

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

\*\*\*\*\*



# **BÁO CÁO**

**Bài tập giữa kỳ**

**Học phần: Thực hành kiến trúc máy tính**

**Giảng viên hướng dẫn: Lê Bá Vui**

**Sinh viên thực hiện: Phạm Huy Cảnh - 20194490**

**Mã lớp: 130938**

**Hà Nội, tháng 5 năm 2022**

**Phần A – Bài 2:** Nhập 2 số nguyên dương M và N từ bàn phím. In ra màn hình bội số chung nhỏ nhất của M và N.

## 1. Phân tích cách thực hiện

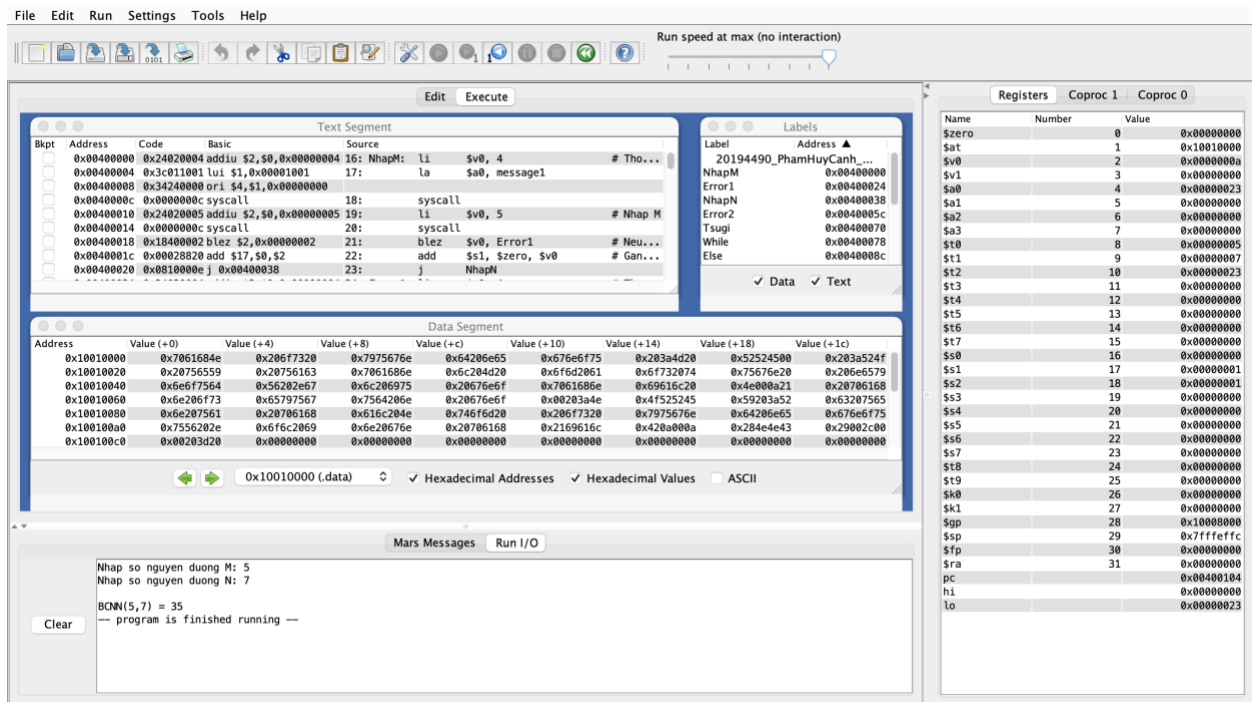
- Bước 1: Yêu cầu người dùng nhập số nguyên dương M và N, kiểm tra điều kiện nhập xem có đúng là số nguyên dương hay không. Nếu không đúng thì báo lỗi và cho phép nhập lại.

- Bước 2: Tìm ước chung lớn nhất của hai số M và N

- Bước 3: Kết quả bội chung nhỏ nhất của hai số M và N bằng tích M.N chia cho ước chung lớn nhất. In ra màn hình kết quả

## 2. Ý nghĩa của các chương trình con nếu có (Không có)

## 3. Ảnh chụp màn hình kết quả thực hiện



## 4. Mã nguồn, có chú thích

<https://drive.google.com/drive/folders/1ToUA4HZSozHSb4BZGC0QHihTWaXgOe5x?usp=sharing>

```
#-----#
# Computer Architecture Lab - Midterm #
# Author: Pham Huy Canh - 20194490 #
# A. So nguyen - Bai 2 #
#-----#

.data
message1: .asciiz "Nhap so nguyen duong M: "
error1: .asciiz "ERROR: Yeu cau nhap M la mot so nguyen duong. Vui long nhap lai!\n"
message2: .asciiz "Nhap so nguyen duong N: "
error2: .asciiz "ERROR: Yeu cau nhap N la mot so nguyen duong. Vui long nhap lai!\n"
message3: .asciiz "\nBCNN("
message4: .asciiz ", "
message5: .asciiz ")" = "

.text
NhapM: li $v0, 4 # Thong bao nhap M
la $a0, message1
syscall
li $v0, 5 # Nhap M
syscall
blez $v0, Error1 # Neu M <= 0 thi bao loi
add $s1, $zero, $v0 # Gan gia tri M vao $s1
j NhapN
Error1: li $v0, 4 # Thong bao loi
la $a0, error1
syscall
j NhapM # Quay lai nhap lai M
NhapN: li $v0, 4 # Thong bao nhap N
la $a0, message2
syscall
li $v0, 5 # Nhap N
syscall
blez $v0, Error2 # Neu N <= 0 thi thong bao loi
add $s2, $zero, $v0 # Gan gia tri N vao $s2
j Tsugi
Error2: li $v0, 4 # Thong bao loi
la $a0, error2
syscall
j NhapN # Quay lai nhap lai N
Tsugi: add $t0, $zero, $s1 # Gan $t0 = $s1 = M
add $t1, $zero, $s2 # Gan $t1 = $s2 = N
While: beq $s1, $s2, BCNN # Neu M = N thi ket thuc vong lap va nhay den BCNN
blt $s1, $s2, Else
# Neu M > N thi thuc hien if; Neu M < N thi nhay den thuc hien else
sub $s1, $s1, $s2 # M = M - N
j While
Else: sub $s2, $s2, $s1 # N = N - M
j While
BCNN: mul $t2, $t0, $t1
# $t2 = M*N (M va N ban dau), day la ket qua cua tich giua UCLN va BCNN
div $t2, $t2, $s1 # BCNN = $t2 chia $s1 (UCLN)
# -----Hien thi thong bao ket qua-----
# BCNN(M,N) = $t2
li $v0, 4
la $a0, message3
syscall
li $v0, 1 # Hien thi gia tri M khi nhap vao
add $a0, $zero, $t0
syscall
li $v0, 4
la $a0, message4
syscall
li $v0, 1 # Hien thi gia tri N khi nhap vao
add $a0, $zero, $t1
syscall
li $v0, 4
la $a0, message5
syscall
li $v0, 1 # Hien thi ket qua BCNN
add $a0, $zero, $t2
syscall
Exit: li $v0, 10
syscall
```

**Phần B – Bài 7:** Nhập mảng số nguyên từ bàn phím. In ra màn hình phần tử có số lần xuất hiện ít nhất trong mảng và các vị trí của phần tử đó.

## 1. Phân tích cách thực hiện

- Bước 1: Yêu cầu người dùng nhập vào một số nguyên dương là tổng số phần tử của mảng (mảng A), kiểm tra xem số đó có phải số nguyên dương hay không, nếu không thì thông báo lỗi và yêu cầu người dùng nhập lại.

- Bước 2: Sau đó, cho người dùng nhập lần lượt từng giá trị của phần tử trong mảng

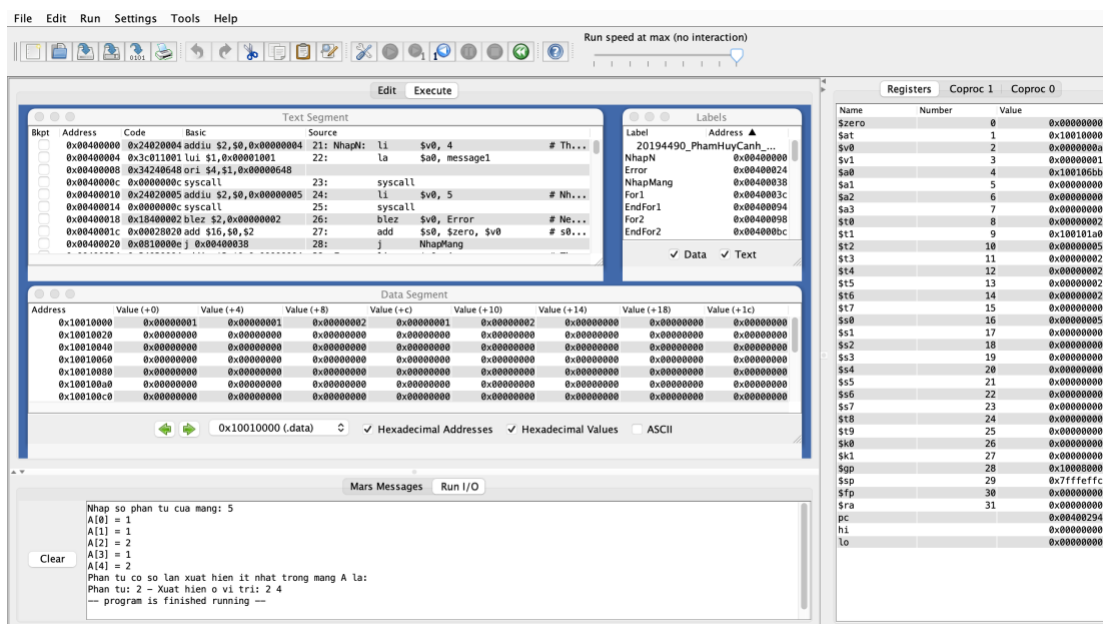
- Bước 3: Tìm số lần xuất hiện của các phần tử trong mảng A và lưu lại giá trị của phần tử đó và số lần xuất hiện của phần tử lần lượt tương ứng vào hai mảng C, D

- Bước 4: Từ mảng D ta tìm ra số lần xuất hiện ít nhất (min)

- Bước 5: Duyệt lần lượt từng giá trị của mảng D nếu phần tử  $D[k]$  nào trong mảng bằng min thì thực hiện in kết quả bằng cách in ra giá trị  $C[k]$  tương ứng. Cùng với đó là in ra từng vị trí xuất hiện của phần tử trong mảng A bằng cách so sánh  $C[k]$  với  $A[j]$  nếu bằng nhau thì j chính là chỉ số cần in.

## 2. Ý nghĩa của các chương trình con nếu có (Không có)

### 3. Ảnh chụp màn hình kết quả thực hiện



## 4. Mã nguồn, có chú thích

<https://drive.google.com/drive/folders/1ToUA4HZSozHSb4BZGC0QHihTWaXgOe5x?usp=sharing>

```
#-----#
# Computer Architecture Lab - Midterm #
# Author: Pham Huy Canh - 20194490 #
# B. Mang - Bai 7 #
#-----#

.data
A: .space 400
B: .space 400
C: .space 400
D: .space 400
arrMes1: .asciiz "A["
arrMes2: .asciiz "]" = "
message1: .asciiz "Nhap so phan tu cua mang: "
message2: .asciiz "Phan tu co so lan xuat hien it nhat trong mang A la:"
message3: .asciiz "\nPhan tu: "
message4: .asciiz " - Xuat hien o vi tri: "
space: .asciiz " "
error: .asciiz "ERROR: So phan tu mang phai la mot so duong. Vui long nhap lai!\n"

.text
NhapN: li $v0, 4 # Thong bao nhap so phan tu mang A
la $a0, message1
syscall
li $v0, 5 # Nhap N
syscall
blez $v0, Error # Neu N <= 0 thi thong bao loi va nhap lai N
add $s0, $zero, $v0 # s0 = N
j NhapMang
Error: li $v0, 4 # Thong bao loi
la $a0, error
syscall
j NhapN # Quay lai nhap lai N
NhapMang: li $t0, -1 # Khoi tao i cho vong For1

For1: addi $t0, $t0, 1 # i = i + 1
bge $t0, $s0, EndFor1 # Neu i = N thi ket thuc vong For1
# Thong bao nhap tung phan tu cua mang
# A[i] =
li $v0, 4
la $a0, arrMes1
syscall
li $v0, 1
add $a0, $zero, $t0
syscall
li $v0, 4
la $a0, arrMes2
syscall

li $v0, 5 # Nhap gia tri phan tu
syscall
la $a0, A # Ghi gia tri vua nhap vao phan tu A[i]
sll $t1, $t0, 2
add $t1, $t1, $a0
sw $v0, 0($t1)
j For1 # Tiep tuc vong lap
EndFor1: li $t0, -1 # Khoi tao i cho vong For2
For2: addi $t0, $t0, 1 # i = i + 1
beq $t0, $s0, EndFor2 # Neu i = N thi ket thuc vong For2
la $a0, B # Gan B[i] bang 1
sll $t1, $t0, 2
add $t1, $t1, $a0
li $v0, 1
sw $v0, 0($t1)
j For2 # Tiep tuc vong lap
EndFor2: li $t0, -1 # Khoi tao i cho vong For3
li $t5, 0 # Khoi tao k = 0
```

```

For3:  addi    $t0, $t0, 1      # i = i + 1
      beq    $t0, $s0, EndFor3 # Neu i = N thi ket thuc vong For3
      li     $v1, 1          # Khoi tao Count = 1
If1:   la     $a0, B          # Lay gia tri B[i] = $v0
      sll    $t1, $t0, 2
      add    $t1, $t1, $a0
      lw     $v0, 0($t1)
      beq    $v0, $zero, For3 # Neu B[i] = 0 bo qua cau lenh trong If1 tiep tục vong lap
      li     $v0, 0          # Gan B[i] = 0
      sw     $v0, 0($t1)
      addi   $t2, $t0, 0      # Khoi tao j cho vong lap For 4 nam trong For3
For4:  addi   $t2, $t2, 1      # j = i + 1
      beq    $t2, $s0, EndFor4 # Neu j = N thi ket thuc For4
If2:   la     $a0, A          # Lay gia tri A[i]
      sll    $t1, $t0, 2
      add    $t1, $t1, $a0
      lw     $t1, 0($t1)
      sll    $t3, $t2, 2
      add    $t3, $t3, $a0
      lw     $t3, 0($t3)      # Lay gia tri A[j]
      bne    $t1, $t3, EndIf2
# Kiem tra, neu A[j] != A[i] thi khong chay lenh trong If2 va tiep tục vong lap
      addi   $v1, $v1, 1      # Count++
      la     $a0, B          # Gan B[j] = 0
      sll    $t3, $t2, 2
      add    $t3, $t3, $a0
      li     $t4, 0
      sw     $t4, 0($t3)
EndIf2: j     For4
EndFor4: la    $a0, A          # Lay gia tri A[i]
      sll    $t1, $t0, 2
      add    $t1, $t1, $a0
      lw     $t1, 0($t1)

      la     $a0, C          # Ghi C[k] = A[i]
      sll    $t2, $t5, 2
      add    $t2, $t2, $a0
      sw     $t1, 0($t2)

      la     $a0, D          # Ghi D[k] = Count
      sll    $t2, $t5, 2
      add    $t2, $t2, $a0
      sw     $v1, 0($t2)
      addi   $t5, $t5, 1      # k = k + 1
      j      For3            # Quay lai For3 tiep tục vong lap
EndFor3:
      li     $t0, 0          # Khoi tao i cho For5
      la     $a0, D          # Gan min = D[0]
      lw     $t6, 0($a0)
For5:  addi   $t0, $t0, 1      # i = i + 1
      beq    $t0, $t5, EndFor5 # Neu i = N thi ket thuc For5
If3:   la     $a0, D          # Lay gia tri D[i]
      sll    $t2, $t0, 2
      add    $t2, $t2, $a0
      lw     $t2, 0($t2)
      bge    $t2, $t6, For5    # Neu D[i] >= min thi bo qua If tiep tục vong lap
      add    $t6, $t2, $zero    # Neu D[i] < min thi gan min = D[i]
      j      For5
# Hien thi ket qua
EndFor5: li    $v0, 4
      la     $a0, message2
      syscall
      li     $t0, -1          # Khoi tao i cho vong For6
For6:  addi   $t0, $t0, 1      # i = i + 1
      beq    $t0, $t5, Exit    # Neu i = k thi thoat
If4:   la     $a0, D          # Lay gia tri D[i]
      sll    $t2, $t0, 2
      add    $t2, $t2, $a0
      lw     $t2, 0($t2)
      bne    $t2, $t6, For6    # Neu D[i] != min thi bo qua If4 va quay lai vong lap

```

```

# In gia tri phan tu
li $v0, 4
la $a0, message3
syscall

la $a0, C          # Lay va in ra gia tri cua phan tu C[i]
sll $t2, $t0, 2
add $t2, $a0, $t2
lw $a0, 0($t2)
li $v0, 1
syscall

# In vi tri cua phan tu trong mang A
li $v0, 4
la $a0, message4
syscall
li $t2, -1          # Khoi tao j cho For7
For7: addi $t2, $t2, 1    # j = j + 1
      beq $t2, $s0, EndFor7 # Neu j = N thi ket thuc For7
If5:  la $a0, A          # Lay gia tri A[j]
      sll $t3, $t2, 2
      add $t3, $a0, $t3
      lw $t3, 0($t3)

      la $a0, C          # Lay gia tri C[i]
      sll $t4, $t0, 2
      add $t4, $a0, $t4
      lw $t4, 0($t4)

      bne $t3, $t4, For7    # Neu A[j] != C[i] thi bo qua If5 va tiep tuc vong lap

      li $v0, 1          # In ra j (chi so cua phan tu thoa man)
      add $a0, $t2, $zero
      syscall
      li $v0, 4
      la $a0, space
      syscall
      j For7
EndFor7: j For6
Exit:  li $v0, 10         # Thoat
      syscall

```



## Phần C – Bài 1: Nhập vào chuỗi ký tự. In ra màn hình từ ngắn nhất có trong chuỗi.

### 1. Phân tích cách thực hiện

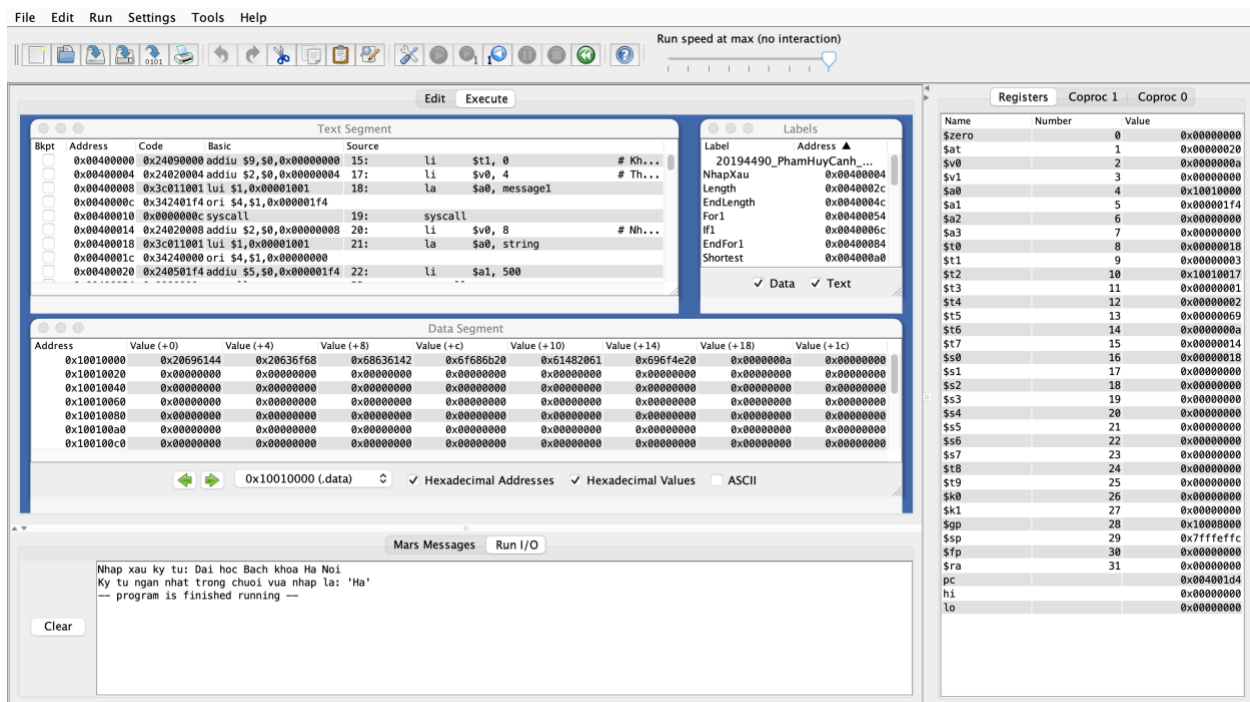
- Bước 1: Yêu cầu người dùng nhập vào một chuỗi ký tự, nếu là chuỗi rỗng hoặc chỉ chứa ký tự cách thì thông báo lỗi và yêu cầu người dùng nhập lại.

- Bước 2: Tìm độ dài ngắn nhất của từ trong chuỗi

- Bước 3: Lần lượt đếm độ dài của từng từ trong chuỗi và kiểm tra với độ dài ngắn nhất tìm được trong bước 2 nếu bằng nhau thì in ra từ đó

### 2. Ý nghĩa của các chương trình con nếu có (Không có)

### 3. Ảnh chụp màn hình kết quả thực hiện





## 4. Mã nguồn, có chú thích

<https://drive.google.com/drive/folders/1ToUA4HZSozHSb4BZGC0QHihTWaXgOe5x?usp=sharing>

```
#-----#
# Computer Architecture Lab - Midterm #
# Author: Pham Huy Canh - 20194490 #
# C. Xau ky tu - Bai 1 #
#-----#

.data
string: .space 500
message1: .ascii "Nhap xau ky tu: "
error: .ascii "ERROR: Khong duoc de trong. Vui long nhap lai\n"
message2: .ascii "Ky tu ngan nhat trong chuoai vua nhap la: "
comma: .ascii ", "
sqm: .ascii ""

.text
li $t1, 0 # Khoi tao check = 0
NhapXau:
li $v0, 4 # Thong bao nhap xau
la $a0, message1
syscall
li $v0, 8 # Nhap xau
la $a0, string
li $a1, 500
syscall
li $t0, 0 # Khoi tao i cho Length
Length: la $a0, string
add $t2, $t0, $a0
lb $t2, 0($t2) # Lay ra string[i] = $t2
beq $t2, 10, EndLength # Kiem tra neu string[i] = '\n' thi ket thuc Length
addi $t0, $t0, 1 # i++
j Length
EndLength:
add $s0, $t0, $zero # Gan $s0 = strlen(string)
li $t0, -1 # Khoi tao i cho For1
For1: addi $t0, $t0, 1 # i++
beq $t0, $s0, EndFor1 # Kiem tra i = strlen(string) thi ket thuc For1
la $a0, string
add $t2, $t0, $a0
lb $t2, 0($t2) # Lay ra string[i] = $t2
If1: beq $t2, 32, For1
beq $t2, 10, For1
li $t1, 1 # Neu string[i] != ' ' && string[i] != '\n' thi gan check = 1
j For1
EndFor1:
bne $t1, 0, Shortest # Kiem tra check khac 0 thi khong thong bao loi
li $v0, 4 # In thong bao loi xau rong va quay lai nhap xau
la $a0, error
syscall
j NhapXau
Shortest:
li $t3, 0 # Khoi tao count = 0
li $t4, 100 # Khoi tao shortest = 100
li $t0, -1 # Khoi tao i cho For2
For2: addi $t0, $t0, 1 # i++
blt $s0, $t0, EndFor2 # Kiem tra neu strlen(string) < i thi ket thuc vong lap
If3: la $a0, string
add $t2, $t0, $a0
lb $t2, 0($t2) # Lay ra string[i]
beq $t2, 32, Else3
beq $t2, 10, Else3
addi $t3, $t3, 1
# Neu string[i] != ' ' && string[i] != '\n' thi count++, neu dieu kien sai thi thuc hien Else
j For2
Else3: bge $t3, $t4, ResetCount # Neu count < shortest && count != 0 thi shortest = count
beq $t3, 0, ResetCount
add $t4, $t3, $zero
ResetCount:
li $t3, 0 # count = 0
j For2
EndFor2:
```

```

    li $t0, -1      # Khoi tao i cho For3
    li $t1, 0       # $t1 = dem = 0
    li $t3, 0       # $t3 = z = 0
    li $v0, 4       # Thông báo in kết quả
    la $a0, message2
    syscall

For3:  addi $t0, $t0, 1      # i++
      beq $t0, $s0, Exit    # Kiểm tra nếu i = strlen(string) thì kết thúc For3
If4:   la $a0, string
      add $t2, $t0, $a0
      lb $t5, 0($t2)        # Lấy ra string[i]
      lb $t6, 1($t2)        # Lấy ra string[i+1]
      beq $t5, 32, Else4    # Kiểm tra nếu string[i] = ' ' thì thực hiện Else4
      addi $t1, $t1, 1      # dem++
If5:   bne $t1, $t4, For3
# Kiểm tra nếu dem = shortest thì kiểm tra tiếp lệnh bên dưới, nếu không bằng thì quay lại For3
      beq $t6, 32, If6      # Kiểm tra string[i+1] = ' ' thì nhảy đến If6
      beq $t6, 10, If6      # Kiểm tra string[i+1] = '\0' thì nhảy đến If6
      j For3               # Nếu các điều kiện trên không thỏa mãn thì quay lại For3
If6:   beq $t3, 0, EndIf6    # Kiểm tra nếu z = 0 thì không thực hiện lệnh trong If6
      li $v0, 4             # In dấu ", "
      la $a0, comma
      syscall
EndIf6: li $v0, 4           # In dấu nhảy đơn
      la $a0, sqm
      syscall

# In kết quả
      sub $t7, $t0, $t1     # Khởi tạo j = i - dem + 1
For4:  addi $t7, $t7, 1      # j++
      bgt $t7, $t0, EndFor4 # Kiểm tra nếu j > i thì kết thúc For4
      la $a0, string
      add $a0, $t7, $a0
      lb $a0, 0($a0)        # Lấy ra string[j]
      li $v0, 11           # In ra string[j]
      syscall
      j For4
EndFor4:
      li $v0, 4           # In dấu nhảy đơn
      la $a0, sqm
      syscall
      add $t3, $t3, 1      # z++
      j For3
Else4: li $t1, 0           # Dem = 0
      j For3
Exit:  li $v0, 10          # Exit
      syscall

```