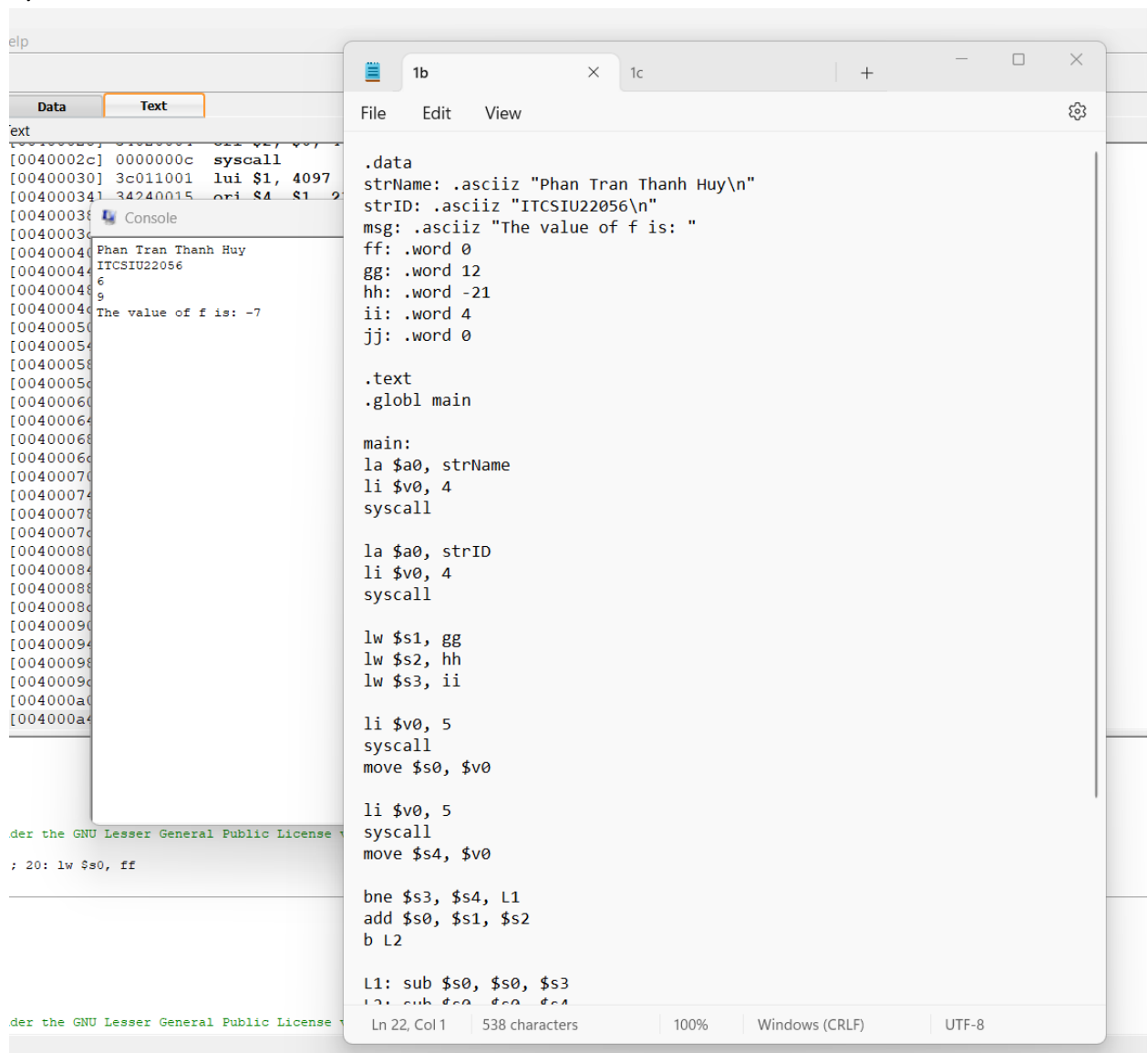


Laboratory Session 3

Conditional Execution

1.

```
a/  
If(s3 != s4)  
S0 -= s3  
Else s0 = s1 + s2  
S0 -= s4  
b/
```



```
.data  
strName: .asciiz "Phan Tran Thanh Huy\n"  
strID: .asciiz "ITCSIU22056\n"  
msg: .asciiz "The value of f is: "  
ff: .word 0  
gg: .word 12  
hh: .word -21  
ii: .word 4  
jj: .word 0  
  
.text  
.globl main  
  
main:  
la $a0, strName  
li $v0, 4  
syscall  
  
la $a0, strID  
li $v0, 4  
syscall  
  
lw $s1, gg  
lw $s2, hh  
lw $s3, ii  
  
li $v0, 5  
syscall  
move $s0, $v0  
  
li $v0, 5  
syscall  
move $s4, $v0  
  
bne $s3, $s4, L1  
add $s0, $s1, $s2  
b L2  
  
L1: sub $s0, $s0, $s3  
L2: sub $s0, $s0, $s4  
li $v0, 1  
syscall  
exit
```

c/

The screenshot shows a debugger interface with two main windows. The left window displays assembly code with memory addresses, hex values, and mnemonics. The right window shows the assembly code in a more readable format, including data labels and control flow instructions.

Assembly Code (Left Window):

```
[00400018] 00000000 nop
[0040001c] 3402000a ori $2, $0, 10
[00400020] 0000000c syscall
[00400024] Phan Tran Thanh Huy
[00400028] ITCIU22056
[00400032] The value of f is: -7
[00400036]
[0040003a]
[0040003e]
[00400040]
[00400044]
[00400048]
[0040004c]
[00400050]
[00400054]
[00400058]
[0040005c]
[00400060]
[00400064]
[00400068]
[0040006c]
[00400070]
[00400074]
[00400078]
[0040007c]
[00400080]
[00400084]
[00400088]
[0040008c]
[00400090]
```

Assembly Code (Right Window):

```
.data
strName: .asciiz "Phan Tran Thanh Huy\n"
strID: .asciiz "ITCIU22056\n"
msg: .asciiz "The value of f is: "
n: .word 5
x: .word 6, 12, -21, 4, 9

.text
.globl main

main:
la $a0, strName
li $v0, 4
syscall

la $a0, strID
li $v0, 4
syscall

la $t1, x
lw $s0, 0($t1)
lw $s1, 4($t1)
lw $s2, 8($t1)
lw $s3, 12($t1)
lw $s4, 16($t1)

bne $s3, $s4, L1
add $s0, $s1, $s2
b L2

L1: sub $s0, $s0, $s3
L2: sub $s0, $s0, $s4

la $a0, msg
li $v0, 4
syscall

move $a0, $s0
li $v0, 1
syscall
```

Console (Left Window):

```
Phan Tran Thanh Huy
ITCIU22056
The value of f is: -7
```

Status Bar (Bottom):

Ln 25, Col 15 | 500 characters | 100% | Windows (CRLF) | UTF-8

2. Sum function

The image shows a code editor window titled "PhanTranThanhHuy_Sum" with a menu bar (File, Edit, View) and a settings icon. The editor contains MIPS assembly code for a program that calculates the sum of five integers. To the left, a console window displays the program's execution output.

```
.data
strName: .ascii "Phan Tran Thanh Huy\n"
strID: .ascii "ITCSIU22056\n"
input: .ascii "Enter an integer: "
output: .ascii "The Sum is: "
strLine: .ascii "\n"
sum: .word 0

.text
.globl main

main:
    la $a0, strName
    li $v0, 4
    syscall

    la $a0, strID
    li $v0, 4
    syscall

    li $t0, 1
    li $s0, 2

LOOP:
    la $a0, input
    li $v0, 4
    syscall

    li $v0, 5
    syscall
    move $t0, $v0
    beq $t0, $0, EXIT

    lw $s1, sum

    div $t0, $s0
    mfhi $t2

    beq $t2, $0, CASE
    mul $t2, $t0, $t0

EXIT:
    li $v0, 10
    syscall
```

Console Output:

```
Phan Tran Thanh Huy
ITCSIU22056
Enter an integer: 3
Enter an integer: 2
Enter an integer: 4
Enter an integer: 5
Enter an integer: 0
The Sum is: 172
```

Footer information: NU Lesser General Public License ver 3.0, Ln 20, Col 1, 665 characters, 100%, Windows (CRLF), UTF-8.

3. Multiplication Function

```
.data
strName: .asciiz "Phan Tran Thanh Huy\n"
strID: .asciiz "ITCSIU22056\n"
input1: .asciiz "Enter the first integer: "
input2: .asciiz "Enter the second integer: "
output: .asciiz "The Multi is: "
strLine: .asciiz "\n"

.text
.globl main

main:
la $a0, strName
li $v0, 4
syscall

la $a0, strID
li $v0, 4
syscall

la $a0, input1
li $v0, 4
syscall
li $v0, 5
syscall
move $t0, $v0

la $a0, input2
li $v0, 4
syscall
li $v0, 5
syscall
move $t1, $v0

li $t2, 0
blt $t1, $0, LOOP1
LOOP:
beq $t1, $0, EXIT
add $t2, $t2, $t0
addi $t1, $t1, 1
EXIT:
li $a0, output
li $v0, 4
syscall
li $v0, 10
syscall
```

Ln 34, Col 1 | 704 characters | 100% | Windows (CRLF) | UTF-8

4.

The screenshot shows a code editor with two tabs: 'PhanTranThanhHuy_Multiply' and 'PhanTranThanhHuy_Lai'. The 'PhanTranThanhHuy_Lai' tab is active, displaying assembly code. To the left, a console window shows the output of the program, including memory addresses and data values. The assembly code defines variables for name, ID, line, point, array, size, brr, and char, and implements a main function that prints these values and a loop that increments a counter.

```
.data
strName: .asciiz "Phan Tran Thanh Huy\n"
strID: .asciiz "ITCSIU22056\n"
strLine: .asciiz "\n"
strPoint: .asciiz ": "
arr: .space 100
size: .word 0
brr: .space 30
char: .space 1

.text
.globl main

main:
la $a0, strName
li $v0, 4
syscall

la $a0, strID
li $v0, 4
syscall

la $a0, arr
li $a1, 100
li $v0, 8
syscall

la $t0, arr
lw $s0, size

LOOP:
lb $t1, 0($t0)
addi $s0, $s0, 1
addi $t0, $t0, 1
bne $t1, 0, LOOP

addi $s0, $s0, -2
sw $s0, size

la $t0, arr
```

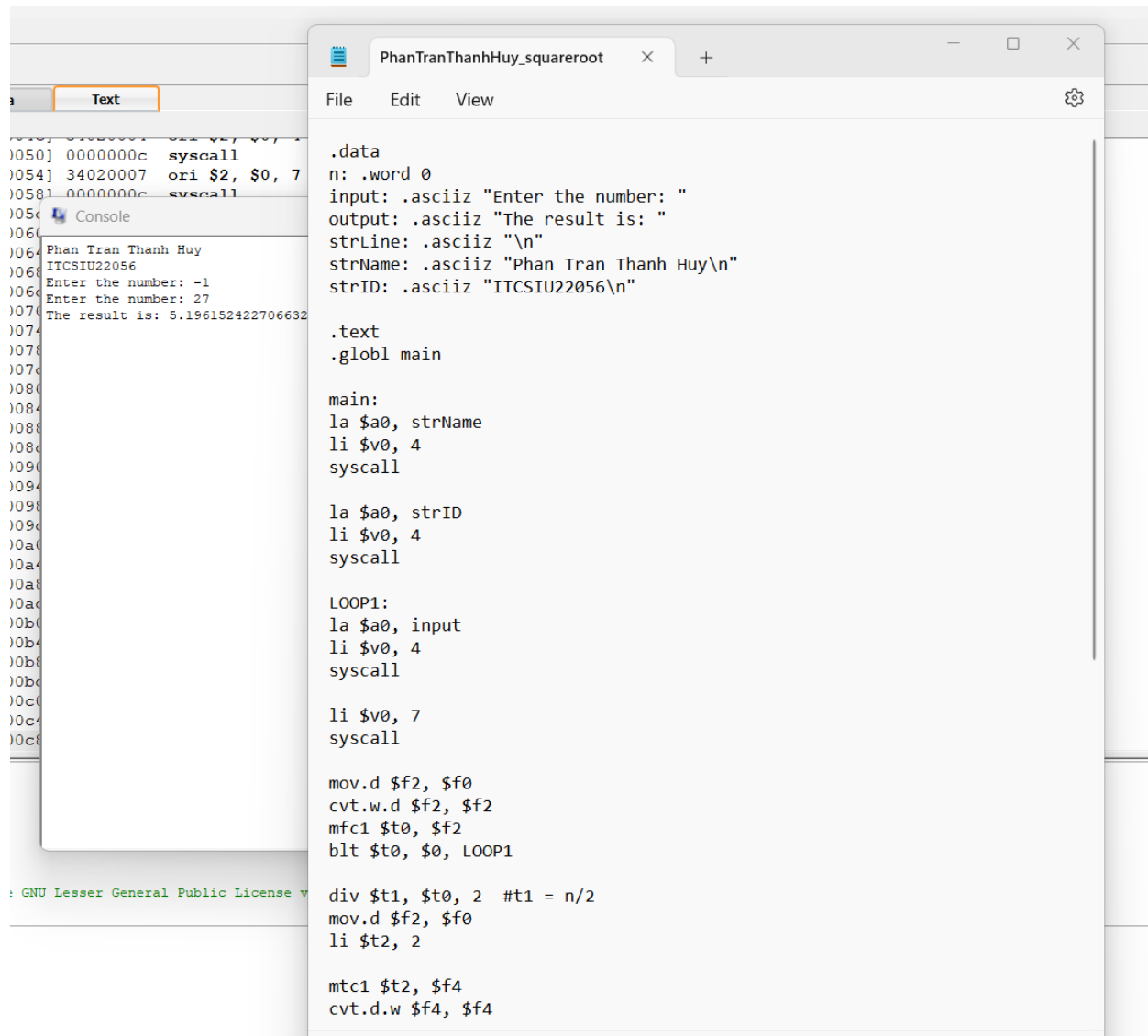
Console output:

```
10c] 2129ffff addi $9, $9, -
110] 2231ffff addi $17, $17, -
114] 1220ffff bne $17, $0, -
118 Console
11c
12c Phan Tran Thanh Huy
12c ITCSIU22056
12c abdeefggff
12c abdeefggff
12c t: 3
13c g: 2
13c e: 2
13c d: 1
13c b: 1
13c a: 1
14c
14c
14c
14c
15c
15c
15c
15c
16c
16c
16c
16c
17c
17c
17c
17c
17c
17c
17c
17c
18c
18c
18c
18c
```

GNU Lesser General Public License v

Ln 3, Col 31 | 1,246 characters | 100% | Windows (CRLF) | UTF-8

5. While conditional statement



```
0050] 0000000c syscall
0054] 34020007 ori $2, $0, 7
0058] 0000000c syscall
005c] Console
0060] Phan Tran Thanh Huy
0064] ITCsIU22056
0068] Enter the number: -1
006c] Enter the number: 27
0070] The result is: 5.196152422706632
0074]
0078]
007c]
0080]
0084]
0088]
008c]
0090]
0094]
0098]
009c]
00a0]
00a4]
00a8]
00ac]
00b0]
00b4]
00b8]
00bc]
00c0]
00c4]
00c8]

.data
n: .word 0
input: .asciiz "Enter the number: "
output: .asciiz "The result is: "
strLine: .asciiz "\n"
strName: .asciiz "Phan Tran Thanh Huy\n"
strID: .asciiz "ITCsiU22056\n"

.text
.globl main

main:
la $a0, strName
li $v0, 4
syscall

la $a0, strID
li $v0, 4
syscall

LOOP1:
la $a0, input
li $v0, 4
syscall

li $v0, 7
syscall

mov.d $f2, $f0
cvt.w.d $f2, $f2
mfc1 $t0, $f2
blt $t0, $0, LOOP1

div $t1, $t0, 2 #t1 = n/2
mov.d $f2, $f0
li $t2, 2

mtc1 $t2, $f4
cvt.d.w $f4, $f4
```

GNU Lesser General Public License v