

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

Bộ môn Kiến trúc máy tính



BÁO CÁO GIỮA KÌ

Giáo viên hướng dẫn: ThS. Lê Bá Vui

Sinh viên thực hiện: Nguyễn Thị Hải Thanh 20176871

Đặng Linh Chi 20176700

Hà Nội, ngày 17 tháng 5 năm 2020

Mục Lục

Nội dung	Trang
Bài 9	3
Đề bài.....	3
Thuật toán	3
Mã nguồn.....	3
Kết quả hiển thị.....	7
Bài 17	8
Đề bài.....	8
Thuật toán.....	8
Mã nguồn.....	8
Kết quả hiển thị.....	14

Bài 9

Sinh viên thực hiện: Đặng Linh Chi

Đề bài: Write a program to:

1. Read in the number of students in class
2. Read information about each student, including: Name, Math mark
3. List the name of all students who have not passed Math exam

Thuật toán:

A, Nhập vào số sinh viên

B, Tạo vùng nhớ trong stack (\$sp) để lưu trữ thông tin của từng sinh viên (tên và điểm môn toán)

Nhập vào tên và điểm của sinh viên, sử dụng vòng lặp để lưu trữ tên và điểm vào stack

Kiểm tra điều kiện nhập điểm của sinh viên, nếu $0 \leq \text{điểm} \leq 10$ thì đúng, nếu không thì nhập lại

- \$Sp[2i] lưu trữ điểm của sinh viên
- \$Sp[2i+1] lưu trữ tên của sinh viên

C, In ra tên của những sinh viên

Sử dụng vòng lặp, duyệt điểm của từng sinh viên. Nếu có điểm toán nhỏ hơn 5 thì in tên ra màn hình, nếu không thì duyệt tiếp.

Mã nguồn:

[Ý nghĩa của các thanh ghi và các hàm]

```
#@Input : $t0 : Lưu giá trị số học sinh ( Nhập vào từ bàn phím)
#Stack sp : Dùng để lưu tên và điểm của sinh viên. sp[2i] lưu điểm, sp[2i+1] lưu tên sinh viên
#$t2: Biến chạy (i=0)
#$a0 lưu địa chỉ Name ( đầu vào khi nhập tên sinh viên)
#@Input_mark: hàm nhập điểm của sinh viên
#@Input_Name: hàm nhập tên sinh viên
#@Loop1, Loop2: Kiểm tra điểm của sinh viên (lớn hơn hoặc bằng 0 và nhỏ hơn bằng 10)
#@Check_Input: Kiểm tra xem nhập đủ số lượng sinh viên chưa.
#@check_notpass: Kiểm tra xem sinh viên nào không qua môn => In kết quả ra màn hình
```

```

=====
.data
Message: .asciiz "Nhap lai diem: "
Diem: .asciiz "Nhap diem: "
Message1: .asciiz "Ten cua nhung hoc sinh not pass toan: \n "
Name: .space 20000          #Nhap ten sinh vien
array: .space 20000
Sohocsinh: .asciiz "Nhap so hoc sinh: "
.text
Main:
    li $v0,51
    la $a0,Sohocsinh      #$a0 luu giá trị so sinh viên
    syscall
# Kiểm tra điều kiện nhập đúng định dạng là số chưa
    beq      $a1, -1,Main      # $a1 status value (if input is invalid => Re-input)
    beq      $a1, -3,Main      # $a1 status value (if input is invalid => Re-input)
    beq      $a1, -2, End
    add $t0,$a0,$zero          # $t0 Lưu giá trị số sinh viên (n)
    #tao vùng nhớ lưu điểm và tên cho sinh viên
    sll $t1,$t0,3              #$t1= 8*$t0
    sub $sp,$sp,$t1            # tao bo nho cho Sp
    addi $t2,$zero,0           # t2=i=0
    la $s0,Name                # Load địa chỉ của Name vào $s0
Input_Mark:
    sll $t3,$t2,3              #t2=t3*8=8i (Tạo vùng nhớ để lưu điểm)
    add $a2,$sp,$t3            #a2 tro toi sp[2i]
    li $v0, 51
    la $a0, Diem
    syscall
    beq $a1, -1,Input_Mark      # $a1 status value (if input is invalid => Re-
input)
    beq $a1, -3,Input Mark      # $a1 status value (if input is invalid => Re-
input)

```

```

        beq $a1, -2, End
        sw $a0,0($a2)                                #lưu điểm vào $sp[2i]

# Xet dieu kien diem nam trong khoang 0-10
addi $s5,$zero,10
Loop1:   lw $s6,0($a2)                                #Lấy giá trị điểm vừa nhập lưu vào $s6
        slt $s4,$s6,$0                                #If điểm nhỏ hơn 0 => $s4=1
        bne $s4,$zero,Loop2                            # Nếu điểm nhập vào nhỏ hơn 0=> Loop2
        slt $s4,$s5,$s6                                #So sanh voi 10
        bne $s4,$zero,Loop2                            # neu lon hon 10 => Loop2
j Input_Name
nop
Loop2:
        # Nhập lại điểm nếu nhập sai
        li $v0, 51
        la $a0, Message
        syscall

#Kiem tra xem nhap đúng định dạng integer chưa
        beq      $a1, -1,Loop2                        # $a1 status value (if input is invalid => Re-input)
        beq      $a1, -3,Loop2                        # $a1 status value (if input is invalid => Re-input)
        beq      $a1, -2, End
        sw $a0,0($a2)
j Loop1
nop
# Nhập tên sinh viên
Input_Name:
        addi $t3,$t3,4                                #$t3=8i+4
        add $a2,$sp,$t3                                #a2 tro toi sp[2i+1]
        move  $a0,$s0                                  # $a0 lấy nội dung của thanh ghi $s0
        li    $a1,16
        li    $v0,8
        syscall

```

```

        sw      $a0,0($a2)          # lưu tên vừa nhập vào mảng sp[2i+1]
        addi    $s0,$s0,20

j Check_Input
nop
# kiểm tra xem nhập đủ sinh viên hay chưa
Check_Input:
        addi    $t2,$t2,1           # i=i+1
        beq     $t2,$t0,End_Input    # nếu i=n kết thúc việc nhập điểm và tên
j Input_Mark          # chưa nhập đủ=>nhập tiếp
nop
# Kết thúc việc nhập
End_Input:
        addi    $t2,$zero,0 # t2=i=0
        addi    $t5,$zero,4        # $t5=4 (lưu điểm để so sánh)
        li      $v0, 4
        la      $a0, Message1
        syscall

check_notpass:
        sll     $t3,$t2,3           # t3= 8*i
        add     $t4,$sp,$t3         # t4 trỏ đến địa chỉ của sp[2i]
        lw      $s0,0($t4)          # s0= sp [2i]
        slt     $s3,$t5,$s0         # so sánh điểm của sinh viên với 5
        bne     $s3,$zero,Check     # Nếu nhỏ hơn 5 => In tên sinh viên
        addi    $t3,$t3,4           # t3=8*i+4
        add     $t4,$sp,$t3         # t4 trỏ đến địa chỉ của sp[2i+1]
#In tên sinh viên not pass môn toán
        lw      $s2,0($t4)          #s2= sp[2i+1]
        li      $v0, 4
        la      $a0, 0($s2)
        syscall

```

```

# Kiểm tra xem đã kiểm tra hết số sinh viên chưa
Check:
    addi $t2,$t2,1
    beq $t2,$t0,End
    j check_notpass
    nop
# Thoát chương trình
End:
    li $v0, 10
    syscall

```

Kết quả:

Giá trị đầu vào:

Tên	Điểm
Chi	4
Thanh	8
Tu	9
Vy	2
Quang	10
Linh	1
Yen	5

Kết quả hiển thị:

```

**** user input : Chi
**** user input : Thanh
**** user input : Tu
**** user input : Vy
**** user input : Quang
**** user input : Linh
**** user input : Yen
Ten cua nhung hoc sinh not pass toan:
Chi
Vy
Linh

-- program is finished running --

```

Bài 17: Giải mã chuỗi kí tự ASCII

Sinh viên thực hiện: Nguyễn Thị Hải Thanh - 20176871

Đề bài: You are taking part in an Escape Room challenge designed specifically for programmers. In your efforts to find a clue, you have found a binary code written on the wall behind a vase and realized that it must be an encrypted message. After some thought, your first guess is that each consecutive 8 bits of the code stand for the character with the corresponding ASCII code.

Assuming that your hunch is correct, decode the message.

Example:

For code = "010010000110010101101100011011000110111100100001",

the output should be `messageFromBinaryCode(code)` = "Hello!".

Thuật toán:

- Cho người dùng nhập một chuỗi string cần giải mã
- Duyệt chuỗi với hai mục đích:
 - + Đếm số lượng kí tự trong chuỗi, kiểm tra độ dài chuỗi có phải là số chia hết cho 8 hay không. Nếu độ dài chuỗi không thỏa mãn yêu cầu chia hết cho 8 thì báo lỗi, yêu cầu người dùng nhập lại
 - + Kiểm tra chuỗi xem trong chuỗi có tồn tại kí tự nào khác ngoài 2 kí tự '0' và '1' hay không. Nếu tồn tại kí tự không đúng yêu cầu thì báo lỗi và yêu cầu người dùng nhập lại
- Sau khi kiểm tra chuỗi xong, thực hiện tính toán giá trị và giải mã từng bộ 8 ký t
 - + Cứ mỗi 8 kí tự, đưa vào làm một nhóm, chuyển đổi giá trị sang giá trị decimal và lưu giá trị đó vào một mảng string mới bằng cách sử dụng lệnh `split` để dịch bit và `add` để tính tổng
 - + Sử dụng vòng lặp tính toán cho đến hết chuỗi
- Bước cuối cùng in ra chuỗi đã được mã hóa

Mã nguồn:

```
.data
string:      .space 1000
output:      .space 1000
Message:     .ascii "Nhap day so can giai ma: "
Message2:    .ascii "Ket qua giai ma la: "

InvalidCharInputMessage: .ascii "Day so Nhap vao phai la day nhi phan!\n"
InvalidLengthMessage:    .ascii "Day so nhap vao phai co do dai chia het cho 8!\n"
InvalidValueMessage:     .ascii "Hay xem lai chuoai so. Dam bao chuoai so co cac ki tu hien thi duoc\n"
```



```

.text
#-----
# @get_string: nhan chuoi nhap vao tu ban phim
# @get_length: luu string vao thanh ghi $s0, duyet chuoi, validate dau vao
#           $s0 luu dia chi co so cua chuoi input
#           #t7 la bien chay cua ky tu, sau get_length la do dai string (n)
#           $t1 dia chi cua ky tu
#           $t2 luu gia tri cua ky tu, luu so du trong ham check_valid_length
# @decode: vong lap doi gia tri moi 8 bit thanh he dem 10
#           $t0 bien chay cua ki tu trong chuoi input (i)
#           $s1 luu gia tri cua ky tu hien tai
#           $t5 luu gia tri bit tuong ung cua ki tu hien tai
#           $s3 luu gia tri decode cua tung nhom 8 ky tu (sum)
#           $t3 luu bien chay cua output (j)
#           $s2 luu dia chi co so cua output
# @add_to_output(ham con cua decode):
#           bien nhom 8 ky tu tro thanh mot ki tu roi dua vao output
#           $t4 luu dia chi hien tai cua output
# @print_output    int ra chuoi da duoc giai ma
#-----

main:
# hien thi hop thoai yeu cau nguoi dung nhap vao string
get_string:
    li $v0, 54
    la $a0, Message
    la $a1, string                #dua string vao $a1
    la $a2, 1000
    syscall
    beq $a1, -2, endmain
end_of_get_str:
# Duyet chuoi
get_length:  la $s0, string                # $s0 = address(string[0])

```

```

        add $t7,$zero,$zero           # $t7 = n = 0
check_char:  add $t1,$s0,$t7           # $t1 = $a0 + $t7 = address (string[n])
            lb $t2, 0($t1)            # $t2 = string[n]
            beq $t2, $zero, end_of_str # $t2==null? j end_of_string | kiem tra ki tu
            addi $t7,$t7,1             # $t0 += 1 =. n += 1
            beq $t2, 48, check_char    # $t2=='1'? j check_char | kiem tra '0' va
'\n'
            beq $t2, 49, check_char    # $t2=='0'? j check_char | kiem tra '\n'
            beq $t2, 10, check_char    # $t2=='\n'? j check_char | InvalidCharInput
invalid_char_input_notice:
            li $v0, 4
            la $a0, InvalidCharInputMessage
            syscall
            j get_string
end_of_str:
            addi $t7, $t7, -1          # $t7 tru di 1 de lay do dai chuoai khong tinh '\n'
end_of_get_length:

# kiem tra do dai chuoai phai chia het cho 8
check_valid_length:
            addi $t3, $zero, 8         # $t3 = 8
            div $t7, $t3               # thuc hien phep chia
            mfhi $t2                   # so du duoc luu vao $t2
            beq $t2, 0, decode         # $t2==0? decode | InvalidLengthNotice
invalid_length_notice:
            li $v0, 4
            la $a0, InvalidLengthMessage
            syscall
            j get_string

# thuc hien giai ma bang cach: Nhom tung nhóm gom 8 ky tu, thuc hien xu ly doi sang he 10
sau
# do dua vao chuoai output
decode:

```

```

#Khoi tao:
addi $s3, $zero, 0
addi $t0, $zero, -1          # $t0 la bien chay cua input, $t0 = -1
addi $t3, $zero, -1          # $t3 = 0, la bien chay cua ouput
la $s2, output                # output duoc luu vao t hanh ghi $s2
add $t5, $zero, $zero        # $t5 luu gia tri bit tuong ung cua ky tu

continue:
    addi $t0, $t0, 1          # $t0+=1 ; i++
    beq $t0, $t7, end         # $t0==n? j end | j func (tinh bit dau)

func1:
    add $t1, $s0, $t0         # $t1=address(string[i])
    lb $s1, 0($t1)            # $s1=string[i]
    beq $s1, 48, func2        # $s1=='0'? j func2 | gan $t5=1, tinh toan sum
    addi $t5, $zero, 1        # gan $t5=1
    sll $t5, $t5, 7           # $t5 dich trai 7 bit
    add $s3, $s3, $t5         # sum += $t5
    addi $t5, $zero, 0        # set lai gia tri $t5 = 0

func2:
    addi $t0, $t0, 1          # $t0+=1; i++
    add $t1, $s0, $t0         # $t1=address(string[i])
    lb $s1, 0($t1)            # $s1=string[i]
    beq $s1, 48, func3        # $s1=='0'? j func3 | gan $t5=1, tinh toan sum
    addi $t5, $zero, 1        # $t5=1
    sll $t5, $t5, 6           # $t5 dich trai 6 bit
    add $s3, $s3, $t5         # sum += $t5
    addi $t5, $zero, 0        # set lai gia tri $t5=0

func3:
    addi $t0, $t0, 1          # $t0+=1; i++
    add $t1, $s0, $t0         # $t1=address(string[i])
    lb $s1, 0($t1)            # $s1=string[i]
    beq $s1, 48, func4        # $s1=='0'? j func4 | gan $t5=1, tinh toan sum
    addi $t5, $zero, 1        # $t5=1

```

```

sll $t5, $t5, 5          # $t5 dich trai 5 bit
add $s3, $s3, $t5        # sum += $t5
addi $t5, $zero, 0       # set lai gia tri $t5=0
func4:
addi $t0, $t0, 1         # $t0+=1; i++
add $t1, $s0, $t0        # $t1=address(string[i])
lb $s1, 0($t1)           # $s1=string[i]
beq $s1, 48, func5       # $s1=='0'? j func5 | gan $t5=1, tinh toan sum
addi $t5, $zero, 1       # $t5=1
sll $t5, $t5, 4          # $t5 dich trai 4 bit
add $s3, $s3, $t5        # sum += $t5
addi $t5, $zero, 0       # set lai gia tri $t5
func5:
addi $t0, $t0, 1         # $t0+=1; i++
add $t1, $s0, $t0        # $t1=address(string[i])
lb $s1, 0($t1)           # $s1=string[i]
beq $s1, 48, func6       # $s1=='0'? j func6 | gan $t5=1, tinh toan sum
addi $t5, $zero, 1       # $t5=1
sll $t5, $t5, 3          # $t5 dich trai 3 bit
add $s3, $s3, $t5        # sum += $t5
addi $t5, $zero, 0       # set lai gia tri $t5
func6:
addi $t0, $t0, 1         # $t0+=1; i++
add $t1, $s0, $t0        # $t1=address(string[i])
lb $s1, 0($t1)           # $s1=string[i]
beq $s1, 48, func7       # $s1=='0'? j func7 | gan $t5=1, tinh toan sum
addi $t5, $zero, 1       # $t5=1
sll $t5, $t5, 2          # $t5 dich trai 2 bit
add $s3, $s3, $t5        # sum += $t5
addi $t5, $zero, 0       # set lai gia tri $t5
func7:
addi $t0, $t0, 1         # $t0+=1; i++

```

```

    add $t1, $s0, $t0          # $t1=address(string[i])
    lb $s1, 0($t1)             # $s1=string[i]
    beq $s1, 48, func8         # $s1=='0'? j func8 | gan $t5=1, tinh toan sum
    addi $t5, $zero, 1         # $t5=1
    sll $t5, $t5, 1            # $t5 dich trai 1 bit
    add $s3, $s3, $t5          # sum += $t5
    addi $t5, $zero, 0         # set lai gia tri $t5
func8:
    addi $t0, $t0, 1           # $t0 += 1; i++
    add $t1, $s0, $t0          # $t1=address(string[i])
    lb $s1, 0($t1)             # $s1=string[i]
    beq $s1, 48, check_output  # $s1=='0'? j check_output | gan $t5=1, tinh toan sum
    addi $t5, $zero, 1         # $t5=1
    add $s3, $s3, $t5          # sum += $t5
    addi $t5, $zero, 0         # set lai gia tri $t5
# vi cac ky tu co the hien thi duoc trong bang ascii co gia tri tu 32 den
# 126 nen can check xem sum $s3 co nam trong khoang nay khong
check_output:
    slti $t2, $s3, 32          # if $s3<32 => $t2=1
    bne $t2, 0, invalid_value_notice # $t2!=0 ($s3<32)? InvalidValueMessage|j
check_output_2
check_output_2:
    slti $t2, $s3, 126         # if $s3<126 => $t2=1
    bne $t2, 0, add_to_output   # $t2!=0? ($s3<126) add_to_output|InvalidValueMessage
invalid_value_notice:
    li $v0, 4
    la $a0, InvalidValueMessage
    syscall
    j get_string
add_to_output:
    addi $t3, $t3, 1           # j++
    add $t4, $s2, $t3          # $t4 = address(output[j])
    sb $s3, 0($t4)             # output[j]=$s3

```

```

        add $s3, $zero, $zero          # set lai sum = 0
        j continue
end:
print_output:                          #in ra chuoai ki tu duoc giai ma
        li $v0, 4
        la $a0, output
        syscall
endmain:

```

Kết quả:

Input	Output
010010000110000101101001001000000101010001101000011000010110111001101000	Hai Thanh
0100110001101001011011100110100000100000010000110110100001101001	Linh Chi
010010000110000101101001001000000101010001101000011000010110111001101000001 000000111011001100001001000000100110001101001011011100110100000100000010000 110110100001101001001000000110001101101000011000010110111100100000011101000 1101000011000010111100100100001	Hai Thanh va Linh Chi chao thay!
00101000010111100010000000110011001000000101111000101001	(^ 3 ^)

Màn hình Kết quả với đầu vào input:

0100110001101001011011100110100000100000010000110110100001101001

```

Reset: reset completed.
Hai Thanh va Linh Chi chao thay!
-- program is finished running (dropped off bottom) --

```