BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH

Giữa kì

Họ tên: Tạ Quang Linh
MSSV: 20194605

A. SỐ NGUYÊN

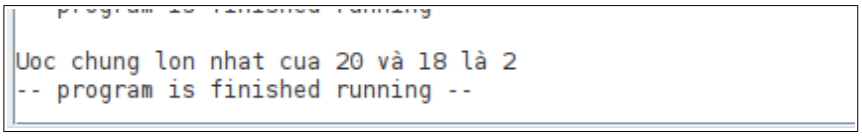
*** Cách thực hiện:**

- Đầu tiên kiểm tra giá trị 0 ở biết a và b, nếu một trong hai biến mang giá trị 0 thì UCLN là tổng của 2 biến a, b.
- Sử dụng vòng lặp duyệt cho đến khi giá trị biến a bằng biến b, trong quá trình duyệt, nếu a lớn hơn b thì gán a bằng a trừ b, ngược lại nếu a nhỏ hơn b thì gán b bằng b trừ a.

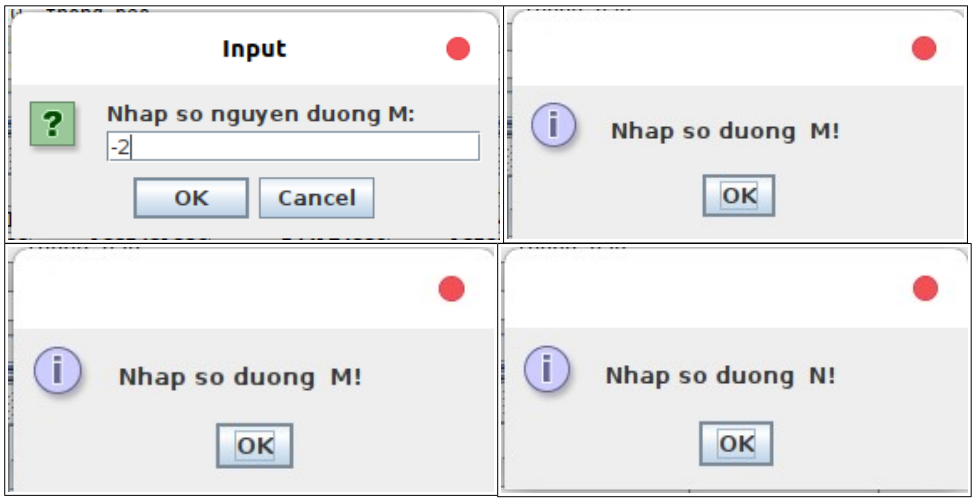
*** Chương trình con UCLN:**

- Với hai giá trị biến đầu vào được đặt trong thanh ghi $a0, $a1 và giá trị trả về được lưu ở thanh ghi $v1
- Chương trình con tìm ước chung lớn nhất của 2 số a và b tương ứng với giá trị ở hai thanh ghi $a0, $a1.

KẾT QUẢ:



Ví dụ nhập giá trị âm:



MÃ NGUỒN:

bai-1.asm

```
.data
        mess1: .asciiz "Nhap so nguyen duong M:"
        mess2: .asciiz "Nhap so nguyen duong N:"
        mess_error: .asciiz "Nhap so duong "
        mess_m: .asciiz " M!"
        mess_n: .asciiz " N!"
        mess_kq1: .asciiz "Uoc chung lon nhat cua "
        mess_kq2: .asciiz " và "
        mess_kq3: .asciiz " là "

.text
        li      $v0, 51                 # read integer
        la      $a0, mess1
        syscall

        slti    $t0, $a0, 0             #if (M < 0) -> Thông báo lỗi nhập số âm
        la      $a1, mess_m
        bnez    $t0, thong_bao

        addi    $s0, $a0, 0             # s0 = M

        li      $v0, 51                 # read integer
        la      $a0, mess2
        syscall

        slti    $t0, $a0, 0             #if (N < 0) -> Thông báo lỗi nhập số âm
        la      $a1, mess_n
        bnez    $t0, thong_bao

        addi    $s1, $a0, 0             # s1 = N

main:
        addi    $a0, $s0, 0             # a0 = M
        addi    $a1, $s1, 0             # a1 = N
        jal     UCLN

        li      $v0, 4                  # print String
        la      $a0, mess_kq1
        syscall

        li      $v0, 1                  # print integer
        addi    $a0, $s0, 0             # M
        syscall

        li      $v0, 4                  # print String
        la      $a0, mess_kq2
        syscall

        li      $v0, 1                  # print integer
        addi    $a0, $s1, 0             # N
        syscall

        li      $v0, 4                  # print String
        la      $a0, mess_kq3
        syscall

        li      $v0, 1                  # print integer
        addi    $a0, $v1, 0             # UCLN
        syscall

end_main:
        li      $v0, 10
        syscall

# Hàm tìm ước chung lớn nhất
# Đầu vào là 2 giá trị M, N tương ứng với hai thanh ghi đầu vào là $a0, $a1
# Đầu ra là $v1

UCLN:
check_value_0:
        beqz    $a0, return
        beqz    $a1, return
        j       else
return:
        add     $v0, $a0, $a1           # c = a+b
        jr      $ra
else:
loop:
        sub     $t0, $a0, $a1           # t0 = a-b
        beqz    $t0, end_loop           # if(a == b) -> end_loop

        slt     $t0, $a1, $a0           # if (b < a)
        beqz    $t0, else_loop
        sub     $a0, $a0, $a1           # a = a - b
        j       loop
else_loop:
        sub     $a1, $a1, $a0           # b = b - a
        j       loop
end_loop:

end_loop:
        addi    $v1, $a0, 0             # return a
        jr      $ra
end_UCLN:


thong_bao:
        li      $v0, 59                 #print string
        la      $a0, mess_error
        syscall
        li      $v0, 10
        syscall
end:
```

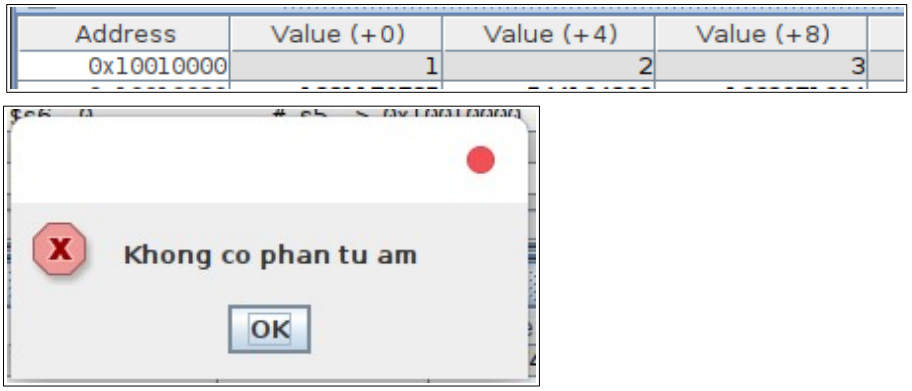
B. MẢNG

*** Cách thực hiện:**

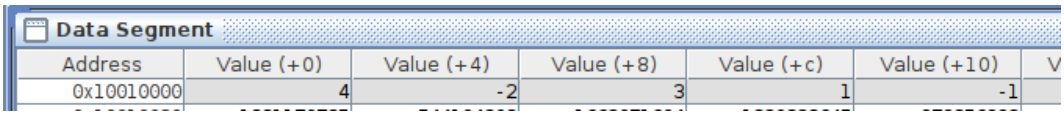
- Duyệt lần lượt các phần tử của mảng, bỏ qua nếu gặp số dương, đồng thời lưu giá thanh ghi $t2 = 1 nếu mảng có số âm
- Kiểm tra nếu giá trị phần tử trong mảng nhỏ hơn max thì gán max bằng giá trị phần tử đó, đồng thời thay đổi vị trí lưu giá trị max
- Nếu không có phần tử âm thì đưa ra thông báo

KẾT QUẢ:

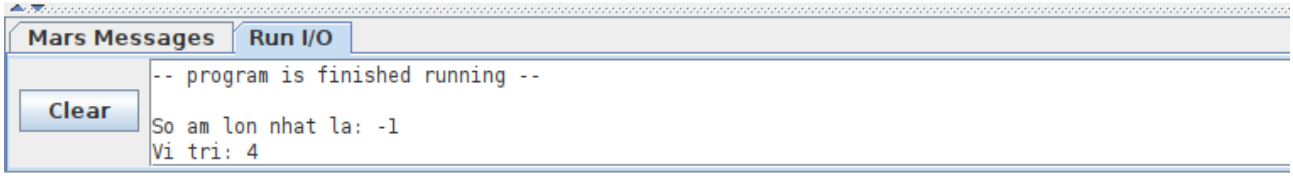
- Khi không có giá trị âm trong mảng



- Khi có giá trị âm trong mảng



Kết quả:



MÃ NGUỒN:

bai-2.asm | 2.1.asm

```
.data
        mess1: .asciiz "Nhap so phan tu cua mang:"
        mess2: .asciiz "Nhap cac phan tu cua mang: "
        mess3: .asciiz "Khong co phan tu am"
        mess4: .asciiz "So am lon nhat la: "
        mess5: .asciiz "\nVi tri: "
.text
        li      $v0, 51                         # read integer
        la      $a0, mess1
        syscall

        addi    $s6, $s6, 0x10010000    # s6 -> 0x10010000 = &A
        addi    $s5, $s6, 0             # s5 -> 0x10010000 = &A

        addi    $s0, $a0, 0             # s0 = N
        addi    $t1, $zero, 0           # i = 0

loop_scan:
        slt     $t0, $t1, $s0           # if (i < N)
        beq     $t0, $zero, end_loop_scan

        li      $v0, 51                 # read integer
        la      $a0, mess2
        syscall
```

```
        sw      $a0, 0($s5)             # s5[i] = v0

        addi    $t1, $t1, 1             # i++
        addi    $s5, $s5, 4             # s5 -> &A[i]

        j       loop_scan
end_loop_scan:

main:
        addi    $t2, $zero, 0           # Lưu giá trị 0 hoặc 1, để kiểm tra mảng chứa phần tử âm hay không \
        addi    $t1, $zero, 0           # i = 0

        addi    $s5, $s6, 0             # s5 -> &A[0]

        addi    $s1, $zero, 0x80000000  # max = giá trị 32 bit nhỏ nhất
        addi    $s2, $zero, -1          # Lưu vị trí giá trị lớn nhất trong mảng
loop:
        slt     $t0, $t1, $s0           # if (i < N)
        beqz    $t0, end_loop

        lw      $a0, 0($s5)             # a0 = A[i]

        slti    $t0, $a0, 0             # if (a0 >= 0) -> loop
        beq     $t0, $zero, continue    # Bỏ qua số dương và 0
```

```
        addi    $t2, $zero, 1           # Có số âm xuất hiện

        slt     $t0, $s1, $a0           # if (max >= A[i]) -> loop
        beqz    $t0, continue

        addi    $s1, $a0, 0             # max = A[i]
        addi    $s2, $t1, 0             # s2 = i

continue:
        addi    $t1, $t1, 1             # i++
        addi    $s5, $s5, 4             # s5 -> &A[i]
        j       loop
end_loop:

        beqz    $t2, thong_bao

        li      $v0, 4          #print string
        la      $a0, mess4
        syscall

        li      $v0, 1          #print integer
        addi    $a0, $s1, 0
        syscall
```

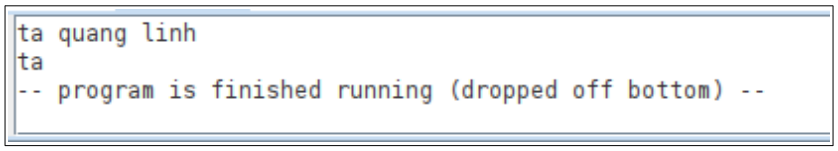
```
        li      $v0, 4          #print string
        la      $a0, mess5
        syscall

        li      $v0, 1          #print integer
        addi    $a0, $s2, 0
        syscall

end_main:
        li      $v0, 10         #exit
        syscall

thong_bao:
        li      $v0, 55         #print mess
        la      $a0, mess3
        syscall
        j       end_main
```

C. XÂU KÝ TỰ

*** Cách thực hiện:**

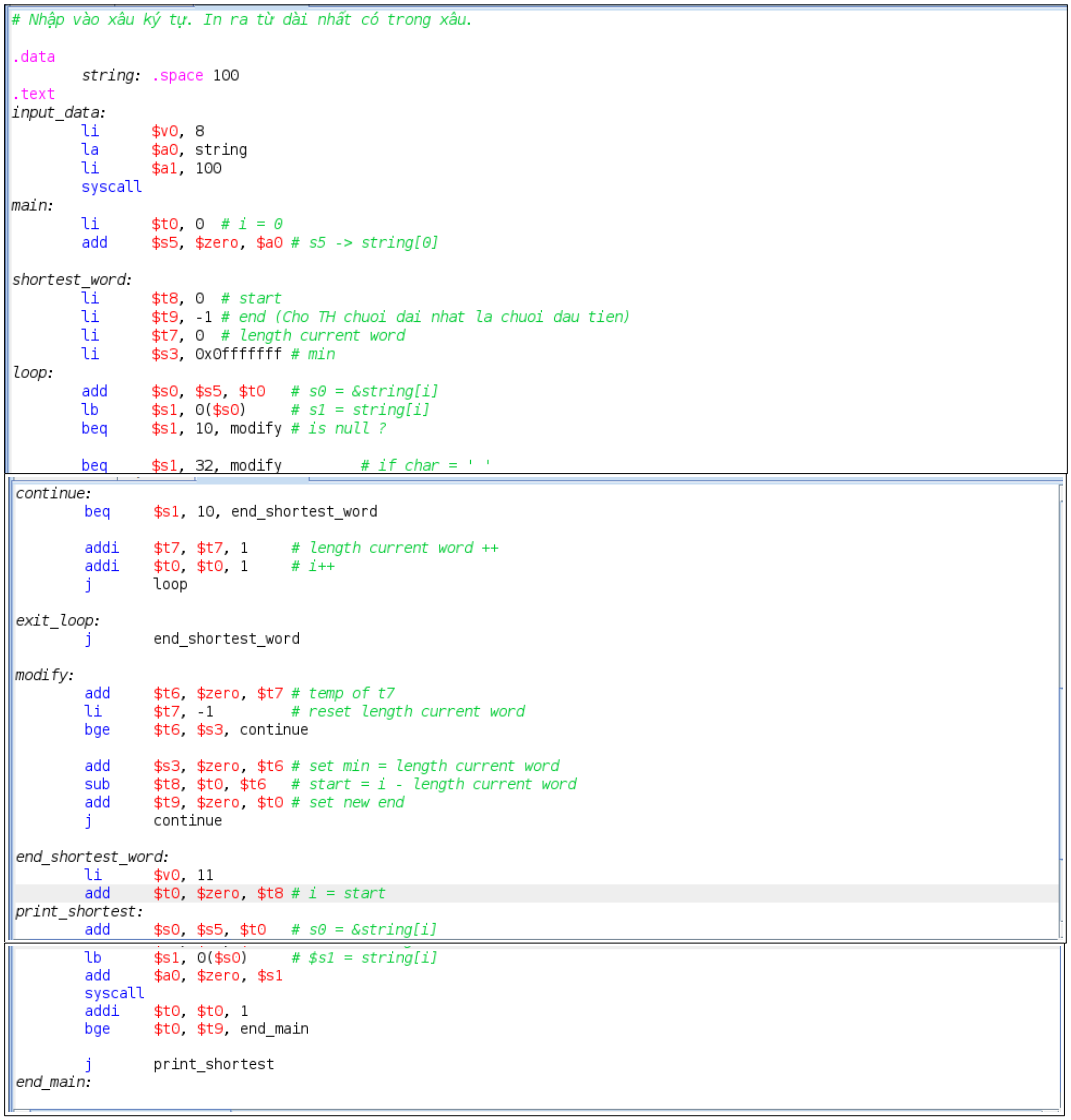
- Khởi tạo 3 loại giá trị: vị trí đầu tiên, vị trí cuối cùng và độ dài của chuỗi hiện tại
- Khởi tạo giá trị min đủ lớn để so sánh với độ dài chuỗi
- Duyệt từng kí tự của chuỗi, đọc được dấu cách hay dấu xuống dòng chuyển đến nhãn modify để so sánh, cắt chuỗi.
- Cuối cùng là duyệt để in các kí tự trong chuỗi con ngắn nhất.

KẾT QUẢ:



```
ta quang linh
ta
-- program is finished running (dropped off bottom) --
```

MÃ NGUỒN:



```
# Nhập vào xâu ký tự. In ra từ dài nhất có trong xâu.

.data
        string: .space 100
.text
input_data:
        li      $v0, 8
        la      $a0, string
        li      $a1, 100
        syscall
main:
        li      $t0, 0  # i = 0
        add     $s5, $zero, $a0 # s5 -> string[0]

shortest_word:
        li      $t8, 0  # start
        li      $t9, -1 # end (Cho TH chuoi dai nhat la chuoi dau tien)
        li      $t7, 0  # length current word
        li      $s3, 0x0fffffff # min
loop:
        add     $s0, $s5, $t0   # s0 = &string[i]
        lb      $s1, 0($s0)     # s1 = string[i]
        beq     $s1, 10, modify # is null ?

        beq     $s1, 32, modify         # if char = ' '
continue:
        beq     $s1, 10, end_shortest_word

        addi    $t7, $t7, 1     # length current word ++
        addi    $t0, $t0, 1     # i++
        j       loop

exit_loop:
        j       end_shortest_word

modify:
        add     $t6, $zero, $t7 # temp of t7
        li      $t7, -1         # reset length current word
        bge     $t6, $s3, continue

        add     $s3, $zero, $t6 # set min = length current word
        sub     $t8, $t0, $t6   # start = i - length current word
        add     $t9, $zero, $t0 # set new end
        j       continue

end_shortest_word:
        li      $v0, 11
        add     $t0, $zero, $t8 # i = start
print_shortest:
        add     $s0, $s5, $t0   # s0 = &string[i]
        lb      $s1, 0($s0)     # $s1 = string[i]
        add     $a0, $zero, $s1
        syscall
        addi    $t0, $t0, 1
        bge     $t0, $t9, end_main

        j       print_shortest
end_main:
```