**BÁO CÁO MÔN THỰC HÀNH KIẾN TRÚC MÁY TÍNH**

Nhóm: 7

Thành viên: Đặng Bá Dũng MSSV: 20176731

Nguyễn Bách Tùng MSSV: 20176899

1. **Bài miniproject 13**

**Đề bài:**

Ticket numbers usually consist of an even number of digits. A ticket number is  
considered lucky if the sum of the first half of the digits is equal to the sum of the  
second half. Given a ticket number n, determine if it's lucky or not.  
Example  
For n = 1230, the output should be isLucky(n) = true;  
For n = 239017, the output should be isLucky(n) = false.

**Code:**

# Bài mini-project 13

.data

inputMessage: .asciiz "Input number want to check: "

errorMessage: .asciiz "Input value isn't valid!"

luckyMessage: .asciiz "isLucky(n) = true"

notluckyMessage: .asciiz "isLucky(n) = false"

.text

START:

li $t0, 10

li $s1, 0

li $s4, 0

li $s5, 0

MAIN:

jal INPUT

nop

jal COUNT\_SUM

nop

jal CHECK

nop

div $s1, $s1, 2

jal SUM\_HALF

nop

add $s5, $s5, $s5

beq $s5, $s4, LUCKY

j NOT\_LUCKY

INPUT:

li $v0, 51

la $a0, inputMessage

syscall

bne $a1, $zero, ERROR

add $s0, $a0, $zero

add $t1, $s0, $zero

add $t2, $s0, $zero

jr $ra

COUNT\_SUM:

div $t1, $t0

mflo $t1

mfhi $t3

add $s4, $s4, $t3

add $s1, $s1, 1

bne $t1, $zero, COUNT\_SUM

jr $ra

SUM\_HALF:

div $t2, $t0

mflo $t2

mfhi $t3

add $s5, $s5, $t3

sub $s1, $s1, 1

bne $s1, $zero, SUM\_HALF

jr $ra

CHECK: andi $t4 , $s1 , 1 bne $t4, $zero, ERROR jr $ra

ERROR:

li $v0, 55

la $a0, errorMessage

syscall

j EXIT

LUCKY:

li $v0, 55

la $a0, luckyMessage

syscall

j EXIT

NOT\_LUCKY:

li $v0, 55

la $a0, notluckyMessage

syscall

j EXIT

EXIT:

**Phân tích cách thực hiện:**

* Sử dụng syscall InputDialogInt để nhận giá trị người dùng nhập vào.
* Báo lỗi nếu giá trị nhập vào không phải là số.
* Đếm số lượng chữ số trong số nhập vào và tính tổng các chữ số đó.
* Nếu số lượng số là số lẻ thì thông báo lỗi.
* Tính tổng nửa các chữ số bên phải của số nhập vào.
* So sánh 2 lần nửa tổng với tổng => Nếu 2 kết quả giống nhau thì thông báo số may mắn và ngược lại.

**Các thanh ghi được sử dụng:**

* $t0 = 10 => Thanh ghi $t0 lưu giá trị hằng số 10 để thực hiện phép tính n/10 lấy phần nguyên và phần dư.
* $s1 => Đếm số lượng các chữ số trong số nhập vào.
* $s4 => Tính tổng các chữ số có trong số nhập vào.
* $s5 => Tính tổng nửa các chữ số bên phải của số nhập vào.
* Ngoài ra có các thanh ghi khác sử dụng trong quá trình tính toán, lưu giá trị tạm thời.

**Ý nghĩa các phần trong chương trình:**

* START => Khởi tạo các giá trị ban đầu của các biến.
* MAIN => Chương trình chính
* INPUT => Lấy giá trị người dùng nhập vào và thông báo lỗi nếu thông tin nhập vào không đúng định dạng.
* COUNT\_SUM => Đếm số lượng các chữ số trong số nhập vào và tính tổng các số đó.
* SUM\_HALF => Tính tổng nửa các chữ số bên phải của số nhập vào.
* ERROR => Thông báo nếu dữ liệu nhập vào lỗi.
* LUCKY => Thông báo nếu số nhập vào là số may mắn.
* NOT\_LUCKY => Thông báo nếu số nhập vào không là số may mắn.

1. **Bài miniproject 14**

**Đề bài :** Given two strings, find the number of common characters between them.  
Example: For s1 = "aabcc" and s2 = "adcaa", the output should be  
commonCharacterCount(s1, s2) = 3. Strings have 3 common characters - 2 "a"s and  
1 "c".

**Code:**

.data

s1: .space 50 # xau s1

s2: .space 50 # xau s2

count\_s1: .word 0:26 # mang dem so lan xuat hien cua cac chu cai trong xau s1

count\_s2: .word 0:26 # mang dem so lan xuat hien cua cac chu cai trong xau s2

count\_common\_char\_freq: .word 0:26 # mang luu tong so cac chu cai trung nhau cua 2 xau s1 va s2

Message1: .asciiz "commonCharacterCount(s1, s2) = "

Message2: .asciiz "\n"

Message3: .asciiz " \" "

Message4: .asciiz " Nhap xau thu nhat:"

Message5: .asciiz " Nhap xau thu hai:"

errorMessage: .asciiz "Input value isn't valid!"

.text

string\_1: #ham yeu cau nhap xau s1

li $v0,54 #thuc hien lenh nhap so 54

la $a0, Message4 #gan a0 = mess4 = 'nhap xau thu nhat :'

la $a1, s1 #gan dia chi xau s1 vao thanh ghi a1

li $a2, 50 #a2 la do dai toi da cua xau s1 =50

li $t9, 0 #do dai xau s1 ban dau t9 = 0

syscall

get\_length\_s1: # ham lay do dai cua xau s1

lb $t8, s1($t9) #luu gia tri thu t9 cua xau s1 vao t8

beq $t8, $zero, end\_get\_length\_s1 #neu nhu t8 bang 0 goi ham end\_get\_length\_s1

add $t9,$t9,1 #neu t8 khac 0 thi cong t9 len 1

j get\_length\_s1 #de quy

end\_get\_length\_s1: #ham lay do dai cua xau s1

subi $s0, $t9 , 1 # s0 = t9-1 la do dai thuc te cua xau s1 do cuoi moi xau deu co dau enter

string\_2: #ham yeu cau nhap xau s1

li $v0,54 #thuc hien lenh nhap so 54

la $a0, Message5 #gan a0 = mess45 = 'nhap xau thu hai :'

la $a1, s2 #gan dia chi xau s2 vao thanh ghi a1

li $a2, 50 #a2 la do dai toi da cua xau s2 =50

li $t9, 0 #do dai xau s2 ban dau t9 = 0

syscall

get\_length\_s2: # ham lay do dai cua xau s2

lb $t8, s2($t9) #luu gia tri thu t9 cua xau s2 vao t8

beq $t8, $zero, end\_get\_length\_s2 #neu nhu t8 bang 0 goi ham end\_get\_length\_s2

add $t9,$t9,1 #neu t8 khac 0 thi cong t9 len 1

j get\_length\_s2 #de quy

end\_get\_length\_s2: #ham lay do dai cua xau s2

subi $s1, $t9 , 1 # s1 = t9-1 la do dai thuc te cua xau s2 do cuoi moi xau deu co dau enter

li $s2, 0 # chi so i cua xau s1

li $s3,0 # chi so j cua xau s2

count\_char\_s1: #dem so luong chu cai xuat hien trong xau 1

slt $s4, $s2, $s0 # kiem tra xem chi so hien tai(s2) co <= do dai xau s1 (s0) khong

beq $s4, $zero, end\_count\_char\_s1 # neu s4 = 0 ( s2>s0) thi goi ham end\_count\_char\_s1

lb $s5, s1($s2) # $s5 = s1[i]

sub $s5, $s5, 97 # $s5 = s1[i] - 'a'

add $s5, $s5, $s5

add $s5, $s5, $s5 # $s5 = 4\*(s1[i] -'a') vi trong mang no moi phan tu no luu 4 byte

lw $s6, count\_s1($s5) # $s6 = count\_s1[s1[i]-'a']

addi $s6, $s6, 1 # $s6++

sw $s6, count\_s1($s5) # count\_s1[s1[i]-'a'] = $s6d luu lai

addi $s2, $s2, 1 # tang chi so len 1

j count\_char\_s1 # tiep tuc vong lap

end\_count\_char\_s1: # thoat

count\_char\_s2:

slt $s4, $s3, $s1 # kiem tra xem chi so hien tai(s2) co <= do dai xau s1 (s0) khong

beq $s4, $zero, end\_count\_char\_s2 # neu s4 = 0 ( s2>s0) thi goi ham end\_count\_char\_s1

lb $s5, s2($s3) # $s5 = s2[j]

sub $s5, $s5, 97 # $s5 = s2[j] - 'a'

add $s5, $s5, $s5

add $s5, $s5, $s5 # $s5 = 4\*(s2[i] -'a') vi trong mang no moi phan tu no luu 4 byte

lw $s6, count\_s2($s5) # $s6 = count\_s1[s1[i]-'a']

addi $s6, $s6, 1 # $s6++

sw $s6, count\_s2($s5) # count\_s1[s1[i]-'a'] = $s6d

addi $s3, $s3, 1 # tang chi so len 1

j count\_char\_s2 # tiep tuc vong lap

end\_count\_char\_s2: # thoat

li $t0, 26 # so phan tu n cua mang count\_common\_char\_freq

li $t1, 0 # chi so i cua mang count\_common\_char\_freq

count\_common\_char: #ham dem so chu cai lap trong 2 xau

slt $t2, $t1, $t0 # i< n ?

beq $t2, $zero, end\_count\_common\_char # neu sai ket thuc vong lap

add $t3, $t1, $t1

add $t3, $t3, $t3 # $t3 = 4\*i

lw $t4, count\_s1($t3) # $t4 = count\_s1[i]

lw $t5, count\_s2($t3) # $t5 = count\_s2[i]

slt $t6, $t4, $t5 # count\_s1[i] < count\_s2[i] ?

bne $t6, $zero, add\_count\_s1 # neu t6 = 0 thi thuc hien add\_count\_s1

add\_count\_s2:

add $t7, $t7, $t5 # commonCharacterCount += count\_s2[i]

sw $t5, count\_common\_char\_freq($t3) #count\_common\_char\_freq[i] = count\_s2[i]

addi $t1, $t1, 1 # i++

j count\_common\_char # de quy

add\_count\_s1:

add $t7, $t7, $t4 # commonCharacterCount += count\_s1[i]

sw $t4, count\_common\_char\_freq($t3) # count\_common\_char\_freq[i] = count\_s1[i]

addi $t1, $t1, 1 # i++

j count\_common\_char # de quy

end\_count\_common\_char:

print: # in tong so cac chu cai trung trong 2 xau

li $v0, 4 # print string

la $a0, Message1 #"commonCharacterCount(s1, s2) = "

syscall

li $v0, 1 # print interger

li $a0, 0 # gan a0 = 0

add $a0, $a0, $t7 # a0 = a0 + t7

syscall

li $t1, 0

print\_character: #in so lan xuat hien cua tung ki tu chung o ca hai mang

slt $t2, $t1, $t0 # so sanh chi so voi do dai trong mang count\_common\_char\_freq

beq $t2, $zero, end\_print\_character # neu t2 = 0 thi thuc hien end\_print\_character

add $t3, $t1, $t1

add $t3, $t3, $t3 # $t3 = 4\*i

lw $t4, count\_common\_char\_freq($t3) #lay count\_common\_char\_freq[i] gan vao t4

beq $t4, $zero, add\_index #neu t4 = 0 thi thuc hien add\_index (tuc la ko co chu cai chung tai vi tri do)

li $v0,4 # print string

la $a0, Message2 #"\n"

syscall

li $v0,1 # print interger

li $a0, 0 # $a0 = 0

add $a0, $a0, $t4 #$a0 = $a0 + $t4 = $a0 + count\_common\_char\_freq[i]

syscall

li $v0, 4 # print string

la $a0, Message3 #" \" "

syscall

li $v0, 11 # print character

addi $a0, $t1,97 # in chu cai

syscall

li $v0, 4 # print string

la $a0, Message3 #" \" "

syscall

add\_index:

addi $t1, $t1, 1

j print\_character

end\_print\_character:

**Phân tích cách thực hiện**

* Tạo mảng A1 có 26 phần tử ban đầu đều là 0 để lưu số lần xuất hiện của các chữ cái từ a đến z có trong xâu s1 ( chữ a tương ứng với A[0] … z tương ứng A[25]), ví dụ xâu s1 là ‘aabc’ thì A=[2,1,1,0……0,0]
* Tương tự tạo mảng A2 đối với xâu s2
* Tạo mảng B cũng có 26 phần tử ban đầu là 0 , để lưu số lần xuất hiện của các kí tự chung từ a đến z ở cả 2 xâu , nếu ở cùng 1 vị trí thì phần tử ở bên mảng nào nhỏ hơn thì sẽ lưu vào B

Ví dụ : A1=[3,2,1,0….0,0]

A2=[2,1,0,0,….0,0]

* B = [2,1,0,0,..0,0]
* Kết quả in ra sẽ là

commonCharacterCount(s1, s2) = tổng các phần tử trong mảng B = 3

String have 3 common characters - 2 "a"s and 1 "b".

**Các thanh ghi được sử dụng:**

* $s0 : độ dài của xâu s1
* $s1 : độ dài thanh ghi s2
* Ngoài ra còn nhiều thanh ghi khác để lưu giá trị tạm thời

**Ý nghĩa các phần trong chương trình:**

* Khai báo count\_s1, count\_s2 lần lượt là mảng đếm số lần xuất hiện các chữ cái trong xâu s1 , s2
* count\_common\_char\_freq là mảng đếm số lần xuất hiện chung của các chữ cái trong cả 2 xâu
* string\_1, get\_length\_s1, end\_get\_length\_s1 dùng để lấy độ dài xâu s1 nhập vào . Tương tự string\_2, get\_length\_s2, end\_get\_length\_s2 dùng để lấy độ dài xâu s2 nhập vào
* count\_char\_s1 dùng để đếm số lượng chữ cái xuất hiện trong xâu s1 để lưu vào mảng count\_s1 . Tương tự count\_char\_s2 dùng để đếm số lượng chữ cái xuất hiện trong xâu s2 để lưu vào mảng count\_s2
* count\_common\_char để so sánh lần lượt các phần tử của 2 mảng count\_s1,count\_s2

1. nếu count\_s1[i] < count\_s2[i] thì gọi add\_count\_s1 để gán count\_common\_char\_freq [i] = count\_s1[i]
2. nếu count\_s1[i] >= count\_s2[i] thì gọi add\_count\_s2 để gán count\_common\_char\_freq [i] = count\_s2[i]

* print , print\_character , add\_index để in kết quả