

# THỰC HÀNH VI XỬ LÝ – VI ĐIỀU KHIỂN

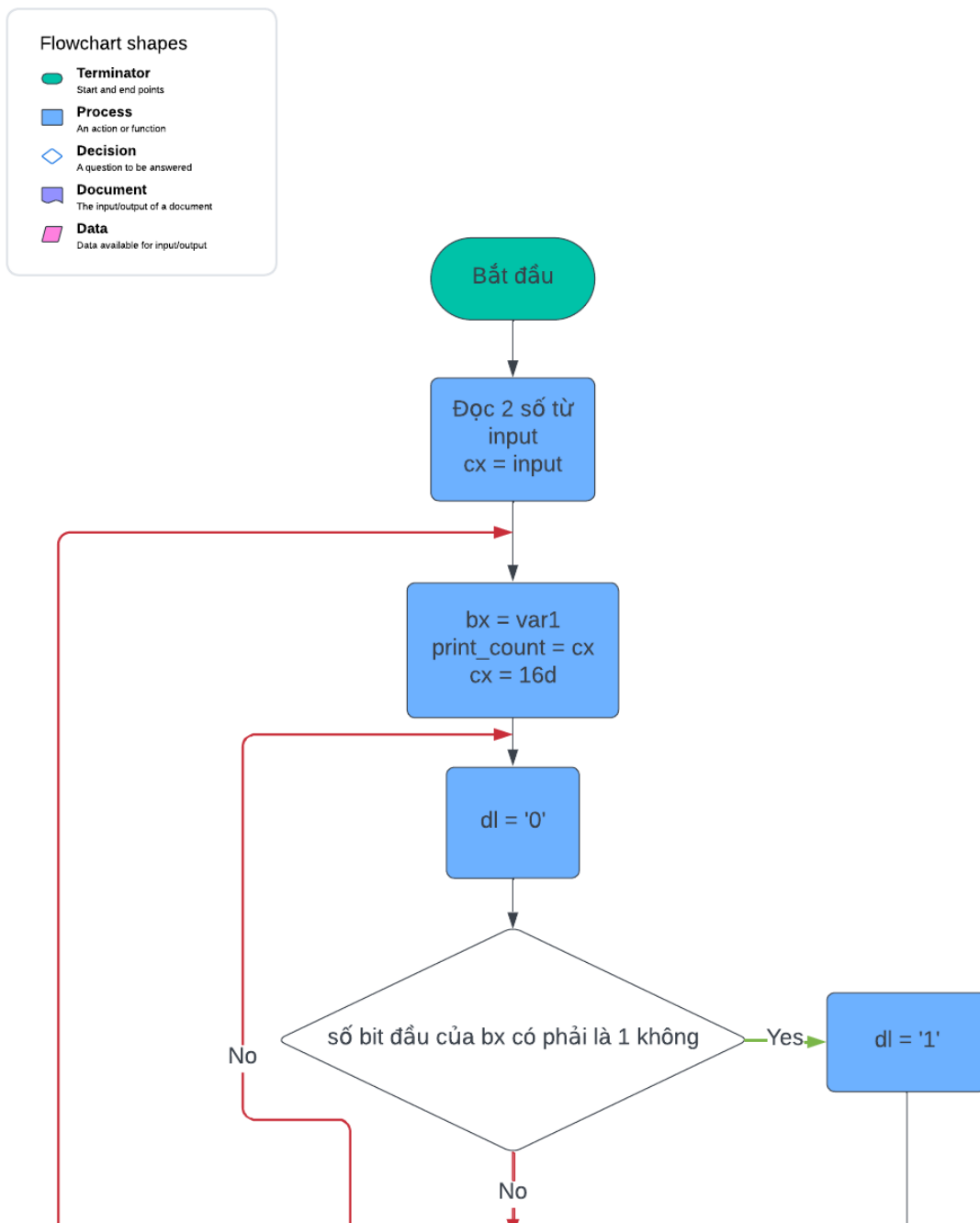
GVHD: Trần Ngọc Đức

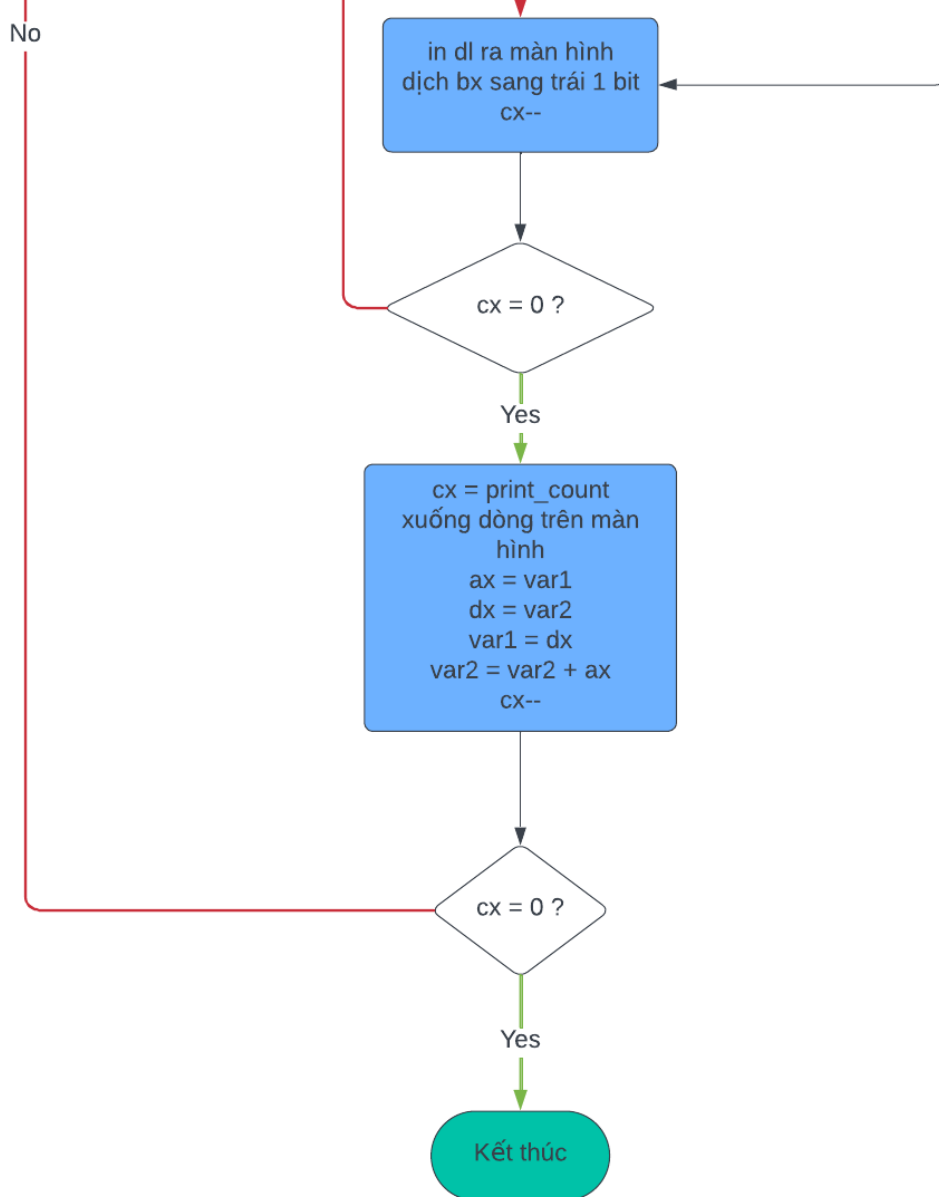
Họ và tên sinh viên thực hiện: Phạm Quốc Tiến

Mã số sinh viên: 22521472

## BÁO CÁO THỰC HÀNH SỐ 3

### I. Lưu đồ thuật toán





## II. Giải thích

### Khái báo biến

```

04 org 100h
05 .data
06     var1 DW 0d
07     var2 DW 1d
08     print_count DW 0d
09
10 .code
11

```

Do interrupt chỉ nhận được 1 ký tự nên em sẽ lấy chữ số hàng chục từ input rồi trừ cho 48 (do là ký tự) nhân cho 10 và đưa vào CX

```

12 mov ah, 1
13 int 21h
14 sub al, 48d
15 mov bl, 10d
16 mul bl
17 mov cx, ax

```

Tiếp đến em sẽ lấy chữ số hàng đơn vị trừ cho 48 rồi cộng với cl và lưu kết quả vào cl

```

18 mov ah, 1
19 int 21h
20 sub al, 48d
21 add cl, al

```

Em tiếp tục gọi interrupt để xuống dòng trên console và gọi Fib

```

22 mov ah, 0eh
23 mov al, 13d
24 int 10h
25 mov al, 10d
26 int 10h
27
28 call Fib

```

Khai báo thủ tục Fib

```

38 Fib PROC
72 Fib ENDP

```

Em gán trị var1 cho bx rồi lưu giá trị cx vào print\_count và gán cx thành giá trị 16 (để có thể xài lệnh loop)

```

39 mov bx, var1
40 mov print_count, cx
41 mov cx, 16d

```

Tiếp theo em sẽ gán cho dl ký tự 0 rồi so sánh bx với 32768d để xem số bit đầu tiên của bx có phải là 1 hay không. Nếu là 1 thì dl sẽ được đặt thành ký tự 1. Sau đó sẽ in ra màn hình ký tự lưu trong dl

```

42 loop_print:
43     mov dl, '0'
44     test bx, 32768d
45     jz print
46     mov dl, '1'
47 print:
48     mov ah, 2
49     int 21h

```

Sau đó em sẽ dịch bx sang trái 1 bit và lặp lại loop\_print

```

50 shl bx, 1
51 loop loop_print

```

Sau khi lặp xong 15 lần thì em sẽ trả giá trị cx về giá trị cũ

```

52 mov cx, print_count

```

Rồi em gọi interrupt xuống dòng trên console

**Em sẽ tính giá trị Fib tiếp theo bằng cách cho `var1 = var2` và cho `var2 = var2 + giá trị cũ của var1`**

**Chương trình sẽ đệ quy khi cx chưa bằng 0**

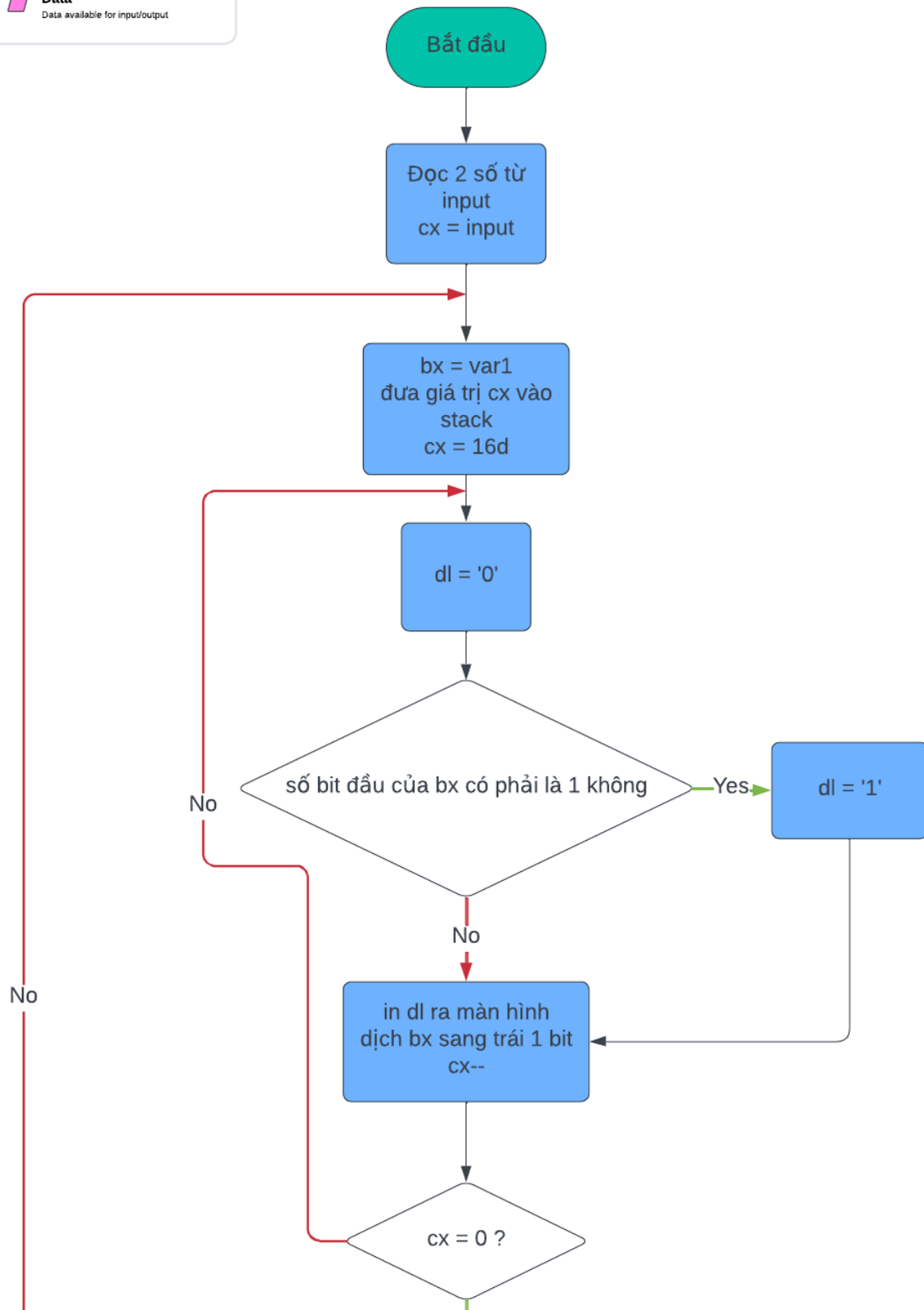
**Khi  $cx = 0$  thì chương trình sẽ return lại chỗ a: call Fib N -1 lần và return chương trình chính 1 lần. Sau đó chương trình chờ input từ bàn phím để kết thúc**

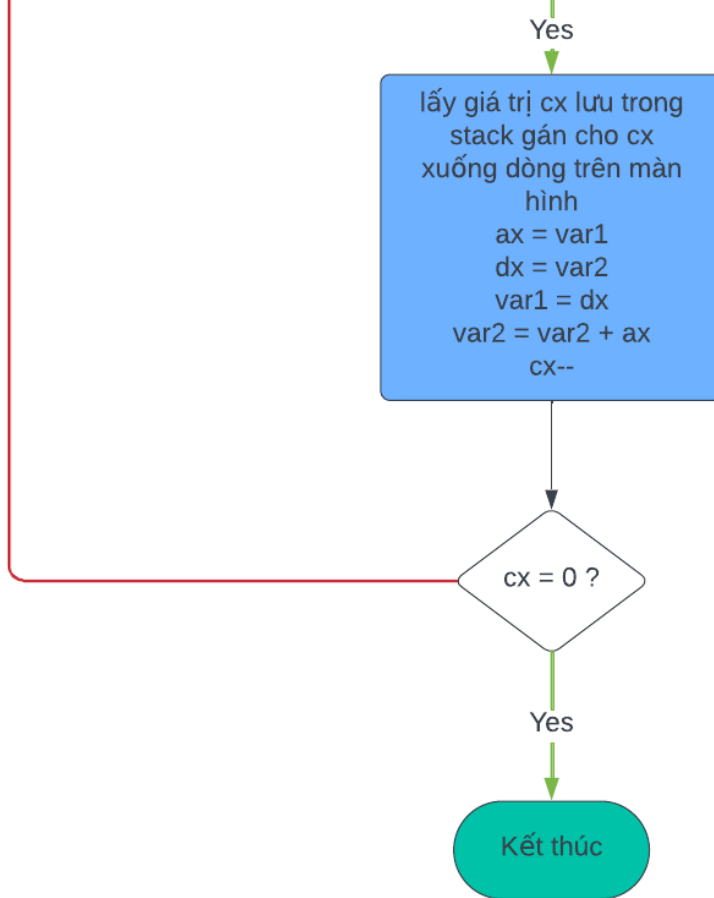
## Kết quả

### III. Lưu đồ và giải thích phần bài tập

### Flowchart shapes

- Terminator**  
Start and end points
- Process**  
An action or function
- Decision**  
A question to be answered
- Document**  
The input/output of a document
- Data**  
Data available for input/output





Trong phần code này thì khá giống với code của phần thực hành chỉ khác một vài chỗ:

Em sẽ không khai báo print count

```

03 , the location of
04 org 100h
05 .data
06     var1 DW 0d
07     var2 DW 1d
08
09 .code
10

```

Em sẽ không tạo thủ tục mà chỉ tạo label fib

```

20
27 fib:
28

```

Em sẽ lưu giá trị cx vào stack và load từ stack vào cx mà không cần qua print count

```

28     mov bx, var1
29     push cx
30
31
32
33
34
35
36
37
38
39
40     loop 100
41     pop cx
42

```

**Phần lặp dưới cùng thì em sẽ không gọi cần gọi thủ tục Fib và ret nữa mà chỉ cần xài lệnh loop**

```
56  
57     loop fib  
58
```