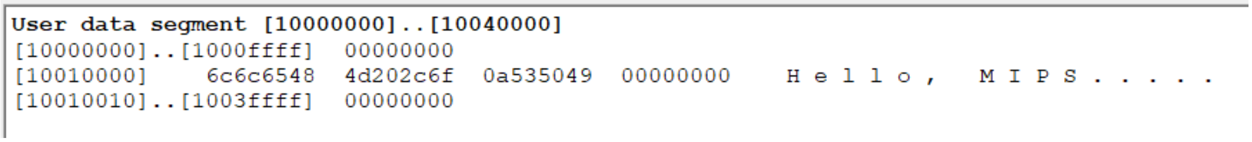
P.1. Introduction to SPIM

P.2. Question and Tasks

3.1. The address where the code resides is 00400000

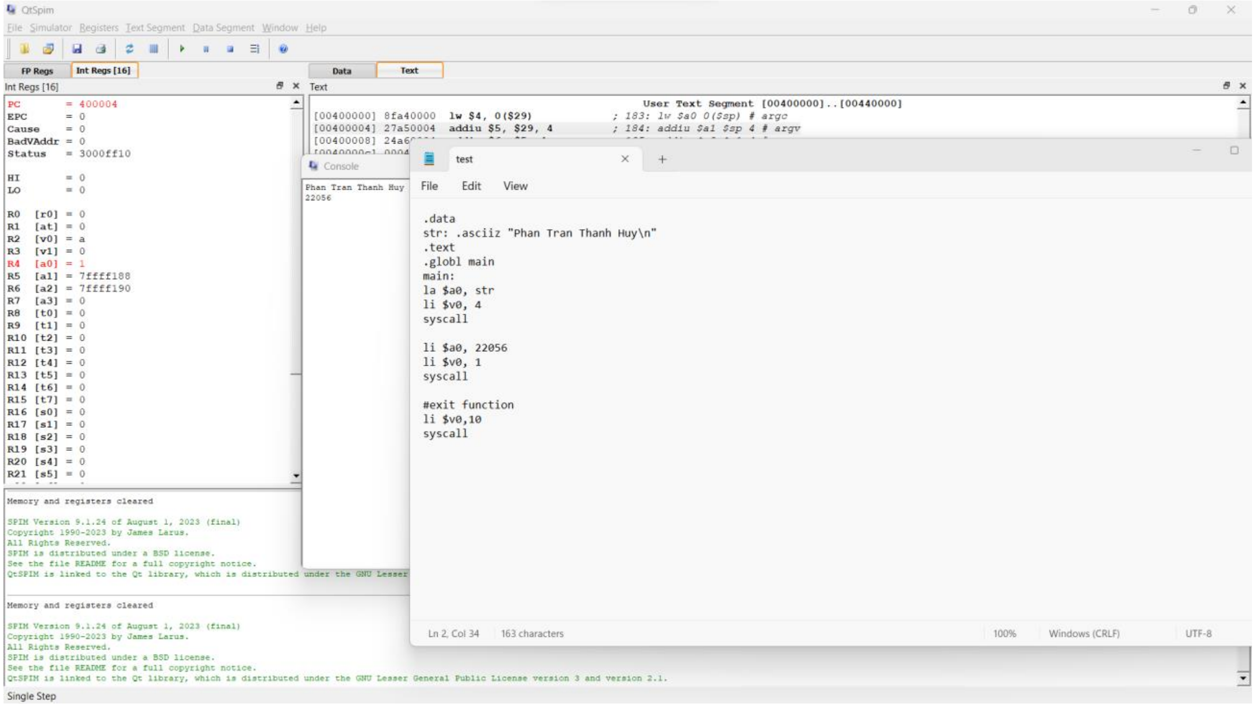
3.2. The instruction which is highlighted after every step show the next instruction. Because after that instruction performed, some values in the Register will change, or the instruction executed. Then the new instruction will be highlighted to determine the next execution.

3.3. The location in memory (in hex): 6c6c6548 4d202c6f 0a535049



```
User data segment [10000000]..[10040000]
[10000000]..[1000ffff]  00000000
[10010000]    6c6c6548  4d202c6f  0a535049  00000000    H e l l o ,   M I P S . . . . .
[10010010]..[1003ffff]  00000000
```

3.4.



```
PC       = 400004
EPC      = 0
Cause    = 0
BadVAddr = 0
Status   = 3000ff10

HI       = 0
LO       = 0

R0  [r0] = 0
R1  [at] = 0
R2  [v0] = a
R3  [v1] = 0
R4  [a0] = 1
R5  [a1] = 7ffff188
R6  [a2] = 7ffff190
R7  [a3] = 0
R8  [t0] = 0
R9  [t1] = 0
R10 [t2] = 0
R11 [t3] = 0
R12 [t4] = 0
R13 [t5] = 0
R14 [t6] = 0
R15 [t7] = 0
R16 [s0] = 0
R17 [s1] = 0
R18 [s2] = 0
R19 [s3] = 0
R20 [s4] = 0
R21 [s5] = 0
```

```
                         User Text Segment [00400000]..[00440000]
[00400000] 8fa40000  lw $4, 0($29)          ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004  addiu $5, $29, 4       ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a6
```

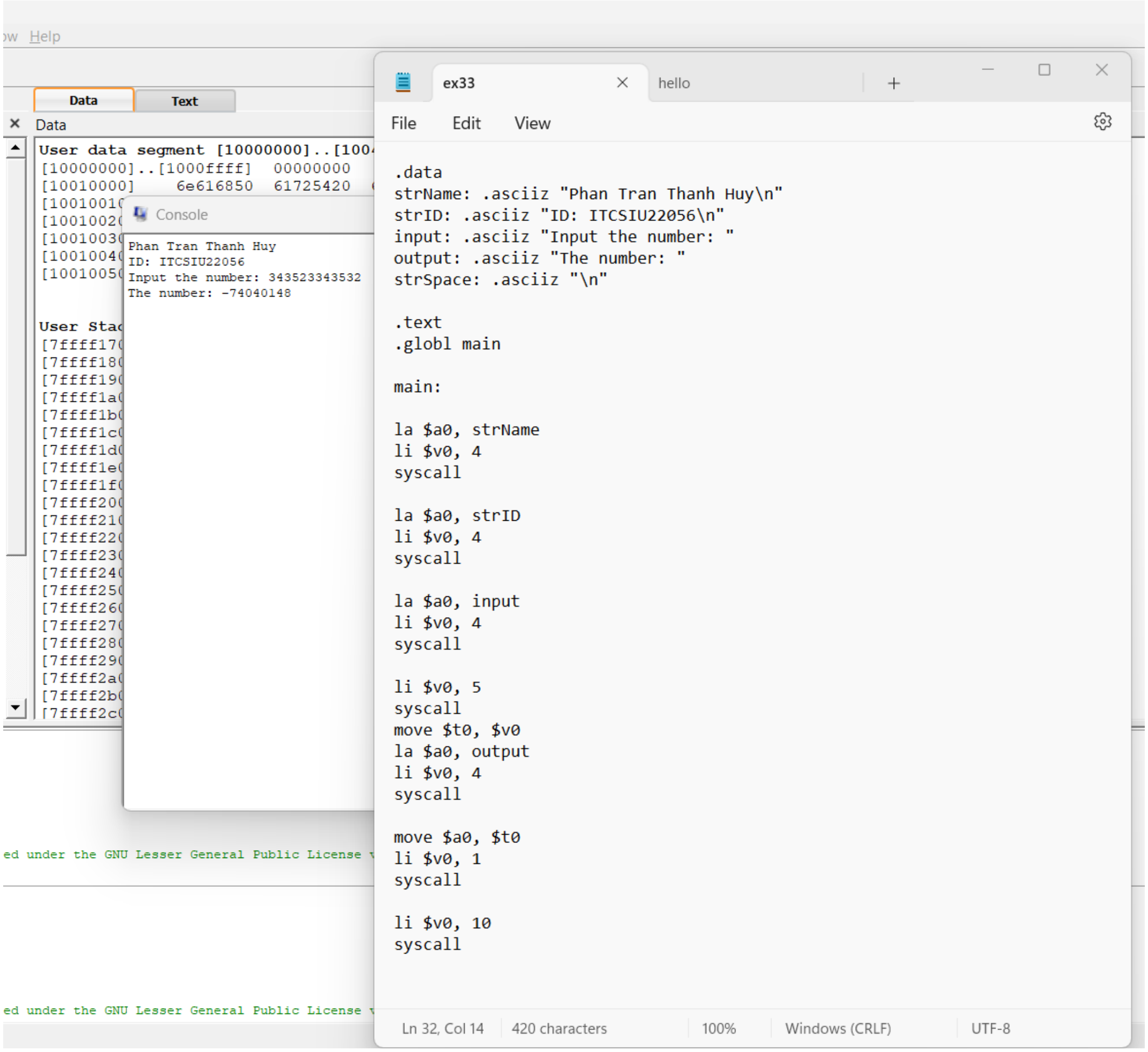
```
Phan Tran Thanh Huy
22056
```

```
.data
str: .asciiz "Phan Tran Thanh Huy\n"
.text
.globl main
main:
la $a0, str
li $v0, 4
syscall

li $a0, 22056
li $v0, 1
syscall

#exit function
li $v0,10
syscall
```

Ln 2, Col 34   163 characters   100%   Windows (CRLF)   UTF-8

```
Memory and registers cleared

SPIM Version 9.1.24 of August 1, 2023 (final)
Copyright 1990-2023 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser

Memory and registers cleared

SPIM Version 9.1.24 of August 1, 2023 (final)
Copyright 1990-2023 by James Larus.
All Rights Reserved.
SPIM is distributed under a BSD license.
See the file README for a full copyright notice.
QtSPIM is linked to the Qt library, which is distributed under the GNU Lesser General Public License version 3 and version 2.1.
```

4.1.



```
User data segment [10000000]..[100
[10000000]..[1000ffff]  00000000
[10010000]    6e616850  61725420
[1001001
[1001002
[1001003
[1001004
[1001005

User Stac
[7ffff17
[7ffff18
[7ffff19
[7ffff1a
[7ffff1b
[7ffff1c
[7ffff1d
[7ffff1e
[7ffff1f
[7ffff20
[7ffff21
[7ffff22
[7ffff23
[7ffff24
[7ffff25
[7ffff26
[7ffff27
[7ffff28
[7ffff29
[7ffff2a
[7ffff2b
[7ffff2c
```

```
Phan Tran Thanh Huy
ID: ITCSIU22056
Input the number: 343523343532
The number: -74040148
```

```
ed under the GNU Lesser General Public License
```

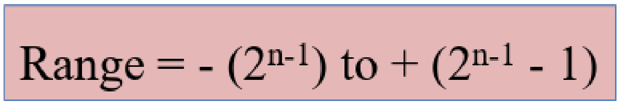
```
.data
strName: .asciiz "Phan Tran Thanh Huy\n"
strID: .asciiz "ID: ITCSIU22056\n"
input: .asciiz "Input the number: "
output: .asciiz "The number: "
strSpace: .asciiz "\n"

.text
.globl main

main:

la $a0, strName
li $v0, 4
syscall

la $a0, strID
li $v0, 4
syscall

la $a0, input
li $v0, 4
syscall

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

move $a0, $t0
li $v0, 1
syscall

li $v0, 10
syscall
```

The output is not correct.

Because the input number is out of range the integer limit

4.2.

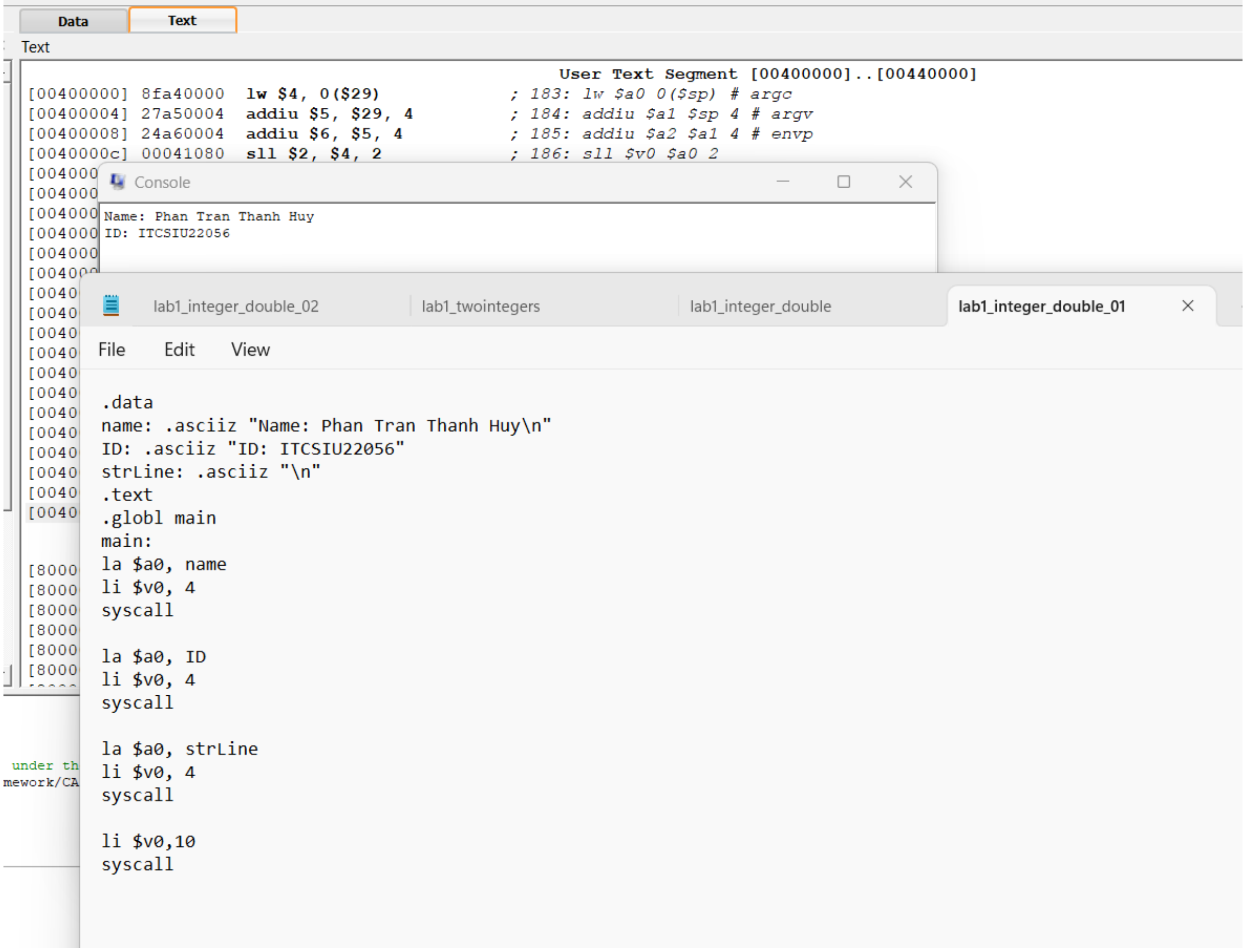
We apply the calculation:



$$\text{Range} = -\left(2^{n-1}\right) \text{ to } +\left(2^{n-1} - 1\right)$$

So the maximum and minimum input values that program can calculate correctly is: -2147483648 to 2147483649

4.3.

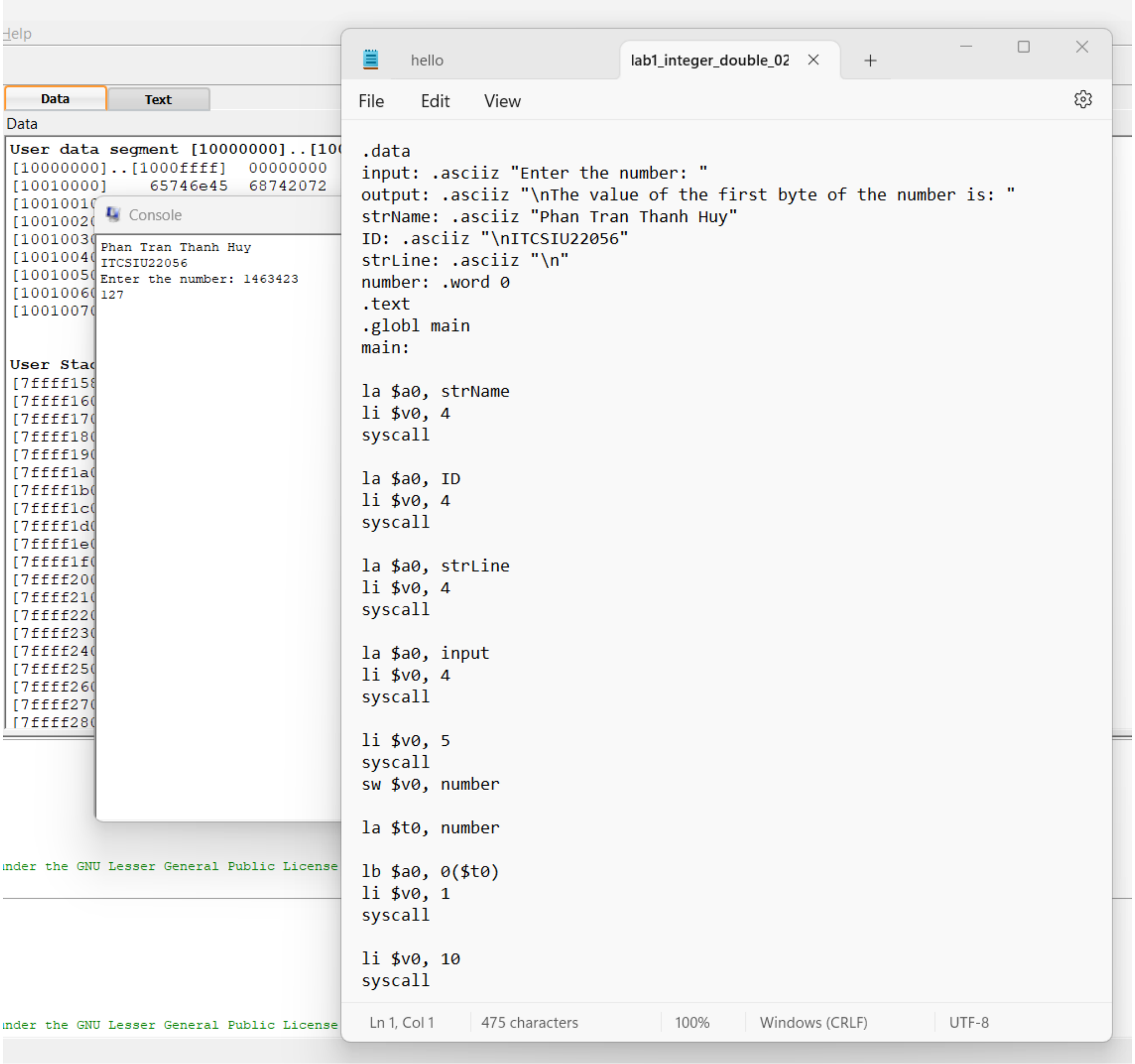


```
                                         User Text Segment [00400000]..[00440000]
[00400000] 8fa40000  lw $4, 0($29)          ; 183: lw $a0 0($sp) # argc
[00400004] 27a50004  addiu $5, $29, 4       ; 184: addiu $a1 $sp 4 # argv
[00400008] 24a60004  addiu $6, $5, 4        ; 185: addiu $a2 $a1 4 # envp
[0040000c] 00041080  sll $2, $4, 2          ; 186: sll $v0 $a0 2
```

```
Name: Phan Tran Thanh Huy
ID: ITCSIU22056
```

```
.data
name: .asciiz "Name: Phan Tran Thanh Huy\n"
ID: .asciiz "ID: ITCSIU22056"
strLine: .asciiz "\n"
.text
.globl main
main:
la $a0, name
li $v0, 4
syscall

la $a0, ID
li $v0, 4
syscall

la $a0, strLine
li $v0, 4
syscall

li $v0,10
syscall
```

4.4.



```
User data segment [10000000]..[10(
[10000000]..[1000ffff]  00000000
[10010000]     65746e45  68742072
[1001001(
[1001002(
[1001003(
[1001004(
[1001005(
[1001006(
[1001007(

User Stac
[7ffff15(
[7ffff16(
[7ffff17(
[7ffff18(
[7ffff19(
[7ffff1a(
[7ffff1b(
[7ffff1c(
[7ffff1d(
[7ffff1e(
[7ffff1f(
[7ffff20(
[7ffff21(
[7ffff22(
[7ffff23(
[7ffff24(
[7ffff25(
[7ffff26(
[7ffff27(
[7ffff28(
```

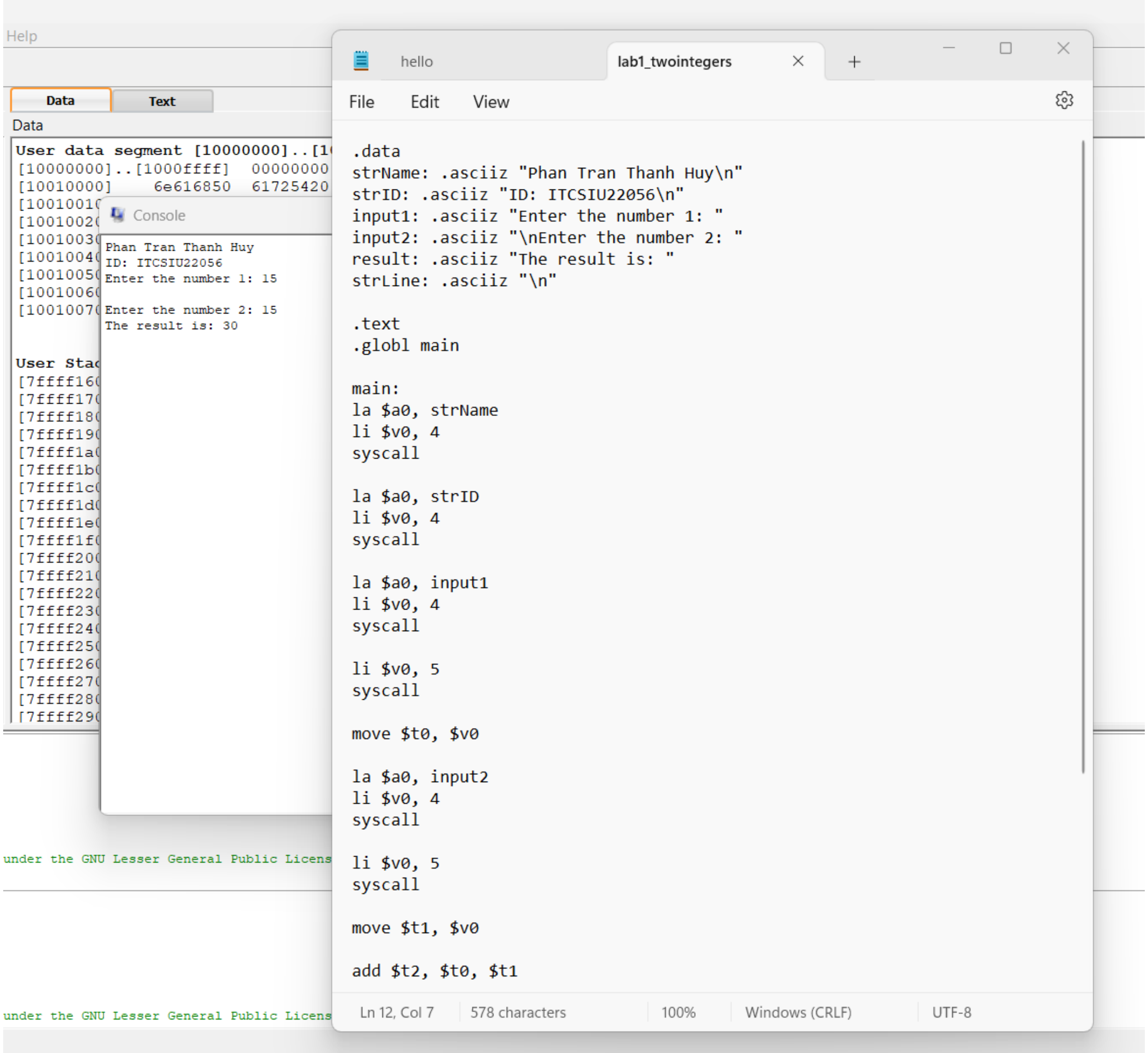
```
Phan Tran Thanh Huy
ITCSIU22056
Enter the number: 1463423
127
```

```
.data
input: .asciiz "Enter the number: "
output: .asciiz "\nThe value of the first byte of the number is: "
strName: .asciiz "Phan Tran Thanh Huy"
ID: .asciiz "\nITCSIU22056"
strLine: .asciiz "\n"
number: .word 0
.text
.globl main
main:

la $a0, strName
li $v0, 4
syscall

la $a0, ID
li $v0, 4
syscall

la $a0, strLine
li $v0, 4
syscall

la $a0, input
li $v0, 4
syscall

li $v0, 5
syscall
sw $v0, number

la $t0, number

lb $a0, 0($t0)
li $v0, 1
syscall

li $v0, 10
syscall
```

```
nder the GNU Lesser General Public License
```

```
nder the GNU Lesser General Public License
```

5.



```
User data segment [10000000]..[1
[10000000]..[1000ffff]  00000000
[10010000]      6e616850  61725420
[1001001
[1001002
[1001003
[1001004
[1001005
[1001006
[1001007


User Stac
[7ffff16
[7ffff17
[7ffff18
[7ffff19
[7ffff1a
[7ffff1b
[7ffff1c
[7ffff1d
[7ffff1e
[7ffff1f
[7ffff20
[7ffff21
[7ffff22
[7ffff23
[7ffff24
[7ffff25
[7ffff26
[7ffff27
[7ffff28
[7ffff29
```

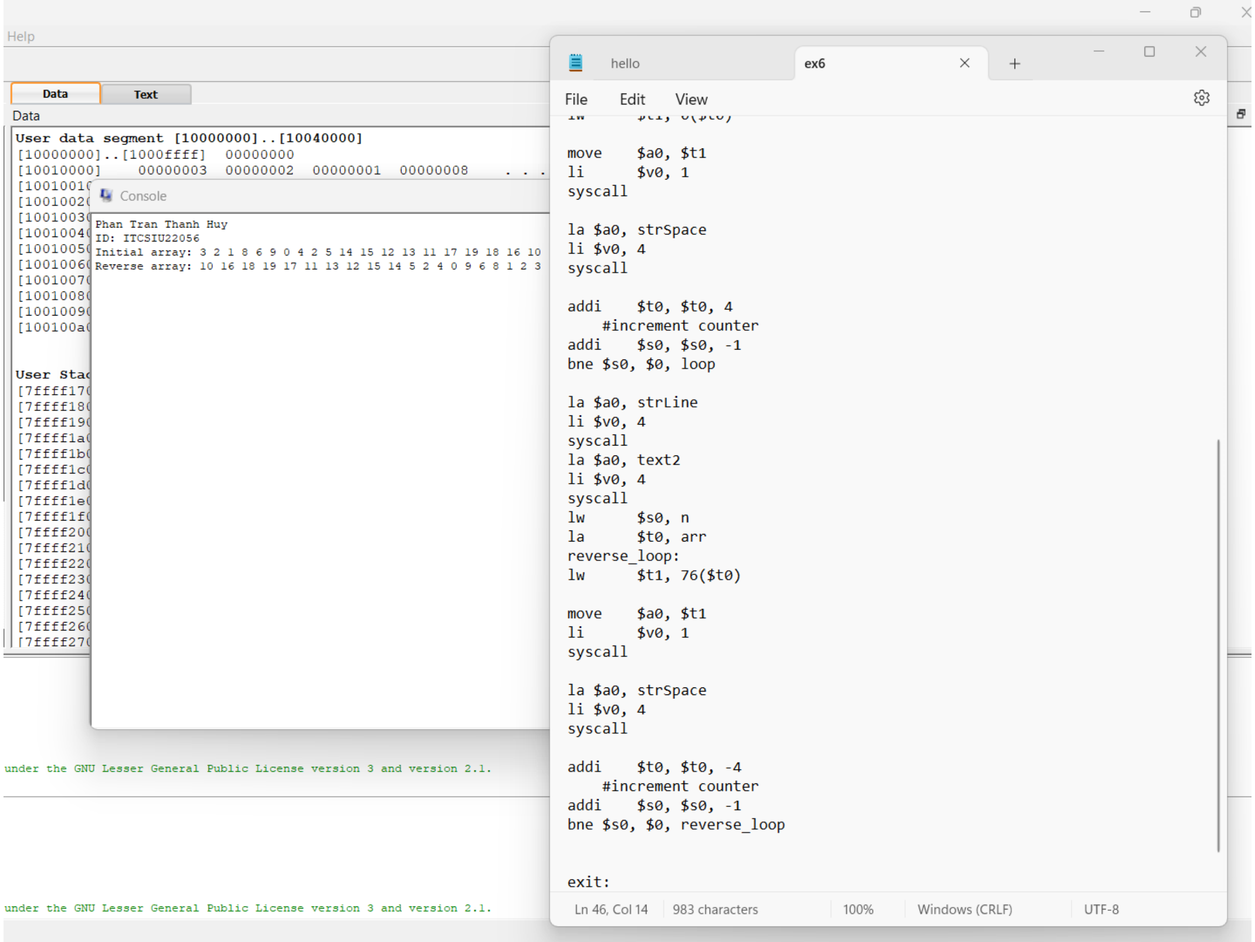
```
Phan Tran Thanh Huy
ID: ITCSIU22056
Enter the number 1: 15

Enter the number 2: 15
The result is: 30
```

```
under the GNU Lesser General Public Licens

under the GNU Lesser General Public Licens
```

```
.data
strName: .asciiz "Phan Tran Thanh Huy\n"
strID: .asciiz "ID: ITCSIU22056\n"
input1: .asciiz "Enter the number 1: "
input2: .asciiz "\nEnter the number 2: "
result: .asciiz "The result 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

add $t2, $t0, $t1
```

Ln 12, Col 7 | 578 characters | 100% | Windows (CRLF) | UTF-8

6.



```
User data segment [10000000]..[10040000]
[10000000]..[1000ffff]  00000000
[10010000]     00000003  00000002  00000001  00000008    . . .
[1001001(
[1001002(
[1001003(
[1001004(
[1001005(
[1001006(
[1001007(
[1001008(
[1001009(
[100100a(


User Stac
[7ffff17(
[7ffff18(
[7ffff19(
[7ffff1a(
[7ffff1b(
[7ffff1c(
[7ffff1d(
[7ffff1e(
[7ffff1f(
[7ffff20(
[7ffff21(
[7ffff22(
[7ffff23(
[7ffff24(
[7ffff25(
[7ffff26(
[7ffff27(
```

```
Phan Tran Thanh Huy
ID: ITCSIU22056
Initial array: 3 2 1 8 6 9 0 4 2 5 14 15 12 13 11 17 19 18 16 10
Reverse array: 10 16 18 19 17 11 13 12 15 14 5 2 4 0 9 6 8 1 2 3
```

```
under the GNU Lesser General Public License version 3 and version 2.1.
```

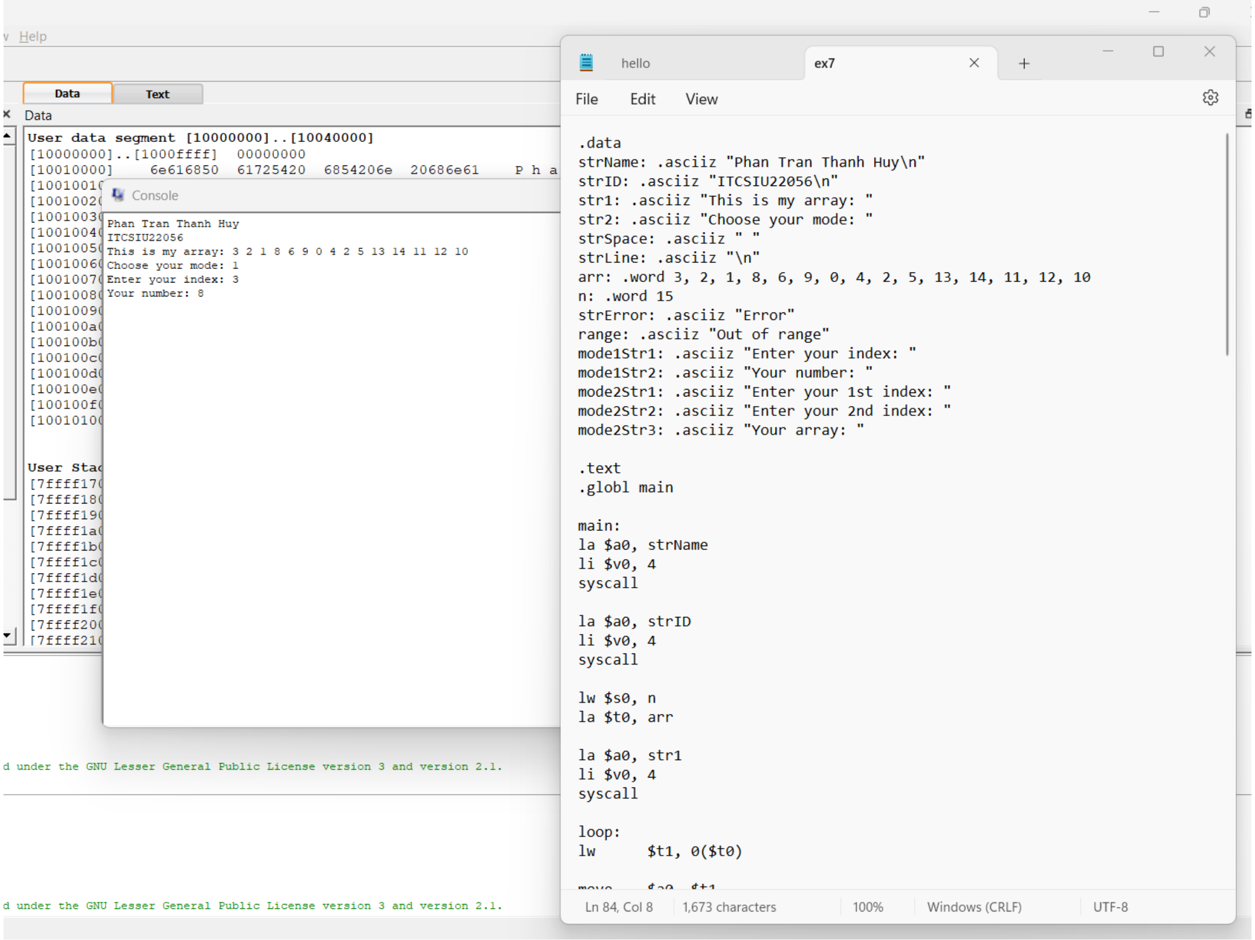
```
under the GNU Lesser General Public License version 3 and version 2.1.
```

```
lw      $t1, 0($t0)

move    $a0, $t1
li      $v0, 1
syscall

la $a0, strSpace
li $v0, 4
syscall

addi    $t0, $t0, 4
    #increment counter
addi    $s0, $s0, -1
bne $s0, $0, loop

la $a0, strLine
li $v0, 4
syscall
la $a0, text2
li $v0, 4
syscall
lw      $s0, n
la      $t0, arr
reverse_loop:
lw      $t1, 76($t0)

move    $a0, $t1
li      $v0, 1
syscall

la $a0, strSpace
li $v0, 4
syscall

addi    $t0, $t0, -4
    #increment counter
addi    $s0, $s0, -1
bne $s0, $0, reverse_loop


exit:
```

Ln 46, Col 14 | 983 characters | 100% | Windows (CRLF) | UTF-8

7.

a/



```
User data segment [10000000]..[10040000]
[10000000]..[1000ffff]  00000000
[10010000]    6e616850  61725420  6854206e  20686e61    P h a
[1001001(
[1001002(
[1001003(
[1001004(
[1001005(
[1001006(
[1001007(
[1001008(
[1001009(
[100100a(
[100100b(
[100100c(
[100100d(
[100100e(
[100100f(
[1001010(

User Stac
[7ffff17(
[7ffff18(
[7ffff19(
[7ffff1a(
[7ffff1b(
[7ffff1c(
[7ffff1d(
[7ffff1e(
[7ffff1f(
[7ffff20(
[7ffff21(
```

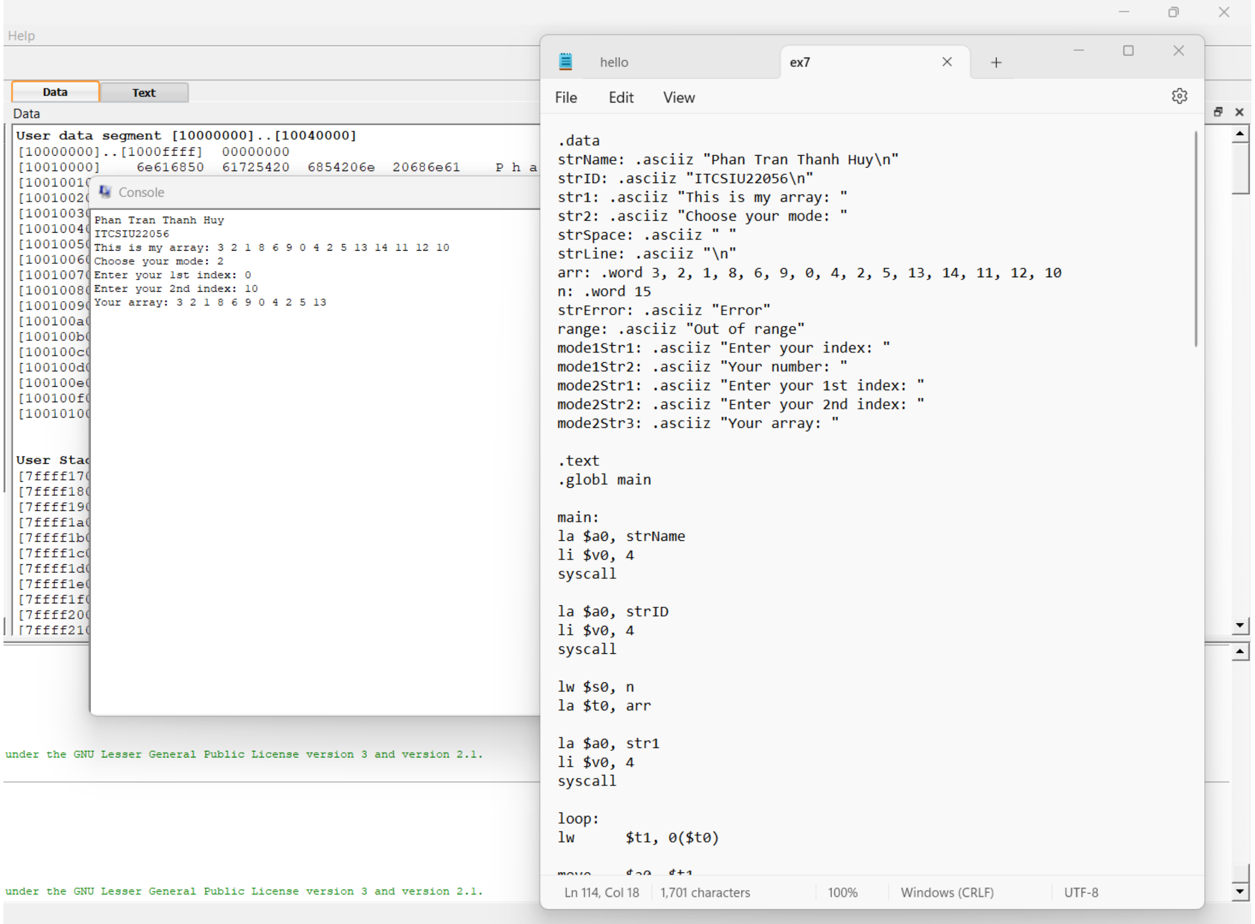
```
Phan Tran Thanh Huy
ITCSIU22056
This is my array: 3 2 1 8 6 9 0 4 2 5 13 14 11 12 10
Choose your mode: 1
Enter your index: 3
Your number: 8
```

```
d under the GNU Lesser General Public License version 3 and version 2.1.
```

```
d under the GNU Lesser General Public License version 3 and version 2.1.
```

```
.data
strName: .asciiz "Phan Tran Thanh Huy\n"
strID: .asciiz "ITCSIU22056\n"
str1: .asciiz "This is my array: "
str2: .asciiz "Choose your mode: "
strSpace: .asciiz " "
strLine: .asciiz "\n"
arr: .word 3, 2, 1, 8, 6, 9, 0, 4, 2, 5, 13, 14, 11, 12, 10
n: .word 15
strError: .asciiz "Error"
range: .asciiz "Out of range"
mode1Str1: .asciiz "Enter your index: "
mode1Str2: .asciiz "Your number: "
mode2Str1: .asciiz "Enter your 1st index: "
mode2Str2: .asciiz "Enter your 2nd index: "
mode2Str3: .asciiz "Your array: "

.text
.globl main

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

la $a0, strID
li $v0, 4
syscall

lw $s0, n
la $t0, arr

la $a0, str1
li $v0, 4
syscall

loop:
lw      $t1, 0($t0)

move    $a0, $t1
```

b/



```
User data segment [10000000]..[10040000]
[10000000]..[1000ffff]  00000000
[10010000]    6e616850  61725420  6854206e  20686e61    P h a
[1001001
[1001002
[1001003
[1001004
[1001005
[1001006
[1001007
[1001008
[1001009
[100100a
[100100b
[100100c
[100100d
[100100e
[100100f
[1001010


User Stac
[7ffff17
[7ffff18
[7ffff19
[7ffff1a
[7ffff1b
[7ffff1c
[7ffff1d
[7ffff1e
[7ffff1f
[7ffff20
[7ffff21
```

```
Phan Tran Thanh Huy
ITCSIU22056
This is my array: 3 2 1 8 6 9 0 4 2 5 13 14 11 12 10
Choose your mode: 2
Enter your 1st index: 0
Enter your 2nd index: 10
Your array: 3 2 1 8 6 9 0 4 2 5 13
```

```
under the GNU Lesser General Public License version 3 and version 2.1.
```

```
under the GNU Lesser General Public License version 3 and version 2.1.
```

```
.data
strName: .asciiz "Phan Tran Thanh Huy\n"
strID: .asciiz "ITCSIU22056\n"
str1: .asciiz "This is my array: "
str2: .asciiz "Choose your mode: "
strSpace: .asciiz " "
strLine: .asciiz "\n"
arr: .word 3, 2, 1, 8, 6, 9, 0, 4, 2, 5, 13, 14, 11, 12, 10
n: .word 15
strError: .asciiz "Error"
range: .asciiz "Out of range"
mode1Str1: .asciiz "Enter your index: "
mode1Str2: .asciiz "Your number: "
mode2Str1: .asciiz "Enter your 1st index: "
mode2Str2: .asciiz "Enter your 2nd index: "
mode2Str3: .asciiz "Your array: "

.text
.globl main

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

la $a0, strID
li $v0, 4
syscall

lw $s0, n
la $t0, arr

la $a0, str1
li $v0, 4
syscall

loop:
lw      $t1, 0($t0)

move    $a0, $t1
```

Ln 114, Col 18 | 1,701 characters | 100% | Windows (CRLF) | UTF-8