

ĐẠI HỌC BÁCH KHOA HÀ NỘI



BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH

GIẢNG VIÊN HƯỚNG DẪN : LÊ BÁ VUI

SINH VIÊN : HOÀNG ĐỨC ANH MSSV: 20176688

VŨ MINH HOÀNG ANH MSSV: 20176689



Mục Lục

Nội dung	Trang
Bài 4	2
Đề bài.....	2
Thuật toán	2
Mã nguồn.....	3
Kết quả hiển thị.....	6
Bài 13	7
Đề bài.....	7
Thuật toán.....	7
Mã nguồn.....	8
Kết quả hiển thị.....	11

Bài 4: Tìm phần tử lớn nhất của mảng và in ra số phần tử trong mảng đó

Sinh viên thực hiện: Vũ Minh Hoàng Anh - 20176689

Đề bài : Tạo chương trình

- + Nhập vào mảng số nguyên từ bàn phím
- + Tìm số nguyên lớn nhất của mảng
- + Tính số phần tử trong phạm vi (m, M). Phạm vi m, M là tích hợp bàn phím nhập từ bàn phím.

Thuật Toán :

Tìm max :

B1: Gán max là 0

B2: Mỗi khi nhập phần tử vào mảng sẽ kiểm tra nếu lớn hơn max thì gán lại max

Đếm số lượng phần tử trong khoảng (m,M)

B1: Khởi tạo count = 0

B2: Kiểm tra nếu số vừa nhập lớn hơn m thì tiếp bước 3

B3: Kiểm tra nếu số vừa nhập nhỏ hơn M thì tăng count

Mã Nguồn:

```
.data
input: .asciiz "Nhap n: "
input2: .asciiz "Nhap gioi han duoi m= "
input3: .asciiz "Nhap gioi han tren M= "
Open: .asciiz "[ "
Close: .asciiz " ] = "
output: .asciiz "\nMang da nhap: "
output2: .asciiz "\nMax="
output3: .asciiz "\nCount = "
array: .word 0:100 # int array[100]

.text
main:
# nhap n
    la $a0, input
    addi $v0, $zero, 4
    syscall
    addi $v0, $zero, 5
    syscall
    addi $s0, $v0, 0 # $s0 = n

    la $a0, input2
    addi $v0, $0, 4
    syscall
    addi $v0, $0, 5
    syscall
    addi $s1, $v0, 0 # $s1 = m

    la $a0, input3
    addi $v0, $0, 4
    syscall
    addi $v0, $0, 5
    syscall
    addi $s2, $v0, 0 # $s2 = M
    addi $t6, $zero, 0
    jal NhapMang
    jal XuatMang
    jal end_Max
    jal Count_Element
    addi $v0, $0, 10
```

```
        syscall
end_main:
NhapMang:
# khoi tao
    addi $t3,$zero,0 #max=0
    addi $t1, $zero, 0 # $t1 = 0
    la $a1, array # $a1 = &array
    NhapPhanTu:
        # kiem tra so lan lap
        slt $t2, $t1, $s0
        beq $t2, $zero, KetThucNhap

        # xuat dau nhac nhap
        la $a0, Open
        addi $v0, $zero, 4
        syscall

        addi $a0, $t1, 0
        addi $v0, $zero, 1
        syscall

        la $a0, Close
        addi $v0, $zero, 4
        syscall

        # nhap so nguyen va luu vao array[i]
        addi $v0, $zero, 5
        syscall
        sw $v0, ($a1)
        lw $s6,($a1)
        slt $t2,$s6,$t3 # so sanh so nguyen vua nhap voi max
        beq $t2,$zero,changmax
        slt $t2,$s6,$s1 # so sanh so nguyen vua nhap voi m
        beq $t2,$zero,Max1
        # tang chi so
        addi $t1, $t1, 1
        addi $a1, $a1, 4

    j NhapPhanTu
# thay doi max bang so nguyen vua nhao, neu nguyen vua nhao lon hon
changmax:
    addi $t3,$s6,0
    slt $t2,$s6,$s1
```

```
        beq $t2,$zero,Max1
        # tang chi so
        addi $t1, $t1, 1
        addi $a1, $a1, 4

        j NhapPhanTu

KetThucNhap: jr $ra
Max1:
        slt $t2,$s6,$s2 # so sanh so nguyen vua nhap voi M
        bne $t2,$zero,cou
        addi $t1, $t1, 1
        addi $a1, $a1, 4
        j NhapPhanTu

cou: # tang so luong neu trong khoang (m,M)
        addi $t6,$t6,1
        addi $t1, $t1, 1
        addi $a1, $a1, 4
        j NhapPhanTu

# Xuat
XuatMang:
        la $a0, output
        addi $v0, $zero, 4
        syscall
        la $a1, array
        addi $t1, $0, 0

XuatPhanTu:
        # kiem tra so lan lap
        slt $t2, $t1, $s0
        beq $t2, $0, Exit

        # xuat phan tu array[i]
        lw $a0, ($a1)
        addi $v0, $zero, 1
        syscall

        # xuat khoang trang
        addi $a0, $0, 0x20
        addi $v0, $0, 11
        syscall
```

```

        # tang i
        addi $t1, $t1, 1
        addi $a1, $a1, 4

        j XuatPhanTu
Exit: jr $ra

end_Max:    la $a0, output2
            addi $v0, $0, 4
            syscall
            addi $a0,$t3,0
            addi $v0, $zero, 1
            syscall
            jr $ra

Count_Element:

            la $a0, output3
            addi $v0, $0, 4
            syscall
            addi $a0,$t6,0
            addi $v0, $zero, 1
            syscall
            jr $ra

end_Count:

```

Kết quả hiển thị :

```

Nhap n: 3
Nhap gioi han duoi m= 1
Nhap gioi han tren M= 8
[ 0 ] = 3
[ 1 ] = 14
[ 2 ] = 6

Mang da nhap: 3 14 6
Max=14
Count = 2
-- program is finished running --

```

Bài 13: Tìm kí tự giống nhau giữa 2 chuỗi

Sinh viên thực hiện: Hoàng Đức Anh - 20176688

Đề bài: Viết chương trình nhập vào 2 chuỗi kí tự. In ra số kí tự giống nhau và các kí tự giống nhau đó

Ví dụ: nhập vào S1 = "aabcc", S2 = "adcaa", đầu ra bao gồm

+ commonCharacterCount(S1, S2) = 3

+ Chuỗi có 3 kí tự giống nhau 2 "a" và 1 "c"

Thuật toán:

Để đổi biểu thức trung tố sang hậu tố, ta sẽ dùng ngăn xếp và xâu

B1: Truyền vào 2 chuỗi kí tự, và độ dài của hai chuỗi đó

B2:

+ Tạo 2 mảng có 26 phần tử. Freq1[] và Freq2[]

+ Tăng giá trị của phần tử thứ "i" trong mảng Freq1[] lên 1 đơn vị nếu xuất hiện kí tự có mã ascii là i

Code java :

```
for (i = 0; i < n1; i++)  
    freq1[s1[i] - 'a']++;
```

+ Tăng giá trị của phần tử thứ "j" trong mảng Freq2[] lên 1 đơn vị nếu xuất hiện kí tự có mã ascii là j

Code java :

```
for (j = 0; j < n2; j++)  
    freq2[s2[j] - 'a']++;
```

B3: In ra số phần tử giống nhau và số lần xuất hiện của chúng:

Code java:

```
for (i = 0; i < 26; i++) {  
    count += (min(freq1[i], freq2[i]));  
    System.out.println("có %d kí tự giống nhau", count);  
    if(freq1[i] >= 1 && freq2[i] >= 1)  
        System.out.println("kí tự %s xuất hiện %d lần/n", s1.charAt[i],  
min(freq1[i], freq2[i]));  
}
```


Mã nguồn:

```
.data
s1: .space 50
s2: .space 50
count_s1: .word 0:26          # Counts the number of character in string s1
count_s2: .word 0:26          # Counts the number of character in string s2
count_common_char_freq: .word 0:26 # store the common character
Message1: .asciiz "commonCharacterCount(s1, s2) = "
Message2: .asciiz "\n"
Message3: .asciiz " \" "
Message4: .asciiz " Nhap xau thu nhat:"
Message5: .asciiz " Nhap xau thu hai:"

.text
string_1:
    li $v0,54
    la $a0, Message4
    la $a1, s1
    li $a2, 50
    li $t9, 0
    syscall

get_length_s1:
    lb $t8, s1($t9)
    beq $t8, $zero, end_get_length_s1
    add $t9,$t9,1
    j get_length_s1
end_get_length_s1:
    subi $s0, $t9 , 1          # m = length(s1)

string_2:
    li $v0,54
    la $a0, Message5
    la $a1, s2
    li $a2, 50
    li $t9, 0
    syscall

get_length_s2:
    lb $t8, s2($t9)
    beq $t8, $zero, end_get_length_s2
    add $t9,$t9,1
    j get_length_s2
end_get_length_s2:
    subi $s1, $t9 , 1          # n = length(s2)
```

```

        li $s2, 0                # index i of s1
        li $s3, 0                # index j of s2
count_char_s1:
        slt $s4, $s2, $s0        # i < m ?
        beq $s4, $zero, end_count_char_s1 # if i < m => false => end_count
        lb $s5, s1($s2)          # $s5 = s1[i]
        sub $s5, $s5, 97          # $s5 = s1[i] - 'a'
        add $s5, $s5, $s5        # $s5 = 4*(s1[i] - 'a')
        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'] = $s6
        addi $s2, $s2, 1
        j count_char_s1
end_count_char_s1:
count_char_s2:
        slt $s4, $s3, $s1        # i < n?
        beq $s4, $zero, end_count_char_s2 # if j < n => false => end_count
        lb $s5, s2($s3)          # $s5 = s2[j]
        sub $s5, $s5, 97          # $s5 = s2[j] - 'a'
        add $s5, $s5, $s5        # $s5 = 4*(s2[j] - 'a')
        lw $s6, count_s2($s5)    # $s6 = count_s2[s2[j]-'a']
        addi $s6, $s6, 1         # $s6++
        sw $s6, count_s2($s5)    # count_s2[s2[i]-'a'] = $s6
        addi $s3, $s3, 1
        j count_char_s2
end_count_char_s2:
        li $t0, 26               # n=number member of count_common_char_freq
        li $t1, 0                # index i of array count_common_char_freq
count_common_char:
        slt $t2, $t1, $t0        # i < n ?
        beq $t2, $zero, end_count_common_char # if false => end_count
        add $t3, $t1, $t1        # $t3 = 4*i
        lw $t4, count_s1($t3)    # $s4 = count_s1[i]
        lw $t5, count_s2($t3)    # $s5 = count_s2[i]
        slt $t6, $t4, $t5        # count_s1[i] < count_s2[i] ?
        bne $t6, $zero, add_count_s1
add_count_s2:
        add $t7, $t7, $t5        # if count_s1[i] < count_s2[i] false,
commonCharacterCount += count_s2[i]
        sw $t5, count_common_char_freq($t3)

```

```

        addi $t1, $t1, 1                # i++
        j count_common_char
add_count_s1:
        add $t7, $t7, $t4                # if count_s1[i] < count_s2[i] true,
commonCharacterCount += count_s1[i]
        sw $t4, count_common_char_freq($t3)
        addi $t1, $t1, 1                # i++
        j count_common_char
end_count_common_char:
print:
        li $v0, 4                        # in commonCharacterCount(s1, s2)
        la $a0, Message1
        syscall

        li $v0, 1
        li $a0, 0
        add $a0, $a0, $t7
        syscall

        li $t1, 0
print_character:
        slt $t2, $t1, $t0                # Print the number appear of the same
character
        beq $t2, $zero, end_print_character
        add $t3, $t1, $t1
        add $t3, $t3, $t3
        lw $t4, count_common_char_freq($t3)
        beq $t4, $zero, add_index
        li $v0, 4
        la $a0, Message2
        syscall

        li $v0, 1
        li $a0, 0
        add $a0, $a0, $t4
        syscall

        li $v0, 4
        la $a0, Message3
        syscall

        li $v0, 11
        addi $a0, $t1, 97

```

```

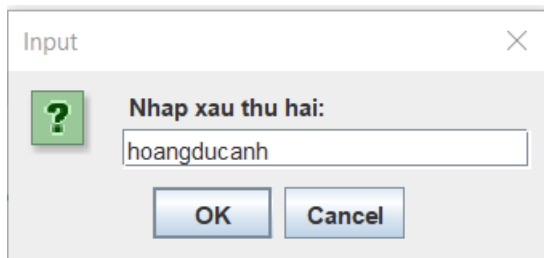
syscall

    li $v0, 4
    la $a0, Message3
    syscall
add_index:
    addi $t1, $t1, 1
    j print_character
end_print_character:

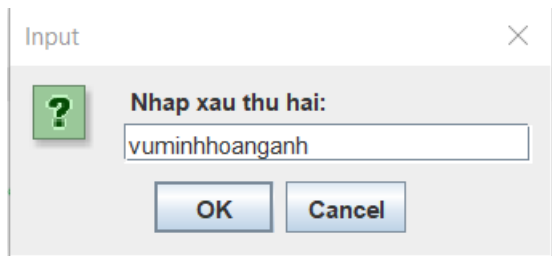
```

Kết quả hiển thị :

String 1:



String 2:



Kết quả:

