

Thực hành Kiến trúc máy tính bài thi Giữa kỳ

Họ tên: Đỗ Hoàng Minh Hiếu

MSSV: 20225837

Bài A.10:

1. Đề bài: Nhập số nguyên dương N từ bàn phím, in ra tổng các chữ số trong biểu diễn nhị phân của N.

2. Thực hiện

- Nhập chữ số N, nếu $N < 0$ thì phải nhập lại
- Sau đó ta dùng vòng lặp chia N với 2 đến khi nào $N = 0$ thì dừng.
- Trong vòng lặp, mỗi lần chia xong thì ta đều lấy phần dư của phép chia cộng vào với tổng sum (do phần dư chính là bit).

3. Code

.data

message: .asciiz "Nhap N: "

message1: .asciiz "Tong cac chu so nhi phan cua N la: "

.text

input: # Nhap N

li \$v0, 51

la \$a0, message

syscall

bltz \$v0, input # n<0 thì nhập lại

move \$t0, \$a0 # lưu vào \$t0

li \$s0, 2

li \$s1, 0 # sum=0

loop:

beqz \$t0, done # if $N == 0$ thì stop

div \$t0, \$s0

mfi \$t1 # Lấy phần dư thành ghi hi cat vào \$t1

add \$s1, \$s1, \$t1 # Cộng phần bit (dư) vào sum

mflo \$t0 # chuyển thương thành số bị chia và tiếp tục

j loop

done:

```
#print
li $v0, 4
la $a0, message1
syscall
```

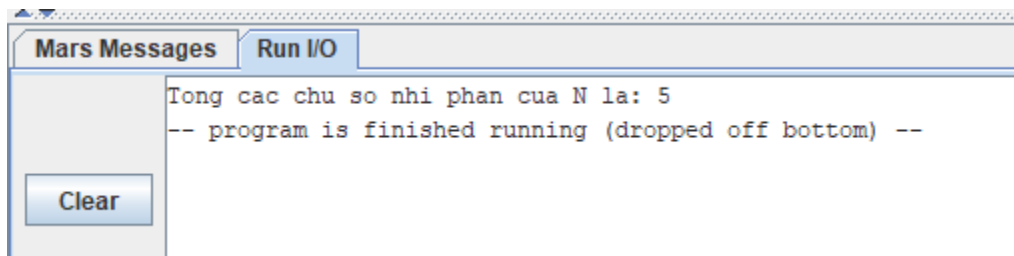
```
move $a0, $s1
li $v0, 1
syscall
```

4. Kết quả chạy chương trình

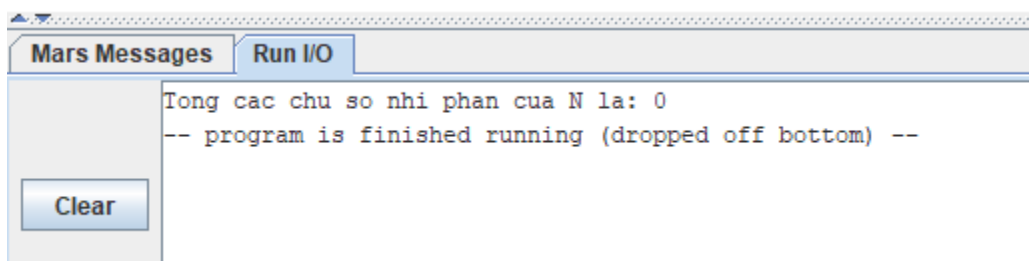
4.1. Với $N = 6445 \rightarrow$ Binary: 1100100101101 \rightarrow sum = 7



4.2. Với $N = 94 \rightarrow$ Binary=1011110 \rightarrow sum = 5



4.3. Với $N=0$



Bài B.1:

1. Đề bài: Nhập mảng số nguyên từ bàn phím. In ra số phần tử của mảng nằm trong đoạn $[M, N]$ với M và N là 2 số nguyên nhập từ bàn phím.

2. Thực hiện

- Nhập số phần tử (không quá 100 phần tử), sau đó dùng hàm loop để nhập từng phần tử một vào mảng. Sau đó nhập M và N .
- So sánh từng phần tử một so sánh xem có nhỏ hơn M và lớn hơn N hay không, nếu có không cộng và lặp lại hàm while đến khi nào hết phần tử thì dừng.
- Hàm compare dùng để so sánh phần tử có nằm trong đoạn $[M, N]$ không.

3. Code

```
.data
arr: .space 100
message: .asciiz "Nhap so phan tu cua mang: "
message1: .asciiz "Nhap M: "
message2: .asciiz "Nhap N: "
message3: .asciiz "So phan tu trong doan [M,N] la: "
.text
    li $v0, 4
    la $a0, message
    syscall # In ra message

    li $v0, 5 # Nhap n
    syscall

    move $s0, $v0 # $s0 = n

    la $a0, arr # arr[0] = $a0
    li $t0, 0 # i = 0

loop: # Nhap vao cac phan tu cua mang
    li $v0, 5
    syscall # Nhap phan tu vao $v0
    sw $v0, 0($a0)
    addi $t0, $t0, 1
    addi $a0, $a0, 4
    blt $t0, $s0, loop
end_loop:
# Input M, N
    li $v0, 4
    la $a0, message1 # In message1
    syscall

    li $v0, 5 # Nhap M
    syscall
    move $t1, $v0 # $t1 = M
```

```
li $v0, 4
la $a0, message2
syscall # In message2
```

```
li $v0, 5 # Nhap N
syscall
move $t2, $v0 # $t2 = N
```

```
la $a0, arr # arr[0] = $a0
li $t3, 0 # index = 0
li $s2, 0 # ketqua = 0
sll $s0, $s0, 2 # chuyen so phan tu thanh offset de so sanh
```

while:

```
bge $t3, $s0, print # chuyen ve nhan print de in ket qua khi het phan tu
add $t4, $a0, $t3 # lay dia chi phan tu arr[index]
lw $s1, 0($t4) # load phan tu arr[index]
jal compare
addi $t3, $t3, 4 # chuyen sang phan tu tiep theo
j while
```

end_while:

compare:

```
blt $s1, $t1, end_compare # neu arr[index] < M chuyen sang end_compare
bgt $s1, $t2, end_compare # neu arr[index] > N chuyen sang end_compare
addi $s2, $s2, 1
```

end_compare:

```
jr $ra
```

print:

```
li $v0, 4
la $a0, message3
syscall # In ra message
move $a0, $s2
li $v0, 1
syscall
```

end:

4. Kết quả thực thi

Với $n = 5$ và mảng 1 9 6 4 7, đoạn [3;9] thì kết quả là

```
1
9
6
4
7
Clear
Nhap M: 3
Nhap N: 9
Sophan tu trong doan [M,N] la: 4
```

Với $n = 12$, mảng -23 -4 0 -5 6 5 4 432 5 63 5 47 và đoạn [-12; 50]:

```
Nhap so phan tu cua mang: 12
-23
-4
0
-5
6
5
4
432
5
63
5
47
Nhap M: -12
Nhap N: 50
So phan tu trong doan [M,N] la: 9
-- program is finished running (dropped off bottom) --
```

Bài C.7:

1. Đề bài: Nhập vào xâu ký tự. Đổi các ký tự chữ hoa thành chữ thường và chữ thường thành chữ hoa. Các ký tự khác giữ nguyên.

2. Thực hiện

- Nhập vào xâu với độ dài tối đa là 100.

- Load từng byte ký tự một, kiểm tra xem có thuộc khoảng (65-90) hay không, nếu có thì chuyển sang chữ thường và lặp. Nếu không thuộc khoảng (65-90) thì kiểm tra xem thuộc khoảng (97-122) không, nếu có thì chuyển về chữ hoa và lặp lại đến khi gặp ký tự enter thì dừng.

3. Code

.data

string: .space 100 # Do dai 100

prompt: .ascii "Nhap mot xau: "

result: .ascii "Sau khi hieu chinh: "

.text

li \$s1, 10 # 10 la dau enter trong ascii

In ra man hinh promp

li \$v0, 4

```
la $a0, prompt
syscall
```

```
# Doc xau
li $v0, 8
la $a0, string
li $a1, 100
syscall
```

```
li $t1, 65 # 65 la 'A'
li $t2, 90 # 90 la 'Z'
li $t3, 97 # 97 la 'a'
li $t4, 122 # 122 la 'z'
la $a0, string
```

loop:

```
lb $t0, 0($a0) # Load byte
beq $t0, $s1, end_loop # Ket thuc khi gap enter
# Kiem tra xem co phai chu hoa khong
blt $t0, $t1, not_uppercase
bgt $t0, $t2, not_uppercase
# Chuyen sang chu thuong
addi $t0, $t0, 32
# Cat lai vao string
sb $t0, 0($a0)
j continue
```

not_uppercase:

```
# Kiem tra xem co phai chu thuong khong
blt $t0, $t3, continue
bgt $t0, $t4, continue
```

```
# Chuyen tu thuong sang hoa
addi $t0, $t0, -32
```

```
# Cat lai vao string
sb $t0, 0($a0)
```

continue:

```
addi $a0, $a0, 1 # Chuyen sang byte tiep theo
```

```
j loop          # Lap
```

end_loop:

```
# In ra ket qua
```

```
li $v0, 4
```

```
la $a0, result
```

```
syscall
```

```
li $v0, 4
```

```
la $a0, string
```

```
syscall
```

4. Kết quả

