

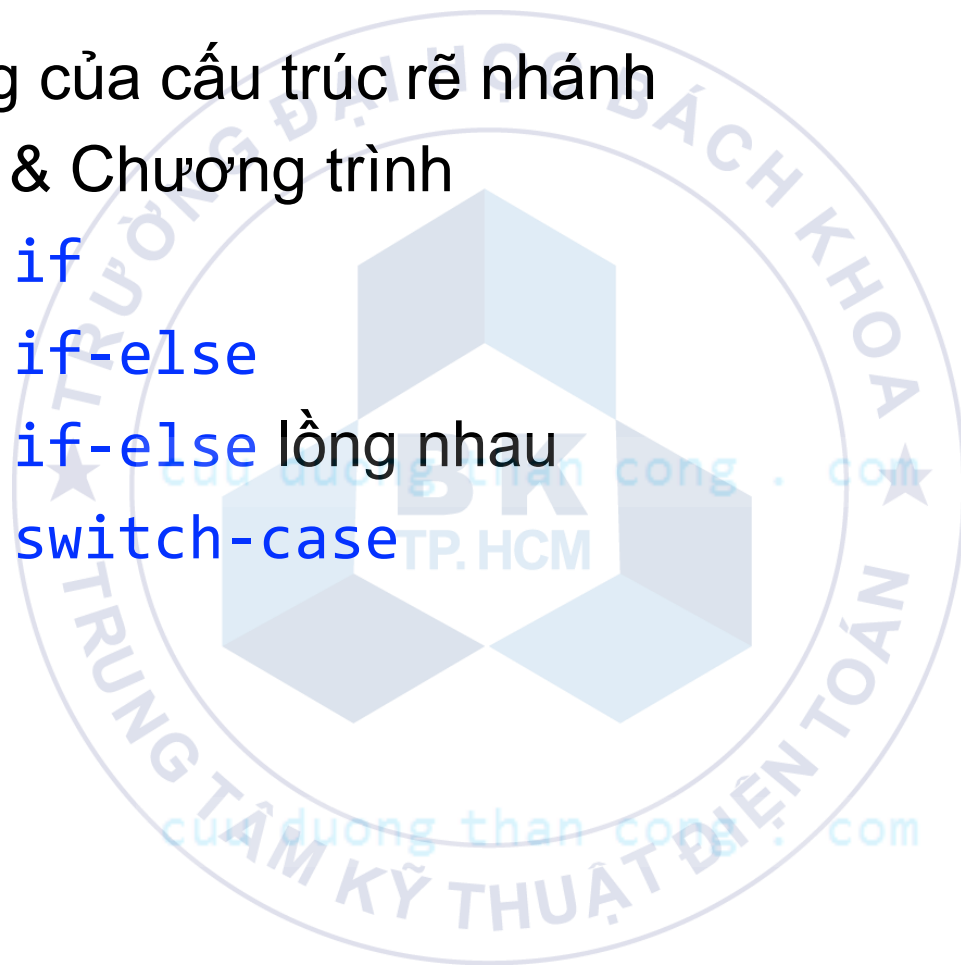


Chương 04 CẤU TRÚC RỄ NHÁNH

Nguyễn Thanh Tùng

Nội dung

- Ứng dụng của cấu trúc rẽ nhánh
- Câu lệnh & Chương trình
- Cấu trúc: `if`
- Cấu trúc: `if-else`
- Cấu trúc: `if-else` lồng nhau
- Cấu trúc: `switch-case`
- Tổng kết



Ứng dụng của cấu trúc rẽ nhánh

- Các ứng dụng hữu ích trong thực tế điều có dùng đến cấu trúc rẽ nhánh
- Ví dụ
 - Bài toán 1: Nhập một ngày (gồm ngày, tháng, năm)
 - Một chương trình tốt KHÔNG giả sử ngày, tháng và năm được đọc vào đã thuộc miền giá trị cho phép
 - Để khắc phục trường hợp lỗi với tháng, có thể cấu trúc sau đã sử dụng

```
if ( (tháng < 0) hoặc (tháng > 12) )  
    - Xử lý lỗi với tháng  
endif
```

Ứng dụng của cấu trúc rẽ nhánh

■ Ví dụ

■ Bài toán 2: Giải Phương trình bậc 2 (PTB2)

- Nếu ứng dụng cho phép nhập các hệ số A, B, C của PTB2
 - A và B: không chắc sẽ khác 0
 - Phương trình đầu vào chưa chắc đã là bậc 2 hay bậc 1
- => Chương trình có thể có lỗi thực thi khi không kiểm tra xem A và B có khác 0 hay không
- => Đã dùng đến cấu trúc rẽ nhánh để kiểm tra điều kiện nói trên

cuuduongthancong.com

cuuduongthancong.com

Ứng dụng của cấu trúc rẽ nhánh

■ Ví dụ

- Bài toán 3: Trong ứng dụng về kế toán, xác định thuế suất của một cá nhân
 - Bảng thuế suất Năm 2016 như bảng (nguồn: Kế toán Thiên ứng)

Bậc	Thu nhập tính thuế /tháng	Thuế suất	Tính số thuế phải nộp	
			Cách 1	Cách 2
1	Đến 5 triệu đồng (trđ)	5%	0 trđ + 5% TNTT	5% TNTT
2	Trên 5 trđ đến 10 trđ	10%	0,25 trđ + 10% TNTT trên 5 trđ	10% TNTT - 0,25 trđ
3	Trên 10 trđ đến 18 trđ	15%	0,75 trđ + 15% TNTT trên 10 trđ	15% TNTT - 0,75 trđ
4	Trên 18 trđ đến 32 trđ	20%	1,95 trđ + 20% TNTT trên 18 trđ	20% TNTT - 1,65 trđ
5	Trên 32 trđ đến 52 trđ	25%	4,75 trđ + 25% TNTT trên 32 trđ	25% TNTT - 3,25 trđ
6	Trên 52 trđ đến 80 trđ	30%	9,75 trđ + 30% TNTT trên 52 trđ	30 % TNTT - 5,85 trđ
7	Trên 80 trđ	35%	18,15 trđ + 35% TNTT trên 80 trđ	35% TNTT - 9,85 trđ

Ứng dụng của cấu trúc rẽ nhánh

■ Ví dụ

- Bài toán 3: Trong ứng dụng về kế toán, xác định thuế suất của một cá nhân
 - Bảng thuế suất Năm 2016 như bảng (nguồn Kế toán Thiên ứng)
 - => Thuế suất phải tính từ “thu nhập tính thuế”
 - Có nhiều trường hợp khác nhau mà áp thuế suất khác nhau
 - => Dùng cấu trúc rẽ nhánh để xác định “thuế suất”

cuduongthancong.com

Ứng dụng của cấu trúc rẽ nhánh

■ Ví dụ

- Bài toán 4: Hiện thực cách tương tác của người sử dụng cho một phần mềm (có giao diện đồ hoạ hay không)
 - Chương trình phải lắng nghe sự kiện xảy ra trên ứng dụng
 - Có giao diện đồ hoạ:
 - Sự kiện, chuột trái, phải, giữa, chọn menu A, menu B, v.v
 - Trên console
 - Đó là mã công việc (con số, chữ) được nhập vào
 - Chương trình phải thực hiện các công việc khác nhau tương ứng với từng sự kiện hay mã công việc.
- => Cần đến cấu trúc rẽ nhánh (switch, vì có nhiều trường hợp)

Câu lệnh & Chương trình

- Câu lệnh là gì?
 - Là một câu chương trình viết bằng ngôn ngữ lập trình
 - Kết thúc của câu này là dấu chấm phẩy (;)
 - Tương tự như câu trong ngôn ngữ tự nhiên là dấu chấm (.)
- Các loại câu lệnh
 - Câu lệnh đơn (câu đơn)
 - Là các câu lệnh đơn giản đã học trước, như:
 - Câu khai báo biến
 - Câu lệnh gán
 - Câu lệnh gọi hàm
 - V.v

Câu lệnh & Chương trình

■ Các loại câu lệnh

- Câu lệnh đơn (câu đơn)
- Câu lệnh phức (câu phức)
 - Một danh sách các câu lệnh được thực hiện cùng nhau. Đánh dấu danh sách này bằng cặp dấu: **{ và }**

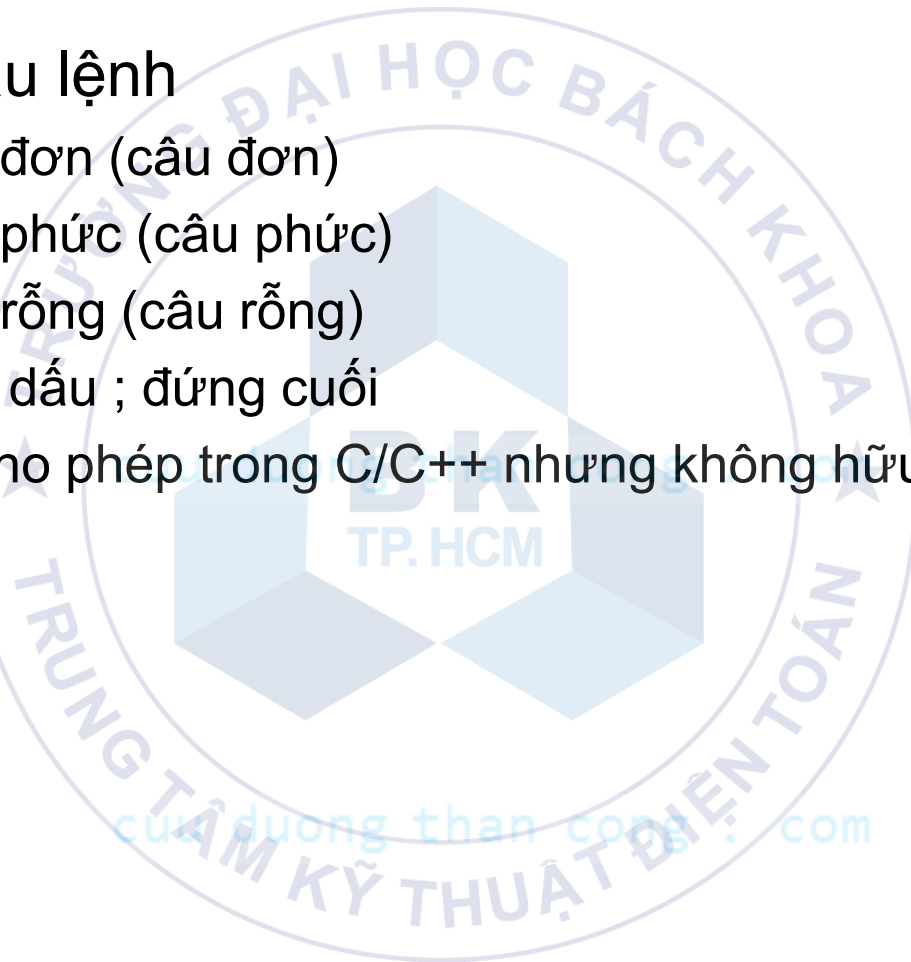
```
{  
    <câu lệnh 1>  
    <câu lệnh 2>  
    //...  
}
```

- Các cấu trúc điều khiển (**if**, **if-else**, **switch**, **for**, **while**, **và do ... while**) cũng được xem là câu lệnh phức

Câu lệnh & Chương trình

■ Các loại câu lệnh

- Câu lệnh đơn (câu đơn)
- Câu lệnh phức (câu phức)
- Câu lệnh rỗng (câu rỗng)
 - chỉ có dấu ; đứng cuối
 - Vẫn cho phép trong C/C++ nhưng không hữu dụng nhiều



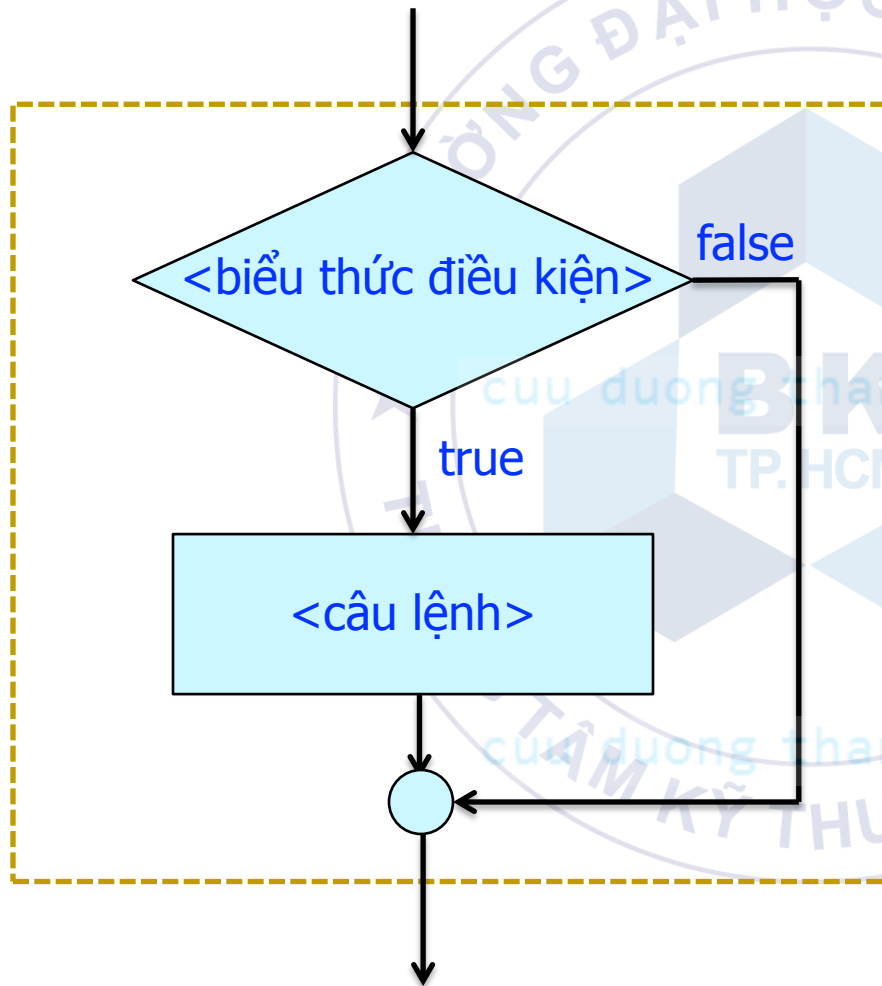
Câu lệnh & Chương trình

■ Chương trình

- Chương trình có thể được xem như một danh sách tuyến tính (có thứ tự) của các câu lệnh
 - Mỗi câu có thể là câu đơn hay câu phức (đã bao hàm cấu trúc điều khiển rẽ nhánh và lặp)
- Máy tính sẽ thực hiện lần lượt từ câu đầu tiên đến câu lệnh cuối cùng.
- Sự điều khiển thực thi này là tự nhiên, gọi là **Điều khiển Tuần tự**.
- Hai kiểu điều khiển khác
 - Rẽ nhánh
 - `if`, `if-else`, `switch`
 - Lặp
 - `for`, `while`, và `do ... while`

Câu lệnh: **if**

Ý tưởng

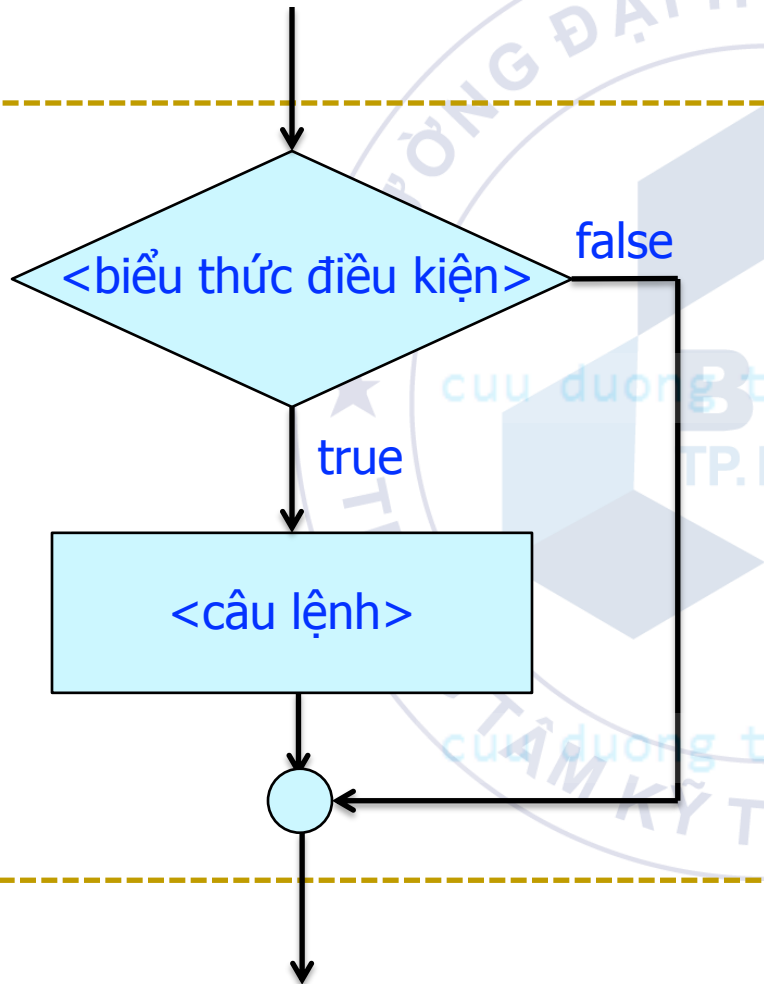


<biểu thức điều kiện>: Biểu thức luận lý hoặc phải chuyển đổi sang luận lý được

<câu lệnh>: một trong các loại câu lệnh: đơn, kép hay rỗng

Câu lệnh: **if**

Ý tưởng



Cách thực hiện câu lệnh **if**:

(1) Biểu thức điều kiện được đánh giá để lấy giá trị

(2) Nếu giá trị này là **true** thì chương trình thực thi <câu lệnh>. Ngược lại thì chương trình đến câu lệnh theo sau câu lệnh **if**

Câu lệnh: **if**

■ Cú pháp:

if (<biểu thức điều kiện>) <câu lệnh>

if (<biểu thức điều kiện>)
<câu lệnh>

if (<biểu thức điều kiện>) {
 <câu lệnh>
 //có thể có 1 hay nhiều câu lệnh
}

if (<biểu thức điều kiện>)
{
 <câu lệnh>
 //có thể có 1 hay nhiều câu lệnh
}

Câu lệnh: **if**

- Cú pháp + **phong cách lập trình**:

```
if (<biểu thức điều kiện>) <câu lệnh>
```

```
if (<biểu thức điều kiện>)  
    <câu lệnh>
```

TAB

```
if (<biểu thức điều kiện>) {  
    <câu lệnh>  
    //có thể có 1 hay nhiều câu lệnh
```

```
if (<biểu thức điều kiện>)  
{  
    <câu lệnh>  
    //có thể có 1 hay nhiều câu lệnh
```

TAB

Câu lệnh: **if**

- Ví dụ

- Bài toán kiểm tra ngày, tháng, năm có hợp lệ

```
if( (thang < 0) || (thang > 12) )  
    exit(1);
```

Chương trình sẽ kết thúc với lỗi khi tháng không hợp lệ

Câu lệnh: **if**

■ Ví dụ

- Bài toán kiểm tra ngày, tháng, năm có hợp lệ
- Cũng có trường hợp người lập trình muốn: nếu dữ liệu (ngày, tháng, năm) không hợp lệ thì sẽ ngày, tháng, và năm chuyển về một ngày đặc biệt nào đó. Ví dụ: 01/01/1970

```
if( (thang < 0) || (thang > 12) ){  
    ngay = 1;  
    thang = 1;  
    nam = 1970;  
}
```

cuu duong than cong . com
BK
TP. HCM

cuu duong than cong . com

Câu lệnh: **if**

■ Ví dụ

- Bài toán kiểm tra ngày, tháng, năm có hợp lệ
- Cũng có trường hợp người lập trình muốn: nếu dữ liệu (ngày, tháng, năm) không hợp lệ thì sẽ ngày, tháng, và năm chuyển về một ngày đặc biệt nào đó. Ví dụ: 01/01/1970

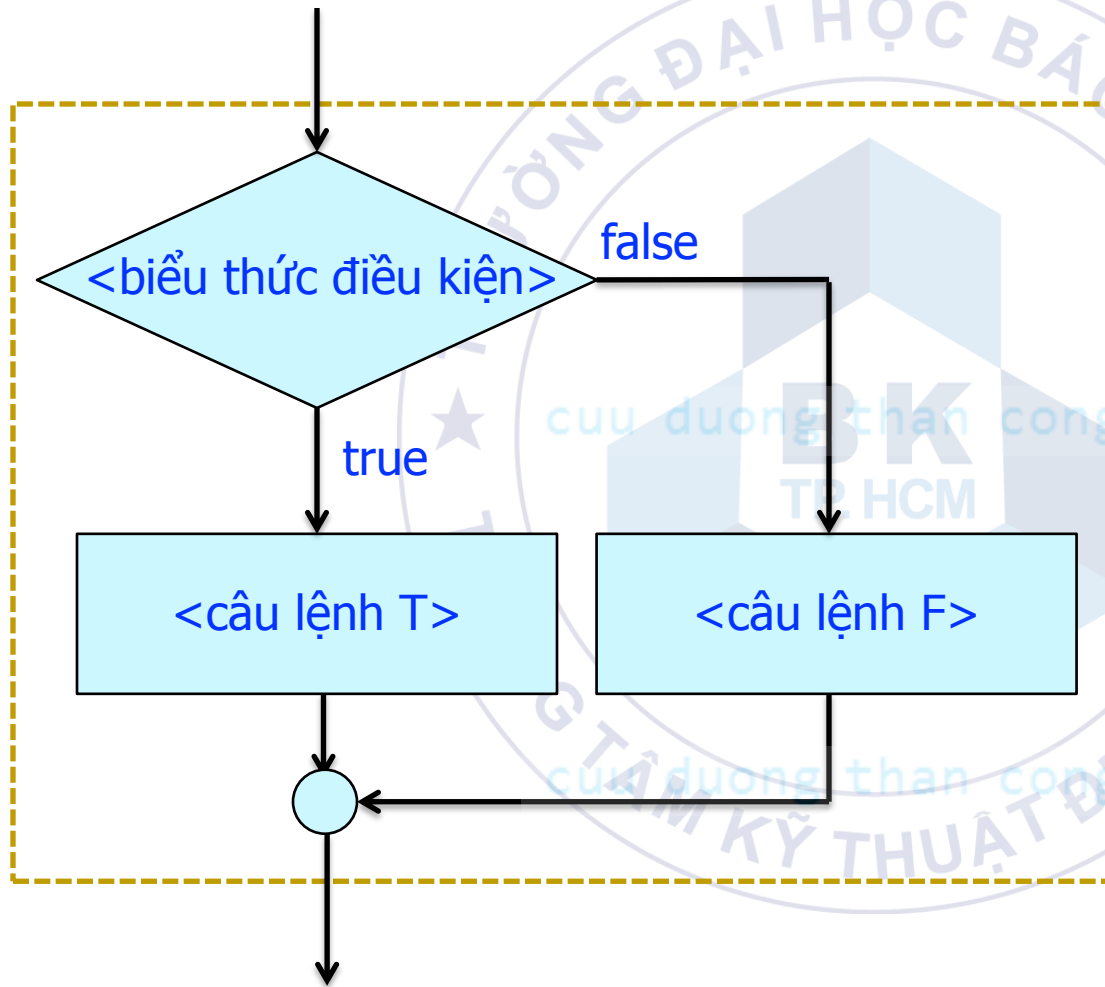
```
if( (thang < 0) || (thang > 12) )  
    ngay = 1;  
    thang = 1;  
    nam = 1970;
```

Sẽ không đúng theo logic nói trên nếu viết như thế này.

Vì tháng và năm luôn luôn bị gán vào giá trị đặc biệt **bất kể** lỗi dữ liệu về tháng hay không

Câu lệnh: **if-else**

Ý tưởng

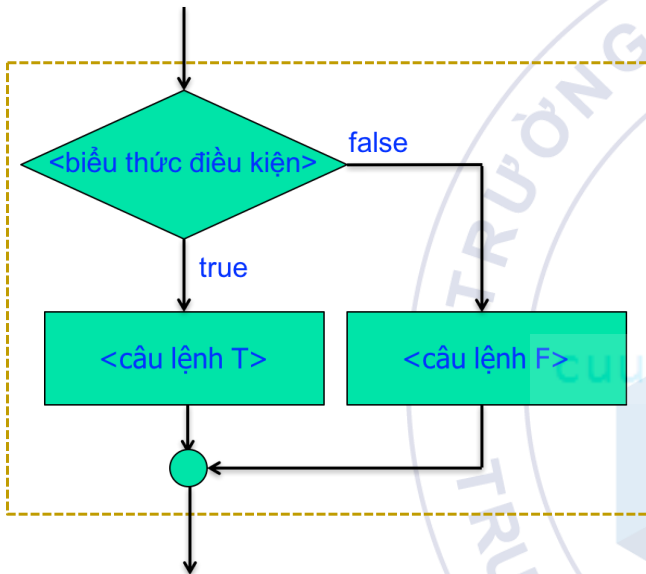


<biểu thức điều kiện>: Biểu thức luận lý hoặc phải chuyển đổi sang luận lý được

<câu lệnh T>, <câu lệnh F>: : một trong các loại câu lệnh: đơn, kép hay rỗng

Câu lệnh: **if-else**

Ý tưởng



Cách thực hiện câu lệnh **if-else**:

(1) Biểu thức điều kiện được đánh giá để lấy giá trị

(2) Nếu giá trị này là **true** thì chương trình thực thi **<câu lệnh T>**. Ngược lại thì chương trình thực thi câu lệnh **<câu lệnh F>**. Chương trình đi đến câu lệnh kế tiếp (sau **if-else**) khi thực thi xong một trong hai câu lệnh nói trên

Lưu ý: chương trình **KHÔNG** thực hiện cả hai câu lệnh T và F trong **BẤT KỲ** trường hợp nào

Câu lệnh: **if-else**

■ Cú pháp:

if (<biểu thức điều kiện>)
 <câu lệnh T>
else
 <câu lệnh F>

if (<biểu thức điều kiện>) <câu lệnh T>
else <câu lệnh F>

Lưu ý: <câu lệnh T> và <câu lệnh F> đã kết thúc bằng dấu **;**

Câu lệnh: **if-else**

■ Cú pháp + phong cách lập trình

```
if (<biểu thức điều kiện>)  
    <câu lệnh đơn khi đúng>  
else  
    <câu lệnh đơn khi sai>
```

```
if (<biểu thức điều kiện>) {  
    <câu lệnh đơn/kép khi đúng>  
    //...  
    <câu lệnh đơn/kép khi đúng>  
}  
else{  
    <câu lệnh đơn/kép khi sai>  
    //...  
    <câu lệnh đơn/kép khi sai>  
}
```

```
if (<biểu thức điều kiện>)  
{  
    <câu lệnh đơn/kép khi đúng>  
    //...  
    <câu lệnh đơn/kép khi đúng>  
}  
else com  
{  
    <câu lệnh đơn/kép khi sai>  
    //...  
    <câu lệnh đơn/kép khi sai>  
}
```

cuu duong than cong
BK
TP.HCM
TRUNG TÂM KỸ THUẬT
cuu duong than cong

Câu lệnh: **if-else**

■ Cú pháp + phong cách lập trình

```
if (<biểu thức điều kiện>)  
    <câu lệnh đơn khi đúng>  
else  
    <câu lệnh đơn khi sai>
```

TAB

```
if (<biểu thức điều kiện>) {  
    <câu lệnh đơn/kép khi đúng>  
    //...  
    <câu lệnh đơn/kép khi đúng>  
}  
else {  
    <câu lệnh đơn/kép khi sai>  
    //...  
    <câu lệnh đơn/kép khi sai>  
}
```

```
if (<biểu thức điều kiện>)  
{  
    <câu lệnh đơn/kép khi đúng>  
    //...  
    <câu lệnh đơn/kép khi đúng>  
}  
else  
{  
    <câu lệnh đơn/kép khi sai>  
    //...  
    <câu lệnh đơn/kép khi sai>  
}
```

TAB

Câu lệnh: **if-else**

- Cú pháp + phong cách lập trình

if (<biểu thức điều kiện>) <câu lệnh T>
else <câu lệnh F>

Phong cách này chỉ nên dùng khi cả hai câu lệnh là câu đơn giản

Câu lệnh: **if-else**

Ví dụ

- Giải Phương trình bậc 2
 - Phân tích
 - Cho phép người dùng nhập ba hệ số của phương trình
 - Do đó,
 - A và B: có thể bằng 0
 - Phương trình suy biến thành Bậc 1 hoặc Bậc 0
 - Nghĩa là có 3 trường hợp cần kiểm tra

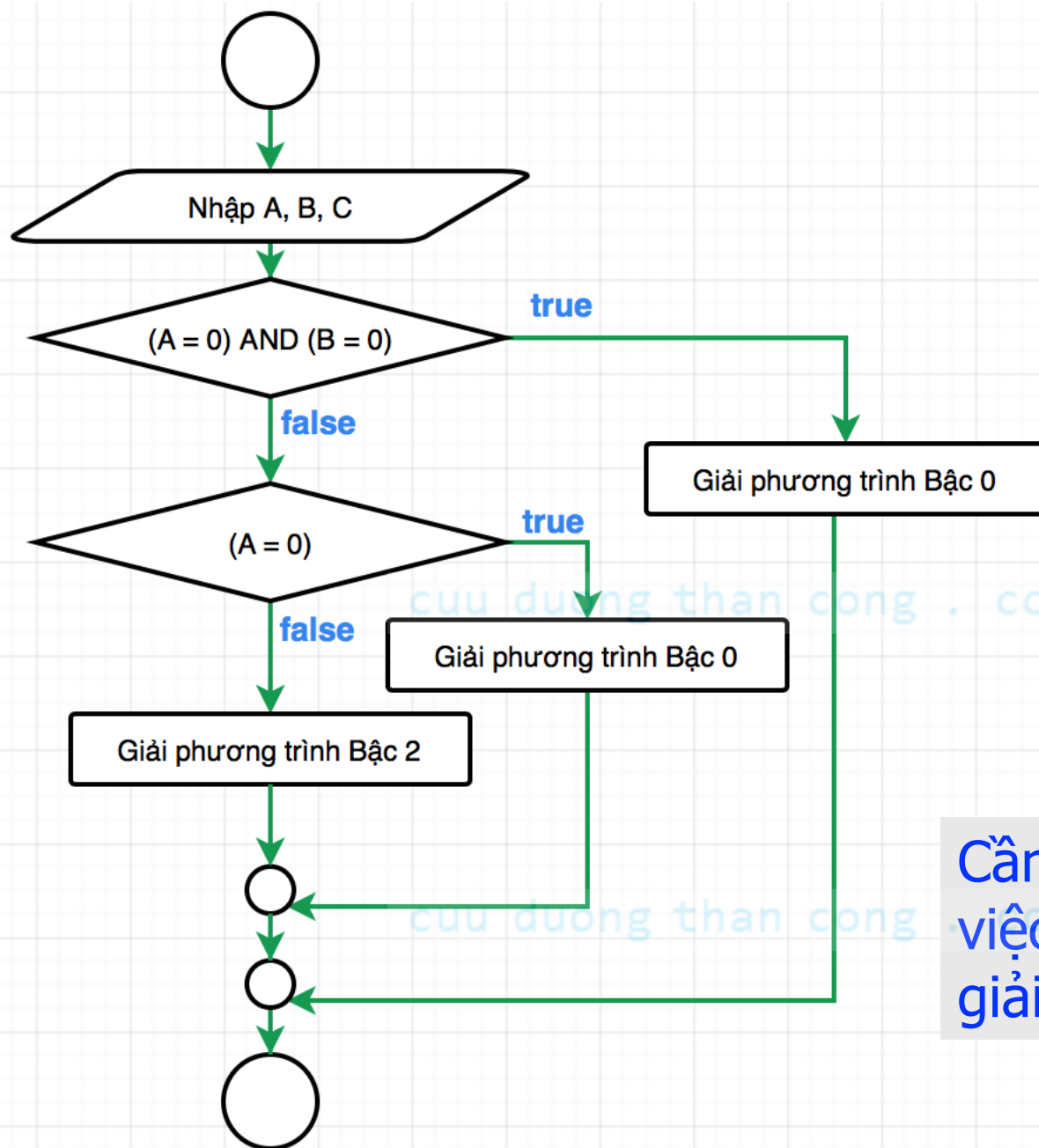
cuiduongthancong.com

Câu lệnh: **if-else**

Ví dụ

- Giải Phương trình bậc 2
 - Phân tích
 - Ba trường hợp đó là
 - (1) **Bậc 0**: $A = 0$ và $B = 0$
 - Dựa vào C để biết vô hay vô số nghiệm
 - (2) **Bậc 1**: $A = 0, B \neq 0$
 - Giải Phương trình Bậc 1
 - (3) **Bậc 2**: $A \neq 0$
 - Giải Phương trình bậc 2

cuuduongthancong.com



Cần phân rã các công việc con trong đó để có giải thuật hoàn thiện

Câu lệnh: **if-else**

Ví dụ

■ PTB2: Chương trình C

```
#include <stdlib.h>
#include <stdio.h>
int main(){
    float a, b, c, delta;

    printf("Nhap vao gia tri cac so a, b, c: \n");
    scanf("%f%f%f", &a, &b, &c);

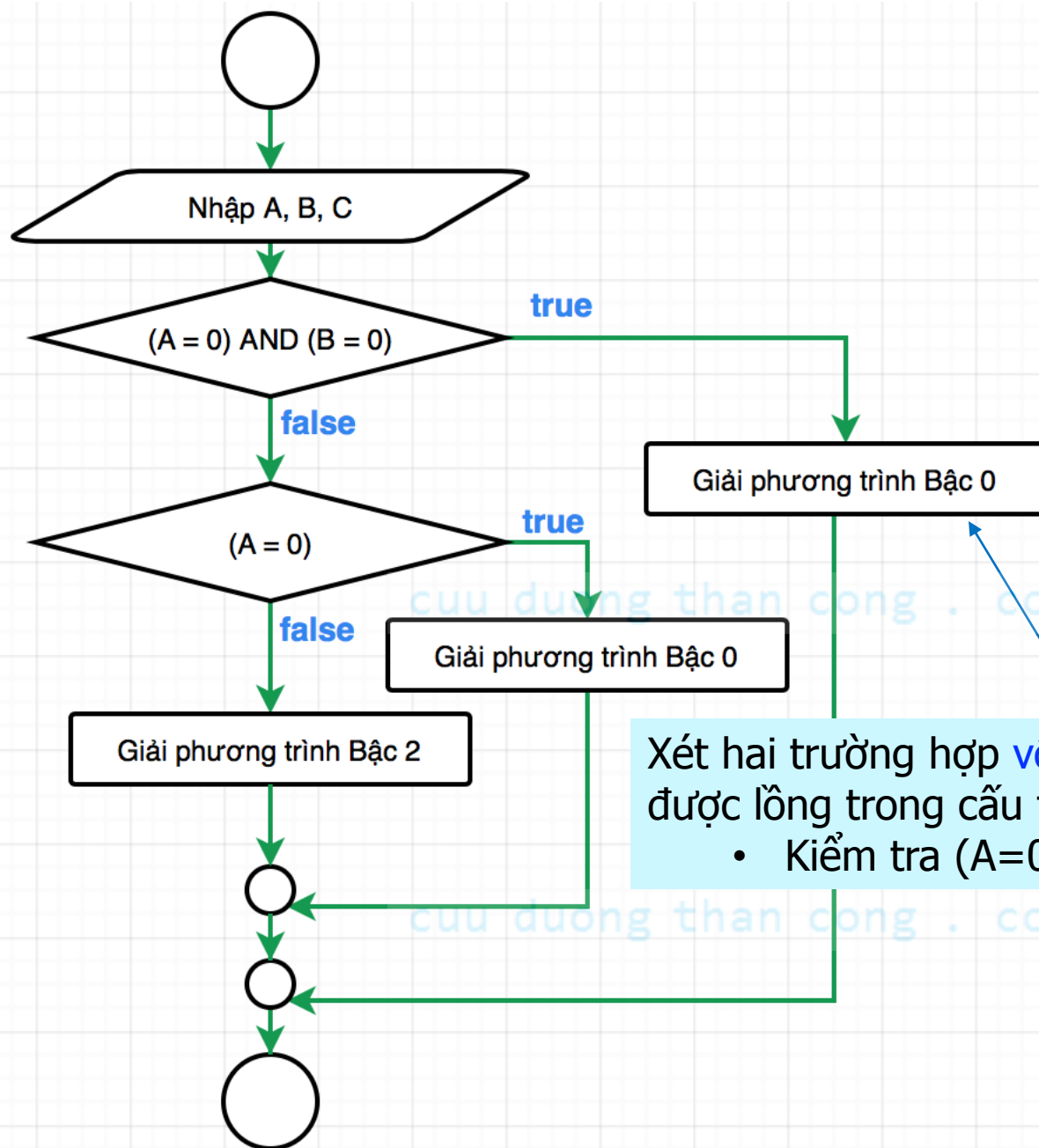
    delta = b*b - 4*a*c;
    if(delta < 0)
        printf("Phuong trinh vo nghiem\n");
    else
        printf("Phuong trinh co it nhat mot nghiem\n");

    return 0;
}
```

Câu lệnh: **if-else** lồng nhau

Ứng dụng

- Hầu hết các ứng dụng hữu ích trong thực tế không chỉ xét các trường hợp đơn giản rời nhau (tuần tự).
- Các ứng dụng cần kiểm tra các điều kiện có yếu tố lồng nhau
 - Ví dụ
 - Giải PTB2:
 - Nếu A và B đã kiểm tra bằng 0
 - Chương trình cần kiểm tra tiếp $C = 0$ hay không
 - Nếu $C = 0$ thì chương trình có vô số nghiệm
 - Ngược lại, chương trình không có nghiệm



Xét hai trường hợp **vô nghiệm** và **vô số nghiệm** được lồng trong cấu trúc lớn hơn:

- Kiểm tra $(A=0) \text{ AND } (B=0)$

Câu lệnh: **if-else** lồng nhau

Ứng dụng

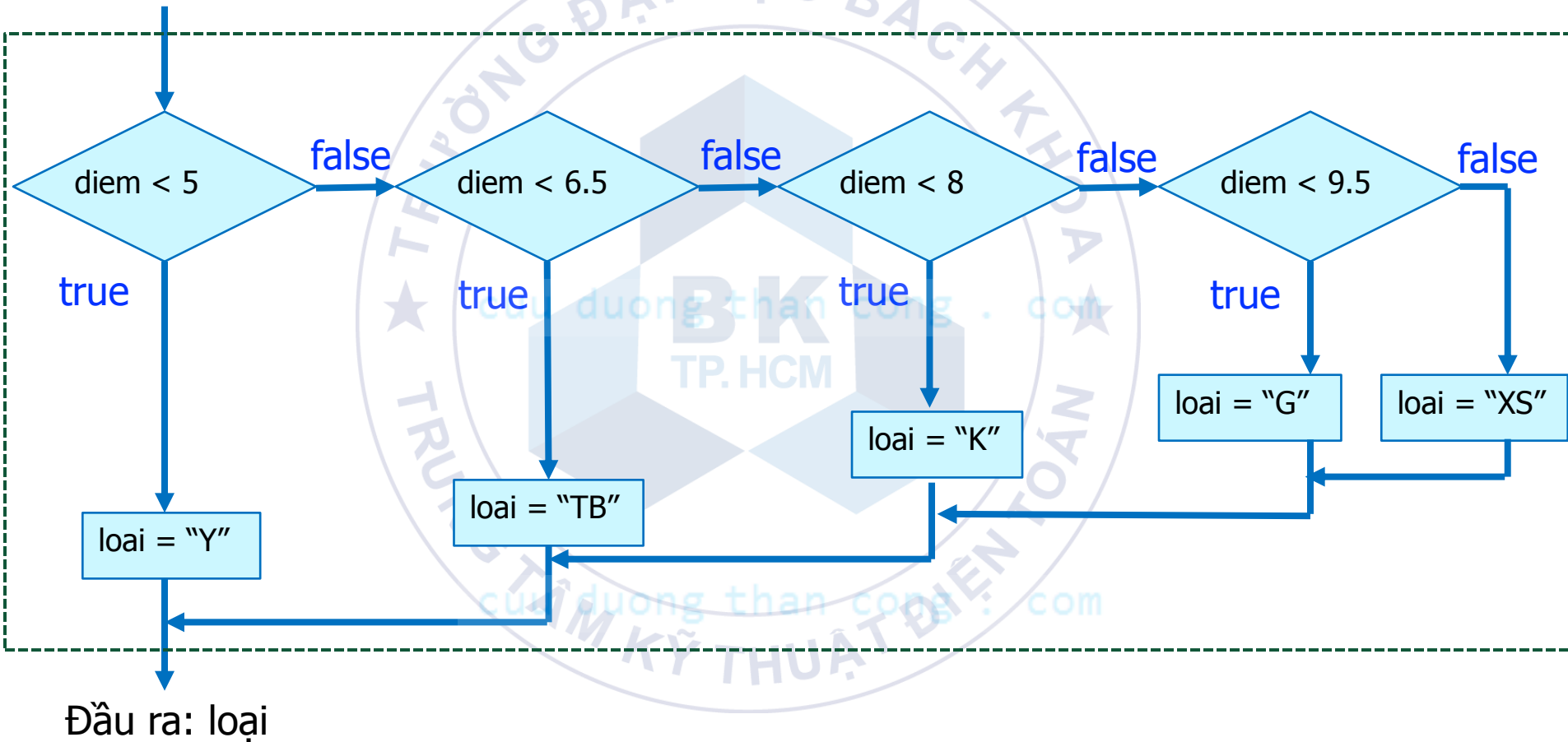
- Bài toán: xác định loại sinh viên theo điểm
 - Có các loại sau:
 - Yếu: điểm thuộc $[0, 5]$
 - Trung bình: điểm thuộc $[5, 6.5)$
 - Khá: điểm thuộc $[6.5, 8)$
 - Giỏi: điểm thuộc $[8, 9.5)$
 - Xuất sắc: điểm thuộc $[9.5, 10]$

cuduongthancong.com

Câu lệnh: **if-else** lồng nhau

Ứng dụng

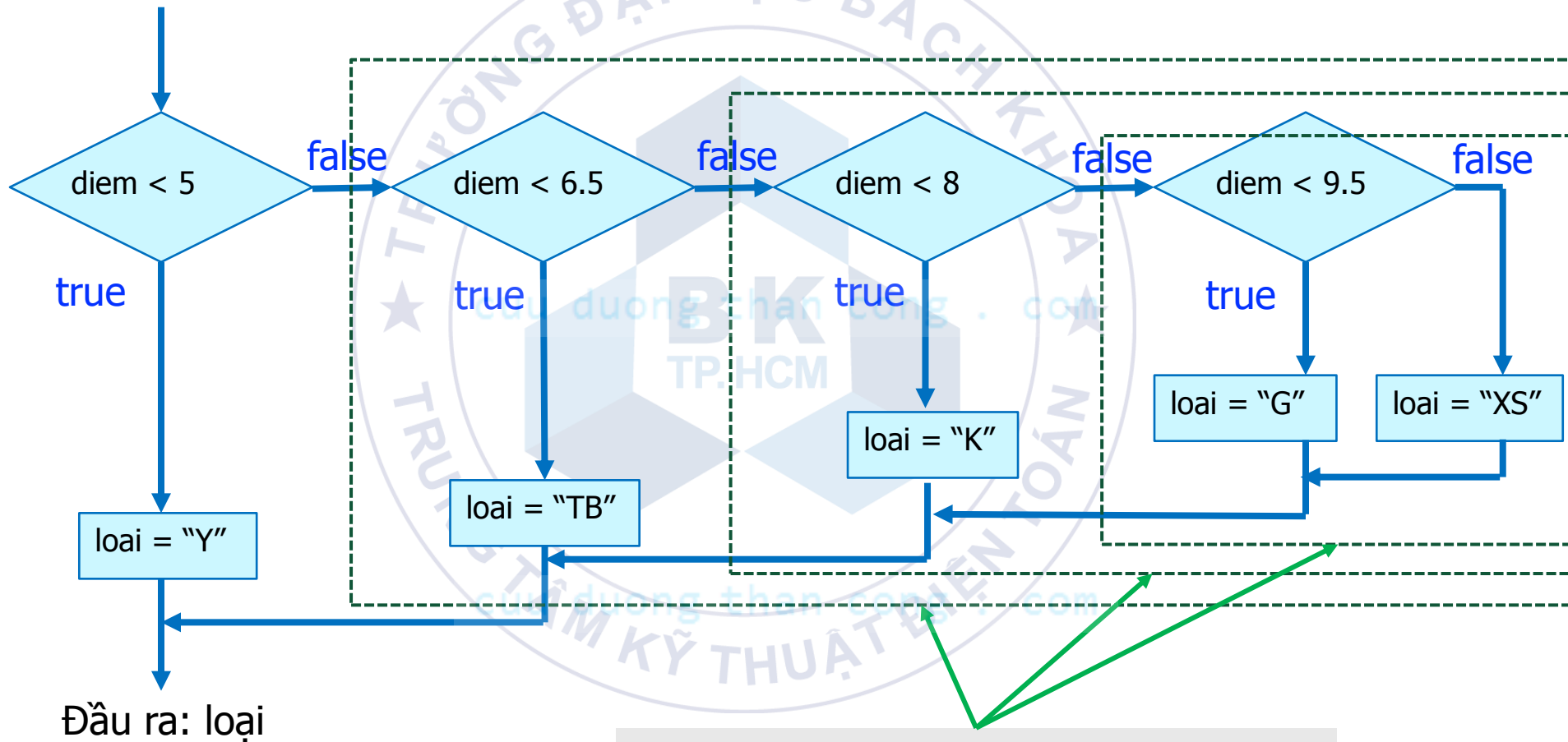
Đầu vào: điểm



Câu lệnh: **if-else** lồng nhau

Ứng dụng

Đầu vào: điểm



Các câu lệnh con lồng bên trong khi điều kiện (diem<5) bị sai

Câu lệnh: **if-else** lồng nhau

Cú pháp - có nhiều cách biểu diễn

```
if (<biểu thức điều kiện 1>)    <Câu lệnh thực thi 1>
else if (<biểu thức điều kiện 2>) <Câu lệnh thực thi 2>
else if (<biểu thức điều kiện 3>) <Câu lệnh thực thi 3>
else    <Câu lệnh thực thi 4>
```

```
if (<biểu thức điều kiện 1>)
    <Câu lệnh thực thi 1>
else if (<biểu thức điều kiện 2>)
    <Câu lệnh thực thi 2>
else if (<biểu thức điều kiện 3>)
    <Câu lệnh thực thi 3>
else
    <Câu lệnh thực thi 4>
```

Câu lệnh: **if-else lồng nhau**

Cú pháp - có nhiều cách biểu diễn


```
if (<biểu thức điều kiện 1>)    <Câu lệnh thực thi 1>
else if (<biểu thức điều kiện 2>) <Câu lệnh thực thi 2>
    else if (<biểu thức điều kiện 3>) <Câu lệnh thực thi 3>
        else    <Câu lệnh thực thi 4>
```

```
if (<biểu thức điều kiện 1>)
    <Câu lệnh thực thi 1>
else if (<biểu thức điều kiện 2>)
    <Câu lệnh thực thi 2>
else if (<biểu thức điều kiện 3>)
    <Câu lệnh thực thi 3>
else
    <Câu lệnh thực thi 4>
```

Câu lệnh: **if-else lồng nhau**

Cú pháp – ví dụ

Phong cách:
Canh lề thẳng đứng



```
#include <stdio.h>
#include <stdlib.h>
int main(){
    float diem = 8.7f;

    if(diem < 5.0f)
        printf("%s", "Yeu");
    else if(diem < 6.5f)
        printf("%s", "Trung Binh");
    else if(diem < 8.5f)
        printf("%s", "Kha");
    else if(diem < 9.5f)
        printf("%s", "Gioi");
    else
        printf("%s", "Xuat sac");

    return 0;
}
```

Câu lệnh: **if-else** lồng nhau

Cú pháp – ví dụ

Phong cách:

Canh lề thẳng đứng

Trường hợp dùng dấu
tạo khối { và }

```
if(diem < 5.0f){  
    printf("%s", "Yeu");  
}  
else if(diem < 6.5f){  
    printf("%s", "Trung Binh");  
}  
else if(diem < 8.5f){  
    printf("%s", "Kha");  
}  
else if(diem < 9.5f){  
    printf("%s", "Gioi");  
}  
else{  
    printf("%s", "Xuat sac");  
}
```



cuu duong than cong . com

cuu duong than cong . com

KỸ THUẬT

Câu lệnh switch-case

Ứng dụng

- Khi chương trình có một số công việc sẽ được thực thi tương ứng những trường hợp (tình huống) xác định trước.
 - Số công việc: hữu hạn & đếm được.
 - Các tình huống: hữu hạn & đếm được
- Ví dụ: chương trình quản lý sách
 - Chương trình cung cấp những menu sau cho người dùng
 - Đọc dữ liệu từ tập tin
 - Nhập dữ liệu vào chương trình
 - Tìm cuốn sách
 - Lấy danh sách các người mượn chưa trả sách
 - V.v.
 - Số các chức năng (công việc) trên là hữu hạn

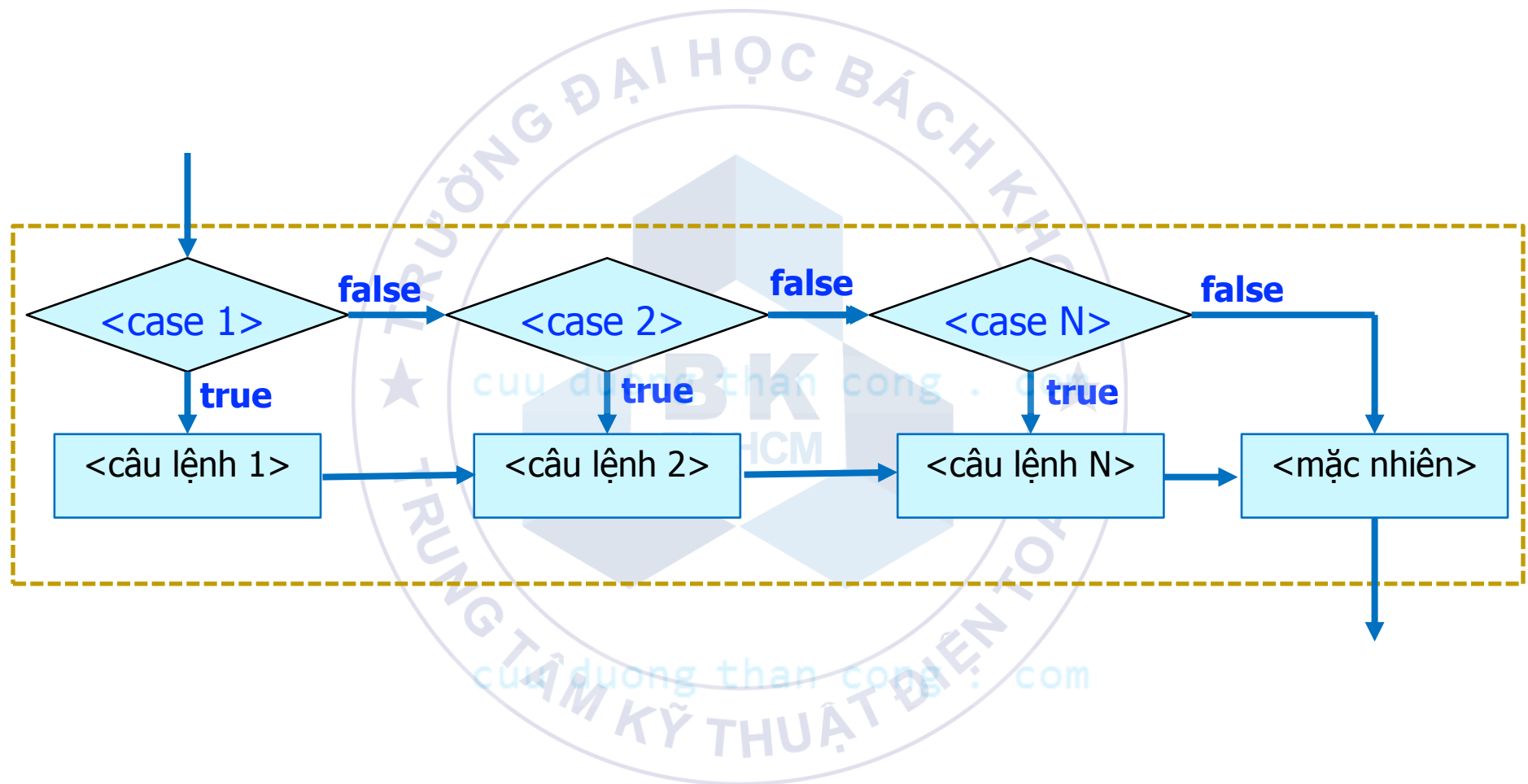
Câu lệnh switch-case

Ứng dụng

- Ví dụ: chương trình quản lý sách
 - Chương trình có thể in ra danh sách các chức năng này cho người dùng chọn
 - Giao diện đồ họa: hiển thị đồ họa thay cho in ra màn hình
 - Khi người dùng chọn một menu
 - Chương trình thực hiện khối công việc tương ứng với menu đã chọn
 - => Khối công việc được thực hiện tương ứng với trường hợp cụ thể
 - => Phù hợp cấu trúc **switch-case**

Câu lệnh switch-case

Ý tưởng



Câu lệnh switch-case

Ý tưởng

Cách thực hiện câu lệnh **switch-case**:

- ❑ Chương trình kiểm tra xem trường hợp nào xảy ra trong số các trường hợp được liệt kê: **<case 1>, <case 2>, .., <case N>**
- ❑ Nếu trường hợp thứ i xảy ra ($i = 1 \dots N$):
 - ❑ Thực hiện lần lượt các câu lệnh từ i đến N (**<câu lệnh i >** → **<câu lệnh N >**), kể cả câu lệnh mặc nhiên **<mặc nhiên>**
 - ❑ Nếu câu lệnh đang thực thi là **break**; chương trình thoát khỏi cấu trúc **switch-case** (nhảy đến câu lệnh liền sau **switch-case**)
- ❑ Nếu không có trường hợp nào xảy ra
 - ❑ Chương trình thực thi câu lệnh mặc nhiên **<mặc nhiên>** và thoát khỏi cấu trúc **switch-case**

Câu lệnh switch-case

Cú pháp

```
switch (<mã trường hợp>){  
case <mã 1>: <câu lệnh 1>  
case <mã 2>: <câu lệnh 2>  
  
case <mã N>: <câu lệnh N>  
default: <câu lệnh mặc nhiên>  
}
```

switch, case, default: Từ khoá

<mã trường hợp>:

PHẢI LÀ biểu thức có một trong các kiểu sau đây

(1) Các kiểu **số nguyên**, hoặc dẫn xuất từ nó thông qua **typedef**

(2) Kiểu **enum**

<mã i>: ($i=1, \dots, N$), các giá trị có thể của mã trường hợp

Câu lệnh switch-case

Cú pháp

```
switch (<mã trường hợp>){  
case <mã 1>:  
case <mã 2>:  
case <mã 3>: <câu lệnh 3>  
  
case <mã N>: <câu lệnh N>  
default: <câu lệnh mặc nhiên>  
}
```

Trường hợp muốn:

<câu lệnh 3> được thực thi cho cả 3 trường hợp 1,2, và 3

Câu lệnh switch-case

Cú pháp

```
switch (<mã trường hợp>){  
case <mã 1>: <câu lệnh 1> break;  
case <mã 2>: <câu lệnh 2>  
  
case <mã N>: <câu lệnh N>  
default: <câu lệnh mặc nhiên>  
}
```

Trường hợp muốn:

<câu lệnh 1> khi được thực thi xong thì thoát khỏi cấu trúc **switch-case** luôn, không thực thi các câu lệnh kế tiếp.

Câu lệnh switch-case

Cú pháp

```
switch (<mã trường hợp>){  
case <mã 1>: <câu lệnh 1> break;  
case <mã 2>: <câu lệnh 2> break;  
  
case <mã N>: <câu lệnh N> break;  
default: <câu lệnh mặc nhiên>  
}
```

Trường hợp muốn:

Muốn mỗi câu lệnh được thực thi với chỉ trường hợp tương ứng.

Câu lệnh switch-case

Cú pháp

```
switch (<mã trường hợp>){  
case <mã 1>: <câu lệnh 1> break;  
case <mã 2>: <câu lệnh 2> break;  
  
case <mã N>: <câu lệnh N> break;  
}
```

Trường hợp muốn:

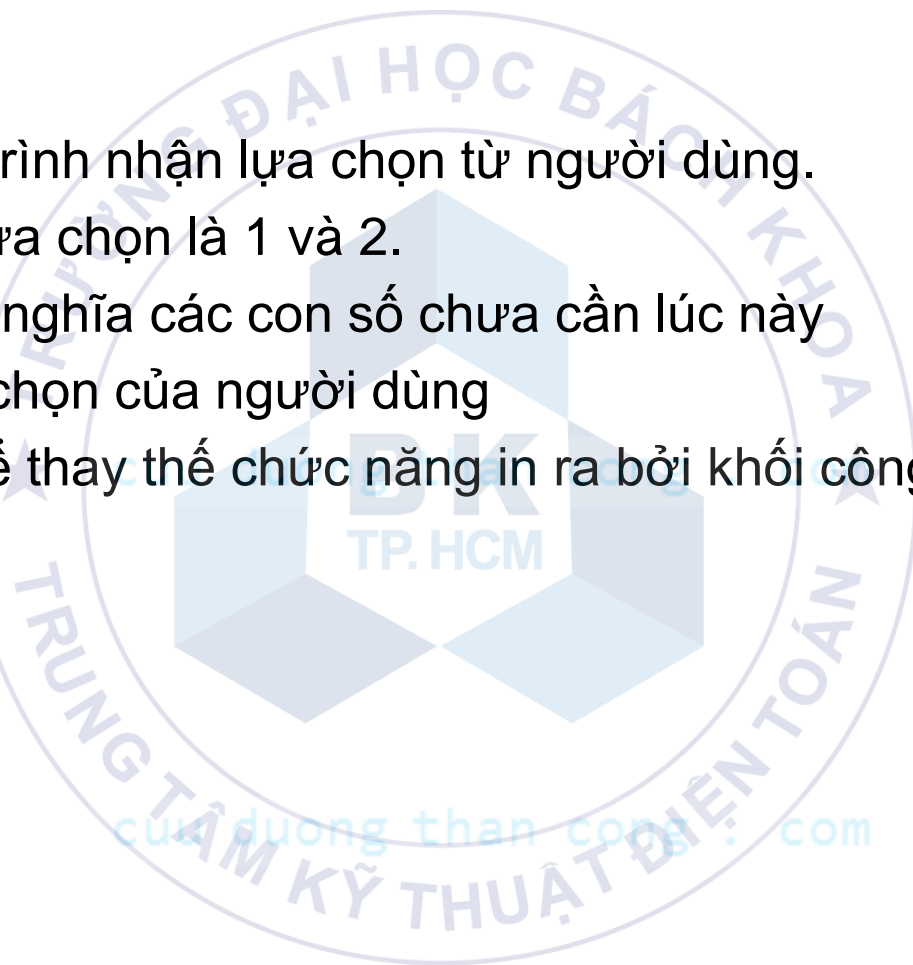
Muốn mỗi câu lệnh được thực thi với chỉ trường hợp tương ứng.
Không có cả phần thực thi mặc nhiên.

Câu lệnh switch-case

Ví dụ

■ Bài toán:

- Chương trình nhận lựa chọn từ người dùng.
 - Các lựa chọn là 1 và 2.
 - Ý nghĩa các con số chưa cần lúc này
- In ra lựa chọn của người dùng
 - Có thể thay thế chức năng in ra bởi khối công việc sau này.



Câu lệnh switch-case

Ví dụ

■ Ví dụ

```
#include <stdlib.h>
#include <stdio.h>
int main(){
    int luachon;
    printf("Nhap vao lua chon: \n");
    scanf("%d", &luachon);
    switch (luachon){
        case 1:
            printf("Truong hop 1\n"); printf("Hanh dong 1\n");
        case 2:
            printf("Truong hop 2\n"); printf("Hanh dong 2\n");
        default:
            printf("Hanh dong mac dinh\n");
    }
    return 0;
}
```


Câu lệnh switch-case

Ví dụ

■ Ví dụ

```
#include <stdlib.h>
#include <stdio.h>
int main(){
    int luachon;
    printf("Nhap vao lua chon: \n");
    scanf("%d", &luachon);
    switch (luachon){
    case 1:
        printf("Truong hop 1\n"); printf("Hanh dong 1\n");
    case 2:
        printf("Truong hop 2\n"); printf("Hanh dong 2\n");
    default:
        printf("Hanh dong mac dinh\n");
    }
    return 0;
}
```

Chương trình in ra tên của cả hai hành động
Vì không có lệnh **break**;

Câu lệnh chuyển đổi

■ Ví dụ

```
#include <stdlib.h>
#include <stdio.h>
int main(){
    int luachon;
    printf("Nhap vao lua chon: \n");
    scanf("%d", &luachon);
    switch (luachon){
        case 1:
            printf("Truong hop 1\n"); printf("Hanh dong 1\n"); break;
        case 2:
            printf("Truong hop 2\n"); printf("Hanh dong 2\n"); break;
        default:
            printf("Hanh dong mac dinh\n");
    }
    return 0;
}
```

break: khiến chương trình chỉ in ra tên hành động tương ứng lựa chọn

Câu lệnh switch-case

Ví dụ

- Một ví dụ khác về menu:
 - Có sử dụng `enum`



Kiểu liệt kê và câu lệnh chuyển đổi

■ Ví dụ

```
#include <stdlib.h>
#include <stdio.h>
enum Luachon {DongY, TuChoi, ChuaQuyetDinh};
int main(){
    enum Luachon luachon;
    printf("Nhap vao lua chon cua ban \n");
    printf("0. Ban dong y \n");
    printf("1. Ban tu choi \n");
    printf("2. Ban chua co quyet dinh \n");
    scanf("%d", &luachon);

    switch (luachon){
        case DongY: printf("Ban dong y\n"); break;
        case TuChoi: printf("Ban tu choi y\n"); break;
        case ChuaQuyetDinh: printf("Ban chua co quyet dinh\n"); break;
        default: printf("Ban khong nhap lua chon dung\n");
    }
    return 0;
}
```

Câu lệnh switch-case

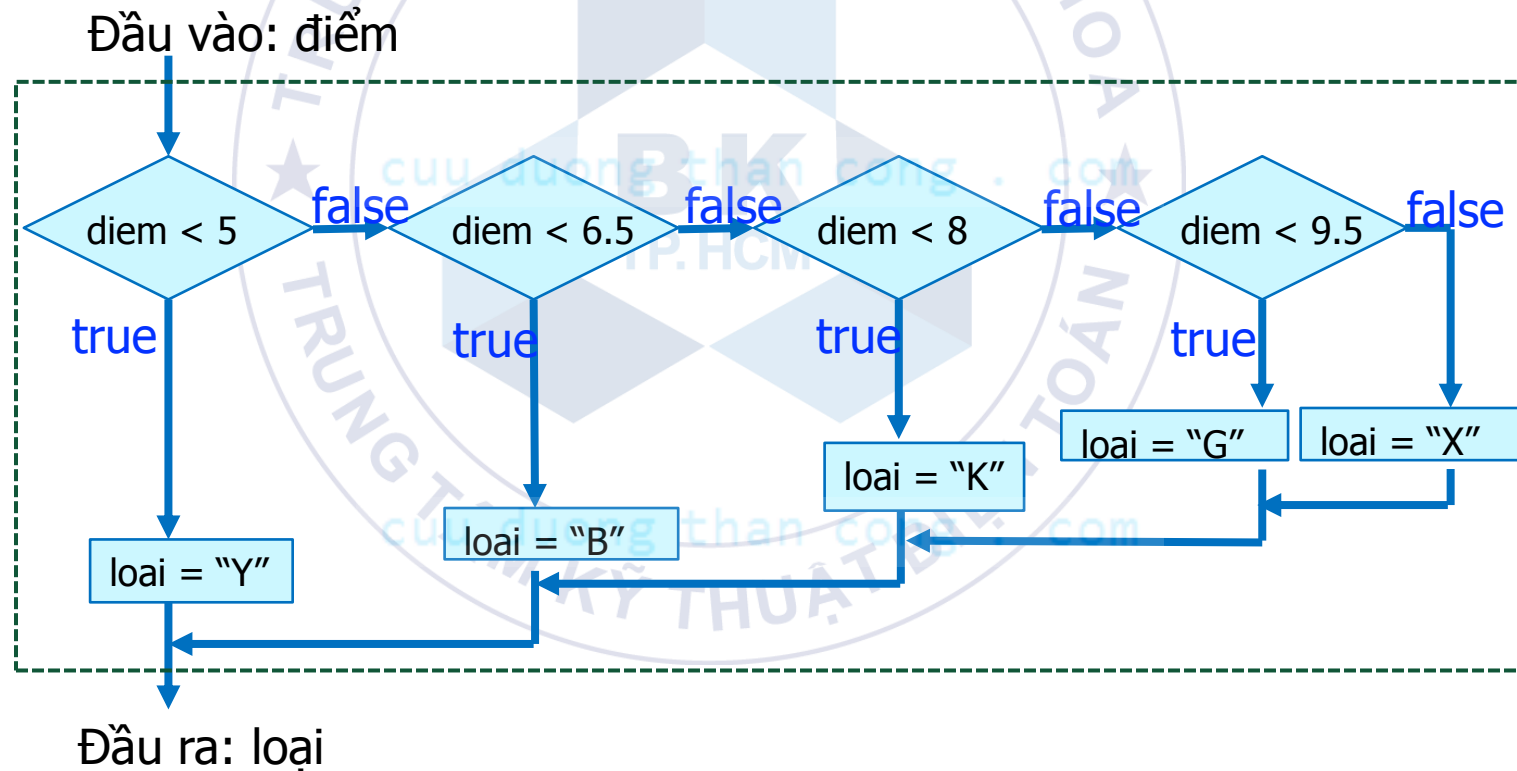
Bài tập

- Bài toán: Quản lý kho hàng
 - Có các chức năng
 - Nhập hàng hoá
 - Lưu hàng hoá
 - Đọc hàng hoá từ tập tin
 - V.v
 - Chương trình in ra menu và chờ người dùng chọn (nhập từ bàn phím)
 - Với mỗi lựa chọn
 - In ra tên chức năng đã chọn như nói trên
 - Sinh viên có thể phát triển các chức năng đầy đủ sau này.

So sánh if – else với switch-case

- Tại sao không dùng switch-case cho bài toán phân loại sinh viên theo điểm

Điểm không phải kiểu nguyên



So sánh `if - else` với `switch`

- Câu lệnh `switch` có thể được biểu diễn lại bằng một trình tự các câu lệnh `if-else`
- Một số trường hợp `switch` tường minh và dễ hiểu hơn.
- Tất cả các cấu trúc điều khiển có thể được biểu diễn lại bằng các câu lệnh `if-else` và các câu lệnh `goto` (Với sự hỗ trợ của các nhãn)

cuu duong than cong . com

cuu duong than cong : com

Tổng kết

- Vận dụng được nguyên tắc phân rã bài toán để giải quyết vấn đề
 - Xem các ví dụ về Giải PTB2 và các ví dụ khác
- Hiểu và vận dụng được các câu lệnh điều khiển của C
 - Nguyên lý của việc thực thi có điều kiện
 - Câu lệnh if-else, câu lệnh điều kiện lồng nhau
 - Câu lệnh switch-case

