

## Chương 6:

# KIỂM THỬ PHẦN MỀM



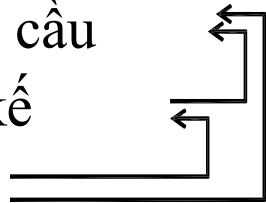
# NỘI DUNG

1. **Một số khái niệm cơ bản**
2. Tiến trình kiểm thử phần mềm
3. Các cách tiếp cận hộp trong kiểm thử p.m
4. Các mức kiểm thử phần mềm

# 1. Một số k.n cơ bản

## a. Xác minh (Verification)

- Kiểm tra sản phẩm có đúng theo bản đặc tả nó hay không (đặc tả yêu cầu, thiết kế, mã nguồn).
- Cơ sở cho hoạt động xác minh
  - Bản đặc tả yêu cầu
  - Các bản thiết kế
  - Mã nguồn
- Các hoạt động xác minh
  - **Rà soát** (*thanh tra, xét duyệt, ...*)
  - **Kiểm thử?** (*kiểm thử đơn vị, kiểm thử tích hợp, kiểm thử hệ thống*)



# 1. Một số k.n cơ bản

## b. Thẩm định (Validation)

- Kiểm tra sản phẩm có đáp ứng yêu cầu của người dùng không.
- Cơ sở cho hoạt động thẩm định
  - Bản đặc tả yêu cầu, mã nguồn
- Các hoạt động thẩm định
  - **Rà soát,**
  - **Kiểm thử chấp thuận?**

# 1. Một số k.n cơ bản

- **Xác minh và thẩm định (V&V)**
  - *=> ~ 2 hoạt động cốt yếu đảm bảo chất lượng sản phẩm, diễn ra trong suốt quá trình phát triển phần mềm.*
  - *Kiểm thử là kỹ thuật chính để thực hiện các hoạt động V&V.*
    - *Kiểm thử?*

# 1. Một số k.n cơ bản

## c. Kiểm thử

- Theo IEEE:

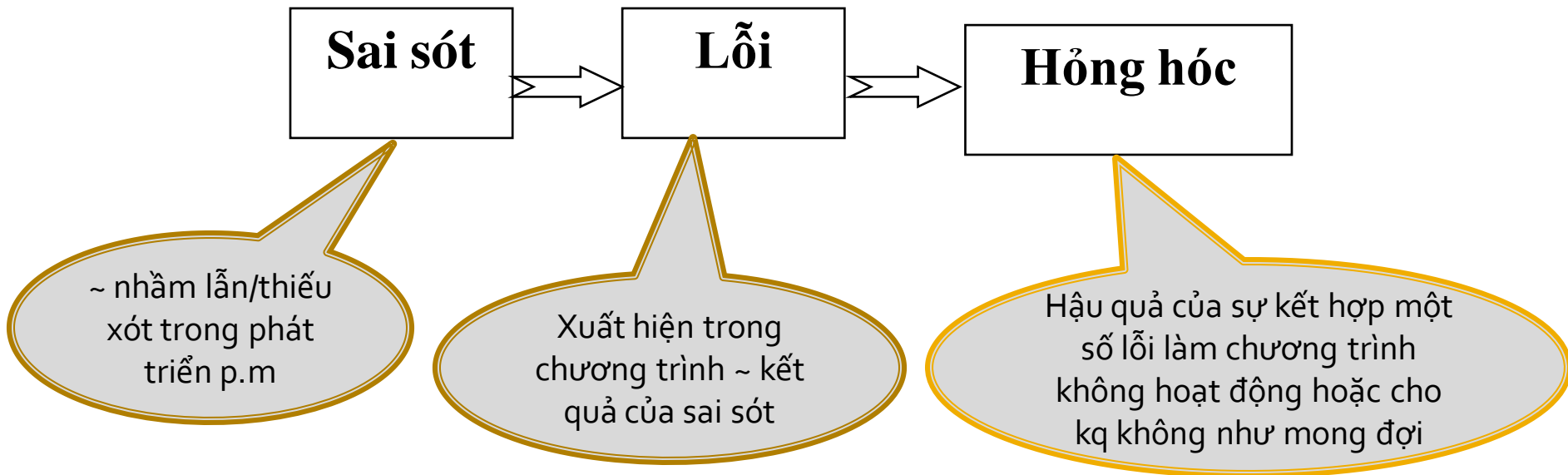
Kiểm thử là tiến trình vận hành hệ thống/thành phần của hệ thống dưới những điều kiện xác định, quan sát hoặc ghi nhận kết quả và đưa ra các đánh giá về hệ thống hoặc thành phần đó.

- Theo Myers[the art of software testing]:

- Kiểm thử là tiến trình vận hành chương trình với mục đích tìm thấy lỗi.

# 1. Một số k.n cơ bản

- Kiểm thử (phát hiện lỗi)  $\neq$  gỡ rối (định vị lỗi, xác định nguyên nhân của lỗi và sửa lỗi)
  - $\Rightarrow$  cần phân biệt:



# 1. Một số k.n cơ bản

## d. Dữ liệu thử (test data)

- Dữ liệu chạy kịch bản kiểm thử

## e. Kịch bản kiểm thử (test scenario):

- Các bước thực hiện khi tiến hành kiểm thử

## f. Phán xét kiểm thử (test oracle):

- Đánh giá kết quả kiểm thử (thủ công hoặc tự động).

## g. Ca kiểm thử (test case):

- Test case gồm:
  - Tập dữ liệu thử;
  - Điều kiện thực thi;
  - Kết quả mong đợi.



# 1. Một số k.n cơ bản

**g. Ca kiểm thử (test case):**

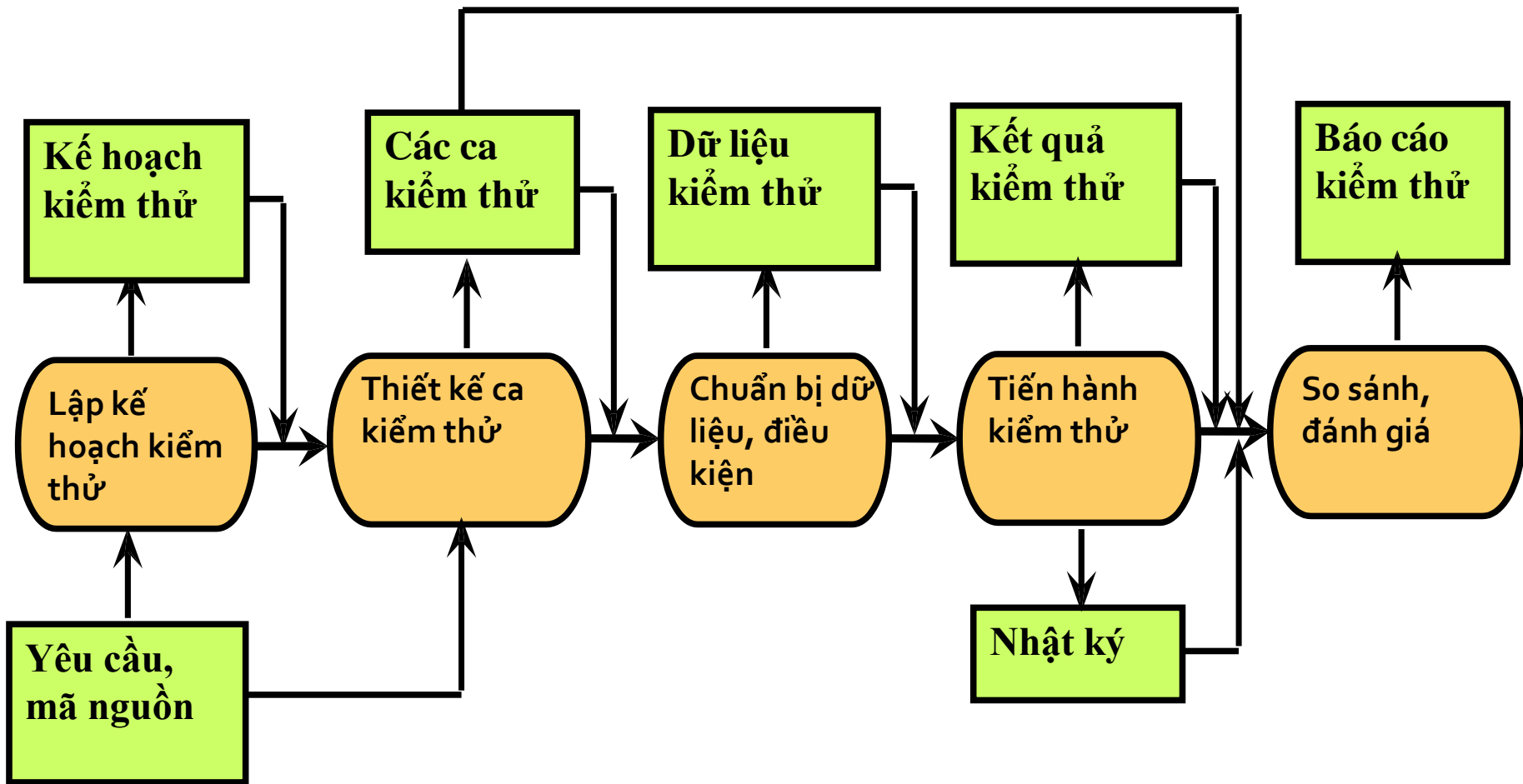
- Ví dụ:
  - Test case thẩm định trường email tại trang đăng nhập

[illegible]

# NỘI DUNG

1. Một số khái niệm cơ bản
2. **Tiến trình kiểm thử phần mềm**
3. Các cách tiếp cận hộp trong kiểm thử p.m
4. Các mức kiểm thử phần mềm

## 2. Tiến trình kiểm thử



## 2. Tiến trình kiểm thử

- Ví dụ: xét chương trình tìm số max trong dãy A gồm n số nguyên
  - \* **Thiết kế ca kiểm thử**
    - Các đầu vào:  $n: \text{Integer}; A = \{a[i] \mid a[i] \text{ thuộc } \mathbb{Z}; i=1 \rightarrow n\}$
    - Điều kiện thử: None
    - Các đầu ra:  $\text{max} = \{a[i] \mid a[i] \geq a[j]; i=1 \rightarrow n; j=1 \rightarrow n\}$
  - \* **Chuẩn bị dữ liệu thử**
    - TD1 =  $\{n=-5; a=\{\};\}$
    - TD2 =  $\{n=5; a=\{7,8,3,6,25\};\}$
  - \* **Chạy chương trình:** với TD2
    - Max = 8 ;
  - \* **Phán xét:** Chương trình chạy sai

# NỘI DUNG

1. Một số khái niệm cơ bản
2. Tiến trình kiểm thử phần mềm
3. **Các cách tiếp cận hộp trong kiểm thử p.m**
4. Các mức kiểm thử phần mềm

### 3. Các cách tiếp cận kiểm thử p.m

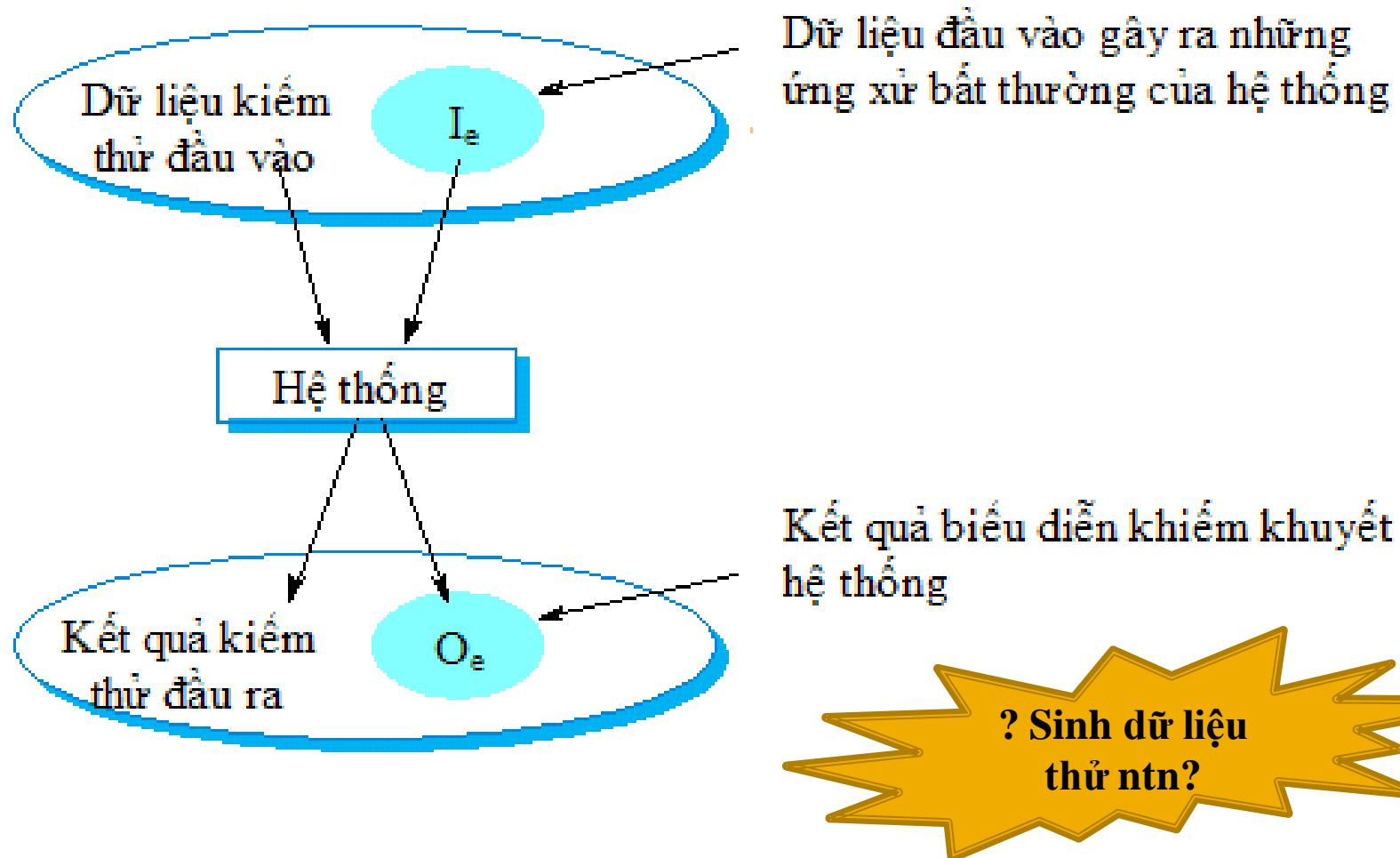
---

- a. Kiểm thử hộp đen
- b. Kiểm thử hộp trắng
- c. Kiểm thử hộp xám

# a. Kiểm thử hộp đen

- Xem đối tượng kiểm thử như một hộp đen,
  - thông qua giao diện của hộp đen để đưa dữ liệu vào và nhận thông tin ra.
- Áp dụng để kiểm thử các chức năng hệ thống
- Đối tượng kiểm thử:
  - Hệ thống/hệ con/chương trình/modun

# a. Kiểm thử hộp đen



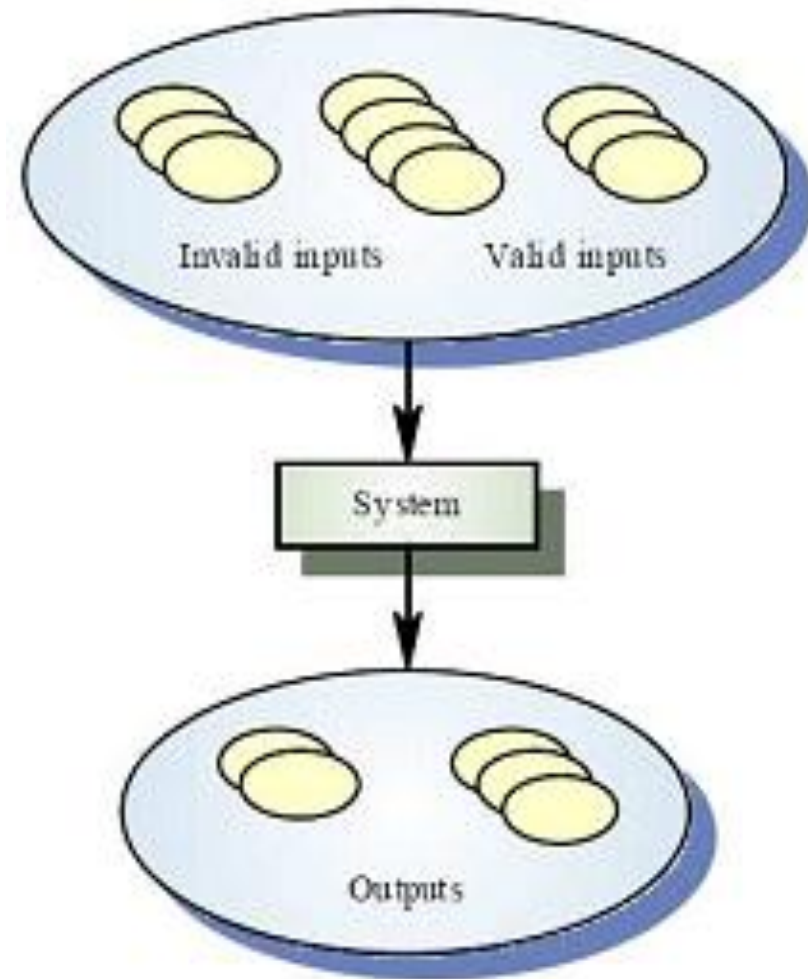


# a. Kiểm thử hộp đen

- Một số kỹ thuật kiểm thử
  - i. Kỹ thuật Phân hoạch lớp tương đương (\*)
  - ii. Kỹ thuật phân tích giá trị biên (\*)
  - iii. Kỹ thuật đồ thị nhân quả (SV tự tìm hiểu)

# i. Phân hoạch lớp tương đương

- Ý tưởng:
  - Chia tập dữ liệu thành một số lớp dữ liệu và chọn các phần tử đại diện của mỗi lớp ra kiểm thử.
  - 2 loại lớp:
    - Một số lớp chứa các giá trị đầu vào hợp lệ;
    - Một số lớp gồm các giá trị đầu vào không hợp lệ



# i. Phân hoạch lớp tương đương

- Các bước tiến hành:
  - Bước 1: với mỗi dữ liệu vào, xác định các lớp tương đương từ miền dữ liệu gồm:
    - *Các lớp dữ liệu vào hợp lệ,*
    - *Các lớp dữ liệu vào không hợp lệ.*
  - Bước 2: chọn dữ liệu đại diện trong mỗi lớp; kết hợp các dữ liệu này lại bởi tích Đề - các đề sinh ra các bộ dữ liệu thử.

# i. Phân hoạch lớp tương đương

## ■ Nguyên tắc phân hoạch

- Nếu dữ liệu thuộc một khoảng, xây dựng:
  - *Một lớp các giá trị lớn hơn giá trị lớn nhất của khoảng*
  - *Một lớp các giá trị nhỏ hơn giá trị nhỏ nhất của khoảng*
  - *Lớp gồm  $N$  giá trị hợp lệ.*
- Nếu dữ liệu vào là tập hợp các giá trị, xây dựng:
  - *1 lớp với tập rỗng*
  - *1 lớp với quá nhiều các giá trị*
  - *Lớp gồm  $N$  giá trị hợp lệ.*
- Nếu dữ liệu vào là điều kiện ràng buộc, xây dựng:
  - *1 lớp chứa các ràng buộc được thỏa mãn*
  - *1 lớp với các ràng buộc không được thỏa mãn.*

# i. Phân hoạch lớp tương đương

## ■ Ví dụ: xét mô đun thống kê kết quả thi

### ■ Dữ liệu vào: điểm 3 môn

- D1 - điểm môn 1;
- D2 - điểm môn 2;
- D3 - điểm môn 3.

*Trong đó:  $D1, D2 \in [0, 10]$ ;  $D3 \in [0, 100]$*

### ■ Kết quả thống kê chia làm 3 loại

- "Dữ liệu sai";
- "Đỗ";
- "Trượt".

### ■ Kết quả mong muốn:

- "Dữ liệu sai" nếu:  $D1, D2 \notin [0, 10]$ ;  $D3 \notin [0, 100]$
- "Đỗ" nếu:  $D1, D2 \in [5, 10]$ ;  $D3 \in [50, 100]$
- "Trượt" nếu:  $D1, D2 \in [0, 5)$ ;  $D3 \in [0, 50)$

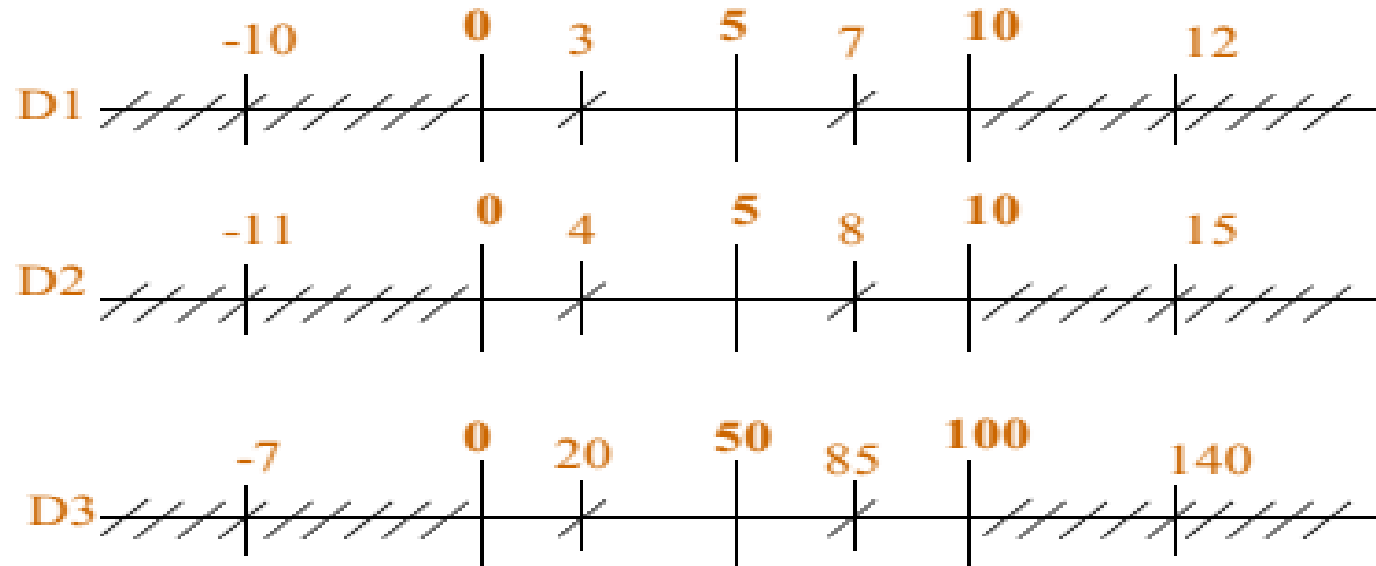
# i. Phân hoạch lớp tương đương

- Bảng liệt kê các lớp tương đương

Điều kiện vào/ra	Các lớp tương đương hợp lệ	Các lớp tương đương không hợp lệ
Điểm môn 1 (D1)	$0 \leq D1 < 5 ; 5 \leq D1 \leq 10$	$D1 < 0 ; D1 > 10$
Điểm môn 2 (D2)	$0 \leq D2 < 5 ; 5 \leq D2 \leq 10$	$D2 < 0 ; D2 > 10$
Điểm môn 3 (D3)	$0 \leq D3 < 50 ; 50 \leq D3 \leq 100$	$D3 < 0 ; D3 > 100$

# i. Phân hoạch lớp tương đương

- Giả sử mỗi phân hoạch (lớp tương đương hợp lệ và không hợp lệ) ta lấy 01 giá trị bất kỳ:



# i. Phân hoạch lớp tương đương

Xác định các test case:

## ❖ Phân lớp tương mạnh

- Mỗi test case là một phần tử của tích đề các phân hoạch con:
  - D1(-10, 3, 7, 12)
  - D2(-11, 4, 8, 15)
  - D2(-7, 20, 85, 140)
- $\Rightarrow$  *số lượng test case =  $4^3=64$  test case*



**Bảng 2.3. Danh sách ca kiểm thử được sinh theo phân lớp tương đương mạn**

TC_ID	Đầu vào			Đầu ra mong muốn
	D1	D2	D3	
TC_01	-10	-11	-7	Dữ liệu sai
TC_02	-10	-11	20	Dữ liệu sai
TC_03	-10	-11	85	Dữ liệu sai
TC_04	-10	-11	140	Dữ liệu sai
TC_05	-10	4	-7	Dữ liệu sai
TC_06	-10	4	20	Dữ liệu sai
TC_07	-10	4	85	Dữ liệu sai
TC_17	3	-11	-7	Dữ liệu sai
TC_18	3	-11	20	Dữ liệu sai
TC_19	3	-11	85	Dữ liệu sai
TC_20	3	-11	140	Dữ liệu sai
TC_21	3	4	-7	Dữ liệu sai
TC_22	3	4	20	Trượt
TC_23	3	4	85	Trượt
...	...	...	...	...
<b>TC_43</b>	<b>7</b>	<b>8</b>	<b>85</b>	<b>Đỗ</b>
...	...	...	...	...
TC_59	12	8	85	Dữ liệu sai
TC_60	12	8	140	Dữ liệu sai
TC_62	12	15	20	Dữ liệu sai
TC_63	12	15	85	Dữ liệu sai
TC_64	12	15	140	Dữ liệu sai

# i. Phân hoạch lớp tương đương

- ❖ Phân lớp tương đương mạnh
  - Gần như bắt được mọi lỗi
  - Tuy nhiên:
    - Số lượng test case rất nhiều, khi có nhiều đầu vào
    - Chi phí test lớn.

# i. Phân hoạch lớp tương đương

## ❖ Phân lớp tương đương yếu

- Việc sinh các test case phải đảm bảo mỗi lớp con được kiểm tra ít nhất một lần.
- VD: 04 test case được sinh ra:

**Bảng 2.4. Danh sách các ca kiểm thử sinh ra theo phân lớp tương đương yếu**

TC_ID	Đầu vào (Input)			Đầu ra mong muốn
	D1	D2	D3	
TC_01	-10	-11	-7	Dữ liệu sai
TC_02	3	4	20	Trượt
TC_03	7	8	85	Đỗ
TC_04	12	15	140	Dữ liệu sai

# i. Phân hoạch lớp tương đương

- Phân lớp tương đương yếu
  - Số lượng test case ít:
    - Không thể khẳng định mọi lỗi nhập liệu đã được bắt hết.
    - => phân lớp tương đương truyền thống

# i. Phân hoạch lớp tương đương

- ❖ Phân lớp tương đương truyền thống
  - Test case cho trường hợp đúng:
    - *Lấy các biến đầu vào nằm trong miền hợp lệ.*
  - Test case cho trường hợp sai:
    - *Lấy các biến đầu vào nằm trong miền không hợp lệ.*
  - Số lượng test case:
    - = Số lượng test case trong phân lớp mạnh-số lượng test case có trên 2 đầu vào sai
    - = Số lượng test case có dưới 1 đầu vào sai
    - =  $3^4 - 3^3 + 1 = 3^3 + 1$
  - $\Rightarrow$  Các test case được sinh ra cho ví dụ trên

TC ID	Đầu vào (Input)			Đầu ra mong muốn
	D1	D2	D3	
TC_01	-10	4	20	Dữ liệu sai
TC_02	-10	4	85	Dữ liệu sai
TC_03	-10	8	20	Dữ liệu sai
TC_04	-10	8	85	Dữ liệu sai
TC_05	3	-11	20	Dữ liệu sai
TC_06	3	-11	85	Dữ liệu sai
TC_07	3	4	-7	Dữ liệu sai
TC_08	3	4	140	Dữ liệu sai
TC_09	3	8	-7	Dữ liệu sai
TC_10	3	8	85	Trượt
TC_11	3	8	140	Dữ liệu sai
TC_12	3	15	20	Dữ liệu sai
TC_13	3	15	85	Dữ liệu sai
TC_14	7	-11	20	Dữ liệu sai
TC_15	7	-11	85	Dữ liệu sai
TC_16	7	4	-7	Dữ liệu sai
TC_17	7	4	85	Trượt
TC_18	7	4	140	Dữ liệu sai
TC_19	7	8	-7	Dữ liệu sai
TC_20	7	8	20	Trượt
TC_21	7	8	85	Đỗ
TC_22	7	8	140	Dữ liệu sai
TC_23	7	15	20	Dữ liệu sai
TC_24	7	15	85	Dữ liệu sai
TC_25	12	4	20	Dữ liệu sai
TC_26	12	4	85	Dữ liệu sai
TC_27	12	8	20	Dữ liệu sai
TC_28	12	8	85	Dữ liệu sai

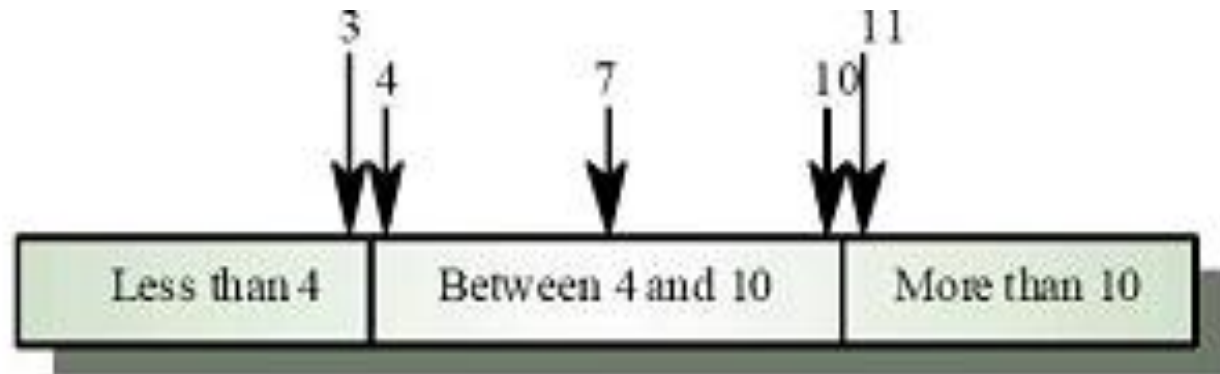
## ii. Phân tích giá trị biên

### ■ Ý tưởng:

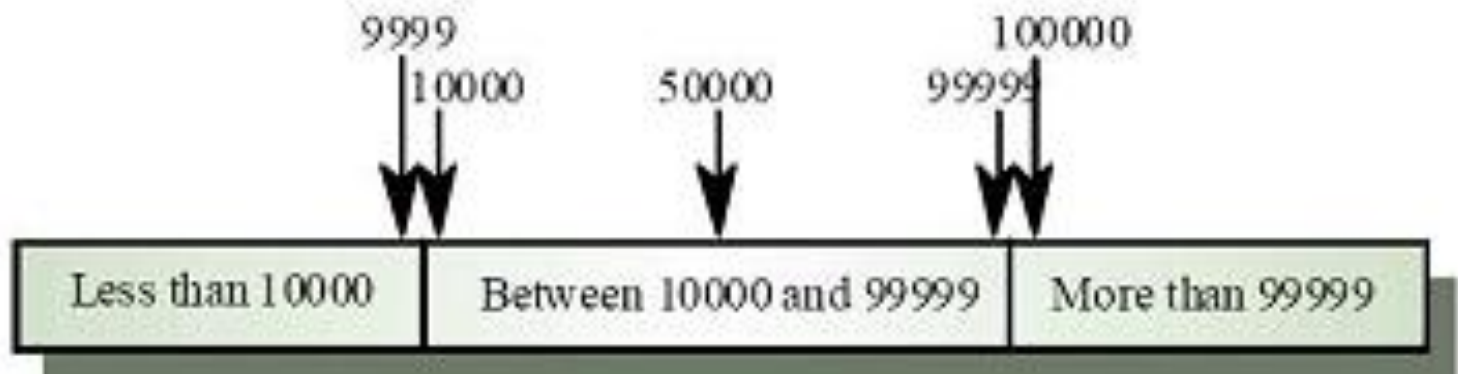
- Thường lỗi xuất hiện gần miền biên của các giá trị.
  - $\Rightarrow$  Nguyên tắc chọn dữ liệu thử:
    - Nếu dữ liệu vào thuộc một khoảng, chọn:
      - Hai giá trị biên
      - Bốn giá trị = 2 giá trị biên  $\pm$  sai số nhỏ nhất
    - Nếu dữ liệu vào thuộc danh sách các giá trị, chọn:
      - Phần tử thứ nhất, Phần tử thứ 2,
      - Phần tử kế cuối, Phần tử cuối.
    - Nếu dữ liệu vào là điều kiện ràng buộc số giá trị, chọn:
      - Số giá trị tối thiểu, Số giá trị tối đa, Một số giá trị không hợp lệ
    - Ngoài ra, chúng ta cần tự vận dụng khả năng thực tế để chọn các giá trị biên cần kiểm thử

## ii. Phân tích giá trị biên

- Ví dụ:



Number of input values





## **b. Kiểm thử hộp trắng**

**(white – box testing)**

- Kiểm thử mã nguồn
- Dữ liệu thử được sinh ra từ mã nguồn
- Thường phát hiện các lỗi liên quan đến lập trình

## **b. Kiểm thử hộp trắng**

**(white – box testing)**

- Một số kỹ thuật kiểm thử hộp trắng
  - **Kiểm thử dựa trên đồ thị luồng điều khiển (\*)**
  - Kiểm thử dựa trên luồng dữ liệu
  - Kiểm thử đột biến

## b. Kiểm thử hộp trắng

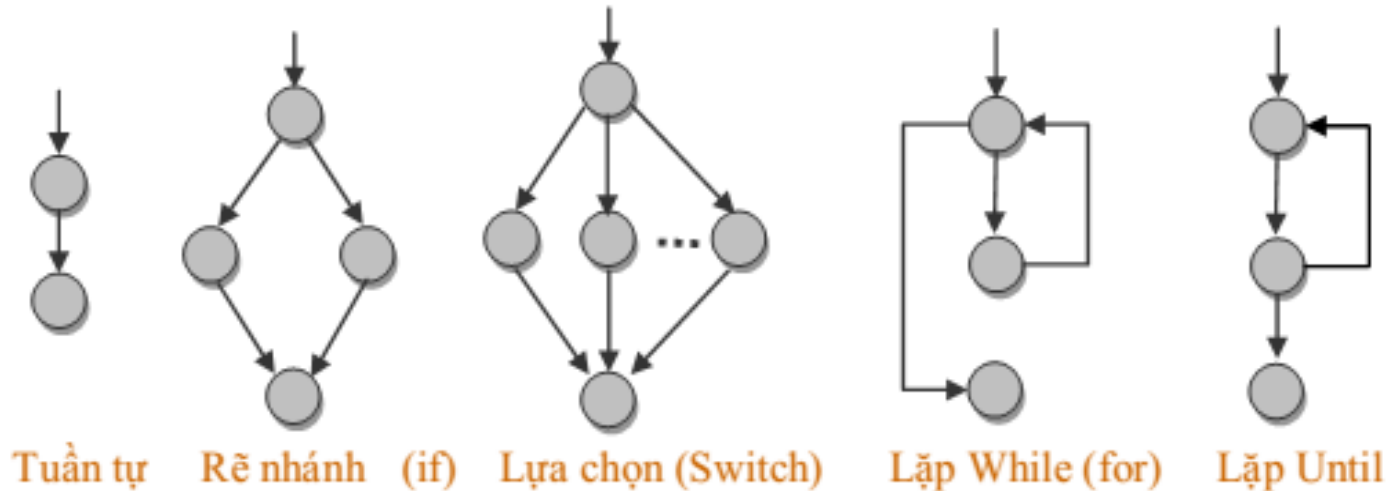
(white – box testing)

- Kiểm thử dựa trên đồ thị luồng điều khiển:
  - Đồ thị luồng/dòng điều khiển
    - ~ Đồ thị có hướng biểu diễn một chương trình/mô đun.
      - Đỉnh của đồ thị biểu diễn 1 lệnh, một khối lệnh.
      - Cung của đồ thị biểu diễn một rẽ nhánh,
      - Một *đỉnh vào* và một *đỉnh ra* được thêm vào để biểu diễn điểm vào, điểm ra của chương trình.
  - Lộ trình:
    - Đường đi từ đỉnh vào đến đỉnh ra.

# b. Kiểm thử hộp trắng

(white – box testing)

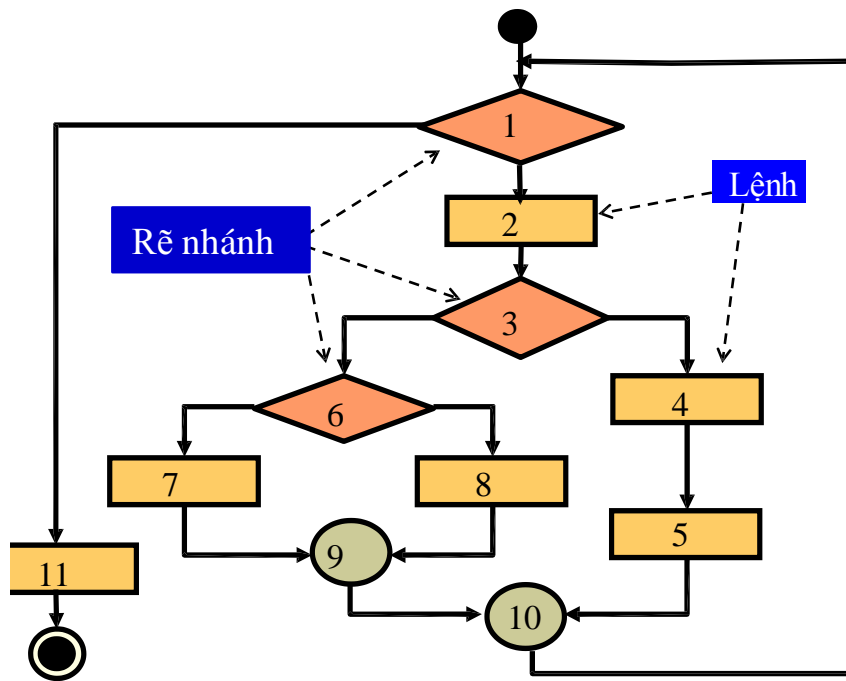
- Kiểm thử dựa trên đồ thị luồng điều khiển:
  - Đồ thị luồng điều khiển
    - Một số cấu trúc điều khiển phổ biến



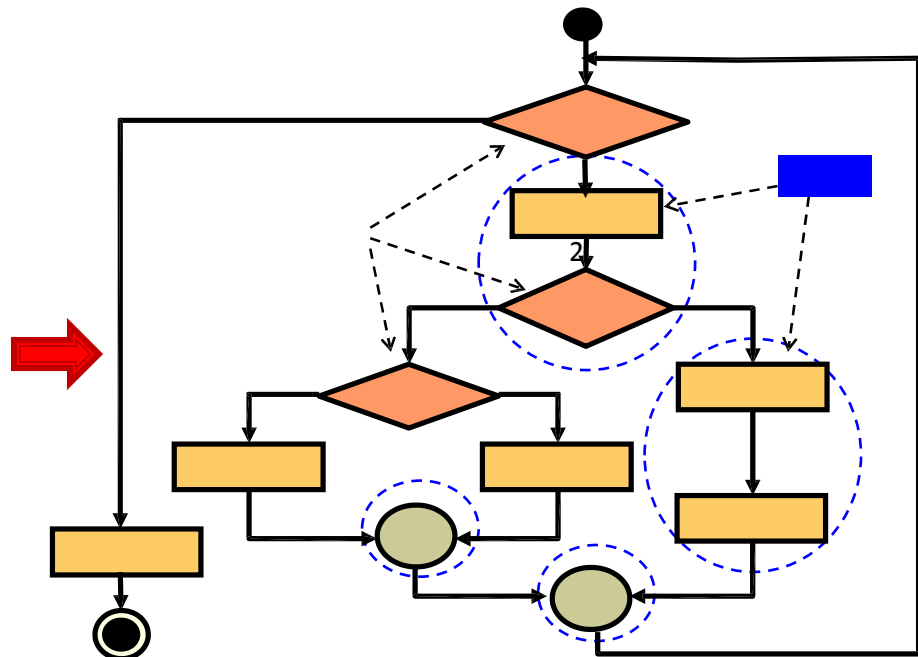
## b. Kiểm thử hộp trắng

(white – box testing)

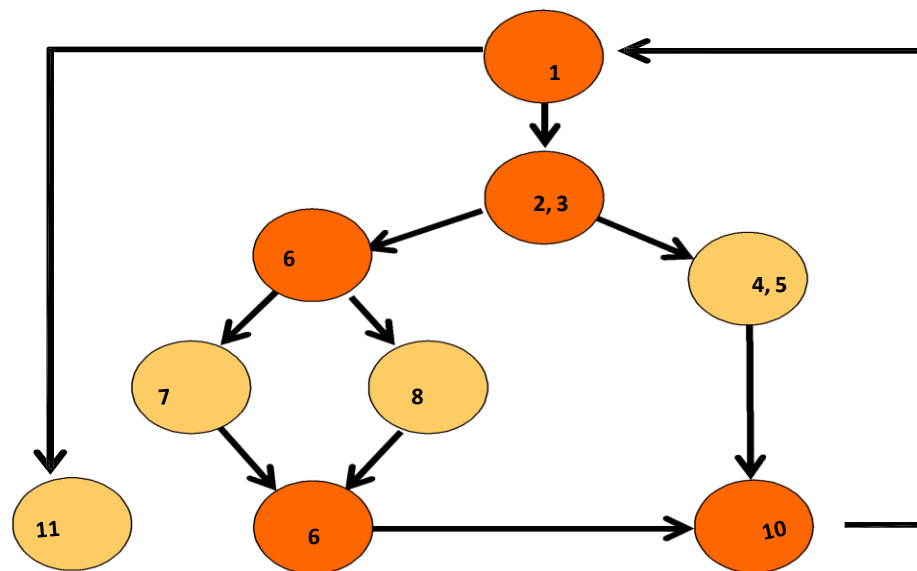
- Kiểm thử dựa trên đồ thị luồng điều khiển:
  - Đồ thị luồng điều khiển
    - Xây dựng đồ thị luồng đk từ chương trình
    - Ví dụ:
      - Xét hình (dưới):
        - (a) chương trình;
        - (b) cách gộp các lệnh/đỉnh;
        - (c) đồ thị luồng đk tương ứng.



(a)



(b)



(c)

## **b. Kiểm thử hộp trắng**

**(white – box testing)**

- Kiểm thử dựa trên đồ thị luồng điều khiển:
  - Thiết kế dữ liệu thử thỏa mãn các tiêu chuẩn phủ
    - Các tiêu chuẩn phủ ĐTLĐK:
      - i. **Phủ mọi đỉnh/lệnh**
      - ii. **Phủ mọi cung**
      - iii. **Phủ mọi quyết định**
      - iv. **Phủ mọi lộ trình**

## i. Phủ mọi đỉnh/lệnh

- Mỗi lệnh được thực thi ít nhất một lần (mỗi đỉnh của đồ thị được phủ ít nhất một lần).
  - Là tiêu chuẩn phủ tối thiểu.
- Ví dụ 1:
  - Xét hàm dưới đây



# i. Phủ mọi đỉnh/lệnh

*Function*  $sum(x, y: integer):integer;$

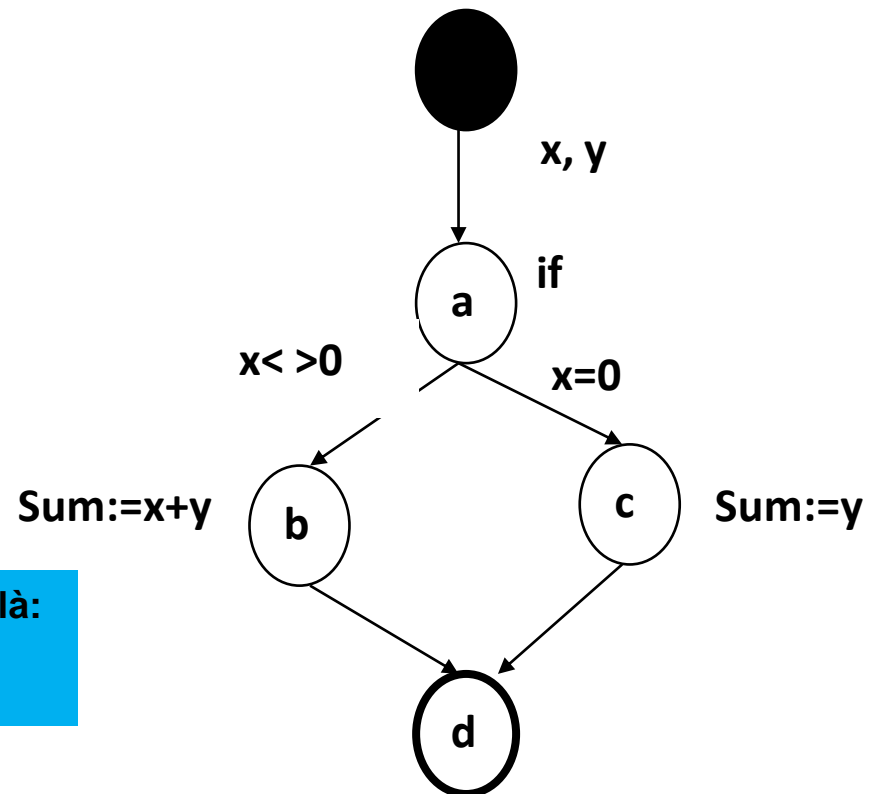
*Begin*

*If*  $(x=0)$  *then*  $sum:=y$

*Else*  $sum:=x+y;$

*End;*

Đồ thị luồng điều khiển tương ứng:



Bộ dữ liệu thử thỏa mãn tiêu chuẩn phủ là:

TD1:  $\{x=5, y=8\}$

TD2:  $\{x=0, y=0\}$

# i. Phủ mọi đỉnh/lệnh

## ■ Ví dụ 2: Xét đoạn mã sau

.....

*Read(x);*

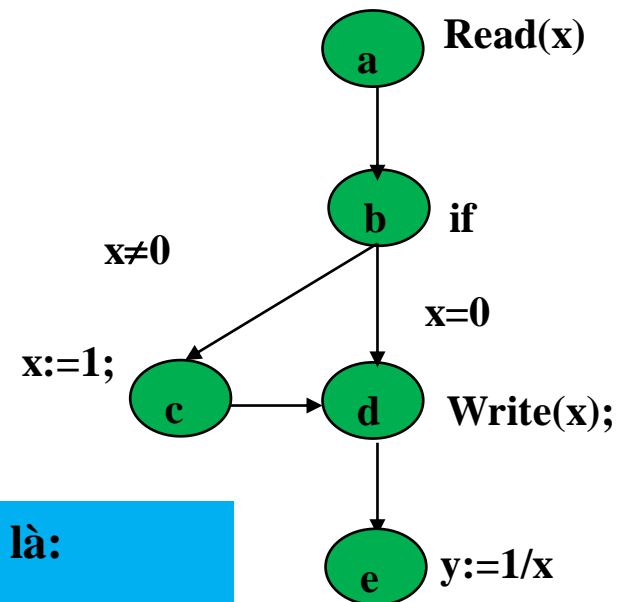
*If (x<>0) then x:=1;*

*Write(x);*

*y:=1/x;*

.....

Đồ thị luồng điều khiển biểu diễn  
đoạn mã là:



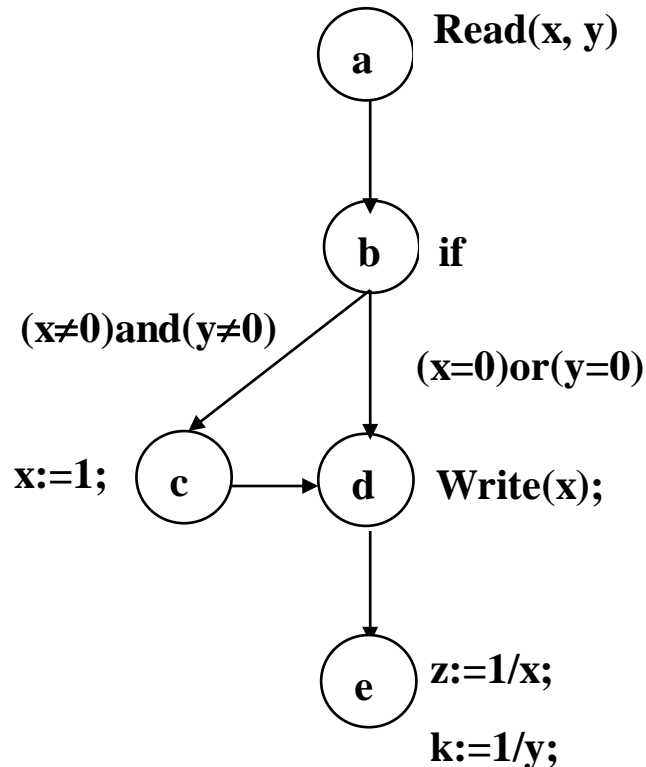
Bộ dữ liệu thử thỏa mãn tiêu chuẩn phủ là:  
TD1: {x=5}, không phát hiện lỗi chia cho 0

## ii. Phủ mọi cung

- Tiêu chuẩn phủ tất cả các cung thỏa mãn khi:
  - Tiêu chuẩn phủ tất cả các đỉnh thỏa mãn,
  - đồng thời mỗi cung phải được phủ ít nhất một lần.
- Xét các ví dụ:
  - Ví dụ 3: Xét đồ thị luồng điều khiển ở ví dụ 2 kể trên
    - Bộ dữ liệu thử thỏa mãn tiêu chuẩn phủ mọi cung:
      - TD1:  $\{x=6\}$
      - TD2:  $\{x=0\}$
    - Khi chạy bộ dữ liệu thử này, ta phát hiện được lỗi chia cho 0 gây ra bởi câu lệnh e.

## ii. Phủ mọi cung

### ■ Ví dụ 4: Xét ĐTLĐK sau



**Bộ dữ liệu thử thỏa mãn tiêu chuẩn phủ mọi cung:**

TD1:  $\{x=5, y=6\}$

TD2:  $\{x=0, y=6\}$

Không phát hiện được lỗi chia cho 0 gây ra bởi câu lệnh  $k := 1/y$

### iii. Phủ mọi quyết định

- Tiêu chuẩn phủ tất cả các quyết định thỏa mãn khi:
  - Tiêu chuẩn phủ tất cả các cung thỏa mãn, đồng thời
  - Mọi quyết định đều được phủ (tức là mỗi biểu thức con của biểu thức điều kiện phải được phủ với mọi giá trị đúng, sai.
- Quyết định?

### iii. Phủ mọi quyết định

- Ví dụ: xét biểu thức logic:  $a \text{ AND } b$

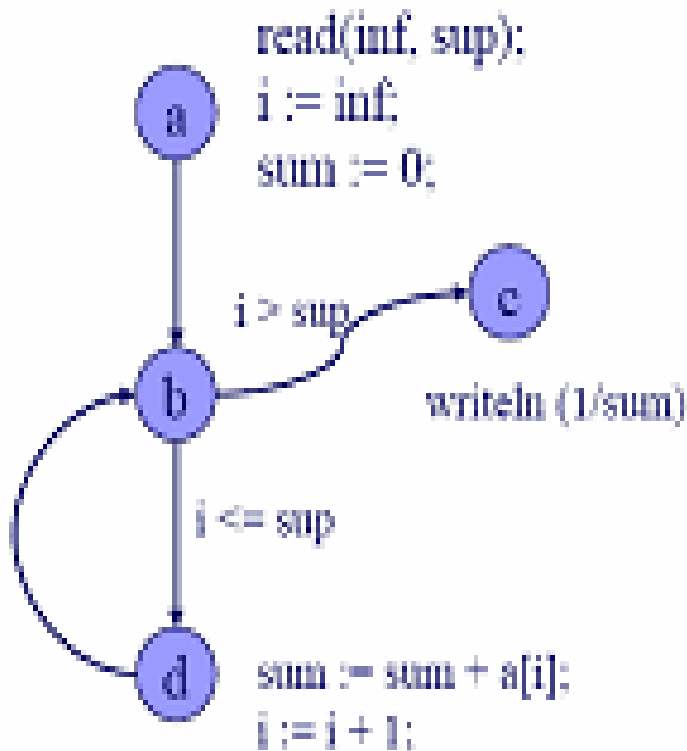
Các quyết định		
$a$	$b$	$a \text{ AND } b$
true	True	True
true	False	False
False	True	False
False	False	False

### iii. Phủ mọi quyết định

- Ví dụ 5: Xét đồ thị luồng điều khiển tại ví dụ 4 kể trên.
  - Bộ dữ liệu thử thỏa mãn tiêu chuẩn phủ mọi quyết định là:  
 $TD1: \{x=0, y=0\}$   
 $TD2: \{x=y=5\}$   
*=> Khi chạy bộ dữ liệu thử này, sẽ phát hiện ra mọi lỗi chia cho 0*

### iii. Phủ mọi quyết định

- Ví dụ 6: Xét đồ thị luồng điều khiển sau:



Bộ dữ liệu thử thỏa mãn tiêu chuẩn phủ mọi quyết định:

***TD: {inf= 1, sup= 5 }***

=> không phát hiện được lỗi chia 0  
gây ra bởi lệnh *writeln(1/sum)* đặt tại  
đỉnh c



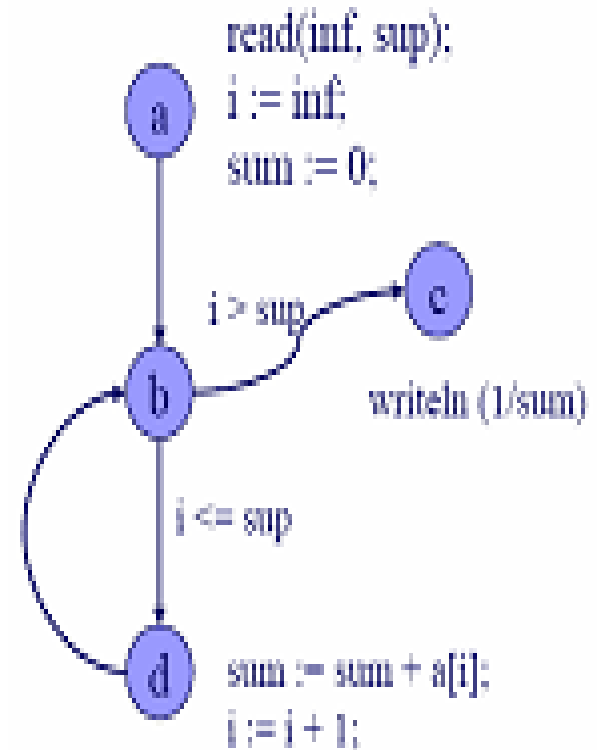
## iv. Phủ mọi lộ trình

- Tiêu chuẩn phủ tất cả các lộ trình thỏa mãn khi:
  - Tiêu chuẩn phủ mọi quyết định thỏa mãn, đồng thời
  - Mỗi lộ trình phải được phủ ít nhất một lần.
- Vấn đề:
  - Khối lượng lộ trình là rất lớn
    - Nhiều vòng lặp, vòng lặp vô hạn
  - => Khắc phục:
    - Chỉ thực hiện một số lần lặp nhất định, hoặc
    - Chỉ phủ hai loại lộ trình:
      - Lộ trình vượt qua vòng lặp mà không thực hiện vòng lặp lần nào
      - Các lộ trình đi qua vòng lặp một số lần

### iii. Phủ mọi quyết định

- Ví dụ 7: Xét đồ thị luồng điều khiển ở ví dụ 6
  - Bộ dữ liệu thử thỏa mãn tiêu chuẩn phủ là:
    - $TD1: \{inf=6; sup=1\}$
    - $TD2: \{inf=1, sup=3\}$

$\Rightarrow$  Lỗi chia cho 0 tại c được phát hiện



# NỘI DUNG

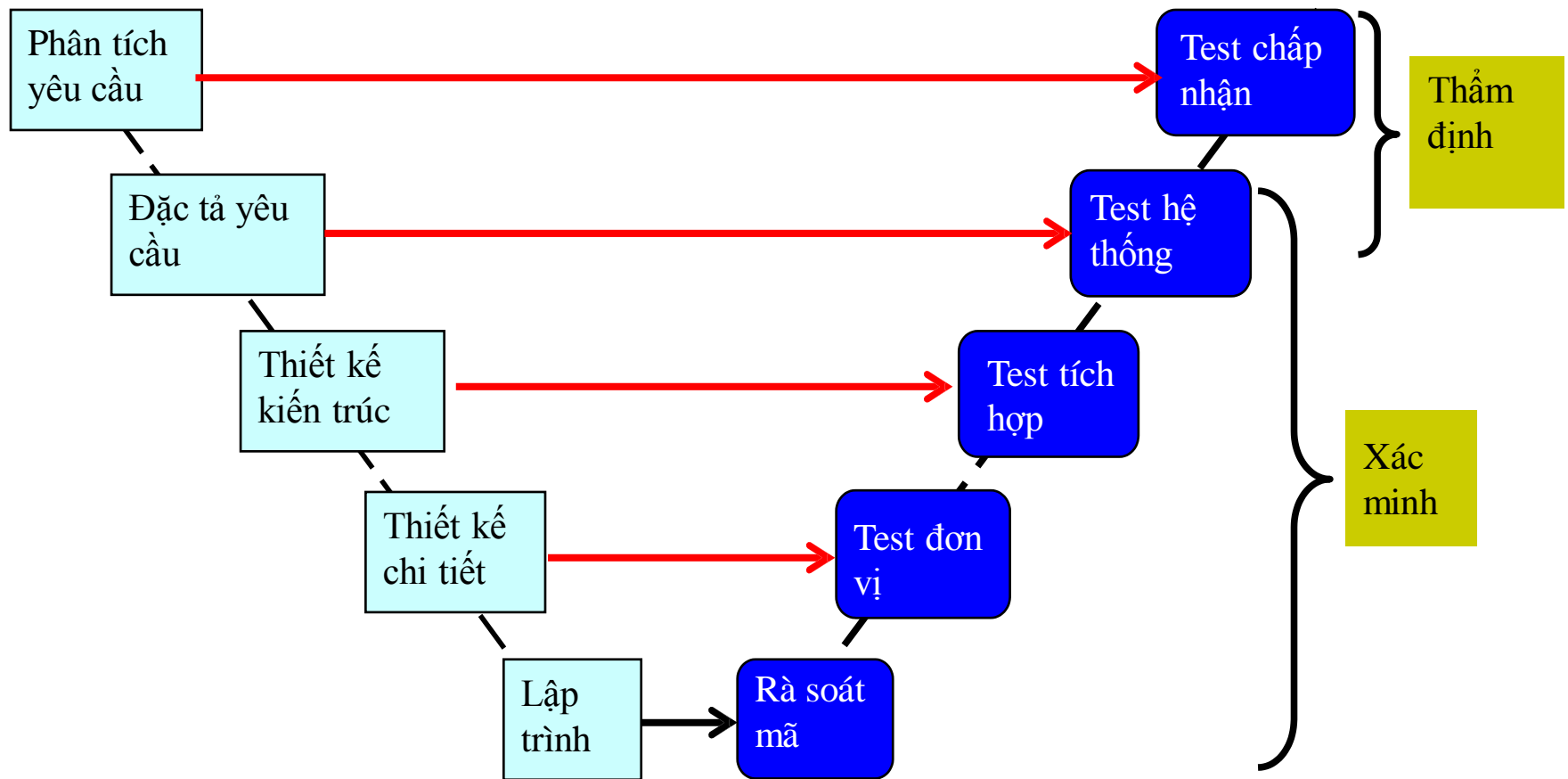
1. Một số khái niệm cơ bản
2. Tiến trình kiểm thử phần mềm
3. Các cách tiếp cận hợp trong kiểm thử p.m
4. **Các mức kiểm thử phần mềm**

## 4. Các mức kiểm thử p.m

- a. Kiểm thử đơn vị
- b. Kiểm thử tích hợp
- c. Kiểm thử hệ thống
- d. Kiểm thử chấp thuận
- e. Kiểm thử hồi quy

# 4. Các mức kiểm thử p.m

- Mô hình chữ V trong kiểm thử phần mềm

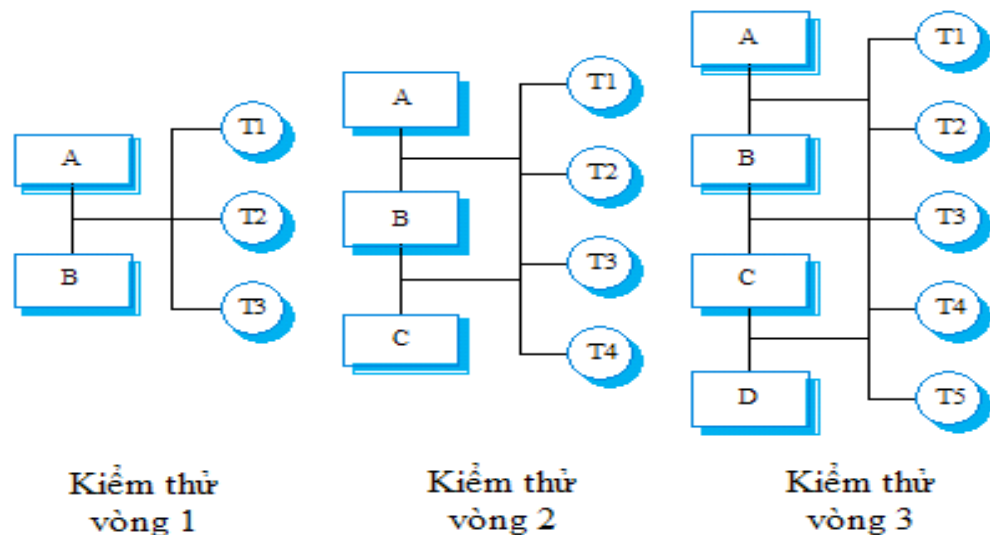


## a. Kiểm thử đơn vị (unit Testing)

- Đơn vị:
  - Các module riêng lẻ trong chương trình
    - Mô đun: chức năng/phương thức, lớp đối tượng, ...
- Kỹ thuật sử dụng:
  - Kiểm thử hộp đen, hộp trắng.
  - Dữ liệu sinh từ tài liệu đặc tả y.c hoặc từ mã nguồn

## b. Kiểm thử tích hợp (Integration Testing)

- Gồm 2 hoạt động:
  - *Tích hợp hệ thống;*
  - *Phát hiện các lỗi tương tác khi tích hợp*
- Thường sử dụng kỹ thuật kiểm thử hộp đen,
  - Dữ liệu thử sinh ra từ tài liệu thiết kế hệ thống tổng thể.



## b. Kiểm thử tích hợp (Integration Testing)

- **Các sai sót có thể gặp khi tích hợp**
  - *Dữ liệu bị mất* khi đi qua giao diện các chức năng.
  - *Hiệu ứng bất lợi* do một mô đun gây ra cho các mô đun khác.
  - *Sự kết hợp các chức năng phụ* có thể không sinh ra chức năng chính mong muốn.
  - *Sự phóng đại*: sự kết hợp các sai sót riêng lẻ làm hệ thống không chấp nhận được.
  - *Vấn đề về cấu trúc dữ liệu*: sự chuyển đổi dữ liệu khi đi qua các chức năng.



# c. Kiểm thử hệ thống

## ■ Các loại kiểm thử hệ thống

### 1. *Kiểm thử chức năng (mức hệ thống)*

- Các chức năng qua giao diện hệ thống có vận hành thông suốt không.

### 2. *Kiểm thử phục hồi (chịu lỗi)*

- Bắt phần mềm thất bại để xem khả năng phục hồi của nó đến đâu.
- Hai mức phục hồi:
  - phục hồi tự động;
  - phục hồi cần đến sự can thiệp của con người.

## c. Kiểm thử hệ thống

### 3. *Kiểm thử an ninh (sức chịu tấn công)*

- Kiểm tra mọi cơ chế bảo vệ được xây dựng xem có đạt hiệu quả đề ra trước các đột nhập hay không.
- Người kiểm thử đóng vai trò của kẻ đột nhập thực hiện mọi đột nhập có thể để đánh giá hệ thống.

### 4. *Kiểm thử hiệu năng*

- Kiểm tra thời gian phản hồi, khả năng chịu tải:
  - Ví dụ: cho hệ thống vận hành với số lượng giao dịch, tần xuất và cường độ bất thường; vận hành mô phỏng với cơ sở dữ liệu với số bản ghi cực lớn, vận hành hệ điều hành mạng với số máy tăng dần, ....

## d. Kiểm thử thẩm định (Validation testing)

- ~ Kiểm thử chấp thuận.

- Mục đích:

- Ktra phần mềm có đáp ứng mọi yêu cầu của người dùng không?

=> Kế hoạch kiểm thử và các thủ tục kiểm thử được thiết kế cần đảm bảo:

- *Tất cả các yêu cầu được thỏa mãn*
- *Các yêu cầu thực thi đã chính xác*
- *Tài liệu đúng đắn và*
- *Mọi yêu cầu khác được thỏa mãn.*

## e. Kiểm thử hồi quy (Regression Testing)

- Kiểm thử lại chương trình sau khi chương trình đã được đưa vào thực tế sử dụng.
- Dữ liệu thử:
  - Dùng lại các bộ dữ liệu thử trước đó.

# Tổng kết

1. Một số khái niệm cơ bản
2. Tiến trình kiểm thử phần mềm
3. Các cách tiếp cận hộp trong kiểm thử p.m
4. Các mức kiểm thử phần mềm

# Bài tập

- Viết các modun chương trình
  - Sắp xếp một dãy số tăng dần/giảm dần
  - Tìm kiếm số lớn nhất/nhỏ nhất trong dãy số
  - Đếm số lượng phần tử trong danh sách
- Thiết kế các test case cho chương trình
  - Kiểm thử lớp tương đương – hộp đen
  - Kiểm thử dựa trên đồ thị luồng điều khiển – hộp trắng.