

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
---∞📖∞---



Báo cáo giữa kì

Học phần : Thực hành kiến trúc máy tính

Giảng viên: Lê Bá Vui

Sinh viên thực hiện: Hà Văn Đức

MSSV : 20194510

Năm học : 2021-2022

Bài C-1 :

Nhập vào xâu ký tự. In ra màn hình từ ngắn nhất có trong xâu

- Source code :

```
# C-1

.data
tmp: .space 32          # save current word that we have just read
input: .space 100       # save user input
start: .asciiz "\n\n----- Let's start -----"
request: .asciiz "\nInput: "
message: .asciiz "\nShortest words is: "
newline: .asciiz "\n\t"
contPrompt: .asciiz "\nPress ENTER to continue..."

.text
li $v0, 4              # print_string
la $a0, start          # start message
syscall

init: li $v0, 4         # print_string call number
la $a0, request
syscall

li $v0, 8
la $a0, input          # pointer to string in memory
la $a1, 100
syscall

li $s0, 0              # i = 0 run through all text
li $s2, 0              # j = 0 is the index of tmp
li $s4, 1111           # current shortest length of result
li $s6, 0              # k = 0: index to clear tmp
la $a2, tmp            # load tmp address

main: jal checkNonAlpha # loop through input string

li $v0, 4              # print_string
la $a0, message        # message to print shortest words
syscall

la $a1, input          # load input address into $a1
j printWord            # print all words with min length

exit: li $v0, 10       # syscall to terminate
syscall

#####
###
# checkNonAlpha method: if current character is non-alphabet --> we got a word #
```

```
#####
###
checkNonAlpha:
    add $t4, $s0, $a0          # address of A[i] in $t4
    lb  $s1, 0($t4)           # load value of A[i]

    slti $t1, $s1, 65          # if ascii code is less than 48
    bne $t1, $zero, checkLength # get a word

    slti $t1, $s1, 91          # if ascii code is greater than 90
                                # and
    slti $t2, $s1, 97          # if ascii code is less than 97
    slt $t3, $t1, $t2
    bne $t3, $zero, checkLength # get a word

    slti $t1, $s1, 123         # if ascii character is greater than 122
    beq $t1, $zero, checkLength # get a word

    addi $s0, $s0, 1           # i = i + 1
    addi $s2, $s2, 1           # j = j + 1
    j    checkNonAlpha         # go to checkNonAlpha
#####
# checkLength method: if current word's length < min -> we have new min length #
#####
checkLength:
    slt $t3, $s2, $s4          # if length of current word is not shorter than current
min (j > min)
    beqz $t3, next             # reset j and move to the next one
    add $s4, $zero, $s2        # else, we have new min length
next: beq $s1, 10, done         # if A[i] = '\n' -> done
    addi $s0, $s0, 1           # i = i + 1
    li $s2, 0                  # j = 0
    j    checkNonAlpha         # proceed next character
done: li $s0, 0
    li $s2, 0
    jr $ra                     # return to main
printWord:
    add $t4, $s0, $a1          # address of A[i] in $t4
    lb  $s1, 0($t4)           # load value of A[i]

    slti $t1, $s1, 65          # if ascii code is less than 65
    bne $t1, $zero, compare    # get a word

    slti $t1, $s1, 91          # if ascii code is greater than 90
                                # and
    slti $t2, $s1, 97          # if ascii code is less than 97
    slt $t3, $t1, $t2
    bne $t3, $zero, compare    # get a word

    slti $t1, $s1, 123         # if ascii character is greater than 122
    beq $t1, $zero, compare    # get a word

    add $t5, $s2, $a2          # address of tmp[i] in $t5
    sb  $s1, 0($t5)           # store current character to tmp[i]
```

```

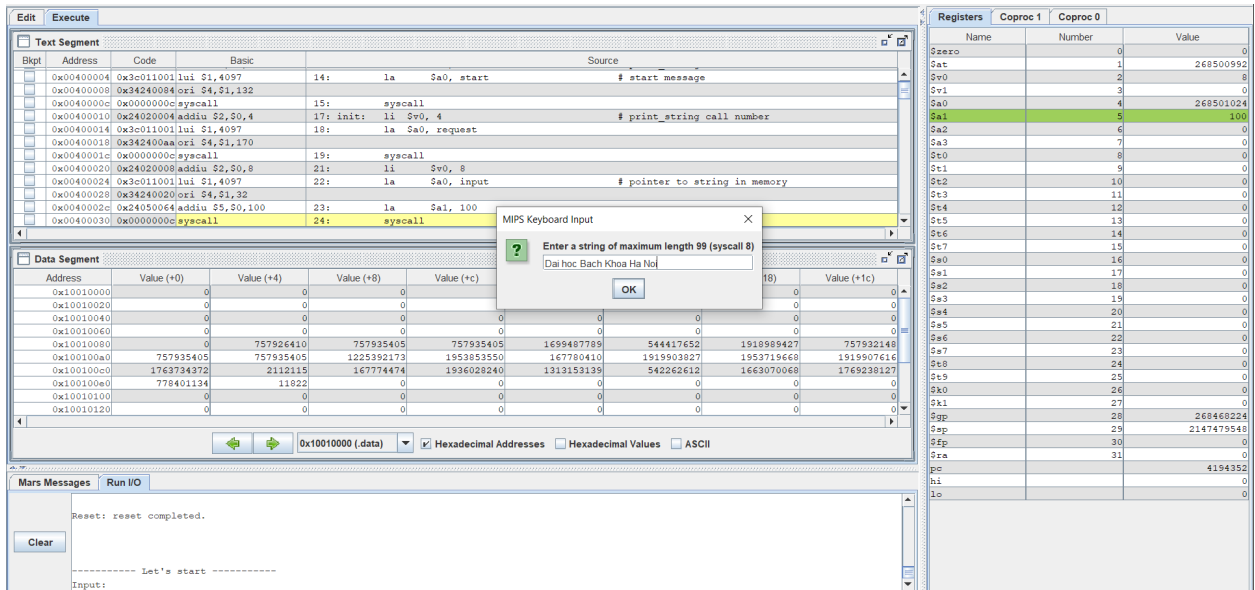
    addi    $s0, $s0, 1      # i = i + 1
    addi    $s2, $s2, 1      # j = j + 1
    j       printWord        # go to checkNonAlpha
#####
#   compare method: if current word's length = min length -> print      #
#####
compare:
    beq     $s2, $s4, print   # if length of current word is equal to current max
    j       reset            # reset j and move to the next one
print:
    li      $v0, 4            # print_word
    la      $a0, newline      # line break and tab
    syscall
    li      $v0, 4            # print_string
    la      $a0, tmp          # word with max length
    syscall
reset:
    addi    $s0, $s0, 1      # i = i + 1
    li      $s6, 0           # k = 0
clear_tmp:
    beq     $s6, $s2, return  # if k = j end
    add     $t4, $s6, $a2     # loop through tmp string
    sb      $0, 0($t4)       # turn tmp[k] into 0
    addi    $s6, $s6, 1      # k = k + 1
    j       clear_tmp        # continue to clear
return:
    li      $s2, 0           # j = 0
    beq     $s1, 10, end_main # if A[i] = '\n' -> done. Press enter to continue
    j       printWord        # proceed next character
end_main:
    j       exit

```

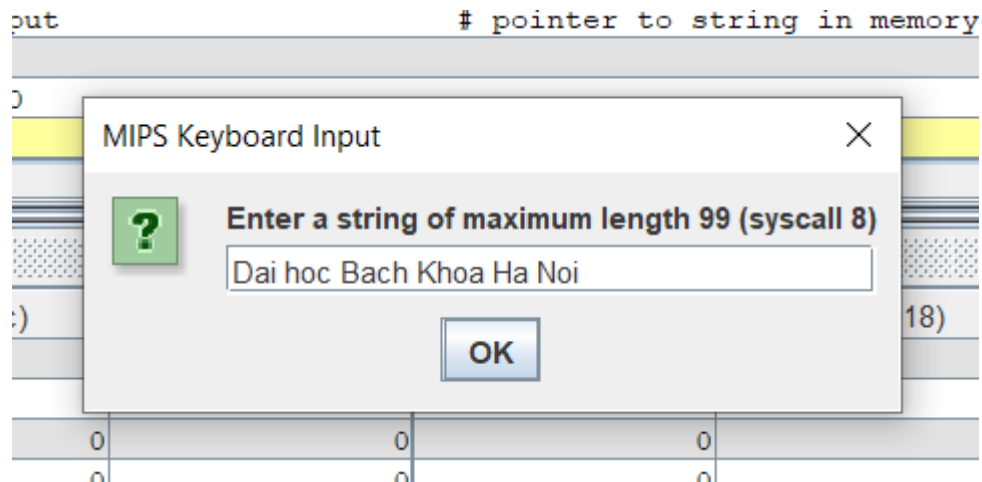
- **Phân tích cách thực hiện :**

Duyệt qua các tất cả các từ có trong xâu , tìm độ dài từ nhỏ nhất . Sau đó so sánh xem những từ nào có độ dài bằng độ dài nhỏ nhất đó thì in ra màn hình

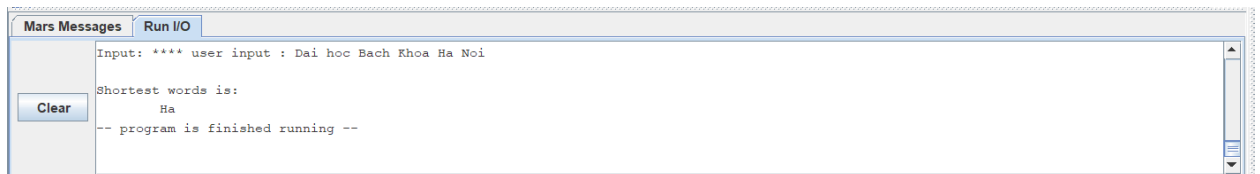
- **Kết quả chạy chương trình :**



- Xâu nhập vào :



- Kết quả chạy chương trình :



- Dựa theo xâu nhập vào , ta thấy kết quả in ra từ có độ dài ngắn nhất là “Hà” với độ dài là 2

Bài A-6 :

Nhập 3 số nguyên a, b, c, kiểm tra đây có phải là 3 cạnh của một tam giác không

- Source code

```
.data
Message1: .asciiz "Input a: "
Message2: .asciiz "Input b: "
Message3: .asciiz "Input c: "
Message4: .asciiz "Khong phai la 3 canh tam giac!"
Message5: .asciiz "La 3 canh tam giac!"
.text
check_input_a: addi $v0, $zero, 51      # Doc gia tri a
               la $a0, Message1
               syscall
               bne $a1, $zero, check_input_a  # Neu $a1 != 0 thi gia tri a khong phai la so nguyen,
nhap lai
               nop                      # Giai quyet van de delay branching
               add $s1, $zero, $a0      # Luu gia tri a vao thanh ghi $s1

check_input_b: addi $v0, $zero, 51      # Doc gia tri b
               la $a0, Message2
               syscall
               bne $a1, $zero, check_input_b  # Neu $a1 != 0 thi gia tri b khong phai la so nguyen,
nhap lai
               nop
               add $s2, $zero, $a0      # Luu gia tri b vao thanh ghi $s2

check_input_c: addi $v0, $zero, 51      # Doc gia tri c
               la $a0, Message3
               syscall
               bne $a1, $zero, check_input_c  # Neu $a1 != 0 thi gia tri c khong phai la so nguyen,
nhap lai
               nop
               add $s3, $zero, $a0      # Luu gia tri c vao thanh ghi $s3

main:
    blt $s1, 1, NotTriangle             # Neu a < 0 thi khong phai la tam giac
    blt $s2, 1, NotTriangle             # Neu b < 0 thi khong phai la tam giac
    blt $s3, 1, NotTriangle             # Neu c < 0 thi khong phai la tam giac

    add $t1, $s1, $s2                   # $t1 = a + b
    bge $s3, $t1, NotTriangle           # Neu c >= a + b thi khong phai la tam giac
    nop
    add $t1, $s2, $s3                   # $t1 = b + c
    bge $s1, $t1, NotTriangle           # Neu a >= b + c thi khong phai la tam giac
    nop
```

```

add $t1, $s1, $s3          # $t1 = a + c
bge $s2, $t1, NotTriangle  # Neu b >= a + c thi khong phai la tam giac
nop

IsTriangle:
addi $v0, $zero, 55        # Hien thi Message5 ra man hinh
la $a0, Message5
addi $a1, $zero, 1
syscall
j done
nop
NotTriangle:
addi $v0, $zero, 55        # Hien thi Message4 ra man hinh
la $a0, Message4
addi $a1, $zero, 0
syscall
done:

```

- **Phân tích cách thực hiện :**

Điều kiện để 3 số nguyên a, b, c tạo thành 3 cạnh của 1 tam giác là 3 số này phải dương và tổng của hai số bất kì trong 3 số a, b, c phải lớn hơn số còn lại . Ngược lại thì không phải 3 cạnh của tam giác.

Tức là cần kiểm tra 3 điều kiện sau :

$$a + b > c$$

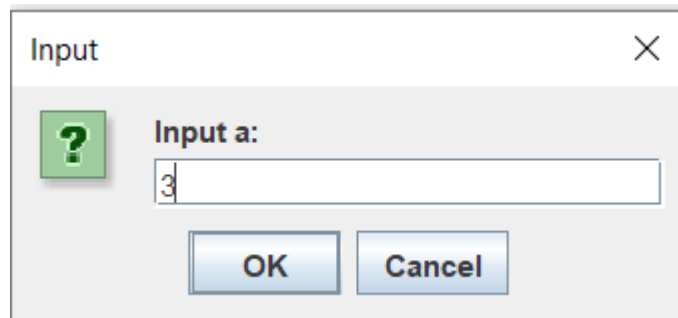
$$b + c > a$$

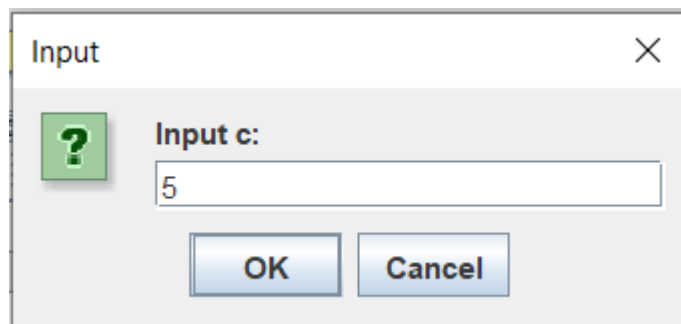
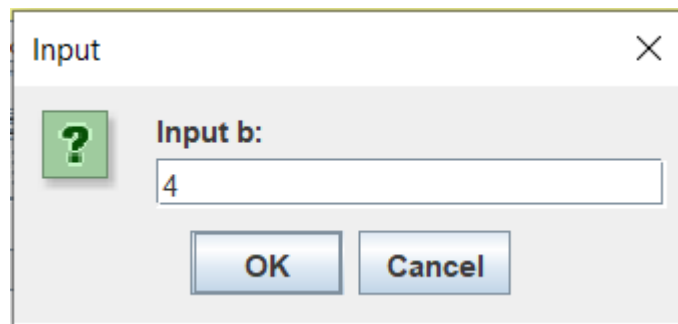
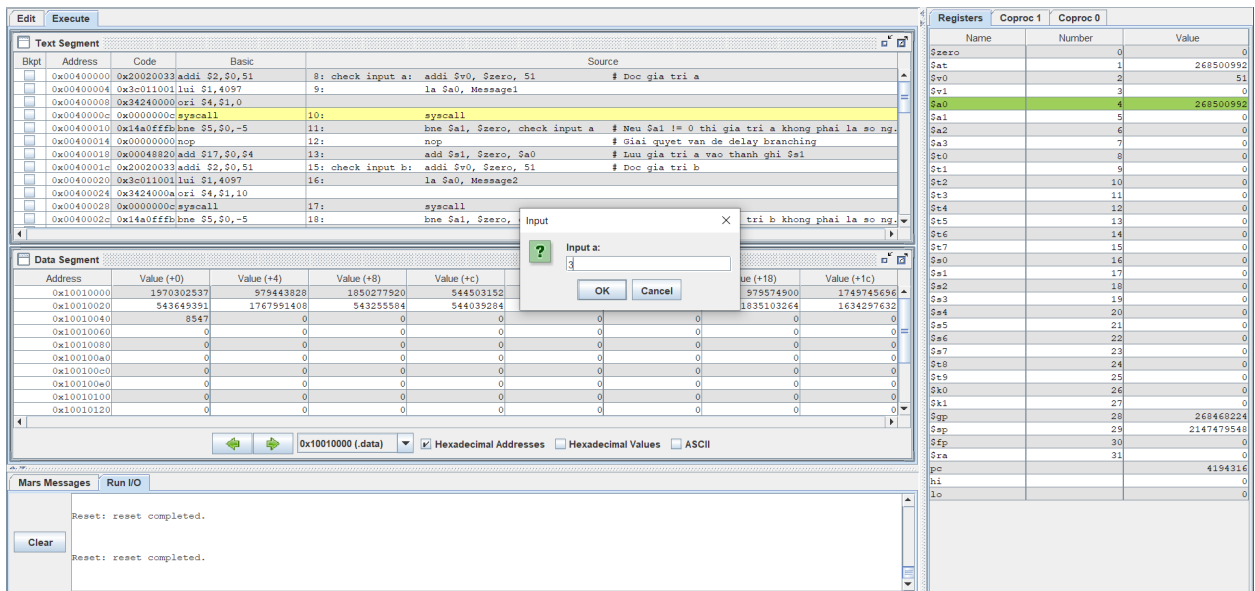
$$c + a > b$$

Nếu thỏa mãn cả 3 điều kiện thì 3 số a, b, c là 3 cạnh của 1 tam giác . Ngược lại , 1 trong 3 điều kiện sai thì không phải là 3 cạnh của 1 tam giác.

- **Kết quả chạy chương trình :**

- Nhập vào 3 số a, b, c

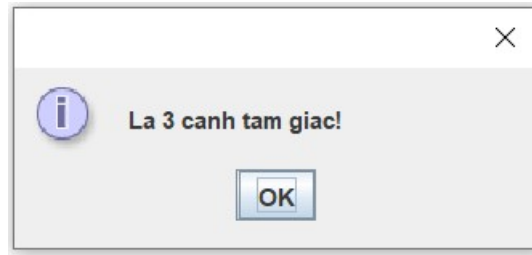




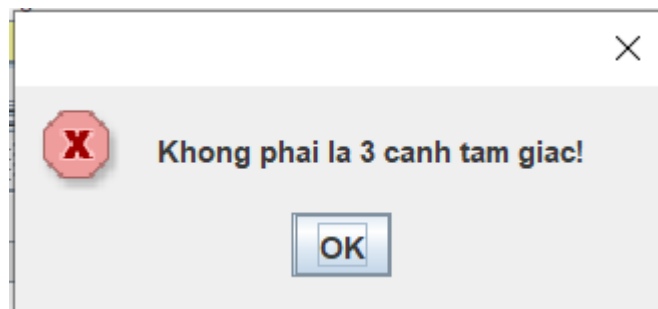
- Kết quả chạy :
In ra màn hình thông báo sau :

La 3 cạnh của tam giác

⇒ Ta thấy kết quả đúng vì 3, 4, 5 tạo thành 3 cạnh của 1 tam giác



- Trường hợp 2 : Nhập vào 3 số a, b, c lần lượt là 1, 3, 7
- Xem kết quả :



⇒ 1, 3, 7 không phải là 3 cạnh của 1 tam giác → chương trình chạy đúng

Bài B-8 :

Nhập mảng số nguyên từ bàn phím. Sắp xếp mảng sao cho các số âm đứng đầu dãy, các số dương đứng cuối dãy

- Source code