

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



TỔ CHỨC VÀ CẤU TRÚC MÁY TÍNH II
LỚP: IT012.N21.2

BÁO CÁO THỰC HÀNH SỐ 3 (LAB 03)

Giảng viên hướng dẫn: Nguyễn Thành Nhân

Sinh viên: Hồ Trọng Hiền

MSSV: 22520414

MỤC LỤC

1. Khai báo và xuất ra cửa sổ 2 chuỗi.....	3
2. Biểu diễn nhị phân chuỗi đã khai báo.....	4
3. Xuất ra chuỗi đã nhập.	5
4. Nhập vào hai số nguyên và in ra tổng	7

1. Khai báo và xuất ra của số 2 chuỗi.

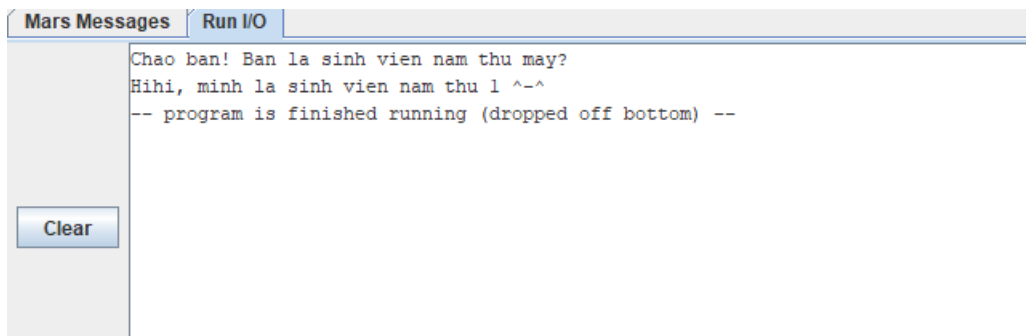
- Chuỗi 1: Chao ban! Ban la sinh vien nam thu may?
- Chuỗi 2: Hihi, minh la sinh vien nam thu 1 ^-^

```
1  .data
2      string1: .ascii "Chao ban! Ban la sinh vien nam thu may?"
3      string2: .ascii "\nHihi, minh la sinh vien nam thu 1 ^-^"
4
5  .text
6  main:  li $v0, 4
7         la $a0, string1
8         syscall
9         la $a0, string2
10        syscall
11
```

- Khai báo 2 biến string1 và string2 có kiểu ascii, 1 ký tự tương đương 1 byte.
- li \$v0, 4: Gán giá trị 4 cho thanh ghi v0.
- la \$a0, string1: Load địa chỉ của string1 vào thanh ghi a0.
- syscall: khi có giá trị 4 trong v0, tiến hành in chuỗi ở trong a0 -> In ra chuỗi string1.

Tương tự với chuỗi 2.

Kết quả:



```
Mars Messages  Run I/O
Chao ban! Ban la sinh vien nam thu may?
Hihi, minh la sinh vien nam thu 1 ^-^
-- program is finished running (dropped off bottom) --

Clear
```

2. Biểu diễn nhị phân chuỗi đã khai báo.

```
01101111011000010110100001000011
01101110011000010110001000100000
01100001010000100010000000100001
01100001011011000010000001101110
01101110011010010111001100100000
01101001011101100010000001101000
01101110001000000110111001100101
01110100001000000110110101100001
01101101001000000111010101101000
00000000001111110111100101100001
01101000011010010100100000001010
01101101001000000010110001101001
00100000011010000110111001101001
01110011001000000110000101101100
00100000011010000110111001101001
01101110011001010110100101110110
01101101011000010110111000100000
01110101011010000111010000100000
01011110001000000011000100100000
000000000000000000101111000101101
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

Chuỗi được biểu diễn theo kiểu ASCII. Trong MIPS, trọng số thấp nhất của chuỗi sẽ nằm ở bên phải. Vì vậy, chuỗi sẽ được biểu diễn ngược lại. Một ký tự trong chuỗi tương ứng với 1 byte. Một giá trị trong MIPS sẽ biểu diễn được 4 byte tương đương với 4 ký tự.

3. Xuất ra chuỗi đã nhập.

Ví dụ:

Nhập: Truong Dai hoc Cong nghe Thong tin

Xuất: Truong Dai hoc Cong nghe Thong tin

```
1 .data
2 string1: .space 100
3 string2: .asciiz "Nhap: "
4 string3: .asciiz "Xuat: "
5
6 .text
7     #print string2
8     li $v0, 4
9     la $a0, string2
10    syscall
11
12    #read string1
13    li $v0, 8
14    la $a0, string1
15    li $a1, 100
16    syscall
17
18    #print string3
19    li $v0, 4
20    la $a0, string3
21    syscall
22
23    #print string1
24    li $v0, 4
25    la $a0, string1
26    syscall
27
```

- Các lệnh :

li \$v0, 4

la \$a0, string2

syscall

-> In ra chuỗi “Nhap: ” tương tự câu 1.

- li \$v0, 8: Gán 8 vào thanh ghi v0

- la \$a0, string1: Load địa chỉ string1 vào thanh ghi a0

- li \$a1, 100: Gán 100 vào thanh ghi a1, là ký tự tối đa cho phép nhập.

- syscall: giá trị v0 là 8 nên tiến hành đọc chuỗi.

- Các lệnh :

li \$v0, 4

la \$a0, string3

syscall

-> In ra chuỗi “Xuat: ” tương tự câu 1.

- Các lệnh :

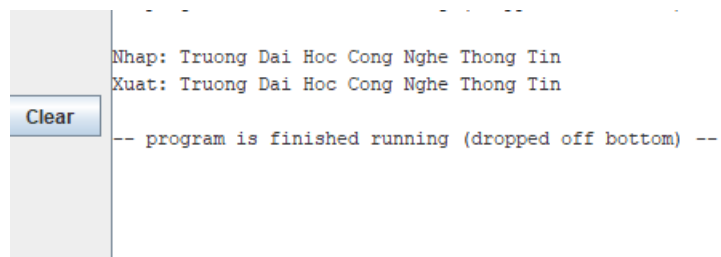
li \$v0, 4

la \$a0, string1

syscall

-> In ra chuỗi vừa được nhập vào trong string1 tương tự câu 1.

Kết quả:

A screenshot of a terminal window with a light gray background. On the left side, there is a vertical toolbar with a blue button labeled "Clear". The terminal text is as follows:

```
-- --  
Nhap: Truong Dai Hoc Cong Nghe Thong Tin  
Xuat: Truong Dai Hoc Cong Nghe Thong Tin  
-- program is finished running (dropped off bottom) --
```

4. Nhập vào hai số nguyên và in ra tổng

```
1  .data
2      string1: .asciiz "Nhap a: "
3      string2: .asciiz "Nhap b: "
4      string3: .asciiz "Tong: "
5  .text
6      #print string1
7      li $v0, 4
8      la $a0, string1
9      syscall
10     #Read a
11     li $v0, 5
12     syscall
13     add $t1, $0, $v0
14     #print string2
15     li $v0, 4
16     la $a0, string2
17     syscall
18     #Read b
19     li $v0, 5
20     syscall
21     add $t2, $0, $v0
22     #add t1, t2
23     add $t3, $t1, $t2
24     #print string3
25     li $v0, 4
26     la $a0, string3
27     syscall
28     #print sum
29     li $v0, 1
30     la $a0, ($t3)
31     syscall
32
```

- Các lệnh :

li \$v0, 4

la \$a0, string1

syscall

-> In ra chuỗi "Nhap a: " tương tự câu 1.

- li \$v0, 5: Gán giá trị 5 cho thanh ghi v0.

- syscall: Tiến hành nhập a vào thanh ghi v0.

- add \$t1, \$0, \$v0: Cộng giá trị trong thanh ghi v0 với 0 vào thanh ghi t1, lưu giá trị vừa nhập vào thanh ghi t1.

Tương tự với số nguyên thứ 2.

- add \$t3, \$t1, \$t2: Cộng hai giá trị vừa nhập và lưu vào thanh ghi t3
- li \$v0, 1: Gán giá trị 1 cho thanh ghi v0
- la \$a0, (\$t3): Load giá trị của t3 lên a0
- syscall: In ra giá trị trong a0 là tổng của hai số.

Kết quả:

```
Nhap a: 25
Nhap b: 36
Tong: 61
-- program is finished running (dropped off bottom) --
```