1
gcc -o Lab7_1 Lab7_1.s
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ ./Lab7_1
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ echo $?
45
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ make Lab7_2
gcc -o Lab7_2 Lab7_2.s
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ ./Lab7_2
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ echo $?
45
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ make Lab7_3
gcc -o Lab7_3 Lab7_3.s
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ ./Lab7_3
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ echo $?
3
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ make Lab7_4
gcc -o Lab7_4 Lab7_4.s
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ ./Lab7_4
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ echo $?
3
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $
```

numbers เก็บเลข 1 2 3 4 5 เป็น bytes , R3 โหลดค่า address ของ numbers มา , R0
โหลดค่า R3 มา โดยเลื่อนตำแหน่งไป 2 bytes แล้ว return

Lab7_5

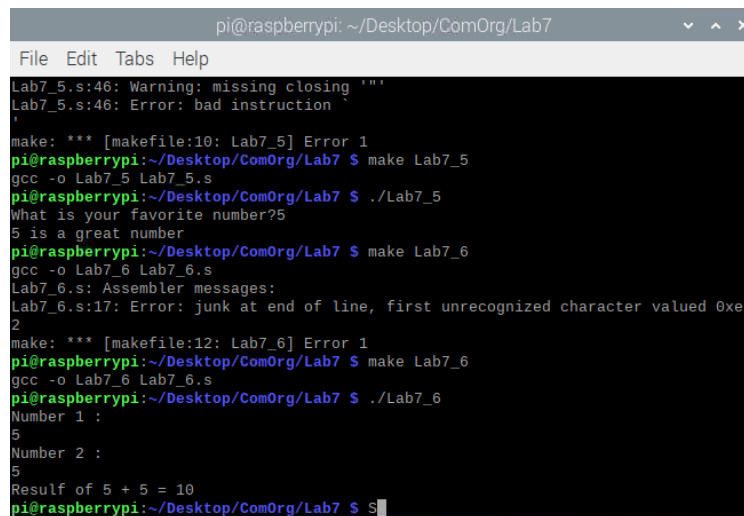
```
pi@raspberrypi: ~/Desktop/ComOrg/Lab7
File Edit Tabs Help
Lab7_5.s:13: Error: junk at end of line, first unrecognized character is '@'
Lab7_5.s:17: Error: junk at end of line, first unrecognized character is '@'
Lab7_5.s:19: Error: garbage following instruction -- `str lr, [R1] @ Mem[addr_lr_bu] <- LR'
Lab7_5.s:20: Error: junk at end of line, first unrecognized character is '@'
Lab7_5.s:24: Error: junk at end of line, first unrecognized character is '@'
Lab7_5.s:28: Error: junk at end of line, first unrecognized character is '@'
Lab7_5.s:33: Error: junk at end of line, first unrecognized character is '@'
Lab7_5.s:35: Error: garbage following instruction -- `ldr lr, [lr] @ LR <- Mem[addr_lr_bu]'
Lab7_5.s:36: Error: garbage following instruction -- `bx lr @ Return to main function'
Lab7_5.s:37: Error: junk at end of line, first unrecognized character is '@'
Lab7_5.s:43: Error: junk at end of line, first unrecognized character is '@'
Lab7_5.s:46: Warning: missing closing '"'
Lab7_5.s:46: Error: bad instruction `
'
make: *** [makefile:10: Lab7_5] Error 1
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ make Lab7_5
gcc -o Lab7_5 Lab7_5.s
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ ./Lab7_5
What is your favorite number?5
5 is a great number
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $
```

กำหนด question เป็น string ที่มีค่า What is your favorite number? , กำหนด message
เป็น string ที่มีค่า %d is a great number\n , กำหนด pattern เป็น string ที่มีค่า %d ,
กำหนด number เป็น word มีค่า 0 , กำหนด lr_bu เป็น word มีค่า 0 , R1 โหลดค่า

address ของ lr_bu มา , เก็บค่าของ R1 ใน lr , R0 โหลดค่า address ของ question มา ,
สั่ง print ตัวแปร question , R0 โหลดค่า address ของ pattern มา , R1 โหลดค่า address
ของ number มา , รับค่าตัวเลขจาก user , R0 โหลดค่า address ของ message มา , R1
โหลดค่า address ของ number มา , R1 โหลดค่าของ R1 มาเก็บไว้ , สั่ง print message
และ number , lr โหลดค่า address ของ lr_bu , lr โหลดค่าของ lr มา , ประกาศที่อยู่ของตัว
แปร มาเป็นตัวแปรแยกไว้ , ประกาศการใช้งาน printf และ scanf

Echo \$? มีไว้เพื่อ มันคือคำสั่งแสดงผลใน terminal โดยนำค่ามาจาก R0

Lab7_6



```
pi@raspberrypi: ~/Desktop/ComOrg/Lab7
File Edit Tabs Help
Lab7_5.s:46: Warning: missing closing '"'
Lab7_5.s:46: Error: bad instruction `
'
make: *** [makefile:10: Lab7_5] Error 1
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ make Lab7_5
gcc -o Lab7_5 Lab7_5.s
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ ./Lab7_5
What is your favorite number?5
5 is a great number
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ make Lab7_6
gcc -o Lab7_6 Lab7_6.s
Lab7_6.s: Assembler messages:
Lab7_6.s:17: Error: junk at end of line, first unrecognized character valued 0xe
2
make: *** [makefile:12: Lab7_6] Error 1
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ make Lab7_6
gcc -o Lab7_6 Lab7_6.s
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $ ./Lab7_6
Number 1 :
5
Number 2 :
5
Result of 5 + 5 = 10
pi@raspberrypi:~/Desktop/ComOrg/Lab7 $
```

เก็บ string ที่มีค่า Number 1 : \n , เก็บ string ที่มีค่า Number 2 : \n , เก็บค่า
string ที่มีค่า %d , เก็บ word ที่มีค่า 0 , เก็บ word ที่มีค่า 0 , เก็บ string ที่มีค่า Result of
%d + %d =%d\n , เก็บ word ที่มีค่า 0 , เก็บ word ที่มีค่า 0 , ส่วนของ sum_func , เอาค่าที่
ได้มา มาบวกกัน , print get_num_1 , รับค่าจาก user , print get_num_2 , รับค่าจาก user
มาเก็บใน num2 , นำ num1 กับ num2 ไปเข้า sum_func เพื่อบวกกัน , ประกาศที่อยู่ของตัว
แปร มาเป็นตัวแปรแยกไว้ , ประกาศการใช้งาน printf และ scanf

4. ระบุซอร์สโค้ดใน Lab7_6.s ว่าตรงกับประโยคภาษา C ต่อไปนี้

```
int num1, num2
```

```
.balign 4
```

```
num_1: .word 0
```

```
.balign 4
```

```
num_2: .word 0
```

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

```
ADD R0,R0,R1
```

6. เหตุใดจึงผู้อ่านจึงไม่ต้องใช้คำสั่ง `echo $?` แล้ว

เพราะเราได้สั่งให้ `printf` ผลลัพธ์แล้ว ไม่จำเป็นต้อง `echo` อีก

6. จงพัฒนาโปรแกรมด้วยภาษา C เพื่อรับตัวเลขจำนวน 2 ตัวจากผู้ใช้ผ่านทางคีย์บอร์ด เรียกว่า A และ B แล้วคำนวณหาค่า หาร่วมมาก (Greatest Common Divisor) หรือ หรม (GCD) และแสดงผลลัพธ์ตามตัวอย่างในตารางต่อไปนี้

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

```

#include <stdio.h>
int main()
{
    int n1, n2, i, gcd;

    printf("Enter two integers: ");
    scanf("%d %d", &n1, &n2);

    for(i=1; i <= n1 && i <= n2; ++i)
    {
        if(n1%i==0 && n2%i==0)
            gcd = i;
    }

    printf("G.C.D of %d and %d is %d", n1, n2, gcd);

    return 0;
}

```

7. จงพัฒนาโปรแกรมด้วยภาษา Assembly เพื่อรับตัวเลขจำนวน 2 ตัวจากผู้ใช้ผ่านทางคีย์บอร์ด เรียกว่า A และ B และแสดงผลลัพธ์ A หรือ B ที่มีค่ามากกว่าด้วยคำสั่งภาษาแอสเซมบลี

```

        .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 "%d is the most value between two int"
@ Variables to backup link register
        .balign 4
lr_bu: .word 0

@ 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

    LDR R0,addr_num_1
    LDR R0,[R0]
    LDR R1,addr_num_2
    LDR R1,[R1]

    BLE end
    MOV R1,R0
    B end

end:
    LDR R0,addr_output
    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

```

ทำการเทียบ num_1 ซึ่งแทนด้วย R0 , num_2 ซึ่งแทนด้วย R1 โดยใช้ CMP R0,R1 แล้วหากน้อยกว่าหรือเท่ากับให้กระโดดไป end แล้วปรี้น output ออกมา