BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH

Nhóm: 1

Thành viên: Đinh Quốc Cường – 20184054

Phạm Tiến Đạt – 20184064

1. **Bài miniproject 8:**

Write a program to:

- Input the number of students in class.

- Input the name of students in class, mark .

- Sort students due to their mark.

1. **Phân tích cách thức thực hiện**
2. **Phân tích đề bài:**

- Input: Số sinh viên và danh sách sinh viên.

- Output: Danh sách sinh viên được sắp xếp theo điểm (tăng dần).

1. **Ý tưởng:**

- Nhập số lượng sinh viên (số nguyên dương)

- Nhập lần lượt mã số sinh viên, tên và điểm sinh viên.

- Nhập lần lượt tên và điểm sinh viên với điều kiện: 0 <= điểm <= 10

- Sắp xếp danh sách sinh viên theo thứ tự tăng dần của điểm sử dụng thuật toán Bubblesort.

1. **Cách thức thực hiện:**

**- Bước 1**: Nhập n (nếu n không phải số nguyên dương thì nhập lại)

**- Bước 2:** Nhập tên và điểm sinh viên (nếu điểm nhập vào không thỏa mãn: 0<=

điểm <=10 thì nhập lại)

**- Bước 3:** Hiển thị thông tin sinh viên vừa nhập vào.

**- Bước 4:** Sắp xếp điểm sinh viên:

+ j = o //xử lý lần đầu tiên

+ i = n //duyệt từ cuối dãy trở về vị trí

Trong khi i > j thực hiện:

Nếu a[i] < a[i-1] thì hoán đổi a[i] và a[i-1]

i = i – 1

+ Nếu j = j + 1

Nếu j > n – 1 thì dừng

Ngược lại lặp lại bước 2.

**- Bước 5:** Hiển thị danh sách sinh viên đã được sắp xếp.

1. **Ý nghĩa của các thanh ghi được sử dụng**

- $s1: lưu địa chỉ của mảng student

- $t1: gán địa chỉ của mảng vào $t1

- $a0: số sinh viên

- $s0, $s6, $s7: số sinh viên(12)

- $a1: chỉ ra vị trí lưu tên

- $f1: đọc giá trị của $s4 (=10.0) vào $f1

- $t0: biến đếm của sinh viên được nhập thông tin

- $t1: chứa giá trị của mảng được gán

- $s3: lưu địa chỉ của students để thực hiện sắp xếp

- $t2: biến i

- $t3: biến j

- $t4: j - 1

1. **Ý nghĩa của chương trình con:**

- \_\_read\_info\_student: nạp địa chỉ của mảng.

- count: nhập số lượng sinh viên.

- name: nhập tên sinh viên với giới hạn là 47 kí tự.

- mark: nhập điểm sinh viên với điều kiện 0 <= điểm <=10.

- \_\_bubble\_sort: thực hiện sắp xếp.

- \_\_show\_student: Hiển thị danh sách sau khi sắp xếp.

1. **Source code**

 # ----------------------------------------------------------

 # Project 8: students

 #- Input the number of students in class .

 #- Input the name of students in class , mark .

 #- Sort students due to their mark .

 # -----------------------------------------------------------

 # -----------------------------------------------------------

 # @idea :

 # - Nhap so luong sinh vien .

 # - Nhap ma so sinh vien .

 # - Nhap ten sinh vien .

 # - Nhap diem sinh vien voi dieu kien 0 <= diem <= 10.

 # - Sap xep nhan vien theo diem dung thuat toan bubble sort .

 # ------------- -------------- -------------- -------------- ----

 .data

 students : .space 5200 # Array to store blocks (52) of 100 student

 input\_count : .asciiz " Nhap so luong sinh vien "

 error\_count : .asciiz "So sinh vien phai la so tu nhien nho hon hoac bang 100 "

 input\_mssv : .asciiz " Nhap ma so sinh vien "

 error\_mssv : .asciiz " 20100000 <= MSSV < 30190000\ nMSSV khong trung nhau ."

 trung : .asciiz " MSSV nay da ton tai "

 input\_name : .asciiz " Nhap ten sinh vien "

 error\_in : .asciiz "Ten ban da nhap qua dai hoac khong nhap gi! Toi da 47 ki tu"

 input\_mark : .asciiz " Nhap diem sinh vien "

 note : .asciiz " Diem duoc nhap phai thoa man : 0 <= diem <=10"

 error : .asciiz "Gia tri nhap vao khong dung kieu !"

 st\_list : .asciiz " Danh sach sinh vien "

 mark\_and\_name : .asciiz "\ nMSSV \t\ tDiem \ tHo va Ten \n"

 sorted\_list : .asciiz "\ nDanh sach da sap xep :\n"

 so\_sinh\_vien : .word

 ten : .float 10.0

 zero : .float 0.0

 number : .space 4

 .text

 # Doc thong tin sinh vien

 main :

 la $s1 , students # Nap dia chi cua mang $s1

 move $t1 , $s1 # Gan dia chi cua mang vao $t1

 la $s4 , ten #$s4 = 10.0

 la $s5 , zero #$s5 = 0.0

 li $t2 , 101 #$t2 = 101

 li $t4 , 20100001 # Cho t4 = 20100001

 li $t5 , 30190000 # Cho t5 = 30190000

 jal \_\_count # Ham nhap so luong SV nop

 la $s2 , number # s2 la dia chi bien number

 lw $s0 , 0( $s2 ) # Doc gia tri number cho vao s0

 move $s6 , $s0 # s6 , s7 = SSV

 move $s7 , $s0

 jal \_\_input\_info # Goi ham de nhap thong tin sinh vien nop

 li $v0 , 4 # In string

 la $a0 , st\_list # In " Danh sach sinh vien "

 syscall

 jal \_\_show\_student # Goi ham show student nop

 jal \_\_bubble\_sort # Sap xep student theo diem tang dan nop

 li $v0 , 4 # In string

 la $a0 , sorted\_list # In " Danh sach sinh vien da sap xep "

 syscall

 jal \_\_show\_student # In lai danh sach sau khi sap xep nop

 li $v0 , 10 # Exit chuong trinh

 syscall

 # nhap so sinh vien

 \_\_count :

 do\_count :

 li $v0 , 51 # Goi hop thoai nhap so luong sinh vien

 la $a0 , input\_count # $a0 tieu de: " Nhap so luong sinh vien "

 syscall # $a0 so sinh vien

 bnez $a1 , error\_input

 slt $t0 , $a0 , $zero # neu gia tri nhap vao < 0 => return 1

 bne $t0 , $zero , error\_input

 slt $v0 , $a0 , $t2 # Neu a0 (so sinh vien ) < t2 ( = 100 ) -->v0 = 1

 beqz $v0 , nhap\_lai # Neu v0 == 0 --> Nhap lai

 la $s7 , number # doc dia chi cua number

 sw $a0 , 0( $s7 ) # ghi SSV vao number

 jr $ra # Thoat khoi ham nhap so SV

 nhap\_lai :

 li $v0 , 55 # Goi hop thoai Thong bao

 la $a0 , error\_count

 li $a1 , 0 # Hop thoai Error

 syscall

 j do\_count # Quay lai ham Nhap so SV

 error\_input :

 la $v0 , 55

 la $a0 , error

 syscall

 j do\_count

 \_\_input\_info :

 li $t0 , 0 # khoi tao $t0 = 0 , $t0[U+FFFD] l bien dem (i) cua sinh vien vua duoc nhap thong tin

 addi $sp , $sp , -4 # Tao vung nho

 sw $ra , 0( $sp ) # luu dia chi thanh ghi $ra

 loop :

 slt $v0 , $t0 , $s0 # So sanh $t0 (So sinh vien nhap thong tin) < $s0 ( Tong so sinh vien ) -> v0 = 1

 beqz $v0 , end\_loop # Thoat vong lap khi nhap du thong tin cho cac sinh vien

  # Nhap mssv

 mssv\_loop :

 li $v0 , 51 # Goi hop thoai nhap diem ( Kieu Int )

 la $a0 , input\_mssv # Tieu de " Nhap MSSV "

 syscall # Tra ve gia tri vua nhap vao a0

 bnez $a1 , re\_input\_mssv

 slt $v0 , $a0 , $t4 # Neu a0 (so sinh vien ) < t4 ( = 20100001 )--> v0 = 1

 bnez $v0 , re\_input\_mssv # v0 != 0 --> Nhap lai

 slt $v0 , $a0 , $t5 # Neu a0 (so sinh vien ) < t5 ( = 30190000 )--> v0 = 1

 beqz $v0 , re\_input\_mssv # Neu v0 == 0 --> Nhap lai

 jal check\_mssv

 nop

 bnez $t9 , re\_mssv # Neu t9 != 0 --> Nhap lai

 j name

 re\_input\_mssv :

 li $v0 , 55 # Goi hop thoai Thong bao

 la $a0 , error\_mssv # Goi xau bao loi

 li $a1 , 0 # Hop thoai Error

 syscall

 j mssv\_loop # Quay lai ham Nhap so SV

 re\_mssv :

 li $v0 , 55 # Goi hop thoai Thong bao

 la $a0 , trung # Goi xau : " MSSV nay da ton tai "

 li $a1 , 0 # Hop thoai Error

 syscall

 j mssv\_loop # Quay lai ham Nhap so SV

 # Kiem tra MSSV co bi trung khong

 check\_mssv :

 addi $sp , $sp , -12 # Tao vung nho cho $t1 , $t2

 sw $ra , 8( $sp ) # Luu gia tri thanh ghi $ra

 sw $t2 , 4( $sp ) # Luu gia tri $t2 vao $sp

 sw $t1 , 0( $sp ) # Luu gia tri $t1 vao sp

 li $t1 , 0 # Cho $t1 = 0

 la $t2 , students # Gan dia chi mang

 li $t9 , 0 # Thanh ghi kiem tra

 loop\_mssv :

 slt $v0 , $t1 , $t0 # Neu t1 ( bien dem ) < t0 ( so SV da nhap) --> v0 = 1

 beqz $v0 , end\_check # v0 == 0 --> Thoat vong lap

 lw $t8 , 0( $t2 ) # Doc mssv SV thu t1

 beq $t8 , $a0 , mssv\_trung # Neu t8 == a0 ( MSSV trung nhau ) thi thoat

 addi $t1 , $t1 , 1 # Tang bien dem

 addi $t2 , $t2 , 52 # Tang vung nho

 j loop\_mssv

 mssv\_trung :

 li $t9 , 1 # t9 = 1 MSSV bi trung

 end\_check :

 lw $t1 , 0( $sp )

 lw $t2 , 4( $sp )

 lw $ra , 8( $sp ) # Tra lai gia tri cho cac thanh ghi $t1 , $t2 ,$ra

 addi $sp , $sp , 12

 jr $ra

 # Nhap ten sinh vien

 name :

 sw $a0 ,0( $t1 ) # Luu mssv

 do\_name :

 li $v0 , 54 # Goi hop thoai nhap ten sinh vien

 la $a0 , input\_name # Tieu de " Nhap ten sinh vien "

 la $a1 , 8( $t1 ) # Chi ra vi tr[U+FFFD] luu ten

 li $a2 , 44 # Gioi han do dai ten 43 ki tu

 syscall

 bnez $a1 , re\_input # Neu a1 != 0 --> Nhap ten qua dai ( a1 =0 la trang thai dung ) --> Nhap lai name

 j mark # Chay den ham nhap diem

 # Nhap lai ten

 re\_input :

 li $v0 , 55 # Goi hop thoai thong bao String

 la $a0 , error\_in # Goi string loi

 li $a1 , 0 # Goi hop thoai error

 syscall

 j do\_name # Quay lai Nhap name

 # Nhap diem

 mark :

 do: li $v0 , 52 # Goi hop thoai nhap diem ( Kieu float )

 la $a0 , input\_mark # Tieu de " Nhap diem sinh vien "

 syscall

 bnez $a1 , err\_input

 l.s $f1 ,( $s4 ) # Doc gia tri cua s4 (= 10.0) vao thanh ghi f1

 c.le.s $f0 , $f1 # f0 <= f1

 li $a0 , 1

 movt $a0 , $zero

 bne $a0 , $zero , condi # Neu a0 != 0 --> incre

 l.s $f1 ,( $s5 )

 c.le.s $f1 , $f0 # f1 <= f0

 li $a0 , 1

 movt $a0 , $zero

 beq $a0 , $zero , exit\_do # Neu a0 != 0 --> incre

 err\_input :

 la $v0 , 55

 la $a0 , error

 syscall

 j do

 condi :

 la $v0 , 55

 la $a0 , note

 syscall

 j do

 # Thoat nhap diem

 exit\_do :

s.s $f0 , 4( $t1 ) # Luu diem[U+FFFD] v omang

addi $t0 , $t0 , 1 # Tang bien dem len 1

addi $t1 , $t1 , 52 # Chuyen sang vung nho tiep theo , moi vung nho 52 bit

j loop # Lap lai vong lap

end\_loop :

lw $ra , 0( $sp ) # Lay gia tri thanh ghi $ra

addi $sp , $sp , 4

jr $ra

## ###############################

# Bat dau sap xep #

## ###############################

# -----------------------------------------------------------

# @case\_study : Sap xep bang phuong phap bubble sort

# @input : danh sach sinh vien - chua duoc sap xep theo diem

# @output : danh sach sinh vien - da duoc sap xep theo thu tu tang dan cua diem so

# ---------------------------------------------------------

\_\_bubble\_sort :

addi $sp , $sp , -4 # Tao 1 o nho

sw $ra , 0( $sp ) # cat gia tri thanh ghi ra

la $s3 , students # Doc dia chi cua mang student vao s3

addi $t2 , $t0 , 0 # So sinh vien la t2 la bien dem

loop1 : # for i = n -1 to 0 ( loop 1 )

addi $t2 , $t2 , -1 # i = i - 1 ( t2 la bien diem tu phai sai i)

add $s0 , $s3 , $zero # Cho dia chi cua mang s0 = s3

li $t3 , 0 #[U+FFFD] g nj = 0 ( t3 la bien dem 2 tu trai sang j)

beq $t2 , 0 , break\_1 # Neu t2 (i) == 0 --> Exit loop 1

loop2 : #for j = 0 to i - 1 ( loop 2 )

beq $t2 , $t3 , loop1 # Neu j == i --> loop1

l.s $f1 , 4( $s0 ) # Load diem students [j] = f1

addi $s0 , $s0 , 52 # Tang dia chi len 52 --> Chuyen sang student khac --> s0 = a[j +1] = f2

l.s $f2 , 4( $s0 ) # Load diem students [j +1] = f2

 c.lt.s $f2 , $f1 # Neu f1 < f2

 li $a0 , 1

 movt $a0 , $zero

 bne $a0 , $zero , incre # --> incre

 jal swap # Neu f2 < f1 --> swap

 nop

 incre :

addi $t3 , $t3 , 1 # j = j + 1

j loop2 # Quay ve Loop 2

break\_1 :

lw $ra , 0( $sp ) # Lay gia tri thanh ghi ra

addi $sp , $sp , 4 # Tra vung nho cho stack

jr $ra # Show Students

#a[j -1] vs a[j]

swap :

addi $sp , $sp , -4 # Tao 1 vung nho trong tron stack

sw $ra , 0( $sp ) # Cat gia tri thanh ghi $ra

li $s7 , 0 # s7 = 0

addi $t4 , $s0 , -52 # t4 = students [j] --> f1 --> Vung nho dau tien

loopx :

slti $v0 , $s7 , 13 # s7 < 13 --> v0 = 1

beqz $v0 , end\_loopx # v0 == 0 --> s7 >= 13 --> Thoat vong lap. 1 phan tu co 52 byte moi o 4 byte --> 13 byte

 lw $a1 , 0( $t4 ) # Load gia tri cua f1 ( a[j] ) vao a1 4 byte sau

 lw $a2 , 52( $t4 ) # Load gia tri cua f2 ( a[j +1] ) vao a2 4 byte lien ke

 sw $a1 , 52( $t4 ) # Ghi gia tri cua a1 vao a[j +1] 4 byte

sw $a2 , 0( $t4 ) # Ghi gia tri cua a2 vao a[j] 4 byte

addi $t4 , $t4 , 4 # Tang vung nho cua students [j] len 4

addi $s7 , $s7 , 1 # Tang bien dem len 1

j loopx # Chay lai loopx

end\_loopx :

lw $ra , 0( $sp ) # Lay gia tri thanh ghi $ra da cat

addi $sp , $sp , 4 # Tra lai vung nho stack

jr $ra # Quay ve incre

#in ra danh sach sinh vien da duoc sap xep

\_\_show\_student :

add $s0 , $s7 , $zero #so sinh vien = 0

la $t1 , students # Nap dia chi dau mang cac block luu thongtin sinh vien

la $a0 , mark\_and\_name # in ra chuoi " Diem Ho va Ten "

syscall

li $t0 , 0 # Khoi tao $t0 = 0 , $t0[U+FFFD] l bien diem so sinh vien da duoc duyet

loop3 :

slt $v0 , $t0 , $s6 # So sanh $t0 (So sv da duyet )[U+FFFD] v $s0 (Tong so sinh vien )

beqz $v0 , exit\_show # Thoat vong lap khi duyet het sinh vien

li $v0 , 1 # Hien thi console

lw $a0 , 0( $t1 ) # In mssv ra[U+FFFD] m nhinh console

syscall

li $v0 , 11 # In char

li $a0 , '\t'  # In dau tab

syscall

 li $v0 , 2

 l.s $f12 , 4( $t1 ) # In diem ra man hinh console

 syscall

li $v0 , 11 #in char

li $a0 , '\t' # In dau tab

syscall

li $v0 , 4 #in string

la $a0 , 8( $t1 ) # In ten sinh vien ra man hinh console

syscall

addi $t0 , $t0 , 1 # Tang bien dem so luong sv da duyet

addi $t1 , $t1 , 52 # tang dia chi t1 len 52

j loop3 # Lap lai vong lap

exit\_show :

jr $ra

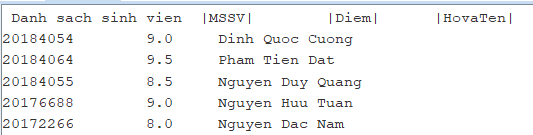
 # -----------------------------------------------------------

# END

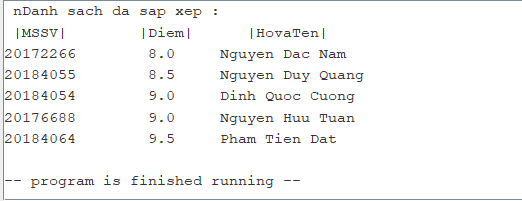
# -----------------------------------------------------------

1. **Kết quả**

**Ban đầu :**

****

**Lúc sau :**

****

1. **Bài miniproject 12:**

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 islucky or not.

Example:

For n = 1230, the output should be true

For n = 239017, the output should be false

1. **Báo cáo trình bày:**
2. **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.

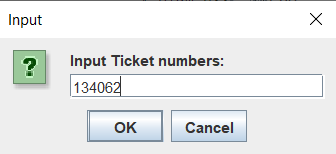
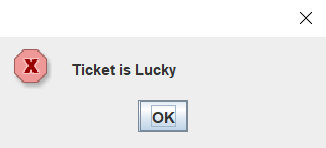
1. **Các thanh ghi được sử dụng:**

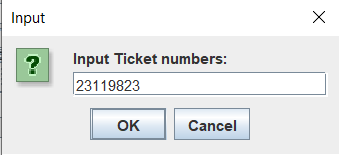
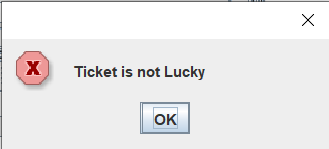
* $t0 = 10 => Thanh ghi $t0 lưu giá trị hằng số 10 để thực hiện phép tính n/10 lấy phần nguyên và phần dư.
* $s1 => Đếm số lượng các chữ số trong số nhập vào.
* $s4 => Tính tổng các chữ số có trong số nhập vào.
* $s5 => Tính tổng nửa các chữ số bên phải của số nhập vào.
* Ngoài ra có các thanh ghi khác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

1. **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. **Kết quả thực hiện:**

1. **Mã ngồn trương trình:**

.data

inputMessage: .asciiz "Input Ticket numbers: "

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

luckyMessage: .asciiz "Ticket is Lucky"

notluckyMessage: .asciiz "Ticket is not Lucky"

.text

# START: Khoi tao cac bien su dung trong chuong trinh

START:

li $t0, 10 # $t0 = 10 la hang so chia lay phan nguyen va phan du

li $s1, 0 # $s1 la bien dem so chu so cua so nhap vao.

li $s4, 0 # $s4 la bien luu tong cac chu so cua so nhap vao

li $s5, 0 # $s5 la bien luu nua tong cac chu so ben phai cua so nhap vao

# MAIN: Chuong trinh chinh

MAIN:

jal INPUT # Goi thu tuc INPUT lay gia tri nhap vao cua nguoi dung

nop

jal COUNT\_SUM # Goi thu tuc COUNT\_SUM dem so luong chu so va tinh tong cac chu so do.

nop

jal CHECK # Goi procedure CHECK de kiem tra so chu so la chan hay le

nop

div $s1, $s1, 2 # Giam so chu so xuong 1 nua

jal SUM\_HALF # Goi thu tuc SUM\_HALF de tinh tong nua cac chu so ben phai cua so nhap vao.

nop

add $s5, $s5, $s5 # Nhan doi tong nua cac chu so ben phai

beq $s5, $s4, LUCKY # Neu tong = 2 \* nua tong => Thong bao so may man

j NOT\_LUCKY # Thong bao so khong may man

# INPUT: Yeu cau nguoi dung nhap so can kiem tra

INPUT:

li $v0, 51 # Goi syscall InputDialogInt(51) Lay so nhap vao tu nguoi dung

la $a0, inputMessage # $v0: ma lenh, $a0: tin nhan, $a0: gia tri ma nguoi dung nhap vao, $a1: trang thai nhap vao

syscall

bne $a1, $zero, ERROR # Truong hop gia tri nhap vao khong phai la so thi thong bao loi

add $s0, $a0, $zero # Luu gia tri vua nhap vao $s0

add $t1, $s0, $zero # Luu gia tri vua nhap vao $s1 => Dung cho thu tuc COUNT\_SUM

add $t2, $s0, $zero # Luu gia tri vua nhap vao $s2 => Dung cho thu tuc SUM\_HALF

jr $ra # Quay tro lai ham MAIN

# COUNT\_SUM: Dem so luong cac chu so va tinh tong cua chung.

COUNT\_SUM:

div $t1, $t0 # Thuc hien chia so vua nhap ($t1) cho 10

mflo $t1 # Gan lai $t1 = phan nguyen cua phep chia

mfhi $t3 # Gan $t3 = phan du cua phep chia => Day la cac chu so cua so nhap vao.

add $s4, $s4, $t3 # Tinh tong cac chu so

add $s1, $s1, 1 # Tang so luong chu so them 1

bne $t1, $zero, COUNT\_SUM # Neu so nhap vao != 0 (chua chia het) thi tiep tuc tinh toan den khi lay het cac chu so

jr $ra # Quay tro lai ham MAIN

# SUM\_HALF: Tinh tong nua cac chu so ben phai

SUM\_HALF:

div $t2, $t0 # Thuc hien chia so vua nhap ($t2) cho 10

mflo $t2 # Gan lai $t2 = phan nguyen cua phep chia

mfhi $t3 # Gan $t3 = phan du cua phep chia => Day la cac chu so cua so nhap vao.

add $s5, $s5, $t3 # Tinh tong cac chu so

sub $s1, $s1, 1 # Giam so luong chu so can phai tinh tong xuong 1

bne $s1, $zero, SUM\_HALF # Neu so luong chu so can tinh != 0 thi tiep tuc tinh toan

jr $ra # Quay tro lai ham MAIN

# CHECK: Kiem tra so luong chu so la chan hay le, neu le thi bao loi

CHECK:

andi $t4 , $s1 , 1 # Kiem tra so chu so cua so nhap vao la chan hay le? $t4 = chan ? 0 : 1

bne $t4, $zero, ERROR # Neu so luong chu so le thi thong bao loi

jr $ra # Quay tro lai ham MAIN

# ERROR: Hien thi loi neu du lieu nhap vao khong phai la so hoac so chu so la so le

ERROR:

li $v0, 55 # Goi syscall MessageDialog(55) => Hien thi loi

la $a0, errorMessage # $v0: ma lenh, $a0: tin nhan

syscall

j EXIT # Thoat chuong trinh

# LUCKY: Hien thi thong tin so nhap vao la so may man

LUCKY:

li $v0, 55 # Goi syscall MessageDialog(55) => Hien thi tin nhan

la $a0, luckyMessage # $v0: ma lenh, $a0: tin nhan

syscall

j EXIT # Thoat chuong trinh

# NOT\_LUCKY: Hien thi thong tin so nhap vao khong la so may man

NOT\_LUCKY:

li $v0, 55 # Goi syscall MessageDialog(55) => Hien thi tin nhan

la $a0, notluckyMessage # $v0: ma lenh, $a0: tin nhan

syscall

j EXIT # Thoat chuong trinh

EXIT: