ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO THỰC HÀNH MÔN VI XỬ LÝ – VI ĐIỀU KHIỂN

BÀI THỰC HÀNH SỐ 2: XỬ LÍ IO, TÍNH TOÁN VÀ BỘ NHỚ TRÊN 8086

Sinh viên thực hiện:

Trần Ngọc Ánh 22520077

Giảng viên hướng dẫn: Phạm Minh Quân

Mã lớp: CE103.O22

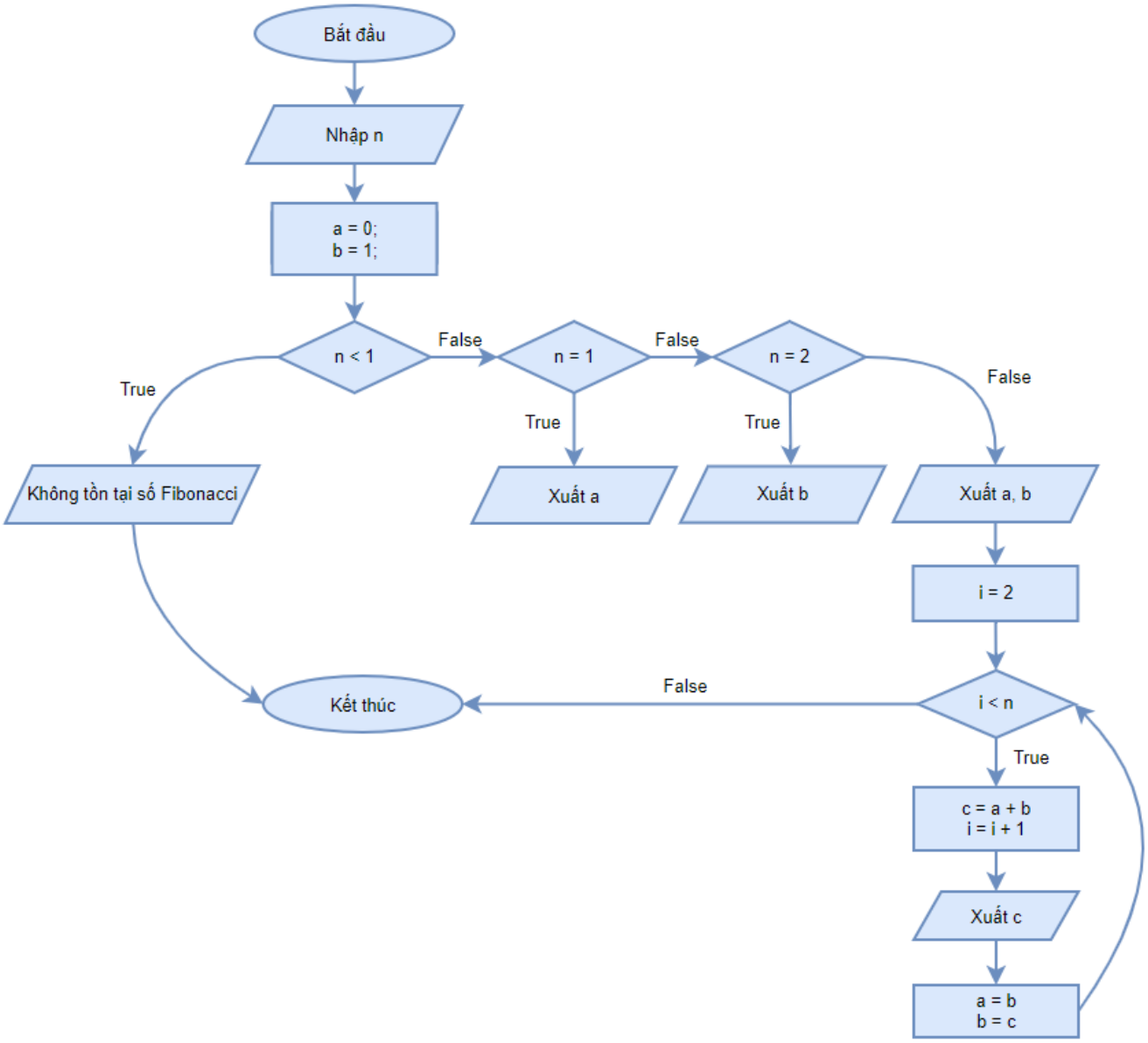
TP. HỒ CHÍ MINH, 25 THÁNG 3 NĂM 2024

YÊU CẦU CHUNG:

1. Nhập một số N có 2 chữ số từ bàn phím thông qua màn hình console.
2. In ra màn hình N số Fibonacci đầu tiên.

I. Phương pháp sử dụng vòng lặp (loop):

1. Lưu dồ thuật toán xử lý:



2. Code và trình bày kết quả:

- Source code:

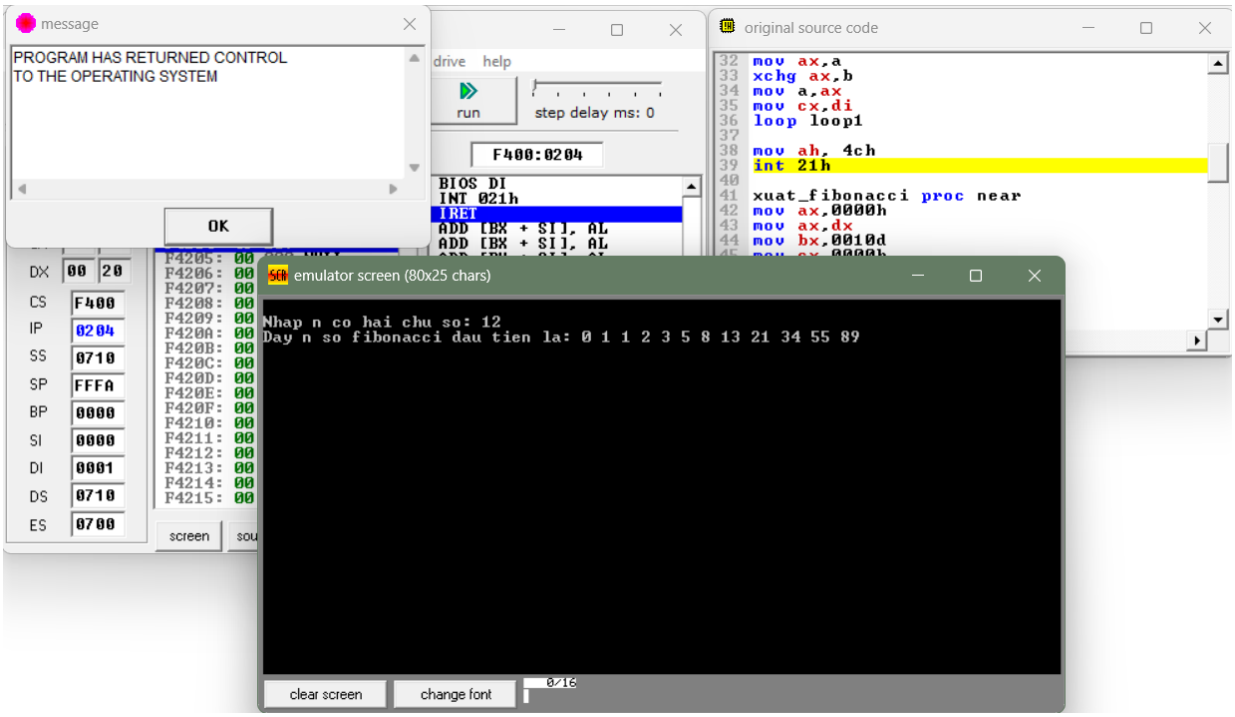
```
.model small
.data
    output1 db 10,13, 'Nhap n co hai chu so: $'
```

```
    output2 db 10,13, 'Day n so fibonacci dau tien la: 0 1 $'
    num dw ?
    a dw 0h
    b dw 01h
.code
    mov ax,@data
    mov ds,ax

    lea dx,output1
    mov ah,09h  ;In chuoi
    int 21h

    call nhap_fibonacci

    mov cx,num
    sub cx,02h

    lea dx,output2
    mov ah,09h
    int 21h
    loop1:
        mov ax,a
        add ax,b
        mov a,ax
        mov di,cx
        mov dx,ax
        call xuat_fibonacci
        mov ax,a
        xchg ax,b
        mov a,ax
        mov cx,di
        loop loop1
```

```
        mov ah, 4ch
        int 21h

xuat_fibonacci proc near
        mov ax,0000h
        mov ax,dx
        mov bx,0010d
        mov cx,0000h

    loop_push:
        mov dx,0000h
        div bx
        push dx
        inc cx
        cmp ax,0000h
        jne loop_push

    loop_pop:
        pop dx
        add dx,0030h
        mov ah,02h
        int 21h
        loop loop_pop

        mov dl,' '
        mov al,02h
        int 21h
    ret
    xuat_fibonacci endp

nhap_fibonacci proc near
```

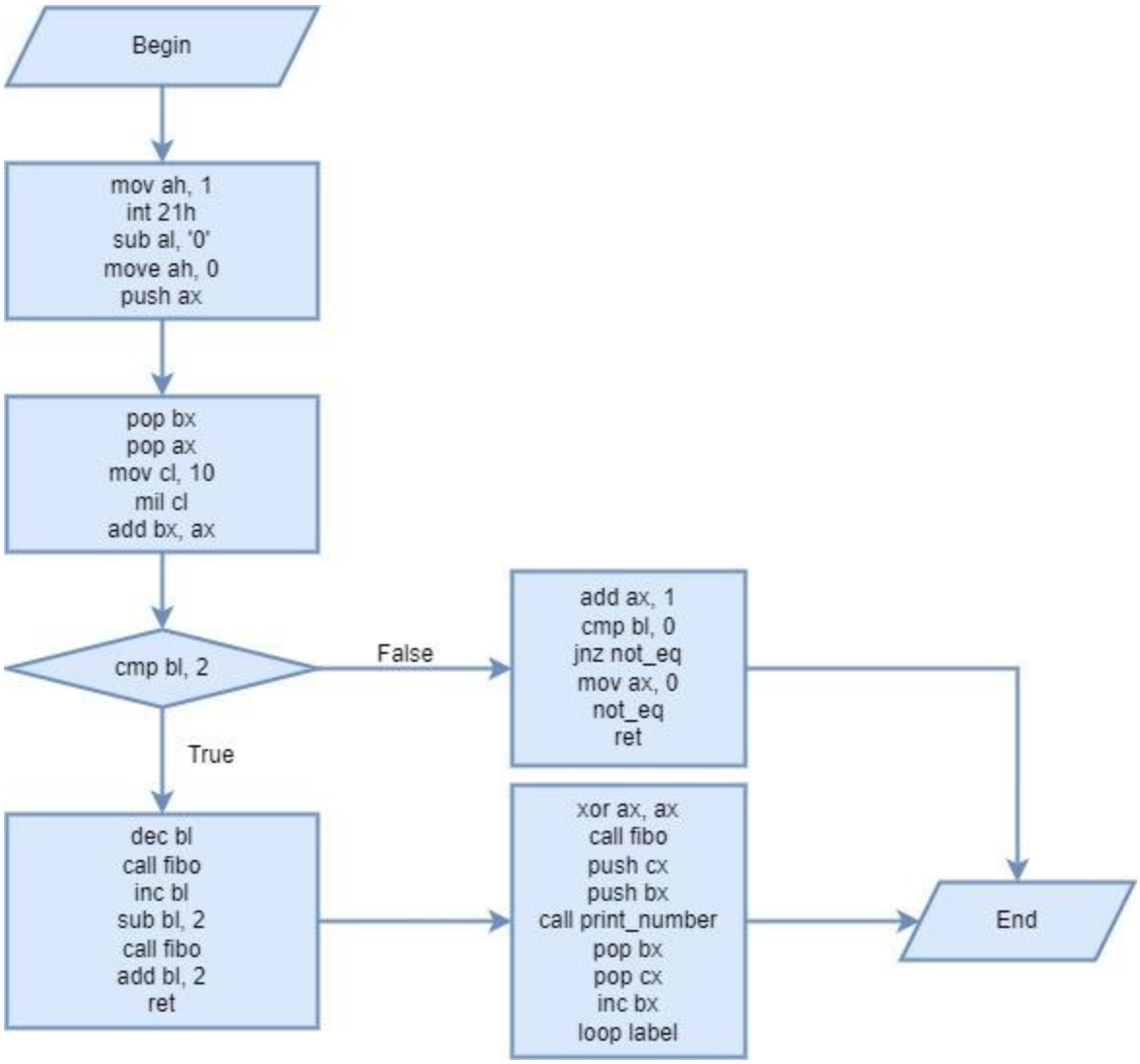
```
    mov ah,01h
    int 21h
    sub al,30h
    mov bl,al
    mov ah,01h
    int 21h
    sub al,30h
    mov ah,bl
    aad
    mov num,ax
    ret
    nhap_fibonacci endp
```

- Kết quả:



II. Phương pháp sử dụng đệ quy (recursion):

1. *Lưu đồ thuật toán xử lý:*



2. ***Giải thích các bước giải thích trong code và trình bày kết quả:***

- Source code và giải thích quá trình xử lý:

```
org 100h

# Nhập N
mov ah, 1 ; lấy kí tự từ bàn phím
int 21h
sub al, '0' ; chuyển đổi ký tự sang số nguyên (trừ ‘0’)
mov ah, 0
push ax  ; đẩy giá trị N vào ngăn xếp

# Nhập M
```

```
mov ah, 1 ; lấy ký tự từ bàn phím
int 21h
sub al, '0' ; chuyển đổi ký tự sang số nguyên (trừ '0')
mov ah, 0
push ax ; đẩy giá trị M vào ngăn xếp

# Lấy M và N từ ngăn xếp
pop bx
pop ax
mov cl, 10
mul cl
add bx, ax

call endl

# Thiết lập bx và cx cho vòng lặp
mov cx, bx ; di chuyển bx → cx (số lượng số hạng để in)
mov bx, 0  ; đặt lại bx thành 0 (số hạng hiện tại)

# Vòng lặp để in số Fibonacci
label:
    xor ax, ax ; xóa thanh ghi ax
    call fibo ; gọi thủ tục fibo
    ; đẩy cx, bx vào ngăn xếp (bảo toàn giá trị vòng lặp)
    push cx
    push bx
    call print_number ; gọi thủ tục print_number
    ; lấy bx, cx ra khỏi ngăn xếp (khôi phục giá trị vòng lặp)
    pop bx
    pop cx
```

```
    inc bx ; tăng bx (bộ đếm số hạng hiện tại)
    loop label ; lặp lại label nếu cx khác 0, tiếp tục in
ret

# Hàm tính fibonacci thứ n
fibo proc
    cmp bl, 2 ; so sánh bl (số hạng hiện tại) với 2
    jg do_calculation ; nếu lớn hơn đi đến do_calculation
# Kiểm tra các trường hợp cơ bản
    add ax, 1 ; nếu bl = 1, cộng 1 vào ax và trả về
    ; nếu bl = 0
    cmp bl, 0
    jnz not_eq
    mov ax, 0 ; đặt ax thành 0 và trả về trường hợp cơ bản
    not_eq:
    ret
fibo endp

do_calculation proc
    dec bl ; giảm bl (số hạng hiện tại)
    call fibo ; gọi thủ tục fibo để lấy số hạng thứ n-1
    inc bl ; tăng bl (khôi phục số hạng hiện tại)

    sub bl, 2 ; giảm bl hai lần (lấy số hạng thứ n-2)
    call fibo ; gọi thủ tục fibo để lấy số hạng thứ n-2
    add bl, 2 ; tặng bl hai lần (khôi phục số hạng hiện tại)
    ret ; trả về số fibonacci được tính toán trong ax
do_calculation endp

print_number proc
```

```
    xor cx, cx ; đặt bộ đếm (cx) thành 0
    mov bx, 10 ; đặt bộ chia (bx) thành 10

# Vòng lặp để chuyển đổi chữ số
.a: xor dx, dx ; xóa thanh ghi dx
    div bx ; chia ax cho bx
    push dx ; đẩy chữ số dx vào ngăn xếp
    inc cx ; tăng bộ đếm cx
    test ax, ax ; kiểm tra ax có bằng 0 hay không
    jnz .a ; nếu khác 0, lặp lại .a
# Vòng lặp để in chữ số
.b:
    pop dx ; lấy dx ra khỏi ngăn xếp
    mov ah, 2
    add dx, 48 ; chuyển đổi chữ số sang ascii bằng cách +48
    int 21h ; in chữ số
    loop .b ; lặp lại .b nếu còn nhiều chữ số

    # In một kí tự khoảng trắng ‘ ’
    mov ah, 2
    mov dx,' '
    int 21h
    ret
print_number endp

# Hàm endl
endl proc
    mov ah,2
    mov dl,0Ah
    int 21h
```

```
    mov ah,2
    mov dl,0Dh
    int 21h
    ret
endl endp
```

- Kết quả: