

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

Viện Công Nghệ Thông Tin Và Truyền Thông



BÁO CÁO GIỮA KỲ

HỌC PHẦN KIẾN TRÚC MÁY TÍNH

Nhóm: 06

Sinh viên: Trần Ngọc Công – 20176707

Hoàng Minh Nguyệt - 20176839

Mã lớp: 113834

Giảng viên hướng dẫn: ThS. Lê Bá Vui

MỤC LỤC

	Trang
Mục lục.....	2
Đề bài.....	3
Hướng giải quyết bài toán	3
Source code	4
Kết quả chương trình.....	7

ĐỀ BÀI

[06]. Given an array of word elements and the number of elements, write a procedure to find the pair of adjacent elements that has the largest product and return that product.

Example: For inputArray = [3, 6, -2, -5, 7, 3]

The output should be the product of 7 and 3 is 21

[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 is lucky or not.

Example: For n = 1230, the output should be isLucky(n) = true

For n = 239017, the output should be isLucky(n) = false

HƯỚNG GIẢI QUYẾT BÀI TOÁN

[06]

Số phần tử của mảng và mảng số nguyên do người dùng nhập từ bàn phím

- Nếu số phần tử âm -> báo lỗi và nhập lại
- Nếu mảng có 0 phần tử -> thông báo mảng rỗng và thoát
- Nếu mảng có 1 phần tử -> thông báo giá trị max chính là phần tử đó và kết thúc
- Nếu mảng có nhiều hơn 1 phần tử tìm MAX PRODUCT theo thuật toán:

```
max = a[0]*a[1];
```

```
for_loop i = 0, i < n - 1, i++
```

```
if (a[i]*a[i+1] > max) max = a[i]*a[i+1];
```

```
end_loop
```

[12]

Kiểm tra xem ký tự nhập vào có phải ký tự số không? Nếu sai -> Nhập lại

Kiểm tra xem số các chữ số trong số nhập vào có phải số chẵn không? Nếu không -> Nhập lại

Khi nó là số và số các chữ số chẵn -> Kiểm tra xem có phải số may mắn không !

SOURCE CODE

[06]

```
# Kiem tra giao ky Kien Truc May Tinh - De so 6

.data
mess1: .asciiiz "Nhap so phan tu cua mang: "
mess2: .asciiiz "Nhap mang\n"
mess3: .asciiiz "Nhap phan tu thu "
mess4: .asciiiz ": "
mess5: .asciiiz "Tich lon nhat duoc tim thay la cua "
mess6: .asciiiz " va "
mess7: .asciiiz " la: "
mess8: .asciiiz "Mang chi co mot phan tu duy nhat! Tich lon nhat la: "
mess9: .asciiiz "Mang rong! Khong tim duoc gia tri max!"
mess10: .asciiiz "So phan tu cua mang phai la mot so nguyen khong am! Moi nhap lai! \n"
A: .word 0:20

#-----
# $s0 : Dia chi mang A
# $s1 : So phan tu cua mang n
# $s2 : Bien chay i
# $s3 : MAX PRODUCT
# $s4 : Gia tri index hien tai
#-----

.text
add $s2, $zero, $zero

main:
li $v0, 4 # "Nhap so phan tu cua mang: "
la $a0, mess1
syscall

li $v0, 5 # Nhap so phan tu n cua mang
syscall

slti $t1, $v0, 0 # Kiem tra neu n < 0 -> Bao loi va nhap lai
bne $t1, $zero, error

beq $v0, $zero, ArrayNull # Neu n = 0 -> Nhay den ArrayNull

slti $t1, $v0, 2 # Kiem tra neu n =1 -> Nhay den OneElement
bne $t1, $zero, OneElement

add $s1, $zero, $v0 # Khi n >= 2, gan n->$s0

li $v0, 4 # "Nhap mang"
la $a0, mess2
syscall

input:
slt $t1, $s2, $s1 # i < n ? $t1 = 1 : $t1 = 0
beq $t1, $zero, end_input # Neu i >= n -> Ket thuc nhap mang

li $v0, 4 # "Nhap phan tu thu "
la $a0, mess3
syscall

li $v0, 1 # In STT cua tung phan tu trong mang
addi $t9, $s2, 1
add $a0, $t9, $zero
syscall

li $v0, 4 # ": "
la $a0, mess4
syscall

li $v0, 5 # Nhap tung phan tu cua mang -> $v0
syscall

add $t2, $s2, $s2 # $t2 = 4i
add $t2, $t2, $t2 #
sw $v0, A($t2) # Phan tu nhap vao tu ban phim -> A[i]
addi $s2, $s2, 1 # i = i+1
j input

end_input:
la $s0, A # Dia chi cua mang A
li $s2, 0 # Bien i = 0
li $s3, 0 # max = 0
sub $t0, $s1, 1 # n-1
j findmaxproduct
nop
```

```

findmaxproduct:
    addi    $t3,$s0,0           # Dia chi cua A[0] -> $t3
    lw      $t4,0($t3)          # Gia tri cua A[0] -> $t4
    add     $t6,$s0,4           # Dia chi cua A[1] -> $t6
    lw      $t7,0($t6)          # Gia tri cua A[1] -> $t7
    mult    $t4, $t7            # A[0] * A[1]
    mflo    $s3                 # Khoi tao max ban dau A[0] * A[1] -> $s3
    move    $s4, $t4
    move    $s5, $t7

loop:
    slt     $t1, $s2, $t0       # i < n-1 ? t1 = 1 : t1 = 0
    beq     $t1, $zero, end_findmaxproduct
    add     $t2,$s2,$s2         # $t2 = 2i
    add     $t2,$t2,$t2         # $t2 = 4i
    add     $t3,$t2,$s0         # Dia chi cua A[i] -> $t3
    lw      $t4,0($t3)          # Gia tri cua A[i] -> $t4
    addi    $t5, $s2, 1         # i = i+1
    add     $t5,$t5,$t5         # $t5 = 2(i+1)
    add     $t5,$t5,$t5         # $t5 = 4(i+1)
    add     $t6,$t5,$s0         # Dia chi cua A[i+1] -> $t6
    lw      $t7,0($t6)          # Gia tri cua A[i+1] -> $t7
    mult    $t4, $t7            # A[i] * A[i+1]
    mflo    $s1                 # Lay tich duoc luu o thanh ghi lo , luu vao $s1
    slt     $s6, $s3, $s1       # max < $s1 ? $s6 = 1 : $s6 = 0
    beq     $s6, $zero, loop_continue
    move    $s3, $s1            # Neu max > $s1 -> tang bien len 1 va tiep tục lap
    move    $s4, $t4            # Neu max < $s1 -> max = $s1
    move    $s5, $t7

loop_continue:
    add     $s2, $s2, 1         # i = i+1
    j       loop

end_findmaxproduct:
    li      $v0, 4              # "Tich lon nhat duoc tim thay la cua "
    la      $a0, mess5
    syscall
    li      $v0, 1              # i [max]
    add     $a0, $s4, $zero
    syscall
    li      $v0, 4              # " va "
    la      $a0, mess6
    syscall
    li      $v0, 1              # i+1 [max]
    add     $a0, $s5, $zero
    syscall
    li      $v0, 4              # " la: "
    la      $a0, mess7
    syscall
    li      $v0, 1              # max
    add     $a0, $s3, $zero
    syscall
    j       end

ArrayNull:
    li      $v0, 4              # "Mang rong! Khong tim duoc gia tri max!"
    la      $a0, mess9
    syscall
    j       end

OneElement:
    li      $v0, 4              # "Nhap phan tu thu "
    la      $a0, mess3
    syscall
    li      $v0, 1
    li      $a0, 1
    syscall
    li      $v0, 4              # ": "
    la      $a0, mess4
    syscall
    li      $v0, 5
    syscall
    add     $s3, $v0, $zero
    li      $v0, 4              # "Mang chi co mot phan tu duy nhat! Tich lon nhat la: "
    la      $a0, mess8
    syscall
    li      $v0, 1              # max
    add     $a0, $s3, $zero
    syscall
    j       end

error:
    li      $v0, 4              # "Error"
    la      $a0, mess10
    syscall
    j       main

end:

```

[12]

Kiểm tra giữa kỳ Kiến Trúc Máy Tính – Đề số 12

```
.data
Message1: .asciiiz "\nNhap so ve cua ban: "
Message2: .asciiiz "So ve khong hop le !\n"
Message3: .asciiiz "So ve phai la mot so tu nhien co so chu so chan ! \nVui long nhap lai !"
Message4: .asciiiz "Ket qua: "
Message5: .asciiiz "Chuc mung ban so huu ve so may man !"
Message6: .asciiiz "Chuc ban may man lan sau !"
Input : .space 20
.text
main:

    li    $t0, 0                # so luong chu so = i = $t0
    li    $v0, 4
    la    $a0, Message1
    syscall

input:
    li    $v0, 12                # nhap lan luot chu so tu ban phim
    syscall
    subi  $s1, $v0, 10           # kiem tra ki tu nhap vao co phai enter khong
    beq   $s1, $zero, end_input  # neu la enter ket thuc viec nhap
check_number:
    subi  $s1, $v0, 48           # kiem tra ki tu nhap vao co phai chu so khong
    slt   $s2, $s1, $zero        # ma ascii cua chu so lon hon hoac bang 48
    bne   $s2, $zero, error      # neu sai bao loi

    subi  $s1, $v0, 57           # ma ascii cua chu so nho hon hoac bang 57
    slt   $s2, $zero, $s1        # neu sai bao loi
    bne   $s2, $zero, error
end_check_number:
    sb    $v0, Input($t0)        # luu ki tu tuong ung vao Input[i]
    addi  $t0, $t0, 1            # i++
    j     input
end_input:

check_number_of_char:
    li    $t1, 2
    div   $t0, $t1
    mflo  $t2
    mfhi  $s1
    bne   $s1, $zero, error      # chia so chu so cho 2
                                # thuong luu trong $t2
                                # so du luu trong $s1
                                # kiem tra xem so chu so co phai so chan khong, neu sai bao loi
check_lucky_number:
    li    $t3, 0
    li    $t6, 0
    li    $t7, 0
first_half:
    slt   $t4, $t3, $t2          # chi so j
    beq   $t4, $zero, end_first_half # tong nua dau
    lb    $t5, Input($t3)        # tong nua sau
    add   $t6, $t6, $t5
    addi  $t3, $t3, 1
    j     first_half            # chi so j chay tu 0 den n/2-1
end_first_half:
second_half:
    slt   $t4, $t3, $t0          # chi so j chay tu n/2 den n-1
    beq   $t4, $zero, end_second_half
    lb    $t5, Input($t3)
    add   $t7, $t7, $t5
    addi  $t3, $t3, 1
    j     first_half
end_second_half:
    sub   $t4, $t6, $t7          # kiem tra $t6 co bang $t7 khong
    beq   $t4, $zero, lucky_number # neu co thong bao so may man
unlucky_number:
    li    $v0, 59                # thong bao so khong may man
    la    $a0, Message4
    la    $a1, Message6
    syscall
    j     end_main

lucky_number:
    li    $v0, 59                # thong bao so may man
    la    $a0, Message4
    la    $a1, Message5
    syscall
    j     end_main

error:
    li    $v0, 59                # thong bao loi
    la    $a0, Message2
    la    $a1, Message3
    syscall
    j     main

end_main:                        # ket thuc chuong trinh
```

KẾT QUẢ CHƯƠNG TRÌNH

[06]

```
Nhap so phan tu cua mang: -1
So phan tu cua mang phai la mot so nguyen khong am! Moi nhap lai!
Nhap so phan tu cua mang: |
```

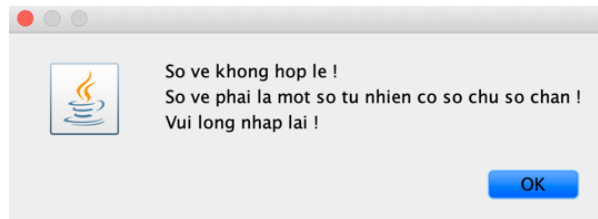
```
Nhap so phan tu cua mang: 0
Mang rong! Khong tim duoc gia tri max!
-- program is finished running (dropped off bottom) --
```

```
Nhap so phan tu cua mang: 1
Nhap phan tu thu 1: 4
Mang chi co mot phan tu duy nhat! Tich lon nhat la: 4
-- program is finished running (dropped off bottom) --
```

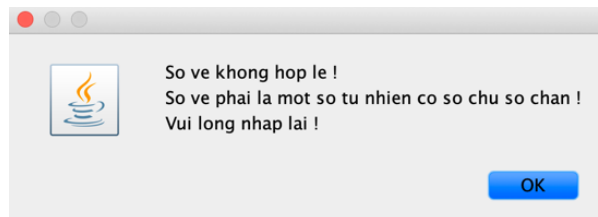
```
Nhap so phan tu cua mang: 5
Nhap mang
Nhap phan tu thu 1: -1
Nhap phan tu thu 2: 2
Nhap phan tu thu 3: -3
Nhap phan tu thu 4: 4
Nhap phan tu thu 5: 2
Tich lon nhat duoc tim thay la cua 4 va 2 la: 8
-- program is finished running (dropped off bottom) --
```

[12]

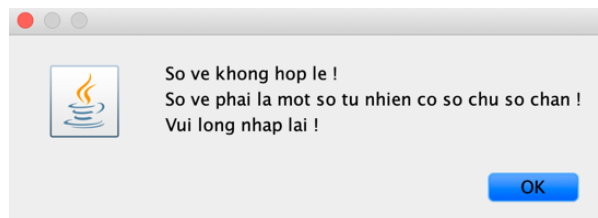
Nhap so ve cua ban: a



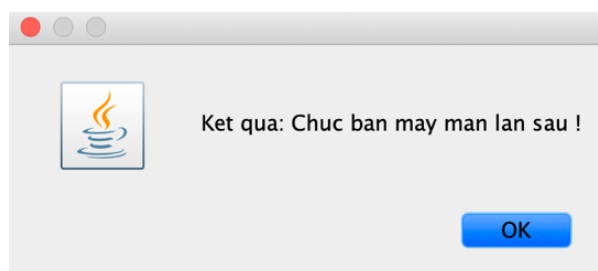
Nhap so ve cua ban: 123s



Nhap so ve cua ban: 123



Nhap so ve cua ban: 1234



Nhap so ve cua ban: 1203

