

# **PRACTICE EXERCISES OF THE MICROPROCESSORS & MICROCONTROLLERS**

**Instructor: The Tung Than**

**Student's name: Lê Nguyễn Hương Tiên**

**Student code: 21521521**

---

## **PRACTICE REPORT #6 IO PROCESSING, CALCULATION AND MEMORY ON THE 8086 MICROPROCESSOR**

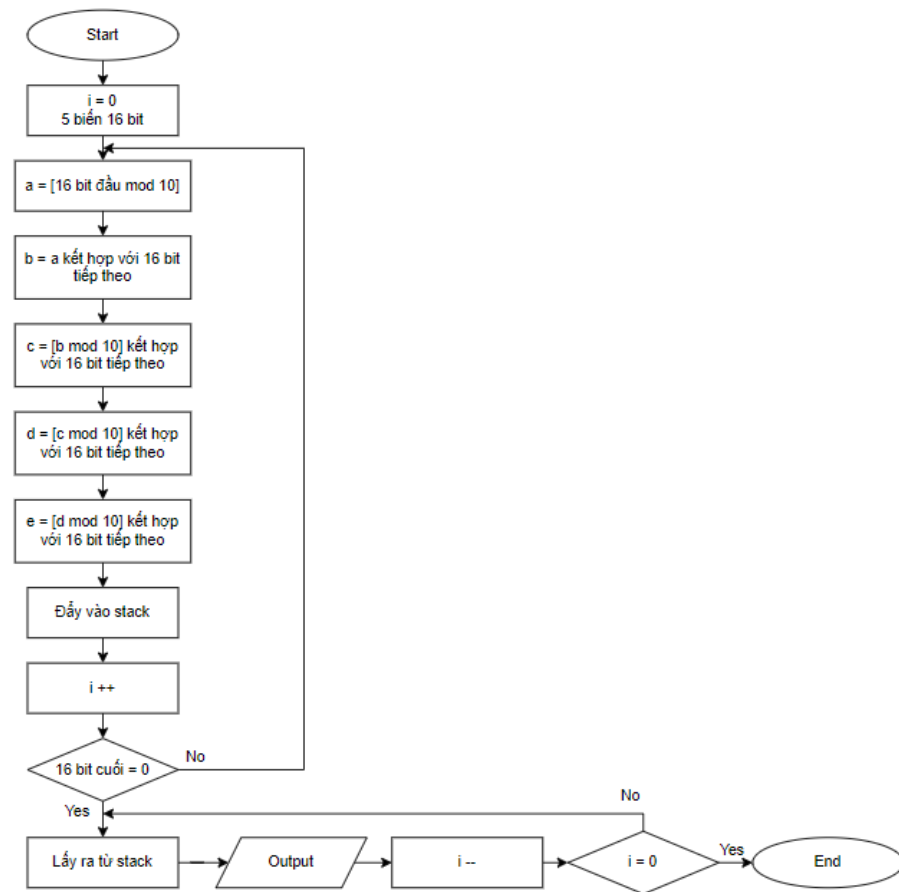
Link drive:

<https://drive.google.com/drive/folders/17ZObYCdtv8kIH6TIUfS6j31OjhQxsL1S?usp=sharing>

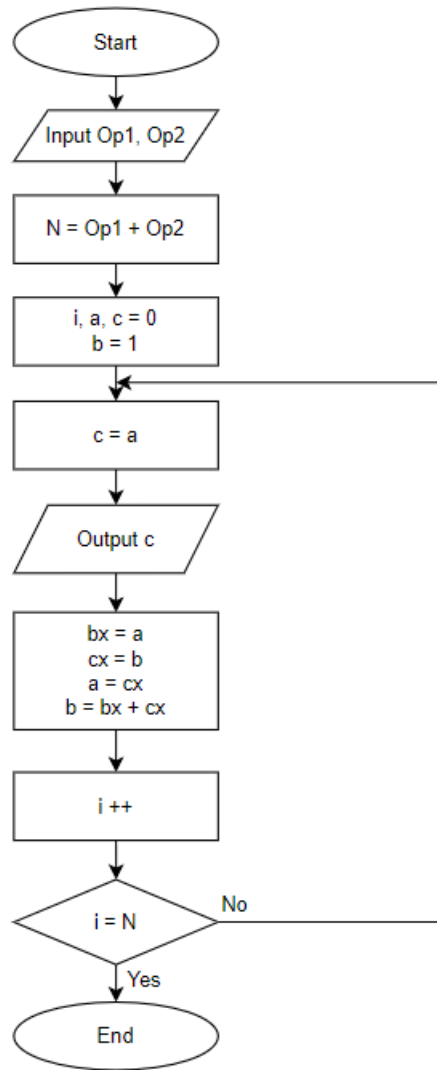
**I. Nhập vào 2 chữ số có 2 chữ số, tính tổng N và in ra màn hình N số đầu tiên của chuỗi Fibonacci**

**1. Flow chart**

**1.1. Xuất ra số 80 bits phù hợp với giá trị của số fibonacci thứ 99 (67 bits)**



**1.2. Xuất chuỗi fibonacci với N phần tử, với N là tổng 2 số 2 chữ số nhập từ bàn phím**



## 2. Source code

```

org 100h
.Data
msg1 db 10, 13, "Nhap vao so thu nhat: $"
msg2 db 10, 13, "Nhap vao so thu hai: $"
msg3 db 10, 13, "Fibonacci: $"
op1 db ?
op2 db ?
count dw ?
;So 80 bit
a1 dw 0 ;a
a2 dw 0
a3 dw 0
a4 dw 0
a5 dw 0

b1 dw 0 ;b
b2 dw 0
b3 dw 0
b4 dw 0
b5 dw 1

c1 dw 0 ;c
c2 dw 0
c3 dw 0
c4 dw 0
c5 dw 0
.Code
Start:
;Nhap so thu nhat
mov dx, offset msg1 ;In thong bao nhap
mov ah, 9
int 21h
call input
mov op1, bl
;Nhap so thu hai
mov dx, offset msg2 ;In thong bao nhap
mov ah, 9
int 21h
call input
mov op2, bl

;Tinh N
mov al, op1
mov bl, op2
add al, bl
mov ah, 100
cmp al, ah
jl Hehe
mov al, 99
Hehe:
xor ah, ah
mov count, ax
mov dx, offset msg3 ;In thong bao xuat chuoai
mov ah, 9
int 21h
call fibonacci
Done:
ret
;=====
;Ham xuat so 80 bit ra man hinh
Output_Num proc
push ax
push bx
push cx
push dx

mov bx, 10
mov cx, 0

```

```

Num_Separation:
    xor dx, dx
    mov ax, c1
    div bx
    mov c1, ax

    mov ax, c2
    div bx
    mov c2, ax

    mov ax, c3
    div bx
    mov c3, ax

    mov ax, c4
    div bx
    mov c4, ax

    mov ax, c5
    div bx
    mov c5, ax

    push dx
    inc cx
    cmp ax, 0
    je Output
    jmp Num_Separation
Output:
    pop dx

    add dl, 30h
    mov ah, 2
    int 21h

    dec cx

    cmp cx, 0
    jne Output

    pop dx
    pop cx
    pop bx
    pop ax

ret
Output_Num endp

```

;Dem so ki tu

```

;Cong 2 so 80 bit
Add_80bit proc
    push bx
    push cx
    ;Cong 16 bit cuoi
    mov bx, a5
    mov cx, b5
    mov a5, cx
    add bx, cx
    mov b5, bx
    ;Cong 16 bit tiep theo
    mov bx, a4
    mov cx, b4
    mov a4, cx
    adc bx, cx
    mov b4, bx
    ;Cong 16 bit tiep theo
    mov bx, a3
    mov cx, b3
    mov a3, cx
    adc bx, cx
    mov b3, bx
    ;Cong 16 bit tiep theo
    mov bx, a2
    mov cx, b2
    mov a2, cx
    adc bx, cx
    mov b2, bx
    ;Cong 16 bit dau
    mov bx, a1
    mov cx, b1
    mov a1, cx
    adc bx, cx
    mov b1, bx

    pop cx
    pop bx
    ret
Add_80bit endp

;=====
Fibonacci proc
    mov bx, 0
fibo:
    mov ax, a1
    mov c1, ax
    mov ax, a2
    mov c2, ax
    mov ax, a3
    mov c3, ax
    mov ax, a4
    mov c4, ax
    mov ax, a5
    mov c5, ax

    mov ah, 2
    mov dl, 9
    int 21h

    call Output_Num
    call Add_80bit

    inc bx
    cmp bx, count
    je exitfibo
    jmp fibo
exitfibo:
    ret
Fibonacci endp

```

```

;=====
;Ham nhap
input proc
    mov ah, 1
    int 21h
    mov bl, al
    sub bl, 30h
    int 21h
    mov bh, al
    cmp bh, 13
    je input1
    sub bh, 30h
    mov al, 10
    mul bl
    add al, bh
    mov bl, al
input1:
    ret
input endp

```

### 3. Nguyên lý hoạt động

Khai báo a, b, c là 3 biến với 80 bit.

Khai báo op1, op2 lần lượt là 2 chữ số nhập vào, count là tổng N của 2 chữ số trên.

Nhập vào 2 số thập phân 2 chữ số, tính tổng N của 2 số trên. Nếu  $N > 99$  thì  $N = 99$ .

In ra dãy số fibonacci với quy tắc:  $c = a$ , output c,  $temp = b$ ,  $b = a + b$ ,  $a = temp$ .

#### 3.1. Hàm input

Nhập kí tự đầu tiên và thứ hai. Kiểm tra kí tự thứ 2: nếu là enter thì số vừa nhập chỉ có 1 chữ số; nếu khác enter thì số vừa nhập là số có 2 chữ số, lấy kí tự đầu nhân cho 10 và cộng với kí tự thứ 2 để ra giá trị của số vừa nhập vào.

#### 3.2. Hàm Fibonacci

Khởi tạo giá trị 0 cho thanh ghi bx. Cho biến  $c = a$ , sau đó in biến c ra màn hình bằng hàm Output\_Num. Tính toán giá trị của số tiếp theo trong chuỗi fibonacci. Tiếp tục thực hiện cho đến khi in ra đủ N chữ số trong chuỗi fibonacci.

#### 3.3. Hàm Add\_80bit

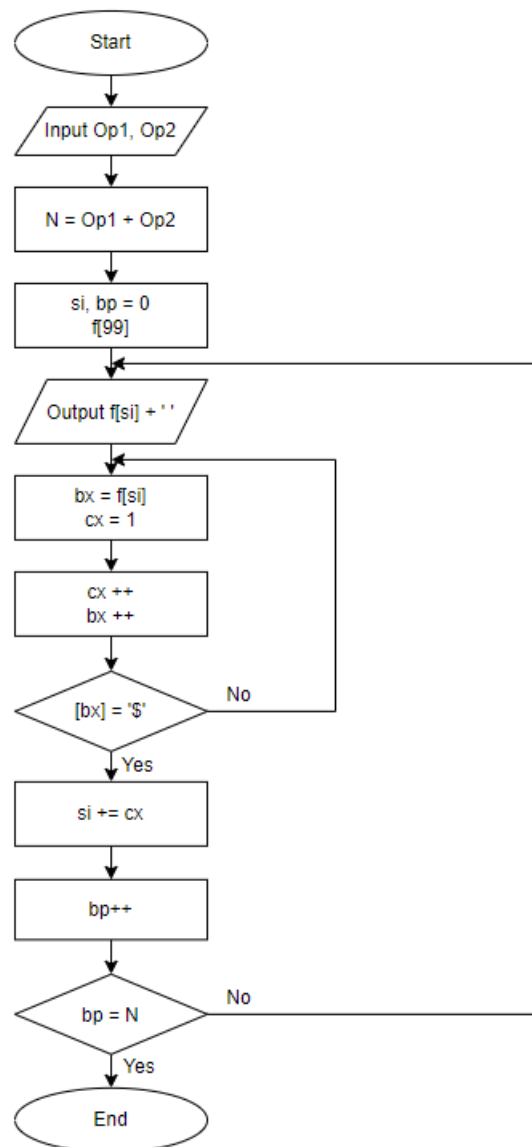
Tính toán giá trị của chữ số tiếp theo với quy tắc  $a = b$ ,  $b =$  tổng giá trị của 2 chữ số trước đó.

#### 3.4. Hàm Output\_Num

Lấy 16 bit đầu chia lấy dư cho 10 rồi kết hợp kết quả đó với 16 bit tiếp theo, tiếp tục chia lấy dư cho 10. Lặp lại cho đến khi chia đến 16 bit cuối, sau đó lấy số dư để vào stack. Khi giá trị 16 bit cuối = 0 thì dừng rồi lấy từng giá trị trong stack để in ra.

## II. Dùng cách khác để thực hiện yêu cầu trên

### 1. Flow chart



### 2. Source code



```

org 100h
.Data
    msg1 db 10, 13, "Nhap vao so thu nhat: $"
    msg2 db 10, 13, "Nhap vao so thu hai: $"
    msg3 db 10, 13, "Fibonacci: $"
    op1 db ?
    op2 db ?
    count dw ?
;Mang tu 0 den 7778742049
    f1 db "0$", "1$", "1$", "2$", "3$", "5$", "8$", "13$", "21$", "34$",
;Mang tu 12586269025 den 135301852344706746049
    f2 db "12586269025$", "20365011074$", "32951280099$", "53316291173$",
.Code
Start:
;Nhap so thu nhat
    mov dx, offset msg1                ;In thong bao nhap
    mov ah, 9
    int 21h
    call input
    mov op1, bl
;Nhap so thu hai
    mov dx, offset msg2                ;In thong bao nhap
    mov ah, 9
    int 21h
    call input
    mov op2, bl
;Tinh N
    mov al, op1
    mov bl, op2
    add al, bl
    mov ah, 100
    cmp al, ah
    jl Hehe
    mov al, 99
Hehe:
    xor ah, ah
    mov count, ax
    mov dx, offset msg3                ;In thong bao nhap
    mov ah, 9
    int 21h
    call fibonacci
Done:
ret

```

```

;Fibonacci
fibonacci proc
    mov si,0
    mov bp, 0
next:
    mov cx, 1
    lea bx, f1[si]
    mov dx, bx
    mov ah, 9
    int 21h
    mov ah, 2
    mov dl, ','
    int 21h
next1:
    cmp [bx], '$'
    je next2
    inc bx
    inc cx
    jmp next1
next2:
    add si, cx
    inc bp
    cmp bp, 50
    je next3
    cmp bp, count
    je Exit_fibo
    jmp next
next3:
    mov si,0
next4:
    mov cx, 1
    lea bx, f2[si]
    mov dx, bx
    mov ah, 9
    int 21h
    mov ah, 2
    mov dl, ','
    int 21h
next5:
    cmp [bx], '$'
    je next6
    inc bx
    inc cx
    jmp next5
next6:
    add si, cx
    inc bp
    cmp bp, count
    je Exit_fibo
    jmp next4
Exit_fibo:
ret
fibonacci endp
;=====
;Ham nhap
input proc
    mov ah, 1
    int 21h
    mov bl, al
    sub bl, 30h
    int 21h
    mov bh, al
    cmp bh, 13
    je input1
    sub bh, 30h
    mov al, 10
    mul bl
    add al, bh
    mov bl, al
input1:
ret
input endp
;=====

```

### 3. Nguyên lý hoạt động

Đưa 99 số fibonacci vào 2 chuỗi f1 và f2: f1 chứa 50 số đầu tiên, từ 0 đến 7778742049; f2 chứa các số còn lại, từ 1258626025 đến 135301852344706746049.

Khai báo op1, op2 lần lượt là 2 chữ số nhập vào, count là tổng N của 2 chữ số trên.

Nhập vào 2 số thập phân 2 chữ số, tính tổng N của 2 số trên. Nếu  $N > 99$  thì  $N = 99$ . Xuất lần lượt n phần tử trong mảng.

### **3.1. Hàm input**

Nhập kí tự đầu tiên và thứ hai. Kiểm tra kí tự thứ 2: nếu là enter thì số vừa nhập chỉ có 1 chữ số; nếu khác enter thì số vừa nhập là số có 2 chữ số, lấy kí tự đầu nhân cho 10 và cộng với kí tự thứ 2 để ra giá trị của số vừa nhập vào.

### **3.2. Hàm fibonacci**

B1: Khai báo thanh ghi si và bp bằng 0

B2: Khai báo thanh ghi cx bằng 1, đây là thanh ghi đếm số kí tự của số fibonacci

B3: In chữ số ở vị trí si ra màn hình

B4: Kiểm tra từng kí tự của số cho đến khi tìm ra kí tự '\$'

B5: Đưa con trỏ đến vị trí của số tiếp theo

B6: Kiểm tra đã in ra đủ N số chưa? Nếu chưa, lặp lại B2

B7: Kết thúc