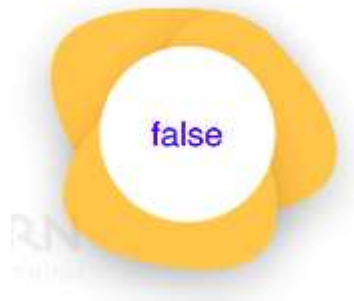


# JavaScript



CYBERSOFT  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

# Kiểu dữ liệu Boolean



CYBERSOFT

- ❖ Boolean: kiểu luận lý có 2 giá trị **true** hoặc **false**
- ❖ Ví dụ:
  - **var** check = true;
  - **var** isLoop = false;
- ❖ Dùng để chứa giá trị của phép so sánh trong lập trình

# Các loại toán tử so sánh

## ❖ Phép so sánh

- Bằng giá trị (==)
- Khác (!=)
- Bằng giá trị và kiểu (===)
- Khác giá trị hoặc kiểu (!==)
- Lớn hơn (>), lớn hơn hoặc bằng (>=)
- Nhỏ hơn (<), nhỏ hơn hoặc bằng (<=)



## Ví dụ biểu thức so sánh

Biểu thức	Ý nghĩa	Giá trị
$8 < 15$	8 bé hơn 15	true
$6 \neq 6$	6 khác 6	false
$2.5 > 5.8$	2.5 lớn hơn 5.8	false
$5.9 \leq 7.5$	5.9 bé hơn bằng 7.5	true
$7 < 10.4$	7 bé hơn bằng 10.4	true

# TOÁN TỬ SO SÁNH

## Toán tử so sánh

Toán tử	Ý nghĩa
!	not - phủ định
&&	and - và
	or - hoặc

# KẾT QUẢ CỦA 2 BIỂU THỨC LOGIC

- A && B đúng khi tất cả đều đúng
- A || B đúng khi một trong 2 đúng

## Kết quả của 2 biểu thức logic

A	B	A&&B	A  B	!A
true	true	true	true	false
false	true	false	true	true
true	false	false	true	false
false	false	false	false	true

# MỘT SỐ VÍ DỤ

$(14 \geq 5) \ \&\& \ ('A' < 'B')$

true

true

true

$(24 \geq 35) \ \&\& \ ('A' < 'B')$

false

true

false

$!('A' > 'B')$

false

true

$(14 \geq 5) \ || \ ('A' > 'B')$

true

false

true

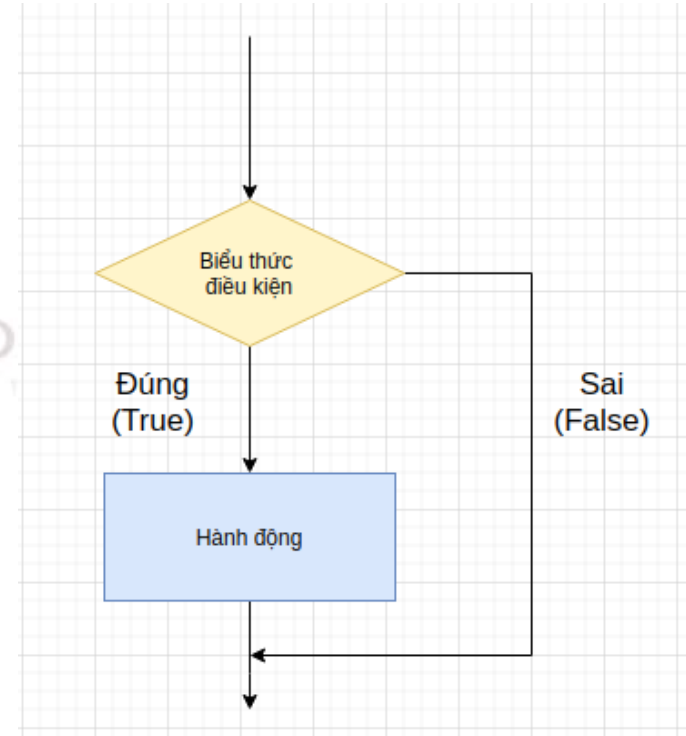
# CẤU TRÚC ĐIỀU KIỆN

## Cú pháp

if (*Biểu thức điều kiện*)

*Hành động*

- **Hành động** có thể là một dòng hoặc nhóm các xử lý gồm nhiều dòng lệnh trong cặp {}
- Nếu **biểu thức điều kiện** là đúng (True) thì sẽ thực hiện **hành động**
- Ngược lại, chương trình sẽ không thực hiện **hành động** và sẽ chạy các dòng lệnh tiếp theo ngoài if



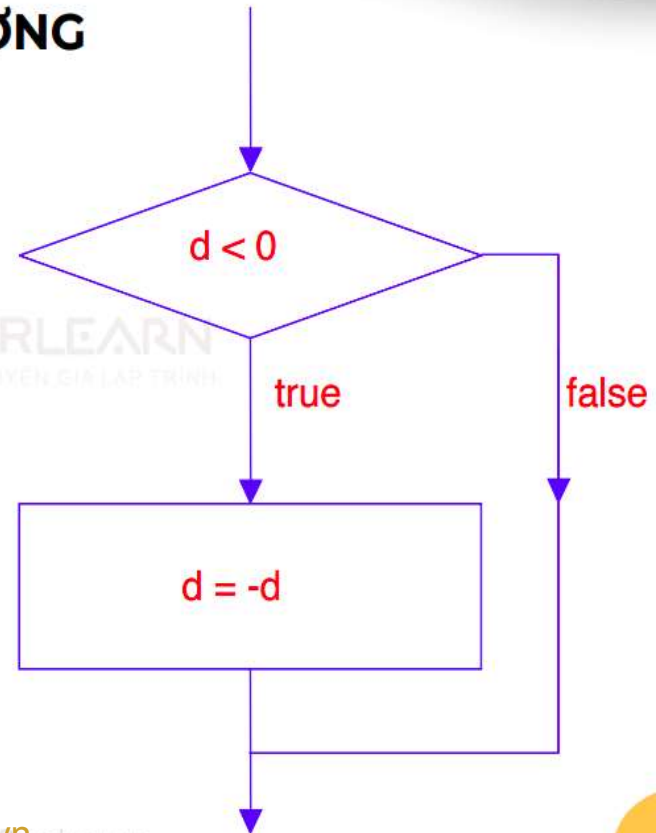


## VÍ DỤ 1: CHUYỂN SỐ ÂM SANG DƯƠNG

```
if (d < 0) {  
    d = -d;  
}
```

\* Giả sử  $d = 4$


\* Giả sử  $d = -9$

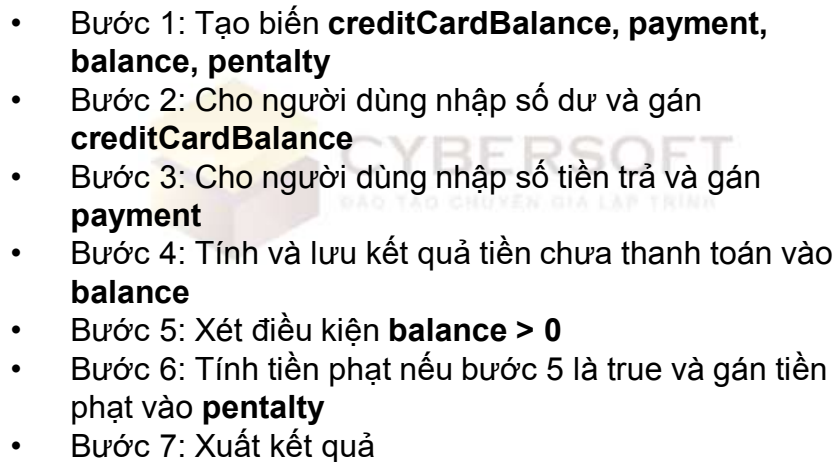


## VÍ DỤ 2: TÍNH PHÍ LÃI SUẤT THẺ TÍN DỤNG

Chương trình cho phép nhập vào số dư thẻ tín dụng, tiền thanh toán trong tháng  
Tính toán tiền phạt thẻ tín dụng chưa thanh toán. Giả định lãi suất là 1.5%/ tháng



- 
- Số dư thẻ tính dụng
  - Tiền thanh toán trong tháng

- 
- Bước 1: Tạo biến **creditCardBalance**, **payment**, **balance**, **pentalty**
  - Bước 2: Cho người dùng nhập số dư và gán **creditCardBalance**
  - Bước 3: Cho người dùng nhập số tiền trả và gán **payment**
  - Bước 4: Tính và lưu kết quả tiền chưa thanh toán vào **balance**
  - Bước 5: Xét điều kiện **balance > 0**
  - Bước 6: Tính tiền phạt nếu bước 5 là true và gán tiền phạt vào **pentalty**
  - Bước 7: Xuất kết quả

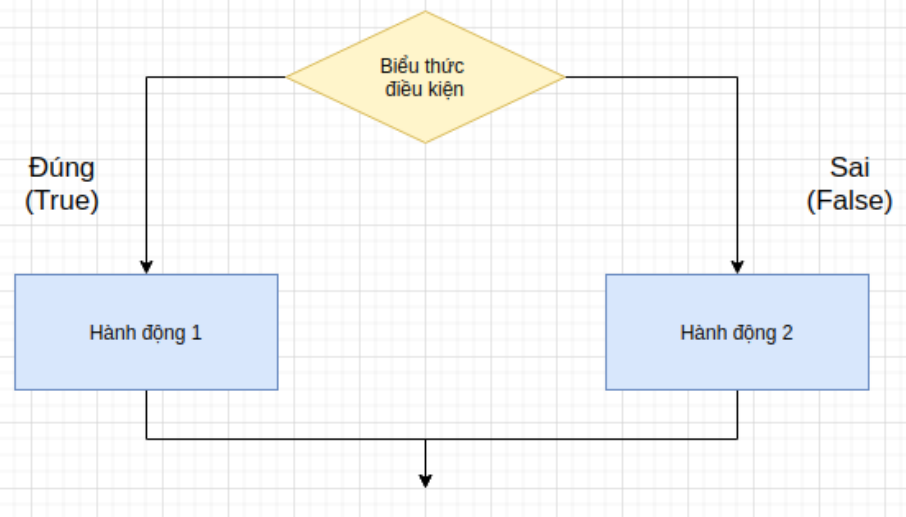


Tiền phạt chưa  
thanh toán

# IF ELSE

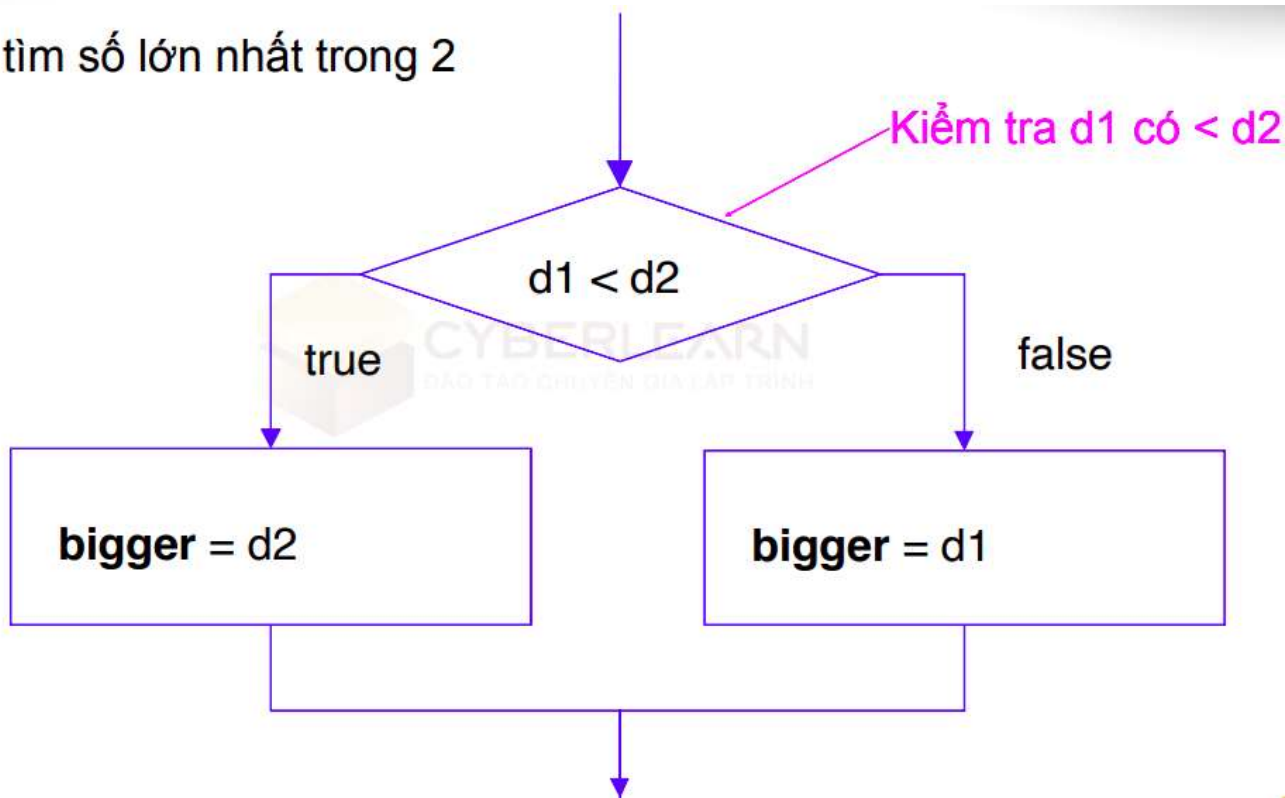
• Cú pháp  
if (*biểu thức điều kiện*)  
    *Hành động 1*  
else  
    *Hành động 2*

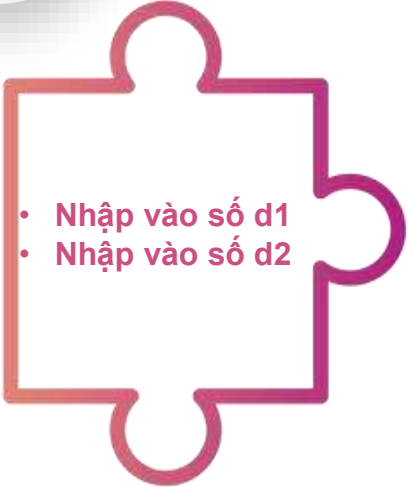
- Nếu **biểu thức điều kiện** là đúng (True) thì sẽ thực hiện **hành động 1** ngược lại sẽ thực hiện **hành động 2**
- **Hành động** có thể là một dòng hoặc nhóm các xử lý gồm nhiều dòng lệnh trong cặp {}

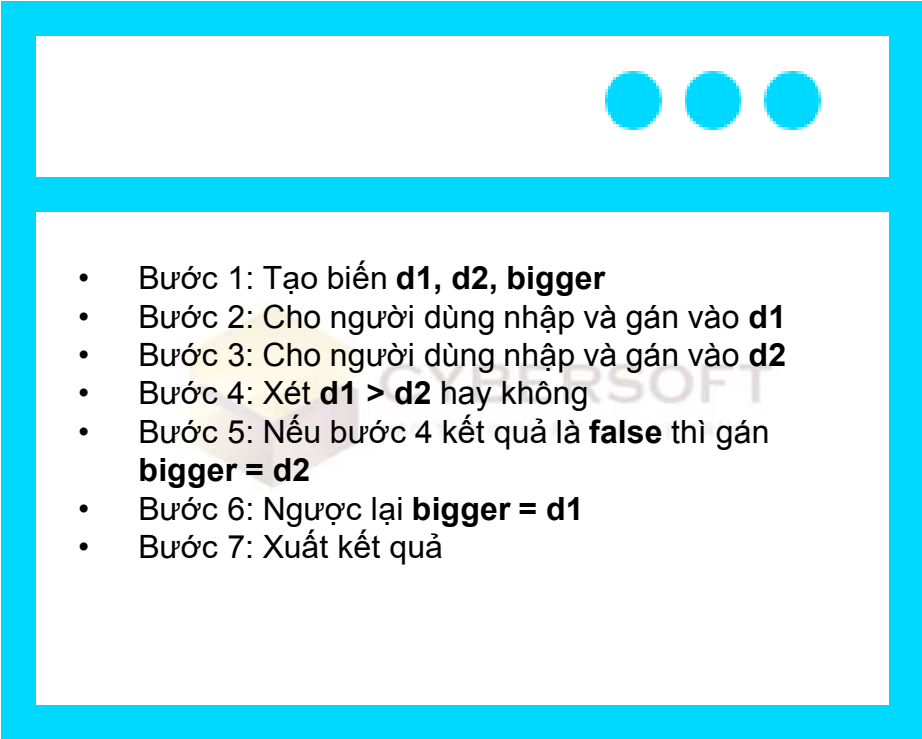


# VÍ DỤ 1

Nhập vào 2 số, tìm số lớn nhất trong 2 số d1 và d2



- 
- Nhập vào số **d1**
  - Nhập vào số **d2**

- 
- Bước 1: Tạo biến **d1**, **d2**, **bigger**
  - Bước 2: Cho người dùng nhập và gán vào **d1**
  - Bước 3: Cho người dùng nhập và gán vào **d2**
  - Bước 4: Xét **d1 > d2** hay không
  - Bước 5: Nếu bước 4 kết quả là **false** thì gán **bigger = d2**
  - Bước 6: Ngược lại **bigger = d1**
  - Bước 7: Xuất kết quả




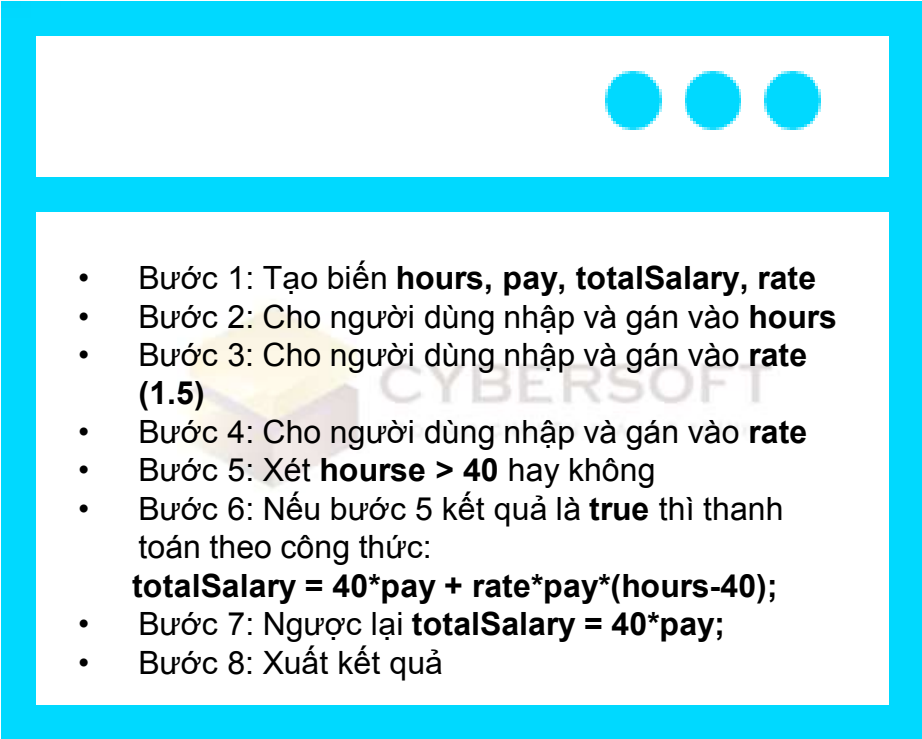
Xuất số lớn nhất

## VÍ DỤ 2

Viết chương trình tính toán tiền phải trả theo tuần cho nhân viên dựa vào số giờ làm, tiền theo giờ. Nếu số giờ lớn hơn 40 giờ 1 tuần thì phải trả giờ OT (Overtime) theo hệ số là 1.5



- 
- Nhập vào giờ làm
  - Nhập vào tiền mỗi giờ

- 
- Bước 1: Tạo biến **hours**, **pay**, **totalSalary**, **rate**
  - Bước 2: Cho người dùng nhập và gán vào **hours**
  - Bước 3: Cho người dùng nhập và gán vào **rate (1.5)**
  - Bước 4: Cho người dùng nhập và gán vào **rate**
  - Bước 5: Xét **hours > 40** hay không
  - Bước 6: Nếu bước 5 kết quả là **true** thì thanh toán theo công thức:  
**totalSalary = 40\*pay + rate\*pay\*(hours-40);**
  - Bước 7: Ngược lại **totalSalary = 40\*pay;**
  - Bước 8: Xuất kết quả



Tính tiền lương trả  
cho nhân viên





CYBERLEARN

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

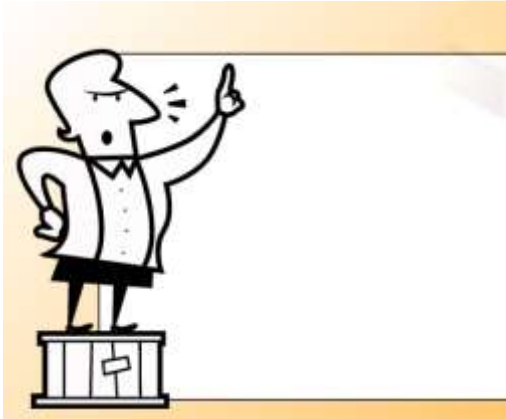


# TOÁN TỬ ĐIỀU KIỆN

- Lệnh *if...else* thường được viết sử dụng toán tử **?** : nếu câu lệnh sau biểu thức đơn giản.
- Cú pháp  
*Biểu thức điều kiện ? Biểu thức1 : Biểu thức 2*
- Nếu *Biểu thức điều kiện* là **True**, giá trị trả về là *Biểu thức1*.
- Nếu *Biểu thức điều kiện* là **false**, giá trị trả về là *Biểu thức2*.

# VÍ DỤ

```
var status = 1 === 1 ? true : false;
```



```
var status;  
if(1 === 1){  
    status = true;  
}else{  
    status = false;  
}
```

# ELSE IF

- Cú pháp

if (*biểu thức điều kiện 1*)

*Hành động 1*

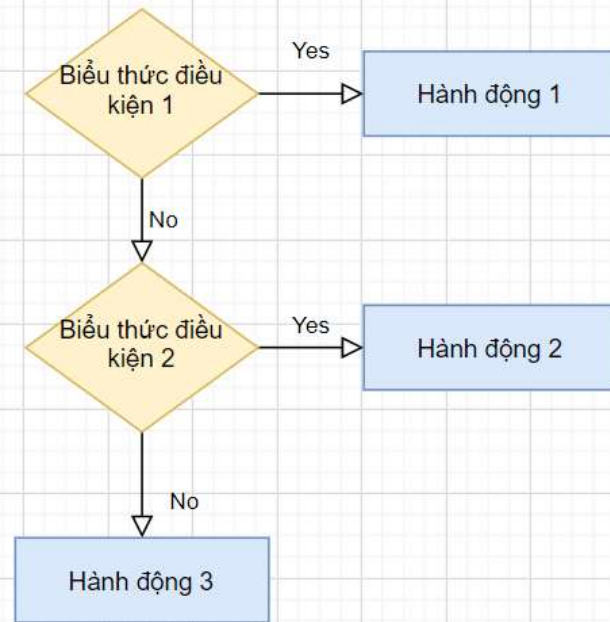
else if (*biểu thức điều kiện 2*)

*Hành động 2*

else

*Hành động 3*

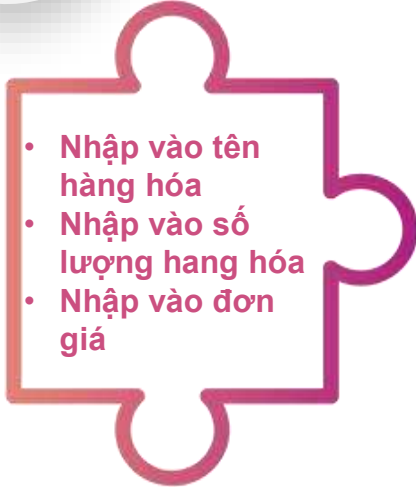
- Nếu **biểu thức điều kiện 1** là đúng (True) thì sẽ thực hiện **hành động 1**
- Ngược lại sẽ kiểm tra **biểu thức điều kiện 2**, nếu điều kiện 2 là đúng thì thực hiện **hành động 2**
- Nếu các điều kiện trước đó đều sai thì sẽ thực hiện **hành động 3**
- **Hành động** có thể là một dòng hoặc nhóm các xử lý gồm nhiều dòng lệnh trong cặp {}

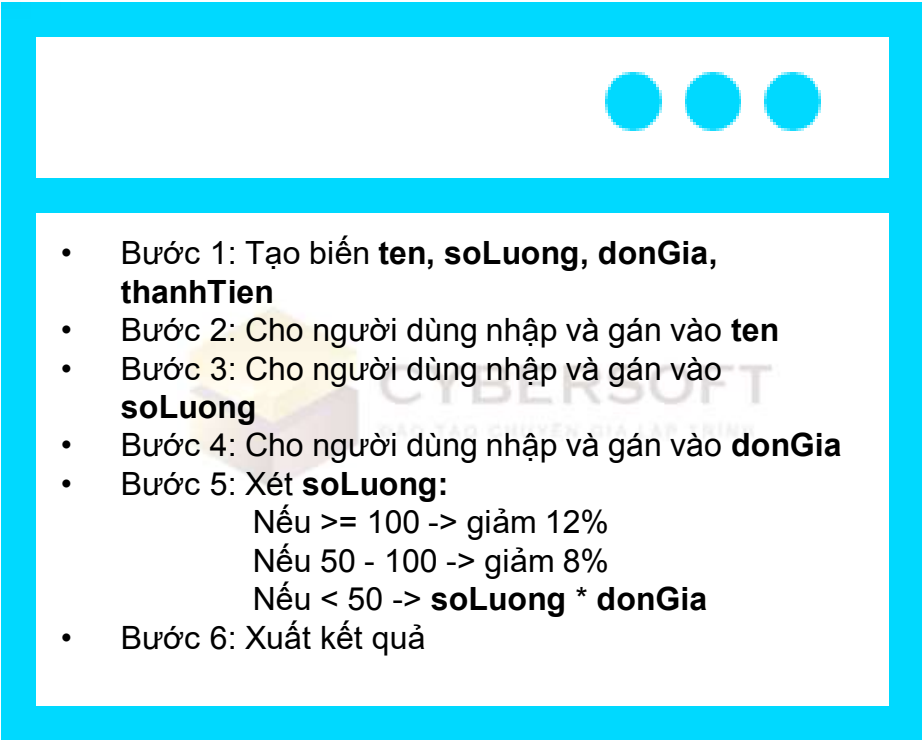


# VÍ DỤ ELSE IF

Viết chương trình nhập vào thông tin 1 mặt hàng Tên, số lượng, đơn giá. Tính và xuất tiền phải trả dựa theo quy tắc: Nếu mua với số lượng từ 50 đến 100 sẽ được giảm 8%, mua với số lượng trên 100 thì số lượng từ 100 trở đi sẽ được giảm 12%.



- 
- Nhập vào tên hàng hóa
  - Nhập vào số lượng hàng hóa
  - Nhập vào đơn giá

- 
- Bước 1: Tạo biến **ten**, **soLuong**, **donGia**, **thanhTien**
  - Bước 2: Cho người dùng nhập và gán vào **ten**
  - Bước 3: Cho người dùng nhập và gán vào **soLuong**
  - Bước 4: Cho người dùng nhập và gán vào **donGia**
  - Bước 5: Xét **soLuong**:
    - Nếu  $\geq 100$  -> giảm 12%
    - Nếu  $50 - 100$  -> giảm 8%
    - Nếu  $< 50$  -> **soLuong** \* **donGia**
  - Bước 6: Xuất kết quả



Tính tiền hàng

# VÍ DỤ ELSE IF



50



Số lượng < 50 → Thành tiền = số lượng \* đơn giá

50



100



$50 \leq \text{Số lượng} < 100$  → Thành tiền =  $50 * \text{đơn giá} + (\text{số lượng} - 50) * \text{đơn giá} * 0.92$

50

100



$100 \leq \text{Số lượng}$  → Thành tiền =  $100 * \text{đơn giá} + (\text{số lượng} - 100) * \text{đơn giá} * 0.88$



CYBERLEARN

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



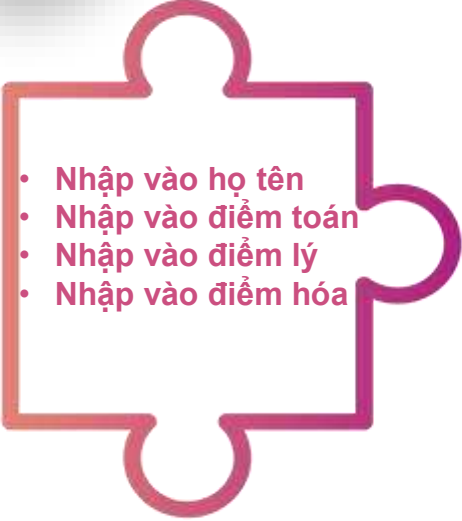


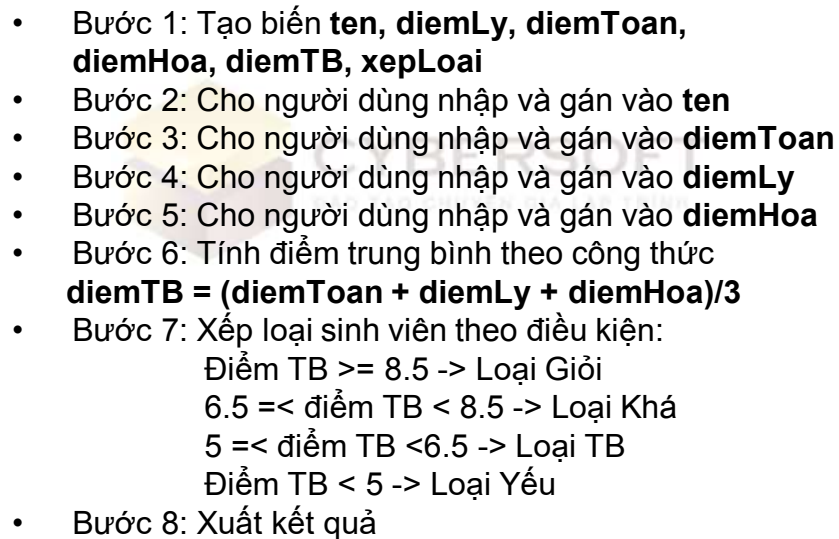
# LUYỆN TẬP IF....ELSE

Viết chương trình nhập vào thông tin 1 sinh viên: Họ tên, Toán, Lý, Hóa

Tính và xuất kết quả xếp loại theo bảng xếp loại sau:

- Loại Giỏi: Điểm TB  $\geq 8,5$
- Loại Khá:  $6,5 \leq \text{Điểm TB} < 8,5$
- Loại TB:  $5 \leq \text{Điểm TB} < 6,5$
- Loại Yếu: Điểm TB  $< 5$

- 
- Nhập vào họ tên
  - Nhập vào điểm toán
  - Nhập vào điểm lý
  - Nhập vào điểm hóa

- 
- Bước 1: Tạo biến **ten**, **diemLy**, **diemToan**, **diemHoa**, **diemTB**, **xepLoai**
  - Bước 2: Cho người dùng nhập và gán vào **ten**
  - Bước 3: Cho người dùng nhập và gán vào **diemToan**
  - Bước 4: Cho người dùng nhập và gán vào **diemLy**
  - Bước 5: Cho người dùng nhập và gán vào **diemHoa**
  - Bước 6: Tính điểm trung bình theo công thức  
**diemTB = (diemToan + diemLy + diemHoa)/3**
  - Bước 7: Xếp loại sinh viên theo điều kiện:  
Điểm TB >= 8.5 -> Loại Giỏi  
6.5 <= điểm TB < 8.5 -> Loại Khá  
5 <= điểm TB < 6.5 -> Loại TB  
Điểm TB < 5 -> Loại Yếu
  - Bước 8: Xuất kết quả



Xếp loại sinh viên



CYBERLEARN

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



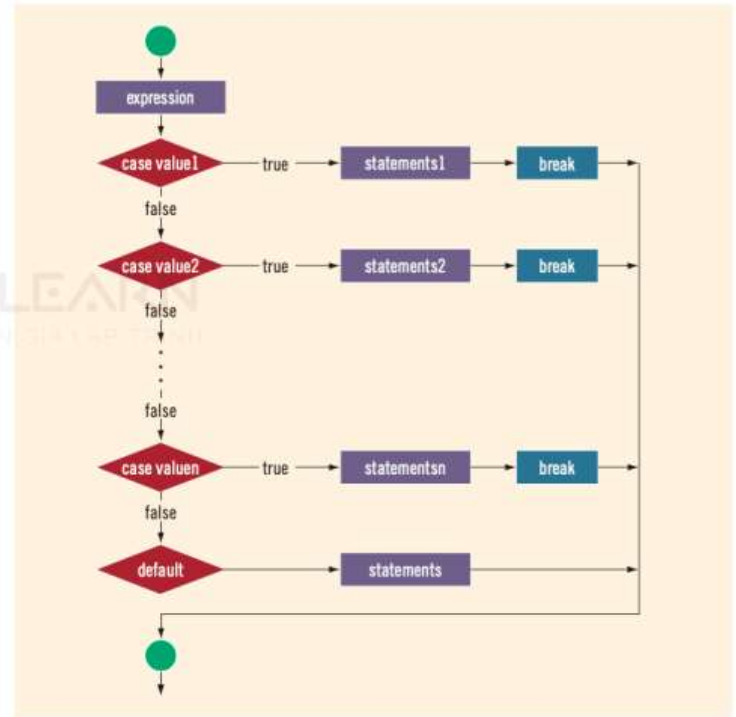
# CẤU TRÚC SWITCH....CASE

Cú pháp

```
switch(Biểu thức switch){  
  case case1 :  
    Action1  
    break;  
  case case2 :  
    Action2  
    break;  
  :  
  :  
  case caseN :  
    ActionN  
    break;  
  default :  
    DefaultAction  
}
```



CYBERLEARN  
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



## CẤU TRÚC SWITCH....CASE

Khi *Biểu thức switch* được xét đến, chương trình sẽ chạy đến kiểm tra các lệnh tương ứng với các giá trị trong CASE.

- Nếu *Biểu thức switch* không khớp với tất cả các CASE sẽ chạy lệnh trong **default**.
- Nếu đúng trường hợp CASE, lệnh sẽ chạy CHO ĐẾN KHI GẶP LỆNH **break** thì mới kết thúc switch
- Gặp lệnh **break** chương trình sẽ nhảy ngay đến cuối lệnh switch .

## ĐỌC SỐ TƯƠNG ỨNG TỪ NGƯỜI DÙNG

Đọc các số do người dùng nhập vào từ 1 đến 4

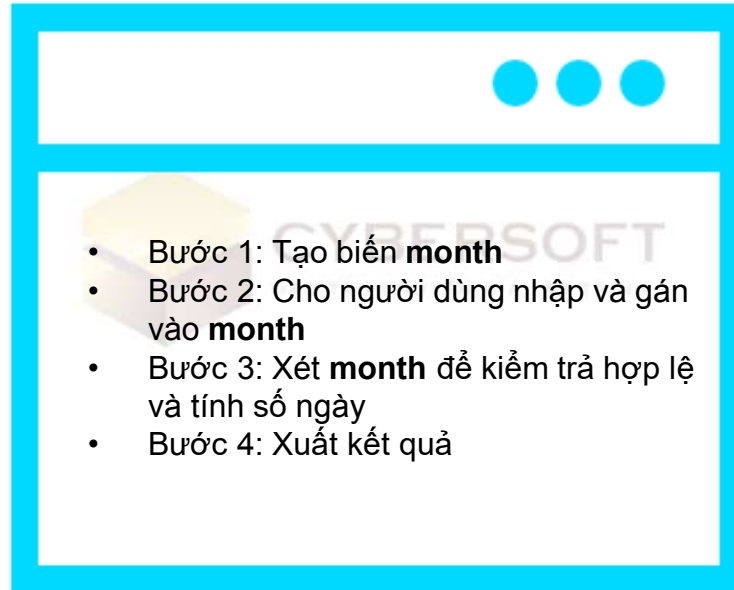
```
switch(n) {  
    case 1:  
        chuoai = "số 1";  
        break;  
    case 2:  
        chuoai = "số 2";  
        break;  
    case 3:  
        chuoai = "số 3";  
        break;  
    case 4:  
        chuoai = "số 4";  
        break;  
    default:  
        chuoai = "Vui lòng nhập số từ 1 đến 4";  
}
```

n = 3



# LUYỆN TẬP CẤU TRÚC SWITCH-CASE

- Nhập vào một tháng, kiểm tra tháng có hợp lệ hay không và xuất số ngày trong tháng đó. Chưa xét đến năm nhuận.



# HOMEWORK

Tóm tắt yêu cầu vẽ mô hình 3 khối, lập trình để xử lý các bài toán sau:

1. Cho người dùng nhập vào 3 số nguyên. Viết chương trình xuất 3 số theo thứ tự tăng dần
2. Viết chương trình “Chào hỏi” các thành viên trong gia đình với các đặc điểm. Đầu tiên máy sẽ hỏi ai sử dụng máy. Sau đó dựa vào câu trả lời và đưa ra lời chào phù hợp. Giả sử trong gia đình có 4 thành viên: Bố (B), Mẹ (M), anh Trai (A) và Em gái (E)
3. Cho 3 số nguyên. Viết chương trình xuất ra có bao nhiêu số lẻ và bao nhiêu số chẵn.
4. Viết chương trình cho nhập 3 cạnh của tam giác. Hãy cho biết đó là tam giác gì?
  - Ví dụ:  $a=2, b=2, c=1 \Rightarrow$  Tam giác cân
  - $a = 3, b=3, c=3 \Rightarrow$  Tam giác đều
  - $a = 3, b = 4, c=5 \Rightarrow$  Tam giác vuông (định lý Pytago)



# BÀI TẬP NÂNG CAO

Tóm tắt yêu cầu vẽ mô hình 3 khối, lập trình để xử lý các bài toán sau:

1. Viết chương trình nhập vào ngày, tháng, năm (Giả sử nhập đúng, không cần kiểm tra hợp lệ). Tìm ngày, tháng, năm của ngày tiếp theo. Tương tự tìm ngày tháng năm của ngày trước đó.
2. Viết chương trình nhập vào tháng, năm. Cho biết tháng đó có bao nhiêu ngày. (bao gồm tháng của năm nhuận).
3. Viết chương trình nhập vào số nguyên có 3 chữ số. In ra cách đọc nó.
4. Cho biết tên và tọa độ nhà của 3 sinh viên. Cho biết tọa độ của trường đại học. Viết chương trình in tên sinh viên xa trường nhất.