

ASSIGNMENT 2:

CODE :

```
.data
A: .asciz "The sum of "
B: .asciz " and "
C: .asciz " is "
.text
    li s0, 18    #s0=18
    li s1, 20    #s1=20
    add t1, s0, s1    #t1=s1+s0
    li a7, 4      #a7=4 in ra chuỗi
    la a0, A      #a0 = địa chỉ của chuỗi A
    ecall
    li a7, 1      #a7=1 in ra số nguyên
    mv a0, s0     #a0=s0
    ecall
    li a7, 4      #a7=4 in ra chuỗi
    la a0, B      #a0 = địa chỉ của chuỗi B
    ecall
    li a7, 1      #a7=1 in ra số nguyên
    mv a0, s1     #a0=s1
    ecall
    li a7, 4      #a7=4 in ra chuỗi
    la a0, C      #a0 = địa chỉ của chuỗi C
    ecall
    li a7, 1      #a7=1 in ra số nguyên
    mv a0, t1     #a0=t1
    ecall
```

Kết quả:

```
The sum of 18 and 20 is 38
-- program is finished running (dropped off bottom) --
```

```
The sum of 18 and 20 is 38
-- program is finished running (dropped off bottom) --
```

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	e h T	m u s	\0 f o	d n a	i \0	\0 \0 s	\0 \0 \0 \0	\0 \0 \0 \0
0x10010020	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010040	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010060	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010080	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100a0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100c0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100e0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010100	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

◀ ▶ ↺ ⌂

☒ Hexadecimal Addresses
 ☒ Hexadecimal Values
 ☒ ASCII

```
# Laboratory Exercise 5, Home Assignment 2
```

Giá trị các thanh ghi:

Label	Address ▲
W5A3.asm	
strcpy	0x00400000
L1	0x00400004
end_of_strcpy	0x00400020
x	0x10010000
y	0x10010020
<input checked="" type="checkbox"/> Data <input checked="" type="checkbox"/> Text	

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010020	1 1 e H	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010040	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010060	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010080	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100a0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100c0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100e0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010100	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

◀ | ▶

0x10010000 (.data) ☒ Hexadecimal Addresses ☒ Hexadecimal Values ☒ ASCII

ASSIGNMENT 4:

Code:

```
# Laboratory Exercise 5, Home Assignment 3

.data
string: .space 50
message1: .asciz "Nhap xau: "
message2: .asciz "Do dai xau la: "

.text
main:
get_string:
    li a7, 54
    la a0, message1
    la a1, string
    li a2, 50
    ecall
    # TODO Nhập chuỗi ký tự từ bàn phím


get_length:
    la    a0, string          # a0 = address(string[0])
    li    t0, 0               # t0 = i = 0

check_char:
    add    t1, a0, t0          # t1 = a0 + t0 = address(string[0]+i)
    lb     t2, 0(t1)           # t2 = string[i]
    beq    t2, zero, end_of_str # Nếu là ký tự NULL thì kết thúc
    addi   t0, t0, 1           # t0 = t0 + 1 -> i = i + 1
    j      check_char

end_of_str:
end_of_get_length:
print_length:
    li a7, 56
    la a0, message2
    mv a1, t0
    ecall
    # TODO In kết quả ra màn hình
```


Kết quả chương trình:

Input

 **Nhap xau:**

anh Gia Huy dep trai

OK Cancel

 **Do dai xau la: 21**

OK

Labels

Label	Address ▲
W5A4.asm	
main	0x00400000
get_string	0x00400000
get_length	0x0040001c
check_char	0x00400028
end_of_str	0x0040003c
end_of_get_length	0x0040003c
print_length	0x0040003c
string	0x10010000
message1	0x10010032
message2	0x1001003d

☒ Data ☒ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	h n a	a i G	y u H	p e d	i a r t	\0 \0 \0 \n	\0 \0 \0 \0	\0 \0 \0 \0
0x10010020	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	h N \0 \0	x p a	:	u a
0x10010040	i a d	u a x	:	a l	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010060	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010080	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100a0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100c0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x100100e0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010100	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

◀ ▶ 0x10010000 (.data) ☒ Hexadecimal Addresses ☒ Hexadecimal Values ☒ ASCII

ASSIGNMENT 5:

Code:

```
.data
msg_input: .asciz "Enter a string: "
buffer: .space 21
.text
_start:
    # In thông báo yêu cầu nhập chuỗi
    la a0, msg_input
    li a7, 4
    ecall

    # Khởi tạo các thanh ghi
    li t0, 0                # t0 = 0 (biến đếm số kí tự nhập vào)
    la t1, buffer           # t1 = Địa chỉ bắt đầu của buffer

input_loop:
    # Đọc một kí tự từ bàn phím
    li a7, 12               # Syscall number for read_char
    ecall
    mv t2, a0               # Lưu kí tự vào t2

    # Kiểm tra xem kí tự nhập vào có phải là Enter (mã ASCII 10) hay không
    li t3, 10               # ASCII code của kí tự Enter
    beq t2, t3, end_input   # Nếu kí tự là Enter thì kết thúc nhập

    # Kiểm tra độ dài chuỗi có vượt quá 20 kí tự không
    li t3, 20
    beq t0, t3, end_input   # Nếu độ dài chuỗi đạt 20 thì kết thúc nhập

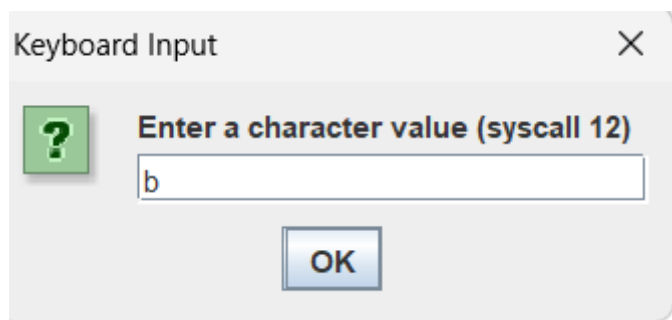
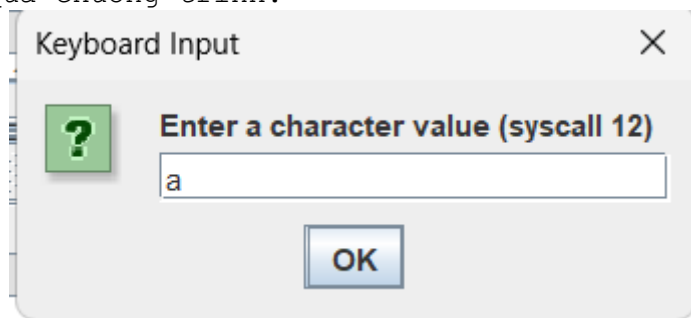
    # Lưu kí tự vào bộ nhớ và tăng biến đếm
    sb t2, 0(t1)            # Lưu kí tự vào buffer tại vị trí t1
    addi t1, t1, 1          # Di chuyển con trỏ t1 đến vị trí tiếp theo trong buffer
    addi t0, t0, 1          # Tăng biến đếm t0
```

```

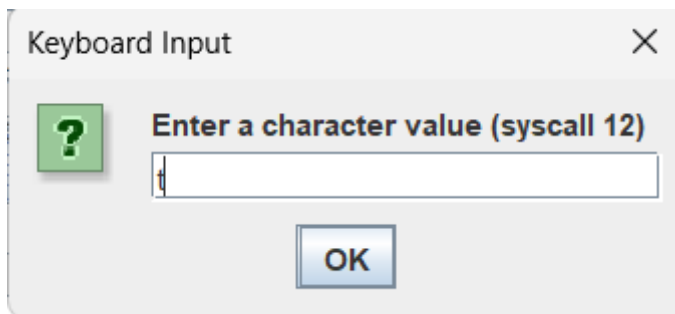
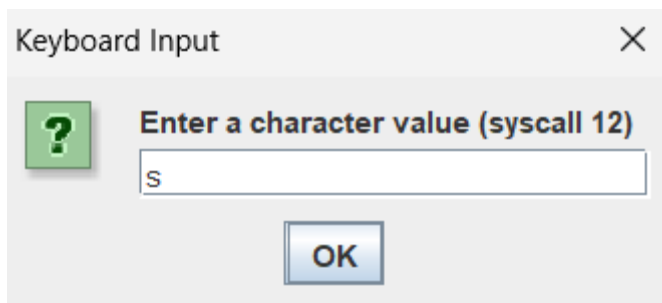
        j input_loop          # Quay lại vòng lặp nhập tiếp kí tự
end_input:
    # In chuỗi ngược lại
    addi t1, t1, -1          # Quay con trỏ t1 về vị trí cuối của chuỗi
    đã nhập
reverse_print_loop:
    lb a0, 0(t1)             # Đọc kí tự từ vị trí hiện tại của buffer
    beq a0, zero, done       # Nếu gặp kí tự null thì kết thúc
    # In kí tự
    li a7, 11
    ecall
    addi t1, t1, -1          # Di chuyển con trỏ ngược lại (lùi 1 byte)
    j reverse_print_loop     # Tiếp tục in kí tự ngược lại
done:
    # Dừng chương trình
    li a7, 10                # exit
    ecall

```

Kết quả chương trình:



...



Enter a string: **** user input : a

**** user input : b

**** user input : c

**** user input : d

**** user input : e

**** user input : f

**** user input : g

**** user input : g

**** user input : h

**** user input : i

**** user input : j

**** user input : k

**** user input : l

**** user input : m

**** user input : m

**** user input : n

**** user input : o

**** user input : p

**** user input : q

**** user input : r

**** user input : s

**** user input : t

tsrqponmlkjihgfedcba