

BÀI GIẢNG CƠ SỞ LẬP TRÌNH

CHƯƠNG 3.

CẤU TRÚC ĐIỀU KHIỂN

NGUYỄN THÀNH THỦY

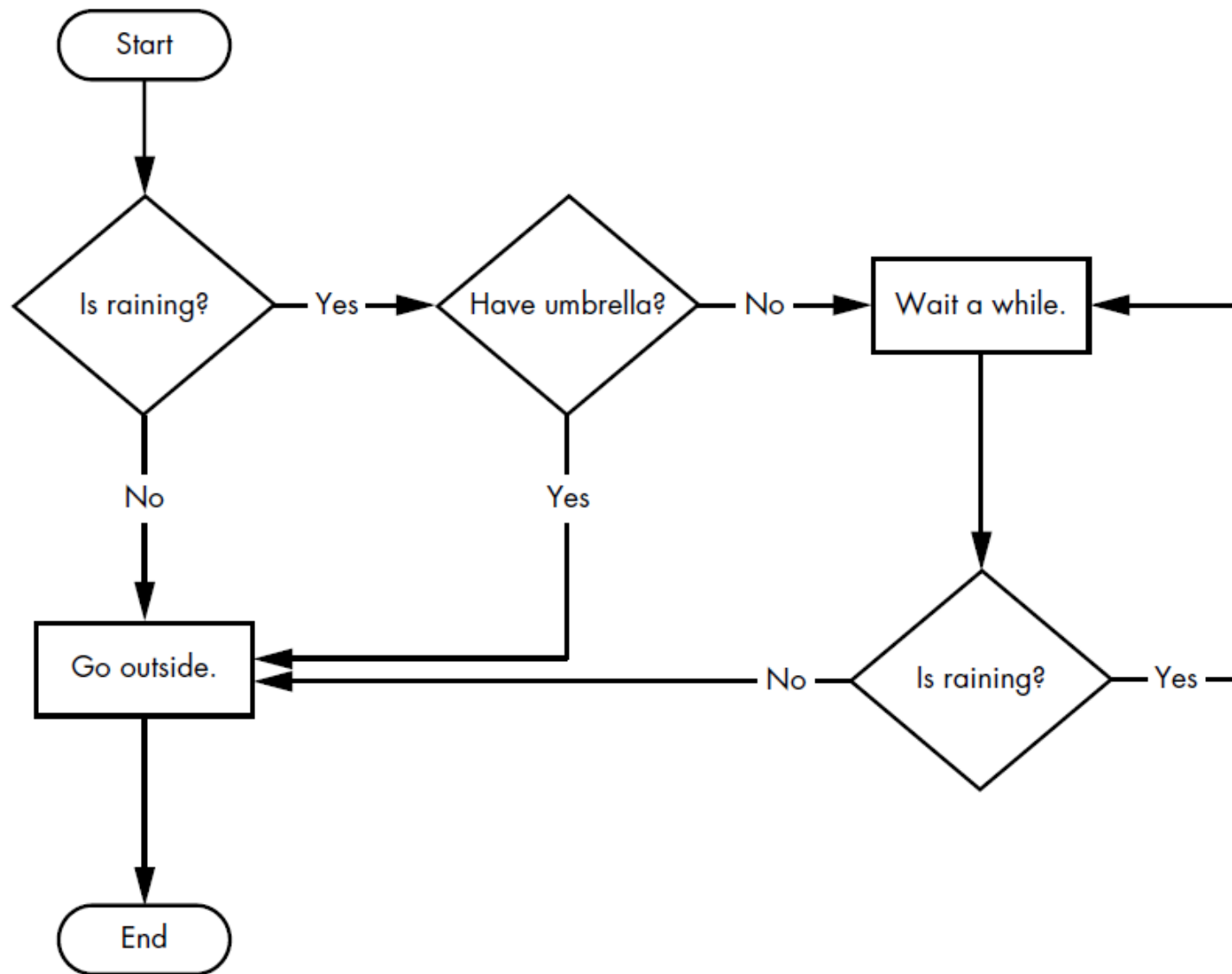
BỘ MÔN TIN HỌC QUẢN LÝ

TRƯỜNG ĐẠI HỌC KINH TẾ, ĐẠI HỌC ĐÀ NẴNG

THUYNT@DUE.EDU.VN

NỘI DUNG

- ❑ Cấu trúc điều kiện **if**
- ❑ Cấu trúc lặp **while**
- ❑ Cấu trúc lặp **for**
- ❑ Câu lệnh nhảy **break, continue, return**



A flowchart to tell you what to do if it is raining

GIÁ TRỊ KIỂU BOOLEAN

❑ Giá trị Boolean (Logic):

- **True**: giá trị đúng
- **False**: giá trị sai

```
>>> spam = True
```

```
>>> spam
```

```
True
```

```
>>> true
```

→ true khác True

```
Traceback (most recent call last):
```

```
File "<pyshell#2>", line 1, in <module>
```

```
    true
```

```
NameError: name 'true' is not defined
```

```
>>> True = 2 + 2
```

→ Không được sử dụng từ khóa của Python đặt tên cho biến

```
SyntaxError: assignment to keyword
```

TOÁN TỬ SO SÁNH

- Toán tử so sánh cho phép so sánh hai **giá trị/ biến/ biểu thức**. Kết quả trả về **True** hoặc **False**.

Operator	Meaning
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

TOÁN TỬ SO SÁNH

■ Ví dụ 3.1-1:

```
>>> 42 == 42
True
>>> 42 == 99
False
>>> 2 != 3
True
>>> 2 != 2
False
```

```
>>> 'hello' == 'hello'
True
>>> 'hello' == 'Hello'
False
>>> 'dog' != 'cat'
True
>>> True == True
True
>>> True != False
True
>>> 42 == 42.0
True
>>> 42 == '42'
False
```

```
>>> 42 < 100
True
>>> 42 > 100
False
>>> 42 < 42
False
>>> eggCount = 42
>>> eggCount <= 42
True
>>> myAge = 29
>>> myAge >= 10
True
```

TOÁN TỬ LOGIC

□ and, or, not

- Được dùng để kết hợp các giá trị/biến/biểu thức kiểu logic.

and	True	False
True	True	False
False	False	False

or	True	False
True	True	True
False	True	False

not	True	False
	False	True

TOÁN TỬ LOGIC

□ AND, OR, NOT

Ví dụ 3.1-2:

```
>>> True and True
```

```
True
```

```
>>> True and False
```

```
False
```

```
>>> False or True
```

```
True
```

```
>>> False or False
```

```
False
```

```
>>> not True
```

```
False
```

```
>>> not not not not True
```

```
True
```

TOÁN TỬ LOGIC

❑ Kết hợp giữa toán tử logic với toán tử so sánh

Ví dụ 3.1-3:

```
>>> (4 < 5) and (5 < 6)
True
>>> (4 < 5) and (9 < 6)
False
>>> (1 == 2) or (2 == 2)
True
```

```
(4 < 5) and (5 < 6)
  ↓
True and (5 < 6)
  ↓
True and True
  ↓
True
```

You can also use multiple Boolean operators in an expression, along with the comparison operators.

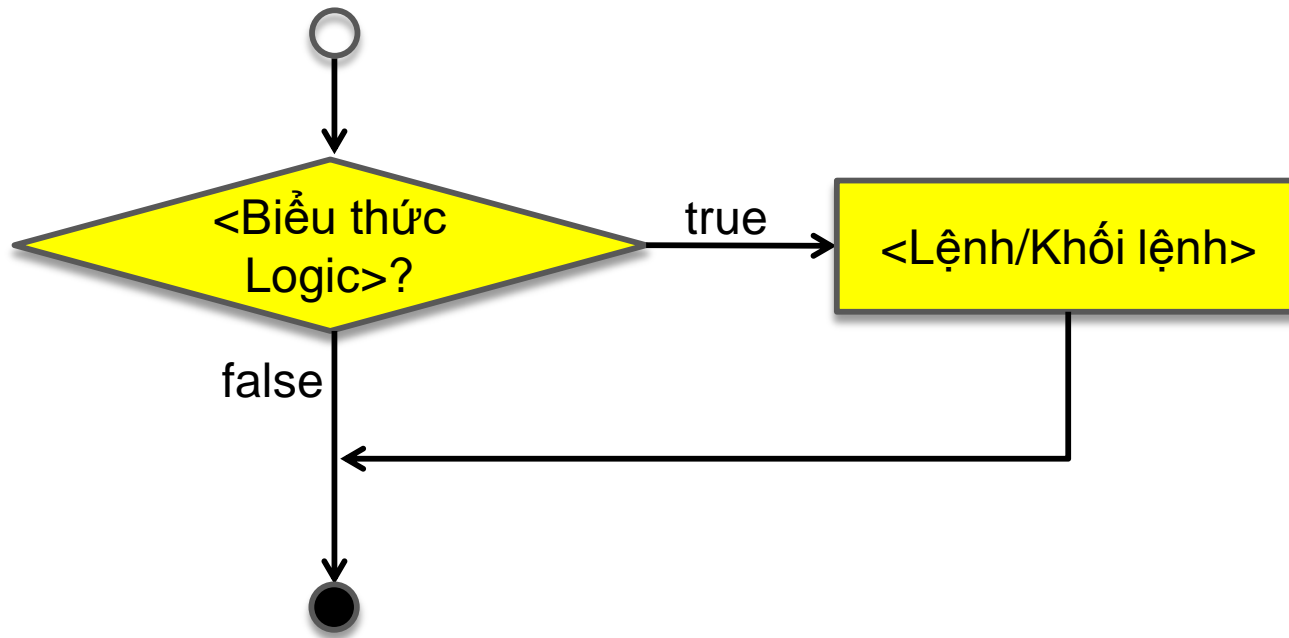
```
>>> 2 + 2 == 4 and not 2 + 2 == 5 and 2 * 2 == 2 + 2
True
```

CẤU TRÚC ĐIỀU KIỆN IF – DẠNG 1

□ Cú pháp

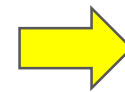
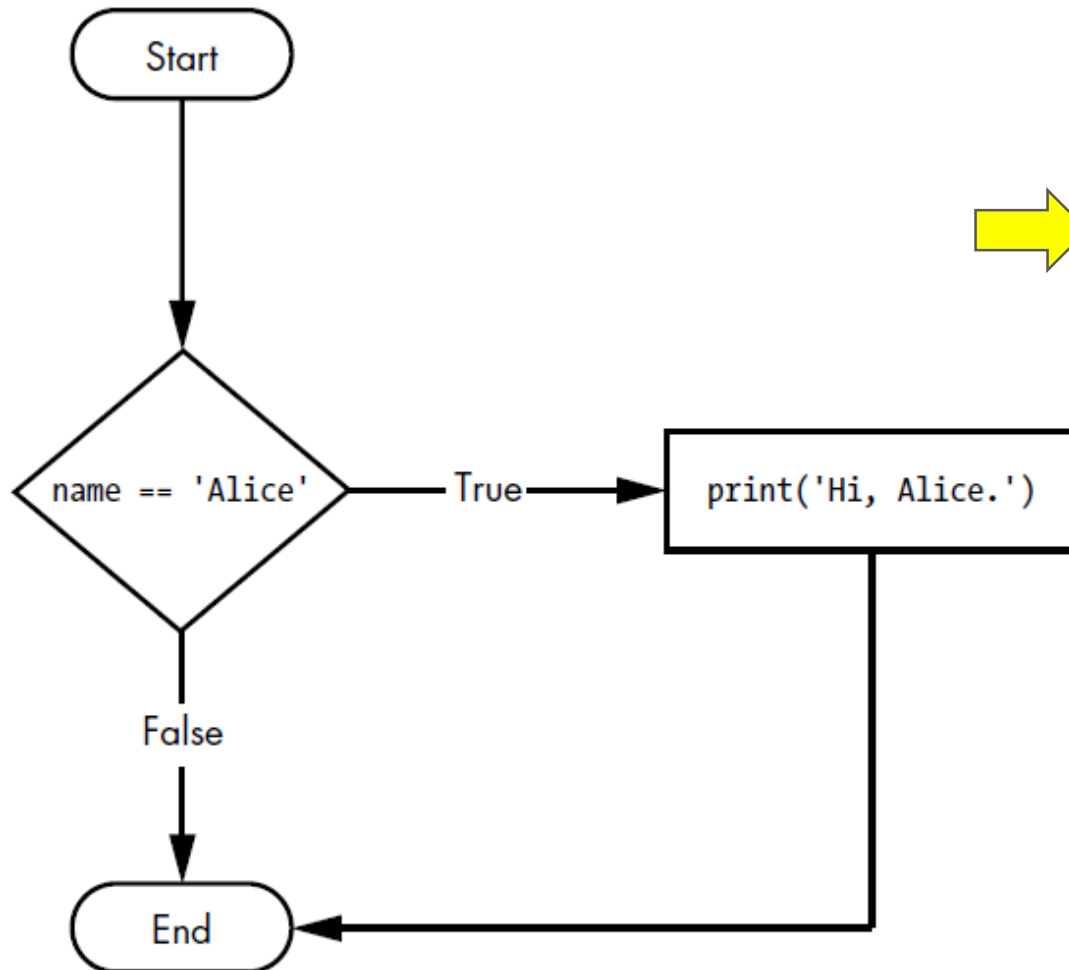
if <Biểu thức Logic>:

//Lệnh hoặc khối lệnh



CẤU TRÚC ĐIỀU KIỆN IF – DẠNG 1

Ví dụ 3.1-4:

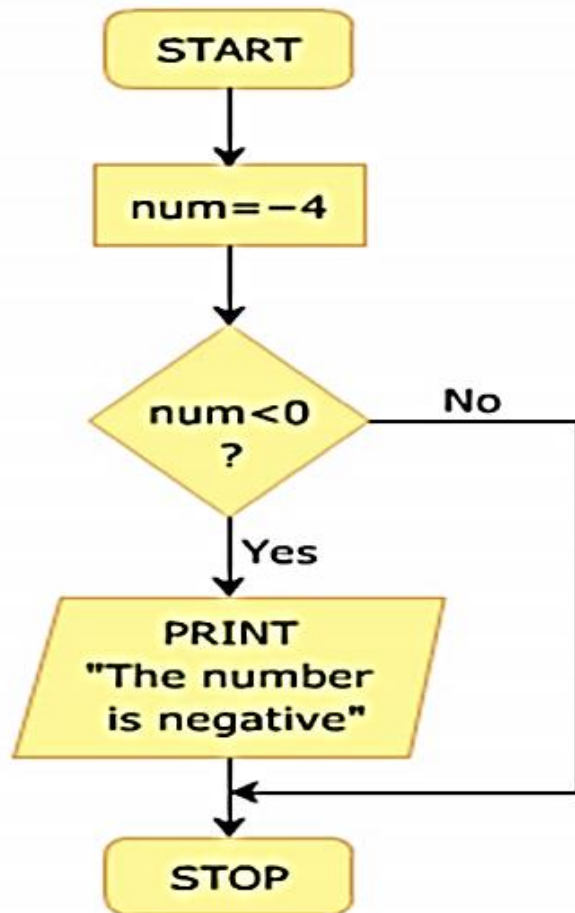


```
if name == 'Alice':  
    print('Hi, Alice.')
```



CẤU TRÚC ĐIỀU KIỆN IF – DẠNG 1

- ❑ **Ví dụ 3.1-5:** viết chương trình giải bài toán theo sơ đồ khối sau



```
num=-4
if num<0:
    print("The number is negative")
```

CẤU TRÚC ĐIỀU KIỆN IF – DẠNG 2

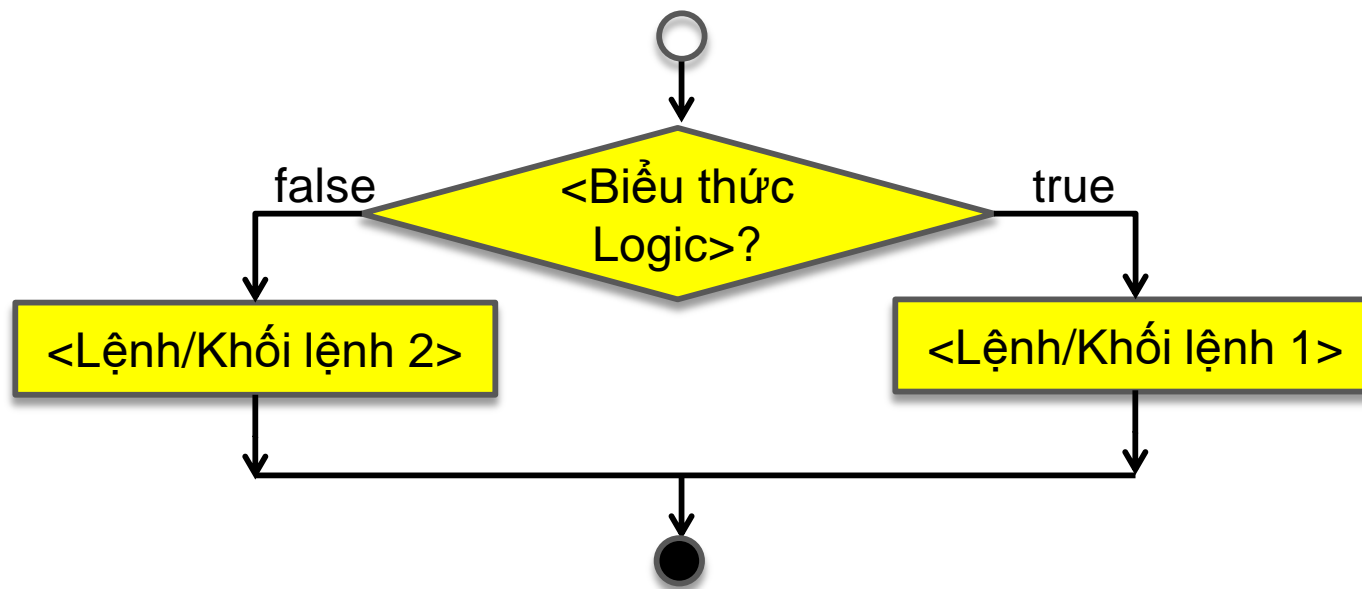
□ Cú pháp

if <Biểu thức Logic>:

//Lệnh hoặc khối lệnh 1

else:

//Lệnh hoặc khối lệnh 2

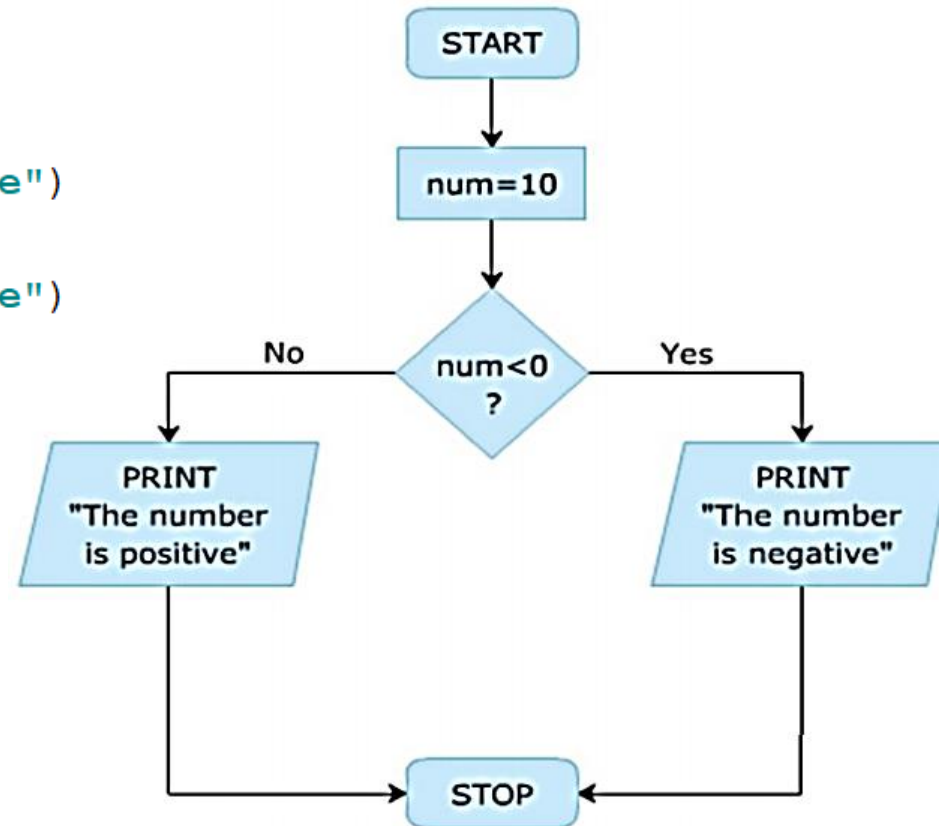


CẤU TRÚC ĐIỀU KIỆN IF – DẠNG 2

- ❑ **Ví dụ 3.1-6:** viết chương trình giải bài toán theo sơ đồ khối sau

```
num=10
if num<0:
    print("The number is negative")
else:
    print("The number is positive")
```

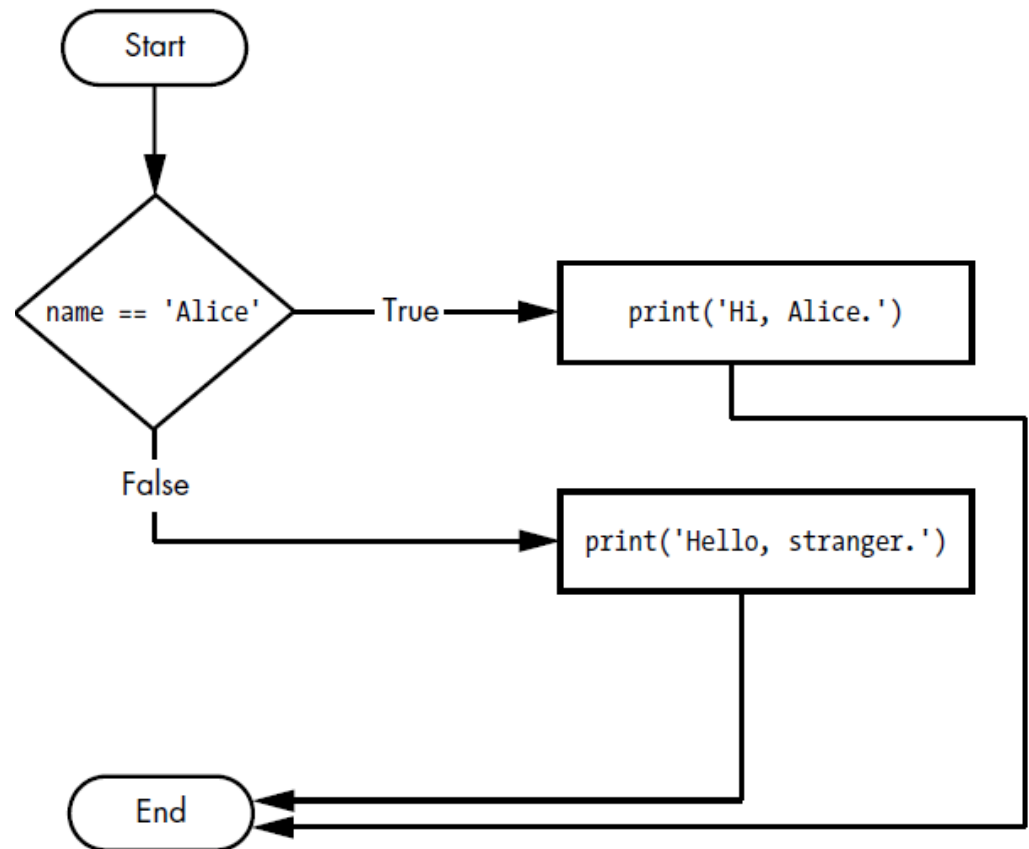
➡ The number is positive



CẤU TRÚC ĐIỀU KIỆN IF – DẠNG 2

- ❑ **Ví dụ 3.1-7:** viết chương trình giải bài toán theo sơ đồ khối sau

```
name="Alice"  
if name == 'Alice':  
    print('Hi, Alice.')  
else:  
    print('Hello, stranger.')
```



CẤU TRÚC ĐIỀU KIỆN IF – DẠNG 3

□ Cú pháp

if <Biểu thức Logic 1>:

//Lệnh hoặc khối lệnh 1

elif <Biểu thức Logic 2>:

//Lệnh hoặc khối lệnh 2

...

elif <Biểu thức Logic n-1>:

//Lệnh hoặc khối lệnh n-1

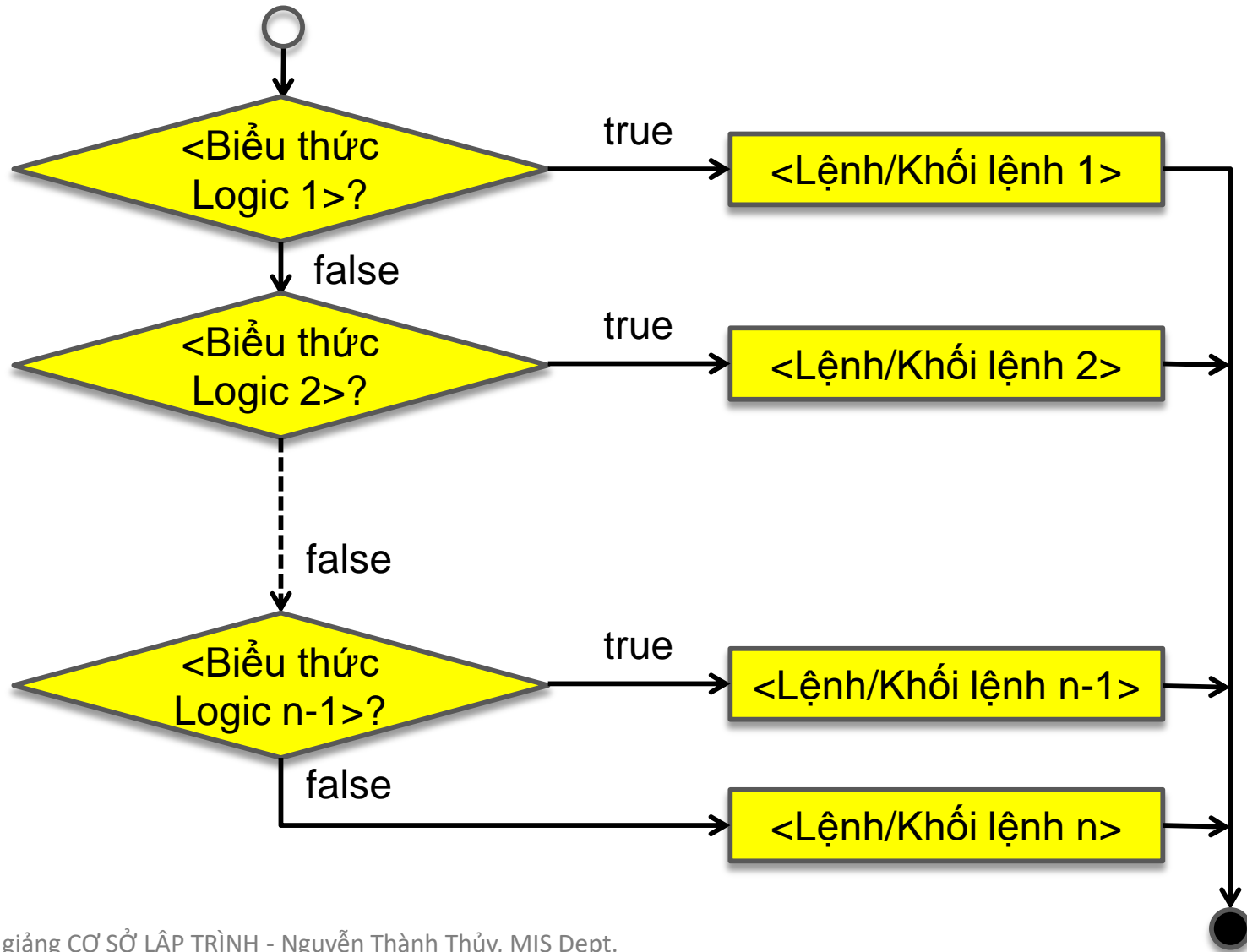
[else:

//Lệnh hoặc khối lệnh n

]

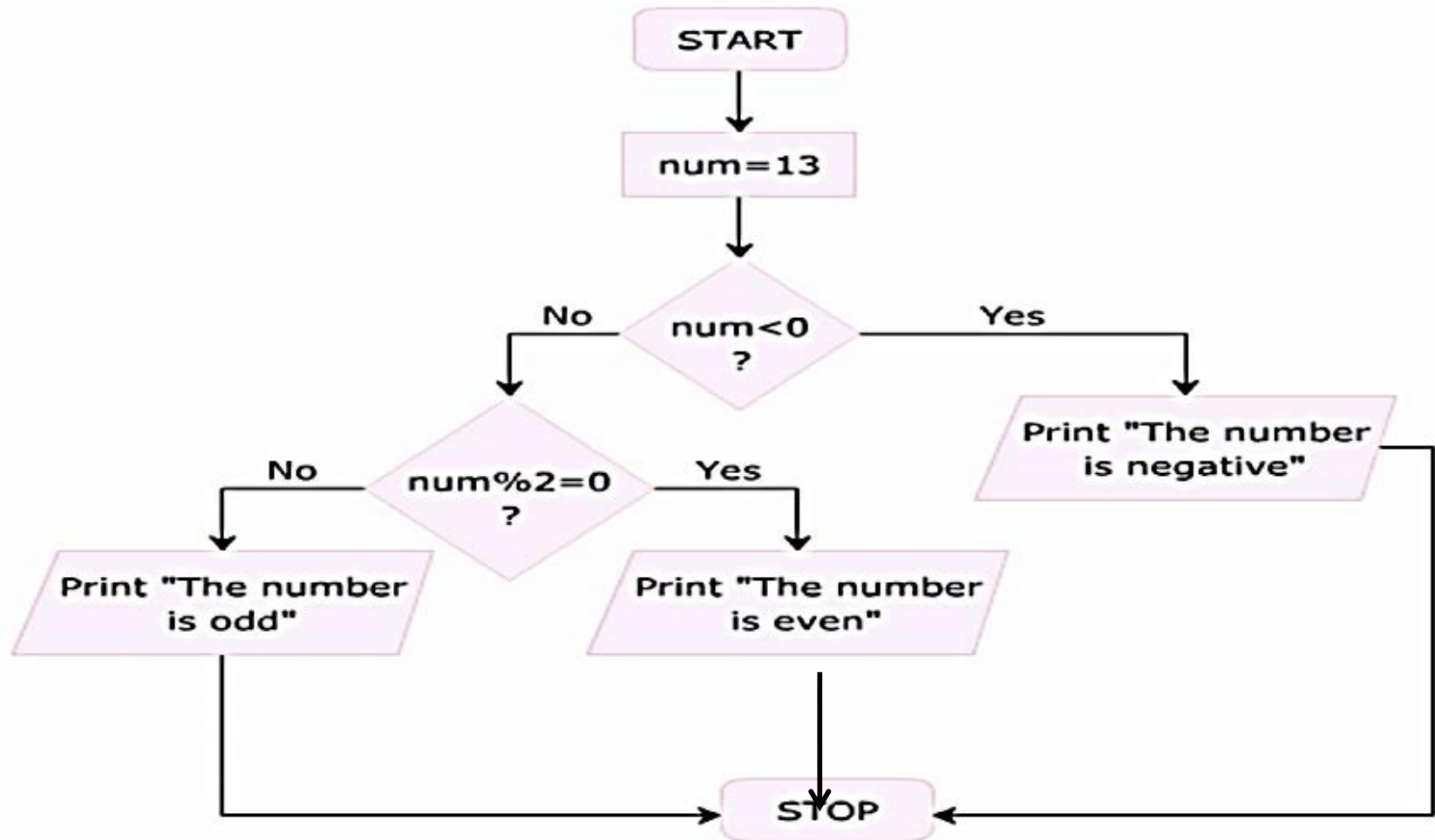
CẤU TRÚC ĐIỀU KIỆN IF – DẠNG 3

□ Cú pháp



CẤU TRÚC ĐIỀU KIỆN IF – DẠNG 3

- ❑ **Ví dụ 3.1-8:** viết chương trình giải bài toán theo sơ đồ khối sau



CẤU TRÚC ĐIỀU KIỆN IF – DẠNG 3

- ❑ **Ví dụ 3.1-9:** viết chương trình giải bài toán theo sơ đồ khối sau

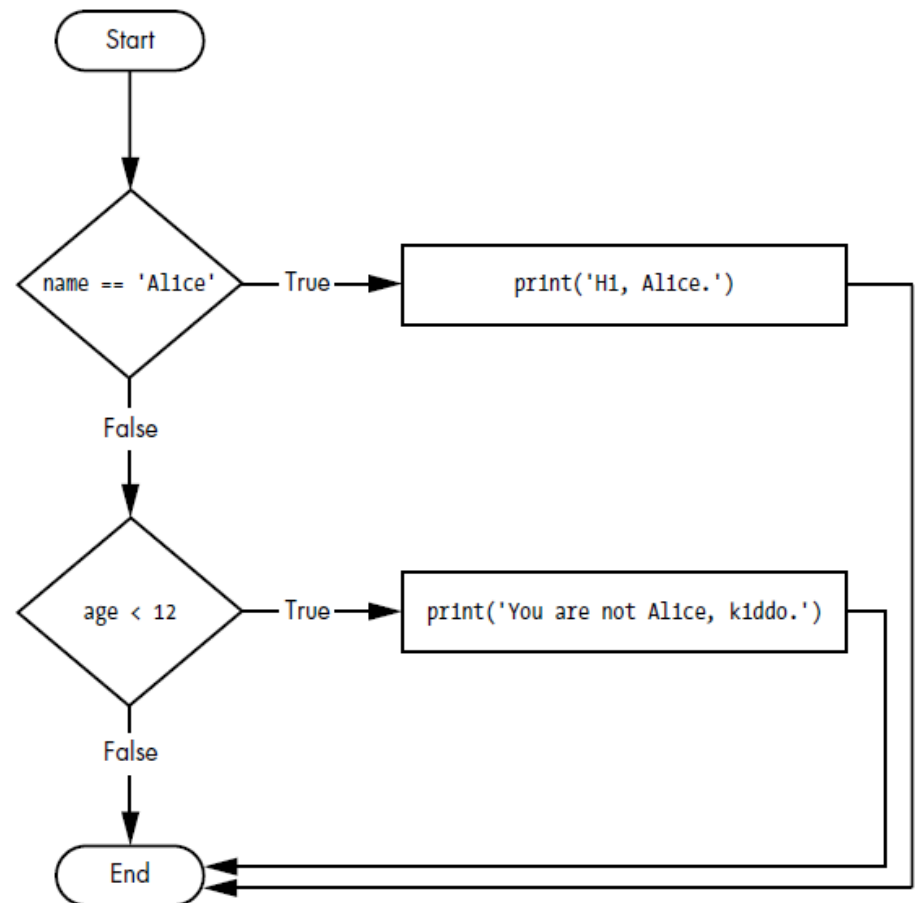
```
num=13
if num<0:
    print("The number is negative")
elif num%2==0:
    print("The number is even")
else:
    print("The number is odd")
```

 The number is odd

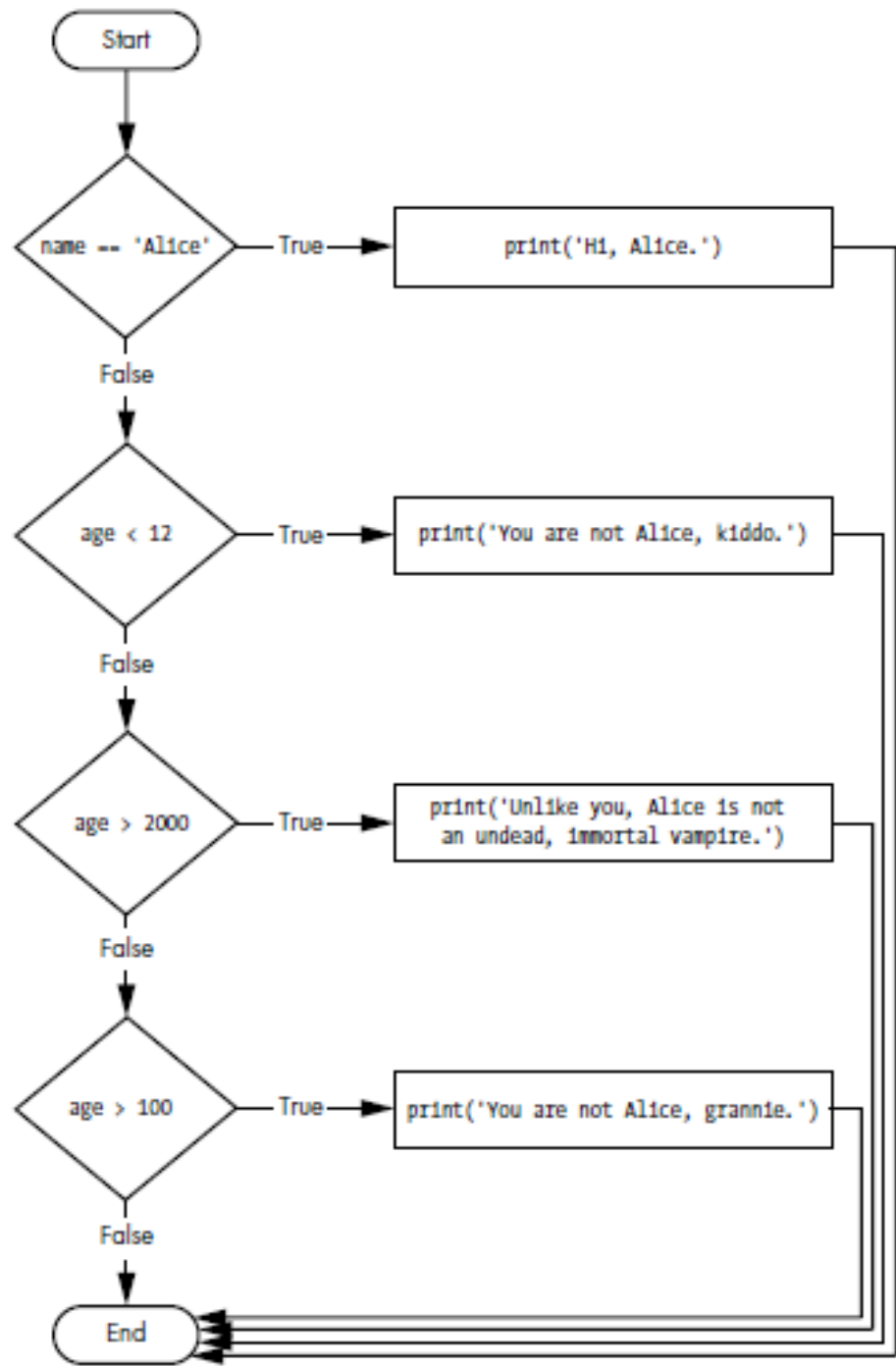
CẤU TRÚC ĐIỀU KIỆN IF – DẠNG 3

- ❑ **Ví dụ 3.1-10:** viết chương trình giải bài toán theo sơ đồ khối sau

```
name='Bob'  
age=10  
if name == 'Alice':  
    print('Hi, Alice.')  
elif age < 12:  
    print('You are not Alice, kiddo.')
```



❑ Ví dụ 3.1-11:



CẤU TRÚC ĐIỀU KIỆN IF – DẠNG 3

- ❑ **Ví dụ 3. 1-11:** viết chương trình giải bài toán theo sơ đồ khối sau

```
name='Alice'  
age=2000  
if name == 'Alice':  
    print('Hi, Alice.')elif age < 12:  
    print('You are not Alice, kiddo.')elif age > 2000:  
    print('Unlike you, Alice is not an undead, immortal vampire.')elif age > 100:  
    print('You are not Alice, grannie.')
```

SỰ LỒNG NHAU CỦA CÁC CẤU TRÚC ĐIỀU KIỆN

□ Cú pháp

if <Biểu thức Logic 1>:

...

if <Biểu thức Logic 2>:

...

if (<Biểu thức Logic 3>):

...

elif:

...

else:

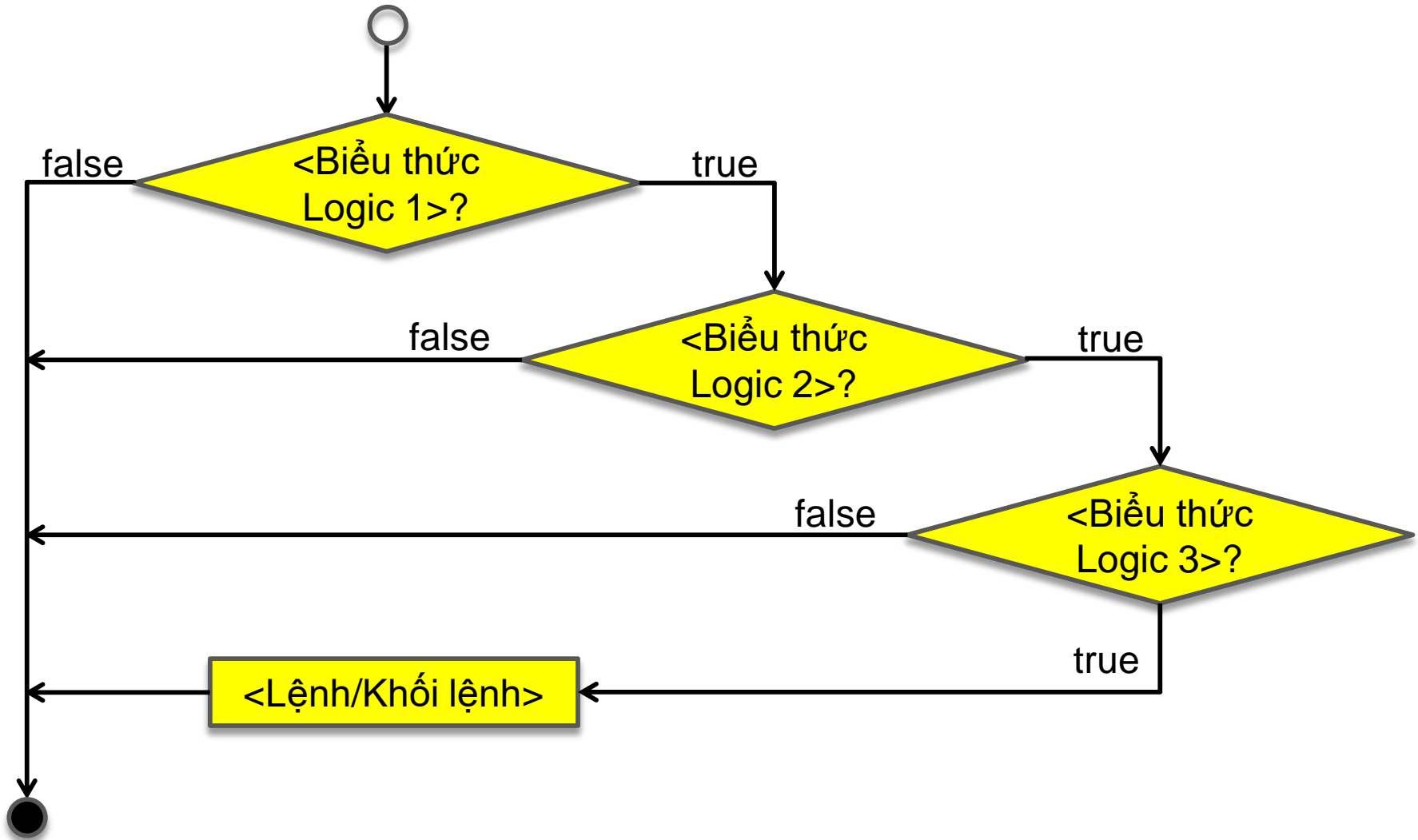
...

...

...

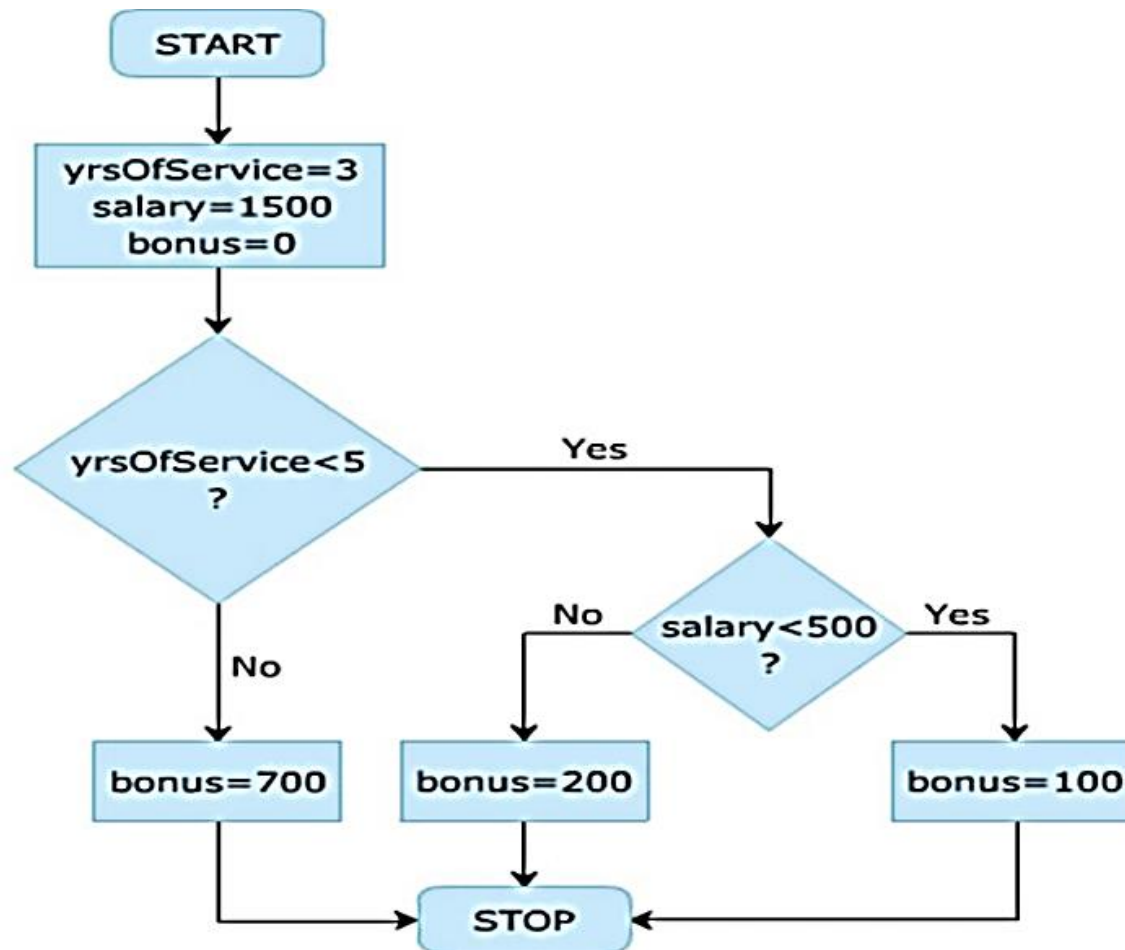
SỰ LỒNG NHAU CỦA CÁC CẤU TRÚC ĐIỀU KIỆN

□ Cú pháp



SỰ LỒNG NHAU CỦA CÁC CẤU TRÚC ĐIỀU KIỆN

❑ **Ví dụ 3.1-12:** viết chương trình giải bài toán theo sơ đồ khối sau



SỰ LỒNG NHAU CỦA CÁC CẤU TRÚC ĐIỀU KIỆN

❑ Ví dụ 3.1-12:

```
yrsOfService = 3
salary = 1500
bonus = 0
if yrsOfService < 5:
    if salary < 500:
        bonus = 100
    else:
        bonus = 200
else:
    bonus = 700

print("Bonus amount: ", bonus)
```



Bonus amount: 200

CẤU TRÚC ĐIỀU KIỆN IF – BÀI TẬP VÍ DỤ

❑ **Ví dụ 3.1-13:** Tìm số lớn nhất giữa hai số a và b

▪ Cách 1:

```
a=5
b=10
if a>b:
    print("So lon nhat la: ", a)
else:
    print("So lon nhat la: ", b)
```

➡ So lon nhat la: 10

CẤU TRÚC ĐIỀU KIỆN IF – BÀI TẬP VÍ DỤ

❑ **Ví dụ 3.1-13:** Tìm số lớn nhất giữa hai số a và b

▪ Cách 2:

```
a=5
```

```
b=10
```

```
if a>b:
```

```
    max=a
```

```
else:
```

```
    max=b
```

```
print("So lon nhat la: ", max)
```



```
So lon nhat la: 10
```

CẤU TRÚC ĐIỀU KIỆN IF – BÀI TẬP VÍ DỤ

❑ **Ví dụ 3.1-13:** Tìm số lớn nhất giữa hai số a và b

▪ Cách 3:

```
a=5
```

```
b=10
```

```
max=a
```

```
if max<b:
```

```
    max=b
```

```
print("So lon nhat la: ", max)
```



```
So lon nhat la: 10
```

CẤU TRÚC ĐIỀU KIỆN IF – BÀI TẬP VÍ DỤ

❑ Ví dụ 3.1-14: Giải phương trình bậc nhất: $ax + b = 0$

```
a=int(input("a="))
b=int(input("b="))
if a==0:
    if b==0:
        print("Phương trình có vô số nghiệm!!!")
    else:
        print("Phương trình vô nghiệm!!!")
else:
    print("Phương trình có nghiệm x=", (-b/a))
```

a=5

b=10



Phương trình có nghiệm x= -2.0

CẤU TRÚC ĐIỀU KIỆN IF – BÀI TẬP VÍ DỤ

❑ Ví dụ 3.1-15: Giải và biện luận phương trình bậc hai: $ax^2 + bx + c = 0$



Nhap a,b,c:

a=4

b=5

c=1

Phuong trinh co hai nghiem:

x1= -0.25

x2= -1.0

```
import math
print("Nhap a,b,c:")
a=int(input("a="))
b=int(input("b="))
c=int(input("c="))
d=b*b-4*a*c
if d<0:
    print("Phuong trinh vo nghiem!!!")
elif d==0:
    print("Phuong trinh co nghiem kep x1, x2=", (-b/2*a))
else:
    print("Phuong trinh co hai nghiem:\n")
    print("x1=", (-b + math.sqrt(d)) / (2*a))
    print("x2=", (-b - math.sqrt(d)) / (2 * a))
```

CẤU TRÚC ĐIỀU KIỆN IF – BÀI TẬP ÔN TẬP

Viết chương trình sử dụng cấu trúc IF:

- **Câu 3.1.** Nhập 3 số thực từ bàn phím, in lên màn hình số lớn nhất và bé nhất trong 3 số trên.

Ví dụ:

**a=5
b=4.5
c=7
SLN=7.0
SBN=4.5**

**a=4.5
b=5
c=7
SLN=7.0
SBN=4.5**

**a=5
b=7
c=4.5
SLN=7.0
SBN=4.5**

CẤU TRÚC ĐIỀU KIỆN IF – BÀI TẬP ÔN TẬP

Viết chương trình sử dụng cấu trúc IF:

- **Câu 3.2.** Một hãng máy tính có chính sách khuyến mại, cứ mua **từ 5 máy trở lên** thì giá một máy sẽ là **450\$** còn không thì giá một máy sẽ là **500\$**. Viết chương trình yêu cầu người dùng nhập vào số máy muốn mua, sau đó in ra màn hình số tiền người đó phải trả cho hãng.

Ví dụ:

So may=10
So tien=4500

So may=4
So tien=2000

So may=0
So tien=0

CẤU TRÚC ĐIỀU KIỆN IF – BÀI TẬP ÔN TẬP

Viết chương trình sử dụng cấu trúc IF:

- **Câu 3.3.** Nhập vào số KW điện tiêu thụ của một hộ gia đình, sau đó in lên màn hình số tiền mà hộ gia đình đó phải trả biết rằng cách tính tiền điện như sau:
 - Từ KW **1** đến KW thứ **100**: giá **550** đ/1KW
 - Từ KW **101** đến KW thứ **150**: giá **750** đ/1KW
 - Từ KW **151** đến KW thứ **200**: giá **950** đ/1KW
 - Từ KW **201** trở đi: giá **1350** đ/1KW
 - Thuế VAT là **10%**.

Thành tiền = Số KW tiêu thụ * Đơn giá + VAT

Ví dụ:

Tieu thu=110

Phai tra=68750.0

Tieu thu=170

Phai tra=122650.0