

CHƯƠNG 3

CÁC CẤU TRÚC LỆNH ĐIỀU KHIỂN



3.1

CẤU TRÚC PHÂN NHÁNH

3.2

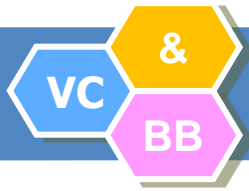
CẤU TRÚC LẶP FOR

3.3

CẤU TRÚC LẶP WHILE VÀ DO...WHILE

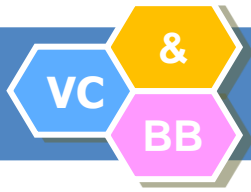
3.4

CÂU LỆNH BREAK, CONTINUE



Nội dung

- 3.1 **CẤU TRÚC PHÂN NHÁNH**
- 3.1.1 **Toán tử if**
- 3.1.2 **Toán tử else if**



Đặt vấn đề

Nếu tôi có nhà mặt phố và xe bốn bánh thì sẽ lấy chân dài
Nếu tôi có điểm tbc = 4.0 thì tôi được xếp hạng giỏi



a. If thiếu

❖ Cú pháp

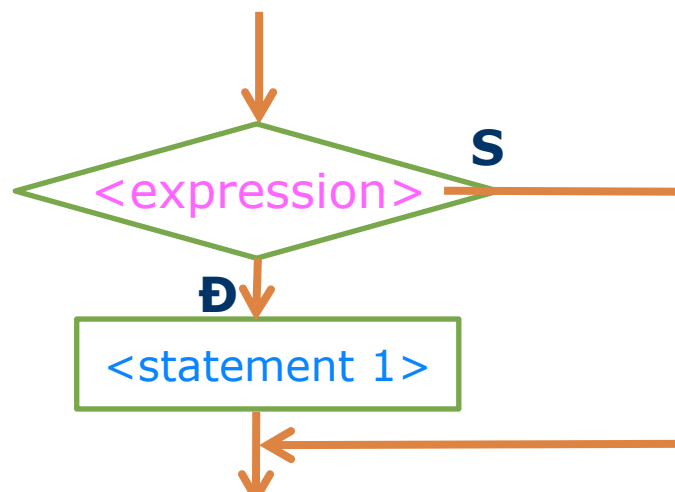
```
if (expression)
    statement 1;
```

```
if (expression)
{
    statement 1;
    statement 2;
    ...
}
```

❖ Ý nghĩa:

- Expression: biểu thức điều kiện
- Statement: lệnh
- Nếu expression đúng thì statement được thực hiện. Nếu sai thì bỏ qua lệnh

❖ Sơ đồ khối



Câu lệnh if (thiếu)

```
vd 1: if (a > 0)
        printf("a lon hon 0");
```

```
vd 2:
    if (a == 0)
    {
        printf("a bang 0");
        a = 2912;
    }
```

Câu lệnh if (thiếu)

```
vd 1: if (a > 0)
    printf("a lon hon 0");
```

Nếu a=5

• ???

a lon hon 0

Nếu a=-5

• ???

Bỏ qua lệnh

Câu lệnh if (thiếu)

```
vd 2:  
    if (a == 0)  
    {  
        printf("a bang 0");  
        a = 2912;  
    }
```

Nếu a=0

• ?????

a bang 0
a=2912

Nếu a=-5

• ?????

Bỏ qua lệnh

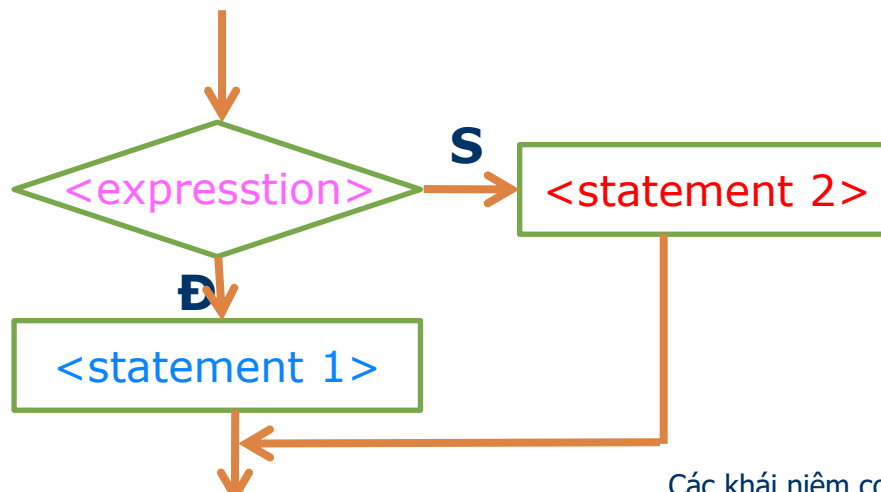
b. If đủ: if ... else❖ **Cú pháp**

```
if (expression)
    statement 1;
else
    statement 2;
```

```
if (expression)
{
    statement block 1;
}
else
{
    statement block 2;
}
```

❖ **Ý nghĩa:**

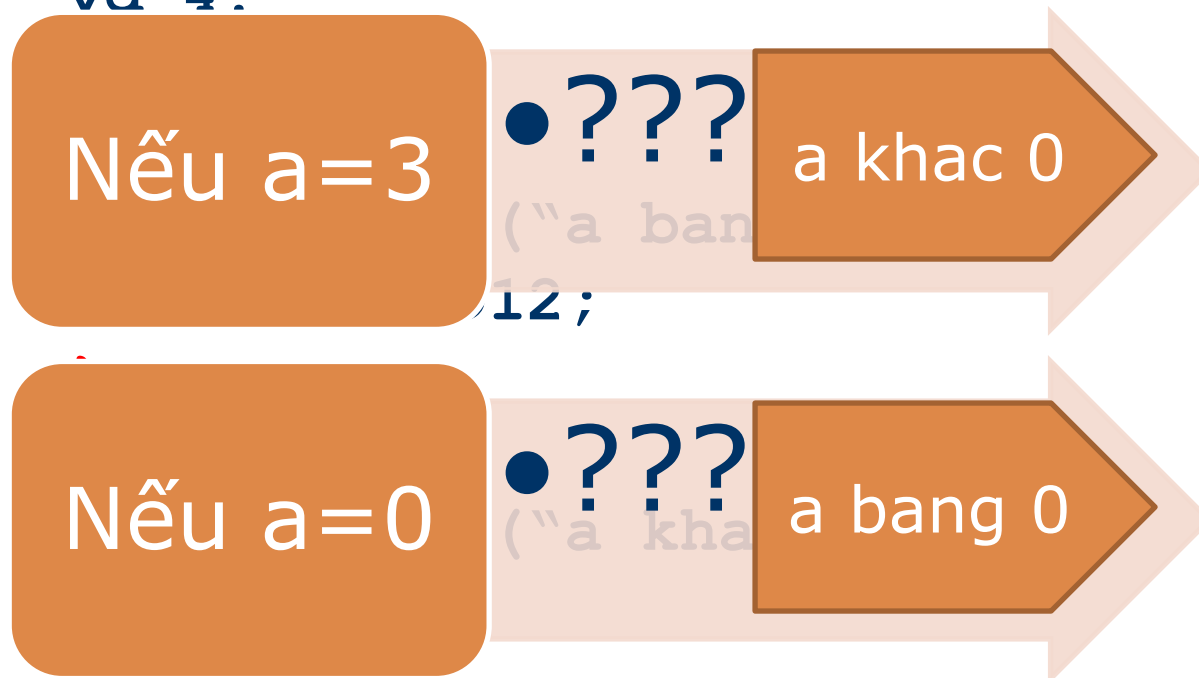
- Expression: biểu thức điều kiện
- Statement: lệnh, block: khối
- Nếu expression đúng thì statement 1 được thực hiện. Nếu sai thì statement 2 thực hiện

❖ **Sơ đồ khối**

Câu lệnh if (đủ)

```
vd 3: if (a == 0)
    printf("a bang 0");
else
    printf("a khac 0");
```

vd 4:



c. if lồng vào nhau

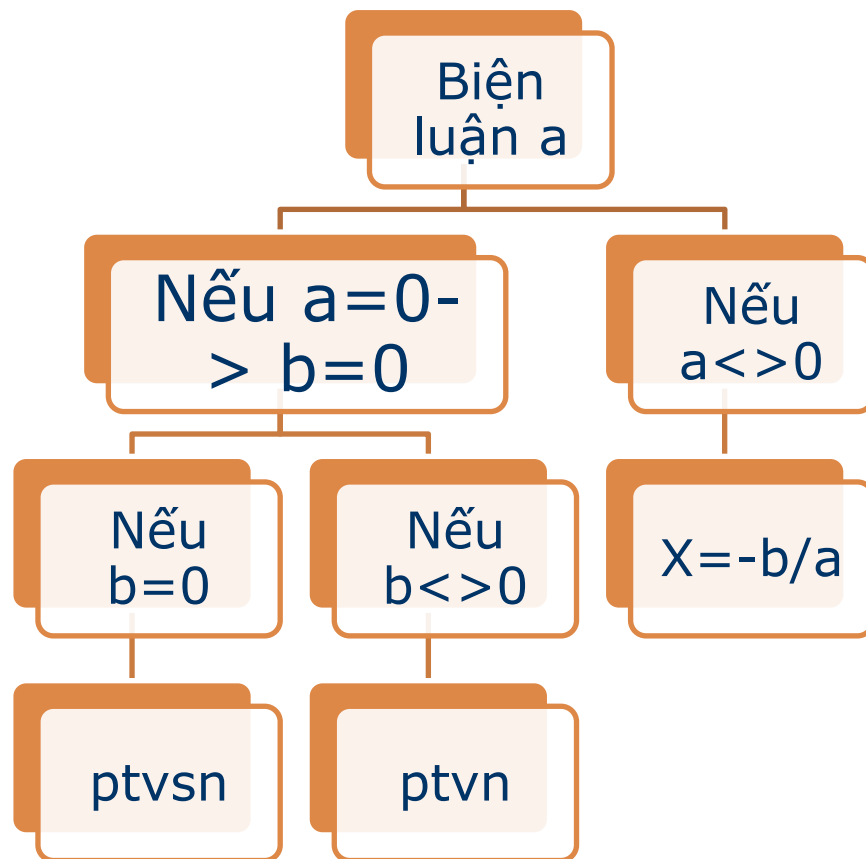
là một lệnh **if** được đặt trong một lệnh **if** hoặc **else** khác

Cú pháp:

```
if( biểu thức điều kiện 1)
{
    if(biểu thức điều kiện 2)
    {
        lệnh;
    }
}
```

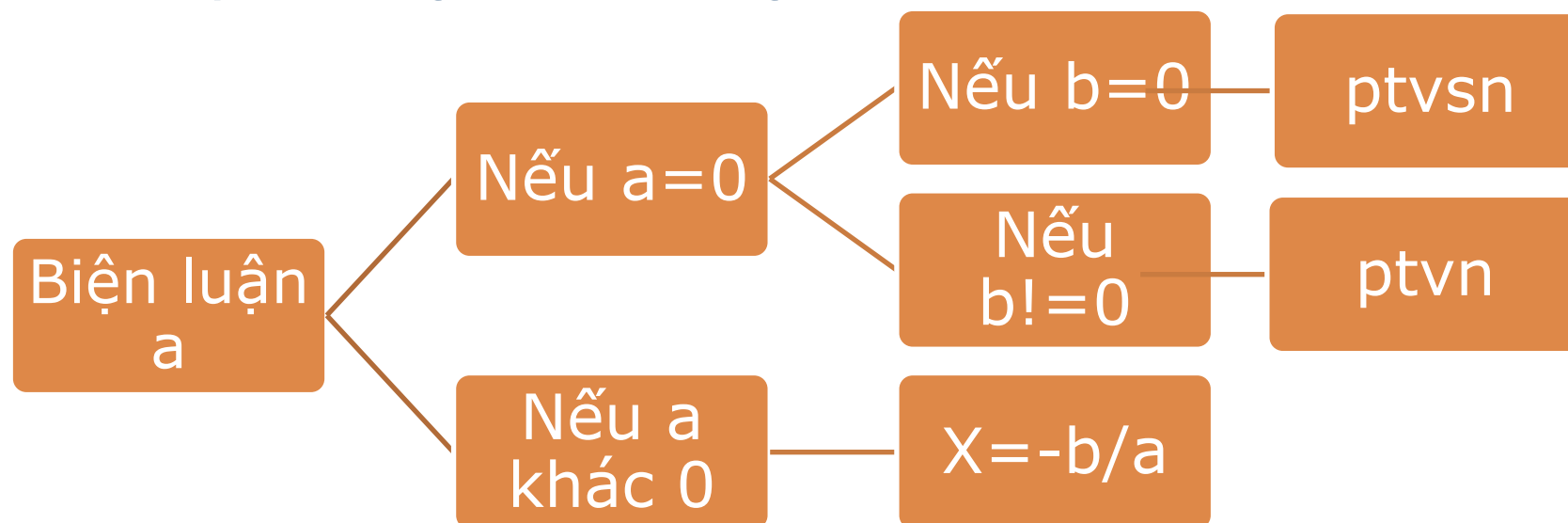
Ví dụ 5

❖ Giải phương trình dạng $ax + b = 0$



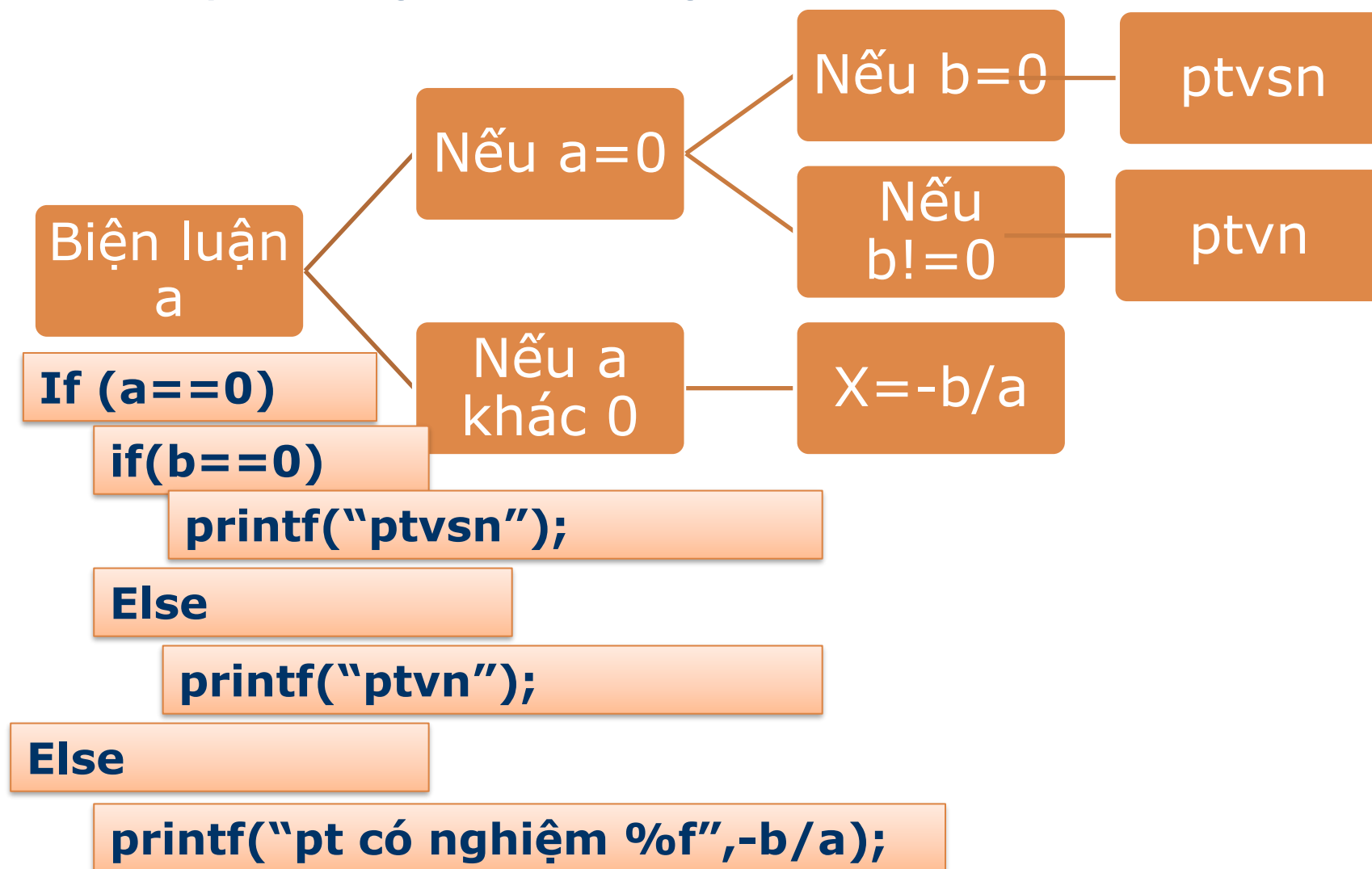
Ví dụ 5

❖ Giải phương trình dạng $ax + b = 0$



Ví dụ 5

❖ Giải phương trình dạng $ax + b = 0$



Có dạng if...else if

Cú pháp

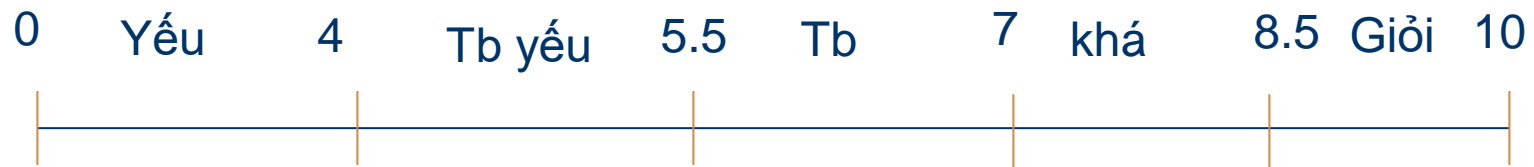
```
if(Biểu_Thức_Điều_Kiện_1)
{
    Lệnh 1;
} else if(Biểu_Thức_Điều_Kiện_2)
{
    Lệnh 2;
} else if(Biểu_Thức_Điều_Kiện_N-1)
{
    Lệnh N-1;
} else
{
    Lệnh N;
}
```

Hoạt động

Nếu biểu thức điều kiện I đúng thì thực hiện lệnh thứ i ($i=1..n-1$)
Nếu xét đến biểu thức điều kiện n -1 sai thì thực hiện lệnh n

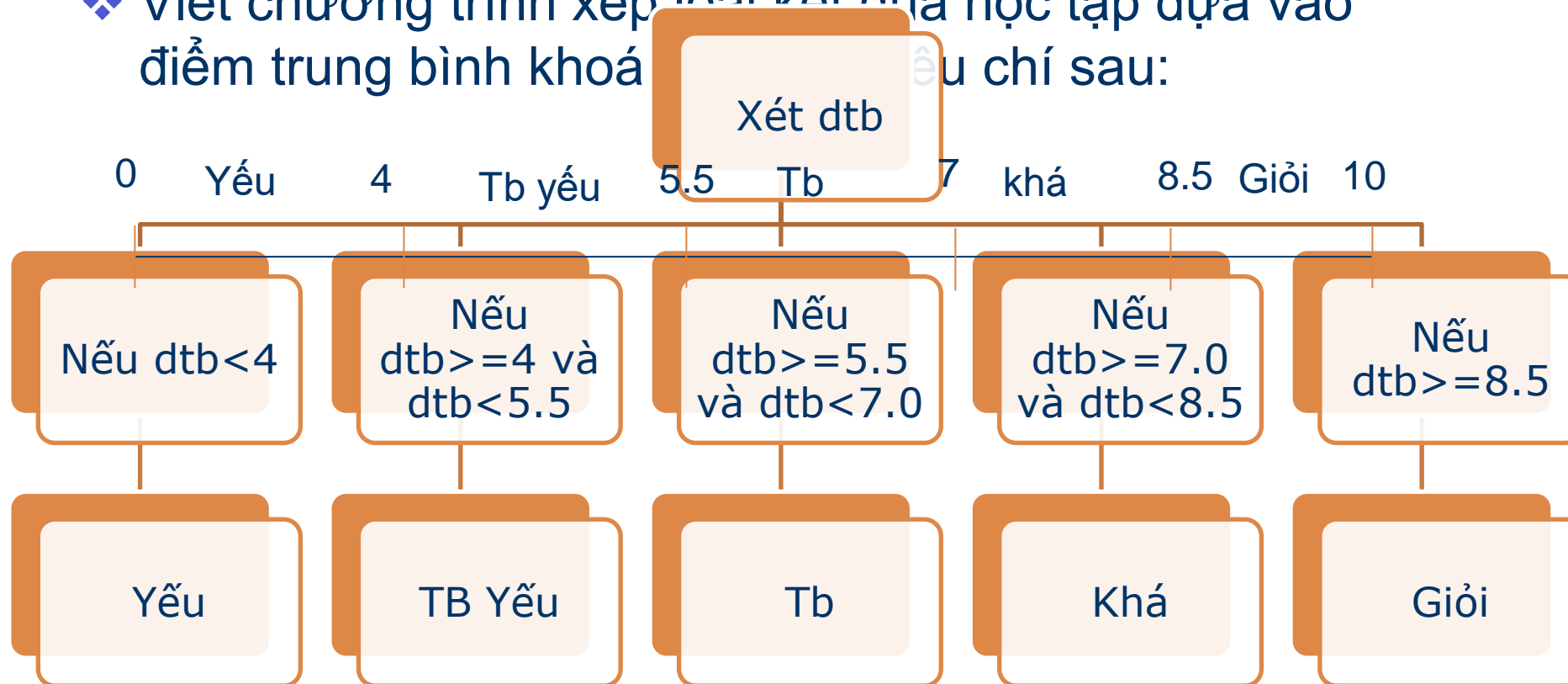
Ví dụ 6:

- ❖ Viết chương trình xếp loại kết quả học tập dựa vào điểm trung bình khoá học theo tiêu chí sau:



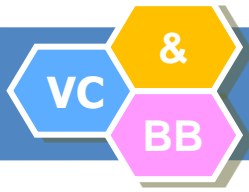
Ví dụ 6:

- ❖ Viết chương trình xếp loại kết quả học tập dựa vào điểm trung bình khoá học theo yêu cầu chỉ sau:



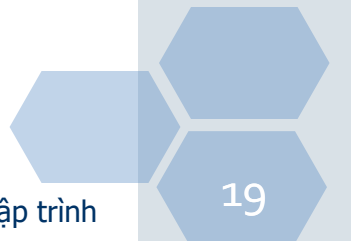
Ví dụ 6:

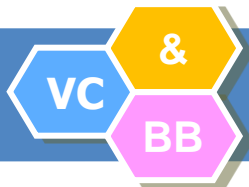
```
//Trường hợp thứ nhất
if(dtb<4) {
    printf("Loại yếu");
//Trường hợp thứ hai
} else if(dtb<5.5) {
    printf("Loại trung bình yếu");
//Trường hợp thứ ba
} else if(dtb<7) {
    printf("Loại tb");
//Trường hợp thứ tư
} else if(dtb<8.5) {
    printf("Loại khá");
//Trường hợp thứ năm
} else {
    printf("Loại giỏi");
}
```



Bài tập

Bài 1: Giải phương trình bậc 2: $ax^2+bx+c=0$





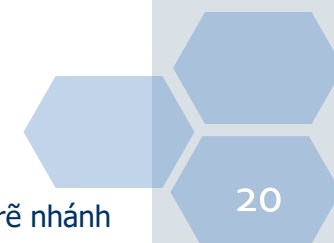
CẤU TRÚC PHÂN NHÁNH



Toán tử switch



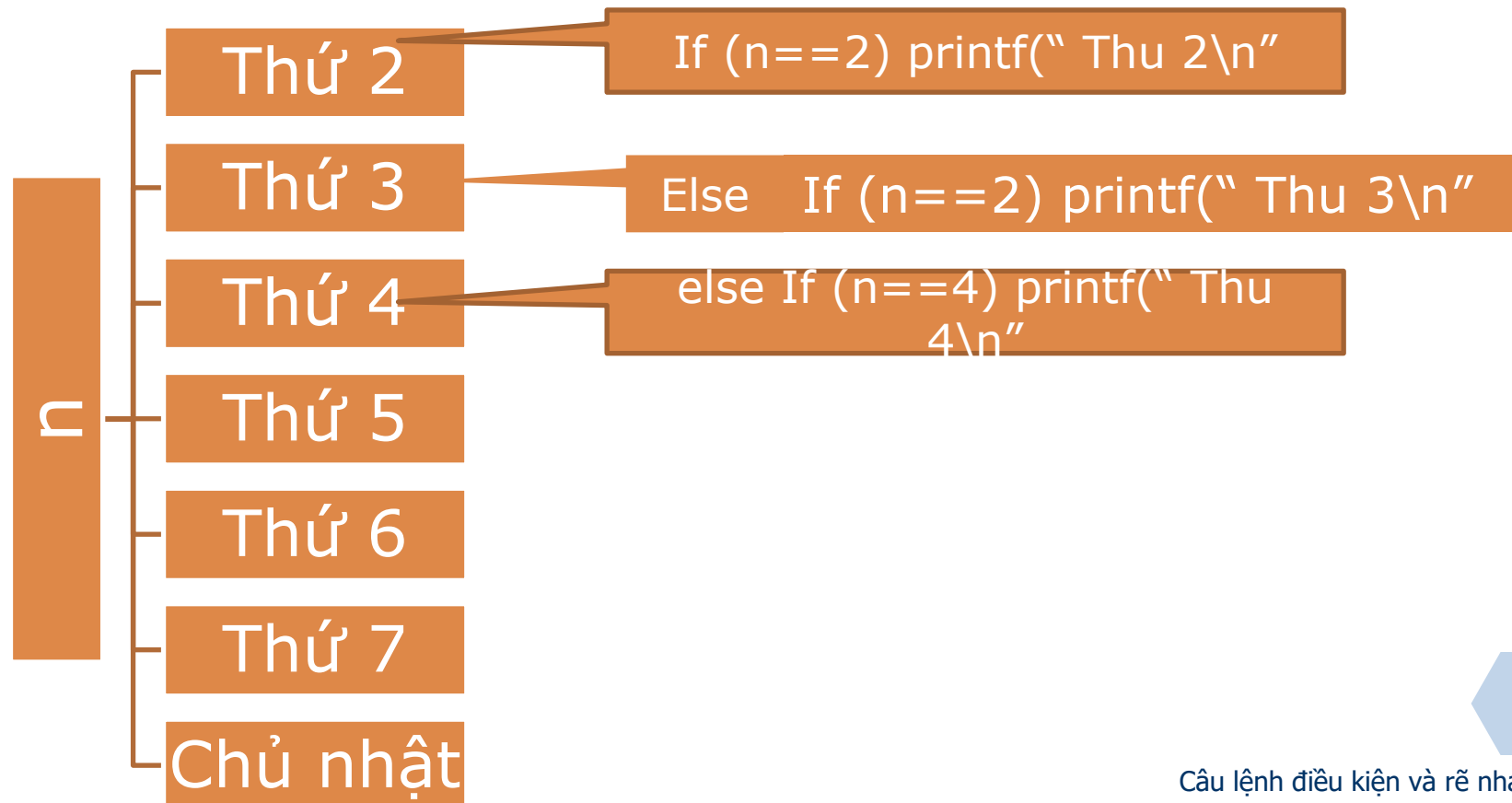
Toán tử goto và nhãn

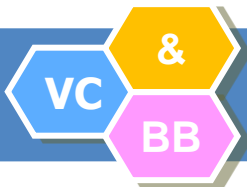


Đặt vấn đề

Kiểm tra bài cũ

Nhập vào một số nguyên, kiểm tra xem đó là thứ mấy trong tuần

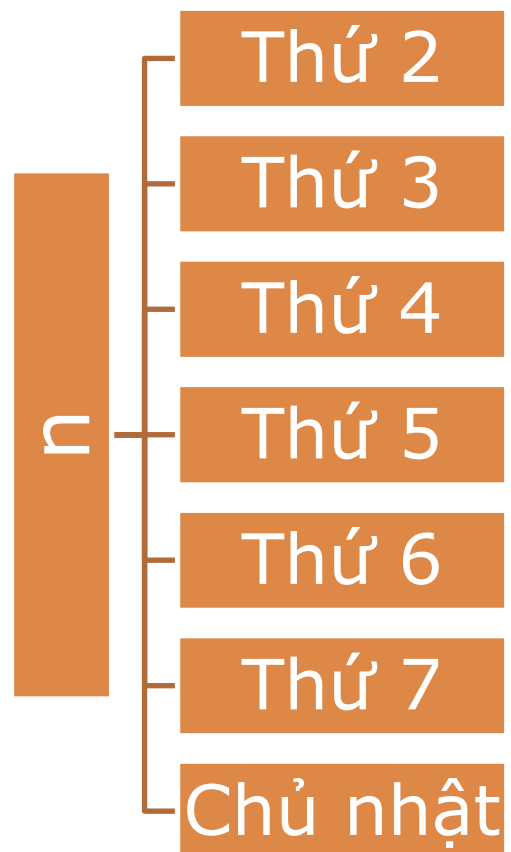




Đặt vấn đề

Kiểm tra bài cũ

Nhập vào một số nguyên, kiểm tra xem đó là thứ mấy trong tuần



Kiểm tra bài cũ

```
#include<stdio.h>
#include<conio.h>
main()
{
int n;
printf("nhap n= ");
scanf("%d",&n);
if (n==2) printf ("thu 2\n");
else if (n==3) printf ("thu 3\n");
        else if (n==4) printf ("thu 4\n");
                else if (n==5) printf ("thu 5\n");
                        else if (n==6) printf ("thu 6\n");
                                else if (n==7) printf ("thu 7\n");
                                        else printf ("cn\n");

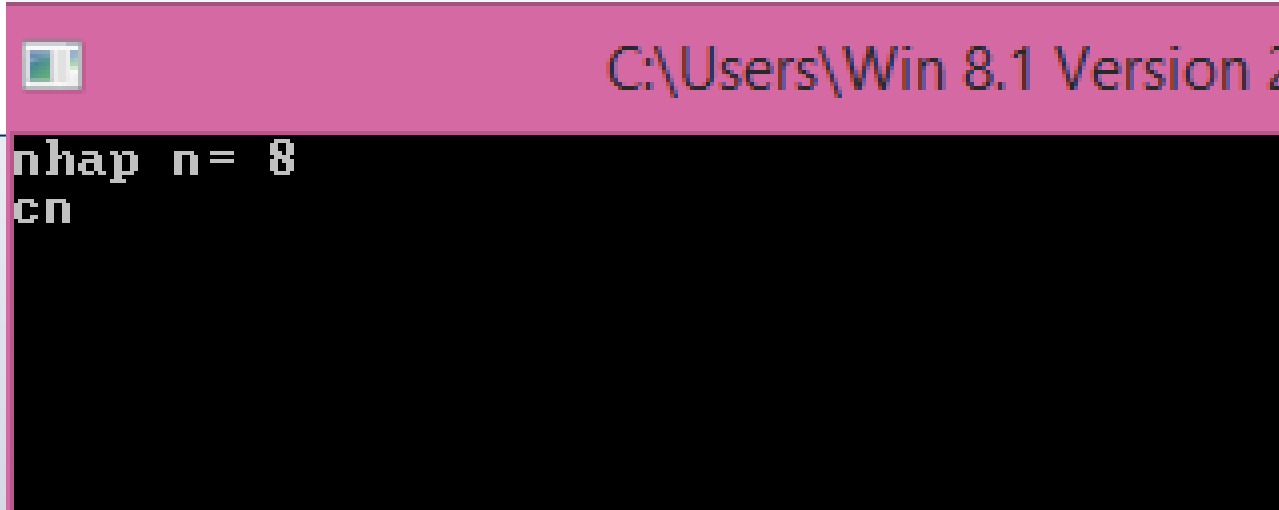
getch();
}
```

Đặt vấn đề

Kiểm tra bài cũ

```
#include<stdio.h>
#include<conio.h>
main()
{
    int n;
    printf("nhap n= ");
    scanf("%d",&n);
    if (n==2) printf ("thu 2\n");
    else if (n==3) printf ("thu 3\n");
        else if (n==4) printf ("thu 4\n");
            else if (n==5) printf ("thu 5\n");
                else if (n==6) printf ("thu 6\n");
                    else if (n==7) printf ("thu 7\n");
                        else printf ("cn\n");

    getch();
}
```



```
C:\Users\Win 8.1 Version 2
nhap n= 8
cn
```


Cú pháp

```
Switch (biến/biểu thức)
{
    case <gt1>: lệnh 1;break;
    case <gt2>: lệnh 2;break;
    ....
    case <gtn>: lệnh n;break;
    [default: lệnh n;]
}
```

Trong đó

- biến/ biểu thức phải trả về là hằng nguyên hoặc kí tự
- gt1, gt2,..., gtn là các biểu thức hằng nguyên hoặc ký tự và chúng phải khác nhau.

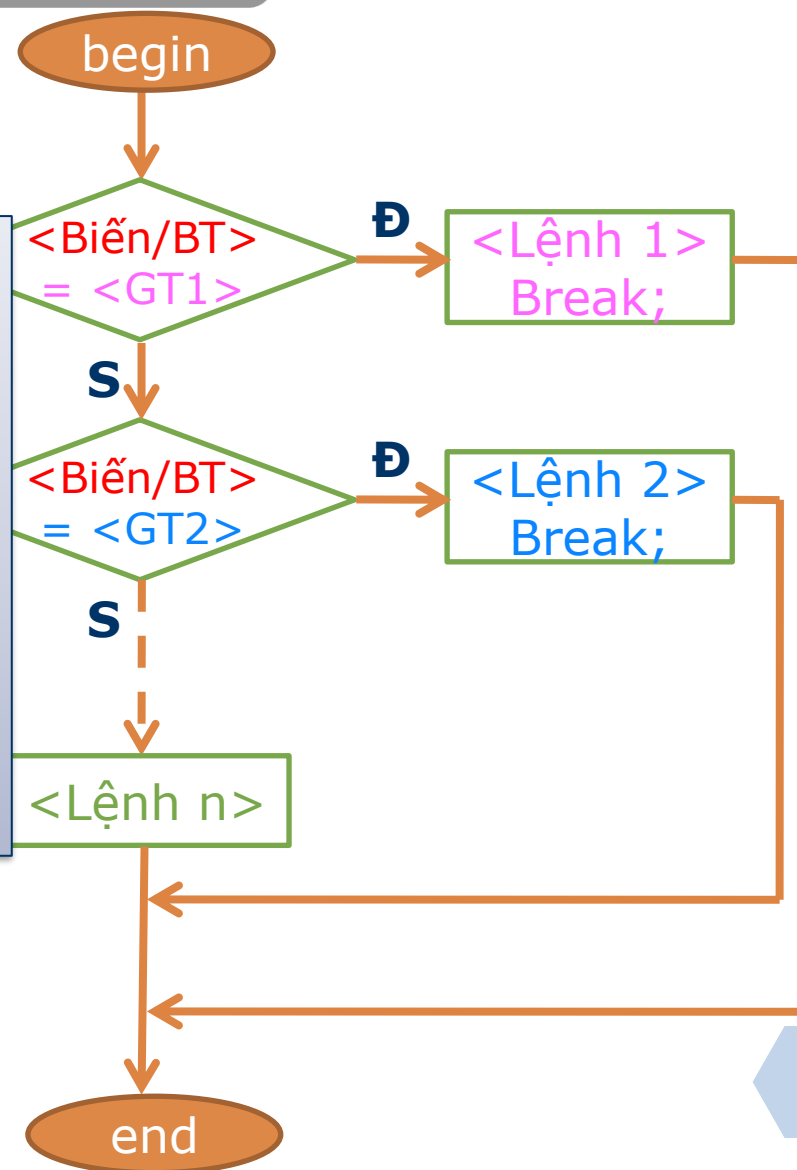


Cú pháp

Switch (biến/biểu thức)

```
{  
  case <gt;1>: lệnh 1; break;  
  case <gt;2>: lệnh 2; break;  
  ....  
  case <gt;n>: lệnh n; break;  
  [default: lệnh n;]  
}
```

Sơ đồ khối





Cú pháp

Switch (biến/biểu thức)

```
{  
    case <gt1>: lệnh 1;break;  
    case <gt2>: lệnh 2;break;  
    ....  
    case <gtn>: lệnh n;break;  
    [default: lệnh n;]  
}
```

Hoạt động

Đầu tiên, máy sẽ kiểm tra biến/bt =gt1.

Nếu đúng thì thực hiện các lệnh nằm trong case đó và gặp lệnh break thì thoát ngay khỏi switch, bỏ qua các case còn lại.

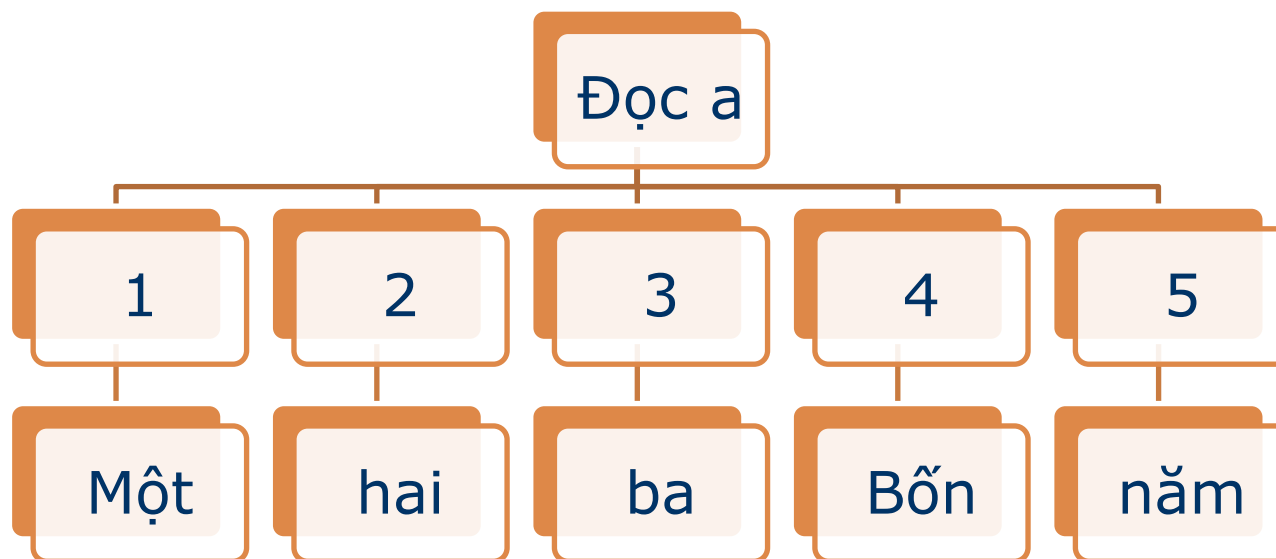
Nếu sai thì kiểm tra biến/bt với gt2...

Tương tự cho đến gtn.

Nếu biến/bt không bằng với bất kỳ giá trị nào trong gt1, gt2,..., gtn thì Lệnh n sẽ được thực hiện nếu có thành phần default.

Ví dụ 1

Nhập vào 1 số từ 1->5 và in ra cách đọc tương ứng:



Ví dụ 1

Nhập vào 1 số từ 1->5 và in ra cách đọc tương ứng:

❖ Câu lệnh if

```
if (a == 1)
    printf("Mot");
if (a == 2)
    printf("Hai");
if (a == 3)
    printf("Ba");
if (a == 4)
    printf("Bon");
if (a == 5)
    printf("Nam");
```

Ví dụ 1

Nhập vào 1 số từ 1->5 và in ra cách đọc tương ứng:

❖ Câu lệnh if

```
if (a == 1)
    printf("Mot");
if (a == 2)
    printf("Hai");
if (a == 3)
    printf("Ba");
if (a == 4)
    printf("Bon");
if (a == 5)
    printf("Nam");
```

❖ Câu lệnh switch

```
switch (a)
{
    case 1:    printf("Mot");
               break;
    case 2:    printf("Hai");
               break;
    case 3:    printf("Ba");
               break;
    case 4:    printf("Bon");
               break;
    case 5:    printf("Nam");
}
}
```

❖ Câu lệnh switch là có thể lồng nhau.

Ví dụ 2:

{

```
switch (a)
```

```
{
```

```
    case 1 : printf("Mot"); break;
```

```
    case 2 : switch (b)
```

```
    {
```

```
        case 1 : printf("A"); break;
```

```
        case 2 : printf("B"); break;
```

```
    } break;
```

```
    case 3 : printf("Ba"); break;
```

```
    default : printf("Khong biet doc");
```

```
}
```

}

❖ Các giá trị trong mỗi trường hợp phải **khác nhau**.

Ví dụ 3

```
switch (a)
{
    case 1 : printf("Mot"); break;
    case 1 : printf("MOT"); break;
    case 2 : printf("Hai"); break;
    case 3 : printf("Ba"); break;
    case 1 : printf("1"); break;
    case 1 : printf("mot"); break;
    default : printf("Khong biet doc");
}
```


❖ Nếu các lệnh giống nhau thì viết gộp;

Ví dụ 4:

```
switch (a)
{
    case 1 : printf("So le"); break;
    case 2 : printf("So chan"); break;
    case 3 : printf("So le"); break;
    case 4 : printf("So chan"); break;
}

switch (a)
{
    case 1 :
    case 3 : printf("So le"); break;
    case 2 :
    case 4 : printf("So chan"); break;
}
```

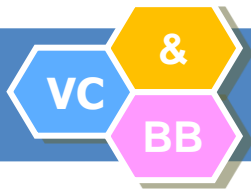
Câu lệnh điều kiện và rẽ nhánh

☐ **Cú pháp**

- ☐ goto nhãn;
- ☐ nhãn : dòng lệnh;

☐ **Hoạt động**

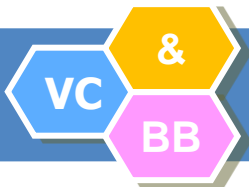
- ☐ Khi gặp lệnh goto chương trình sẽ nhảy đến lệnh nơi chứa từ nhãn và thực hiện lệnh sau từ khóa nhãn



Ví dụ 5:

```
int i=0;  
    quaylai: printf("i= %d", i, "\t");  
    i=i+1;  
    if (i<10)  
        goto quaylai;
```

- ❖ Nếu lệnh **goto** và “nhãn” nằm trong một hàm thì lệnh **goto** chỉ cho phép “nhảy” từ vị trí này sang vị trí khác trong thân của hàm (không được “nhảy” từ hàm này sang hàm khác).



Bài tập

Bài 1: Nhập vào tháng và năm. Cho biết tháng đó có bao nhiêu ngày.

Bài 2: Nhập độ dài 3 cạnh 1 tam giác. Kiểm tra đó có phải là tam giác không và là tam giác gì?



3.2

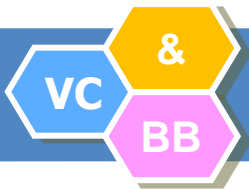
CẤU TRÚC VÒNG LẶP FOR

3.2.1

Cấu trúc vòng lặp for

3.2.2

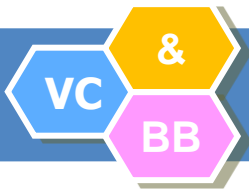
Ví dụ



Đặt vấn đề

❖ Kiểm tra bài cũ

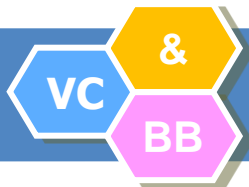
- Viết chương trình in ra 10 dòng “Chào bạn”
Viết chương trình in ra 10000 dòng “Chào bạn”



Đặt vấn đề

❖ Đáp án:

```
printf("Chào bạn\n");  
printf("Chào bạn\n");  
printf("Chào bạn\n");  
printf("Chào bạn\n");  
printf("Chào bạn\n");  
printf("Chào bạn\n");  
printf("Chào bạn\n");  
printf("Chào bạn\n");  
printf("Chào bạn\n");  
printf("Chào bạn\n");
```

Đặt vấn đề

Giải pháp

Sử dụng
vòng lặp

Nhanh, gọn,
dễ sửa chữa

Cấu trúc lặp

- **Cấu trúc lặp** (repetition or loop): lặp lại 1 hay nhiều lệnh cho đến khi biểu thức điều kiện có giá trị sai. Các cấu trúc lặp gồm:
 - for
 - while
 - do ... while.

I

❑ Cú pháp:

```
for (Exp1; Exp2; Exp3)  
statement;
```

❑ Giải thích:

Exp= Expression

Statement: lệnh

Exp1: khai báo và khởi tạo giá trị ban đầu cho biến

Exp2: biểu thức điều kiện

Exp3: cập nhật biến vòng lặp, thay đổi giá trị biến vòng lặp

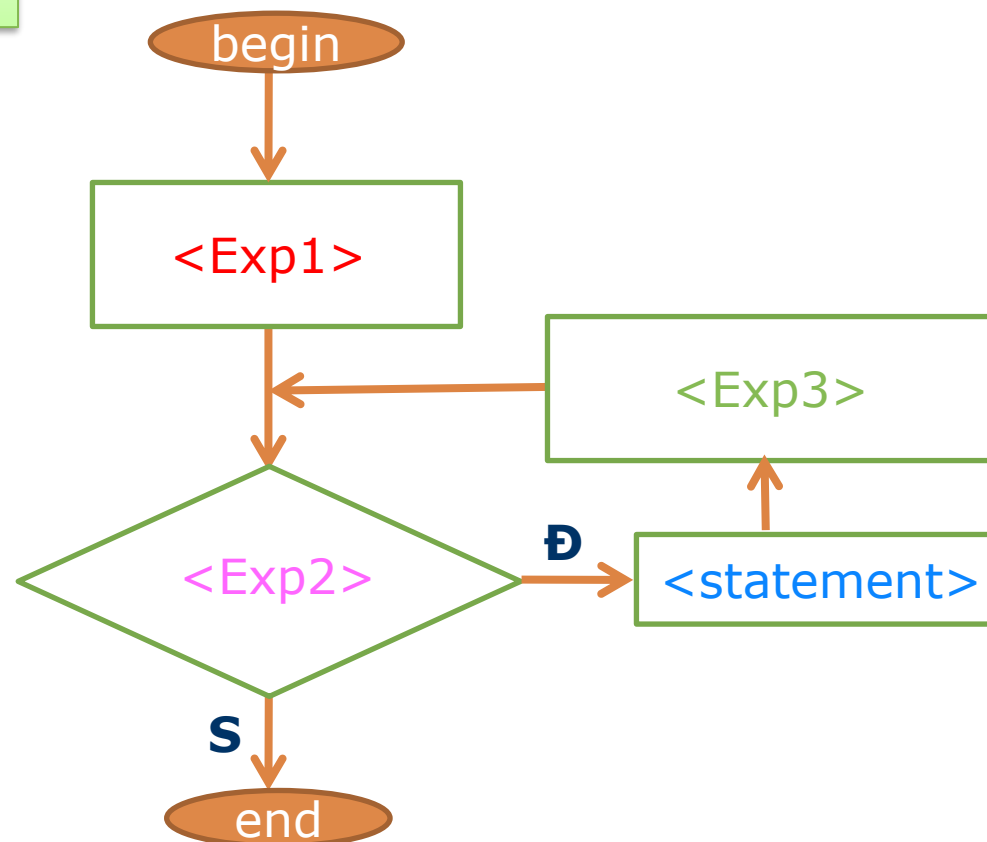
Ví dụ : `i: =1`

Ví dụ :
`i<=10`

Ví dụ `i++`

□ Cú pháp:

```
for (Exp1; Exp2; Exp3)  
statement;
```

□ Sơ đồ khối

☐ **Cú pháp:**

```
for (Exp1; Exp2; Exp3)  
statement;
```

☐ **Hoạt động:**☐ **B1: thực hiện exp1**☐ **B2: kiểm tra điều kiện exp2.**☐ **Nếu đúng thì thực hiện lệnh và sang bước 3.**☐ **Nếu sai sang bước 4**☐ **B3: thực hiện exp3 và quay lại b2**☐ **B4: thoát khỏi vòng lặp**

Ví dụ 1: In ra các số từ 0-9

```
i=0 in ra i  
i=1 in ra i  
i= 2 in ra i  
...  
i=9 in ra i
```

Exp1: $i=0$

Exp 2: $i < 10 \rightarrow$ Lệnh: in i

Exp 3: $i++$

Ví dụ 1: In ra các số từ 0-9

```
i=0 in ra i  
i=1 in ra i  
i= 2 in ra i  
...  
i=9 in ra i
```

Exp1: $i=0$

Exp 2: $i < 10 \rightarrow$ Lệnh: `in i`

Exp 2: $i++$

```
int i;  
for (i = 0; i < 10; ++i)  
    printf("%d\n", i);
```



Ví dụ 1: In ra các số từ 0-9

```
i=0 in ra i  
i=1 in ra i  
i= 2 in ra i  
...  
i=9 in ra i
```

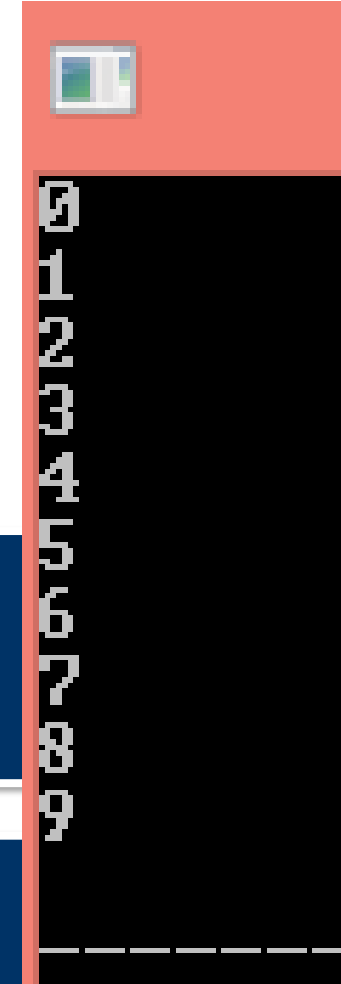
Exp1: i=0

Exp 2: i<10 → Lệnh: in i

Exp 2: i++

```
Ví dụ 2:  int i;  
          for (i = 0; i < 10; i=i+1)  
            printf("%d\n", i);
```

```
Ví dụ 3:  int i;  
          for (i = 0; i < 10; i+=1)  
            printf("%d\n", i);
```



Một số lưu ý

Câu lệnh for lồng vào nhau

Ví dụ 4:

	0	1	2	3	4	5	6	7	8	9
0	*	*	*	*	*	*	*	*	*	*
1	*	*	*	*	*	*	*	*	*	*
2	*	*	*	*	*	*	*	*	*	*
3	*	*	*	*	*	*	*	*	*	*
4	*	*	*	*	*	*	*	*	*	*

Một số lưu ý

Câu lệnh for lồng vào nhau

Ví dụ 4:

0	0	1	2	3	4	5	6	7	8	9
	*	*	*	*	*	*	*	*	*	*
1	*	*	*	*	*	*	*	*	*	*
2	*	*	*	*	*	*	*	*	*	*
3	*	*	*	*	*	*	*	*	*	*
4	*	*	*	*	*	*	*	*	*	*



```
for (int i = 0; i < 5; i++)
```

```
{  
    for(int j = 0; j < 10; j++)
```

```
        printf("* ");
```

```
}
```

Câu lệnh for lồng vào nhau

Ví dụ 4:

0	0	1	2	3	4	5	6	7	8	9
1	*	*	*	*	*	*	*	*	*	*
2	*	*	*	*	*	*	*	*	*	*
3	*	*	*	*	*	*	*	*	*	*
4	*	*	*	*	*	*	*	*	*	*



```
for (int i = 0; i < 5; i++)
```

```
{  
  for(int j = 0; j < 10; j++)
```

```
    printf("* ");
```

```
    printf("\n");
```

```
}
```

❖ Trong câu lệnh for, có thể sẽ không có phần <Khởi đầu>

```
int i;  
int i;  
i=0;  
for (; i < 10; i++)  
    printf("%d\n", i);
```

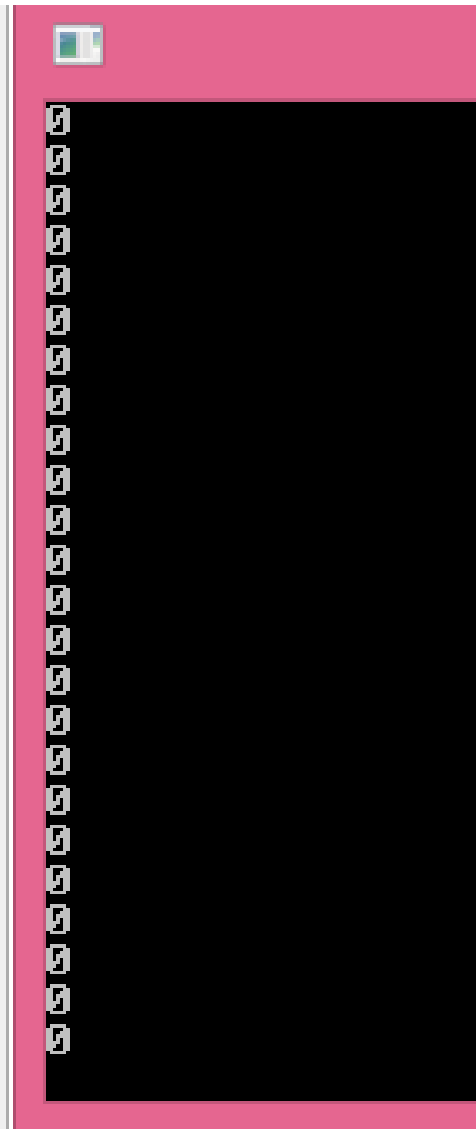
- ❖ Trong câu lệnh for, có thể sẽ không có phần <Đ/K lặp>

```
int i;  
for (i = 0;; i++)  
    printf("%d\n", i);
```

```
for (i = 0; ; i++)  
{  
    if (i <= 10)  
        printf("%d\n", i);  
}
```

❖ Trong câu lệnh for, có thể sẽ
<Bước nhảy>

```
int i;  
for (i = 0; i < 10; )  
{  
    printf("%d\n", i)  
    i=i+1;  
}
```



❖ Trong câu lệnh for, có thể khuyết cả 3

```
int i;  
i=0;  
for ( ; ; )  
{  
    if(i<10)  
        printf("%d\n", i);  
    i++;  
}  
getch();  
}
```

❖ Không được thêm **;** ngay sau lệnh for.

```
for (i = 0; i < 10; i++) ;  
{  
    printf("%d", i) ;  
    printf("\n") ;  
}
```

```
for (i = 0; i < 10; i++)  
{  
};
```


Câu lệnh for - Một số lưu ý

- ❖ Các thành phần <Khởi đầu>, <Đ/K lặp>, <Bước nhảy> cách nhau bằng dấu ;
- ❖ Nếu có nhiều thành phần trong mỗi phần thì được cách nhau bằng dấu ,

```
for (int i = 1, j = 2; i + j < 10; i++, j += 2)
    printf("%d\n", i + j);
```

Viết chương trình giải bài toán cổ sau:

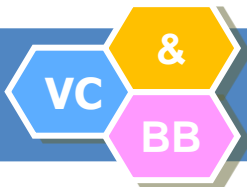
TRĂM TRÂU TRĂM CỎ
TRÂU ĐỨNG ĂN NĂM
TRÂU NẰM ĂN BA
LỤ KHỤ TRÂU GIÀ
BA TRÂU MỘT BÓ



HỎI MỖI LOẠI CÓ MẤY CON?

```
#include <stdio.h>

int main() {
    unsigned int x, y, z; /* khai báo số trâu đúng, trâu
    nằm, trâu già*/
    for( x=0;x<=20;x++){
        for(y=0;y<=33;y++){
            for(z=0;z<100;z+=3){
                if(x + y + z ==100 && (5*x + 3 *y + ( z /
3)) ==100) {
                    printf("\n Trâu đúng:%5d",x);
                    printf(" Trâu nằm:%5d ",y);
                    printf(" Trâu già:%5d", z);
                }
            }
        }
    }
```



Bài tập

Viết chương trình vẽ tam giác các dấu * như sau:

**

*

3.3

Cấu trúc lặp while và do...while

3.3.1

Cấu trúc lặp while

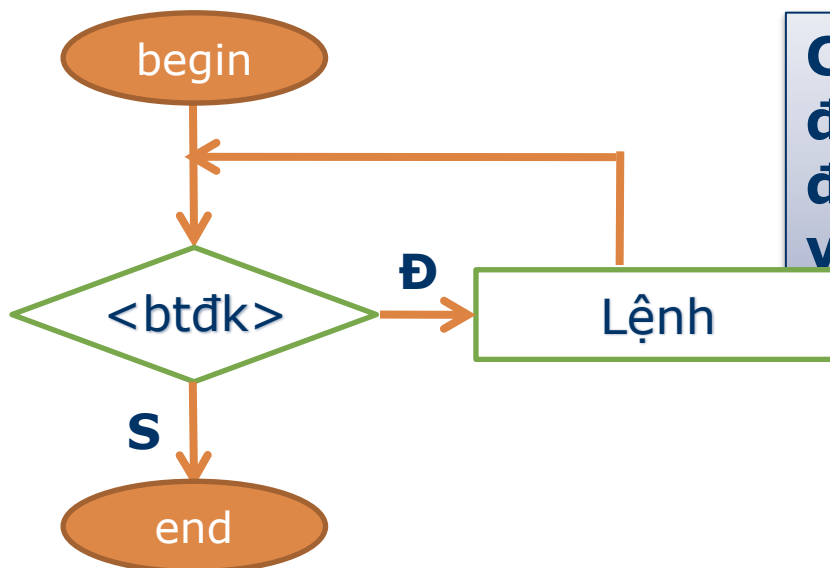
3.3.2

Cấu trúc lặp do... while

Cú pháp:

while (biểu thức điều kiện)
Lệnh;

Sơ đồ khối



Hoạt động

Chừng nào điều kiện còn đúng thì thực hiện lệnh, điều kiện sai thì thoát khỏi vòng lặp

Ví dụ 1:

In ra các số từ 0-9

```
int i;  
for (i = 0; i < 10; i++)  
    printf("%d\n", i);
```

```
int i = 0;
```

```
while (i < 10)
```

```
{  
    printf("%d\n", i);  
    i++;  
}
```



❖ Viết chương trình tính: $S = 1*2*...*n = n!$

$S=1; S=S*1;$
$S=S*2$
$S=S*3;$
$S=S*4;$
...
$S=S*(n-1)$
$S=S*n;$

Lặp S

❖ Viết chương trình tính: $S = 1*2*...*n = n!$

```
int s=1,n;  
printf("n= ");scanf("%d",&n);  
for(int i=1; i<=n ;i++)  
s=s*i;  
printf("s=1*2*3*...*%d= %d",n,s);
```

❖ Viết chương trình tính: $S = 1*2*...*n = n!$

```
int i=1;
int s=1,n;
printf("n= ");scanf("%d",&n);
while(i<=n)
{
    s=s*i;
    i++;}
printf("s=1*2*3*...*%d= %d",n,s);
```

Ví dụ 3

- ❖ 2. Nhập một số nguyên dương n . Xuất ra số ngược lại. Ví dụ: Nhập 1706 → Xuất 6071.

$n=1706$

Lấy 6 $\rightarrow n\%10$

Bỏ 6 ra khỏi $n \rightarrow n/10$. $n=170$

... tiếp tục cho đến khi $n=1$

Lấy 1 $\rightarrow n\%10$

Bỏ 1 ra khỏi $n \rightarrow n/10$. $n=0 \rightarrow$ dừng

Mỗi lần lặp
 $n\%10$
 $n/10$.

Ví dụ 3

- ❖ 2. Nhập một số nguyên dương n. Xuất ra số ngược lại. Ví dụ: Nhập 1706 → Xuất 6071.

n=1706

Lấy 6 -> $n \% 10$

Bỏ 6 ra khỏi n -> $n / 10$.

... tiếp tục cho đến khi n=1

Lấy 1 -> $n \% 10$

Bỏ 1 ra khỏi n -> $n / 10$.

```
int n, donvi;
printf("Nhap n: ");
scanf("%d", &n);
printf("So dao cua %d la ", n);
while (n > 0)
{
    donvi = n % 10;
    n = n / 10;
    printf("%d", donvi);
}
```

❖ Câu lệnh **while** có thể lồng nhau.

```
if (n < 10 && m < 20)
{
    while (n >= 1)
    {
        while (m >= 1)
        {
            printf("%d", m);
            m--;
        }
        n--;
    }
}
```

❖ Không được thêm **;** ngay sau lệnh while.

```
int n = 0;
while (n < 10) ;
{
    printf("%d\n", n);
    n++;
}

while (n < 10)
{
};
```

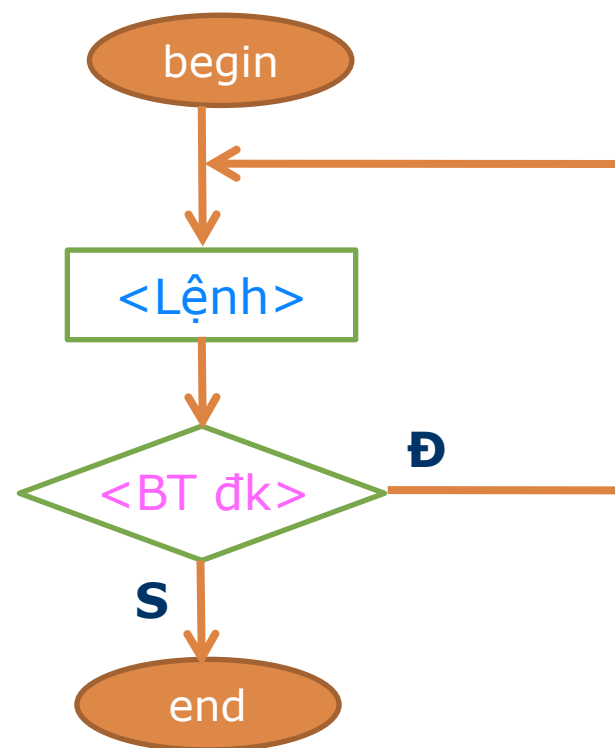
Cú pháp:

```
do
{
    Lệnh;
}
while (biểu thức điều kiện);
```

Hoạt động

B1: thực hiện lệnh
B2: kiểm tra điều kiện, chừng nào điều kiện còn đúng thì thực hiện lệnh, điều kiện sai thì thoát khỏi vòng lặp

Sơ đồ khối



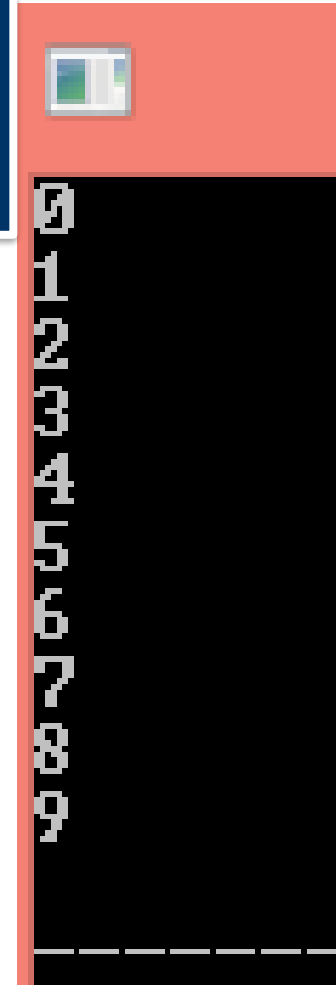
Ví dụ 1:

In các số từ 0-9

```
int i;  
for (i = 0; i < 10; i++)  
    printf("%d\n", i);
```

```
int i = 0;  
while (i < 10)  
{  
    printf("%d\n", i);  
    i++;  
}
```

```
int i = 0;  
do  
{  
    printf("%d\n", i);  
    i++;  
}  
while (i < 10);
```



Ví dụ 2

Nhập vào một số $n > 0$, đếm xem có bao nhiêu số chẵn $< n$

```
printf("so chan < %d la: ",n);
    for(int i=1;i<n;i++)
        if (i%2==0)
        {
            printf("%d",i);
            dem++;
        }
    printf("\nSo cac so chan la
%d",dem);

    getch();
}
```

```
printf("so chan < %d la: ",n);
    do
    {
        if (i%2==0)
        {
            printf("%d ",i);
            dem++;
        }
        i++;}
    while (i<n);
    printf("\nSo cac so chan la
%d",dem);
```

❖ Câu lệnh **do... while** có thể lồng nhau.

```
Ví dụ 4: int a = 1, b;  
do  
{  
    b = 1;  
    do  
    {  
        printf("%d\n", a + b);  
        b = b + 2;  
    }  
    while (b < 20);  
    a++;  
}  
while (a < 20);
```

Một số lưu ý

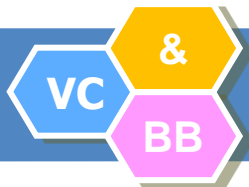
- ❖ Câu lệnh do... while sẽ được thực hiện ít nhất 1 lần

```
Ví dụ 5: int n;  
do  
{  
    printf("Nhap n: ");  
    scanf("%d", &n);  
}  
while (n < 1 || n > 100);  
}
```

Viết chương trình nhập vào từ bàn phím một số nguyên dương n . Trả ra màn hình kết quả tính:

- a. $S = 1^2 + 2^2 + \dots + n^2$
- b. $S = 1 + 1/2 + \dots + 1/n$
- c. $S = 1! + 2! + \dots + n!$





Nội dung

3.4

CÂU LỆNH BREAK, CONTINUE

3.4.1

Cấu lệnh break

3.4.2

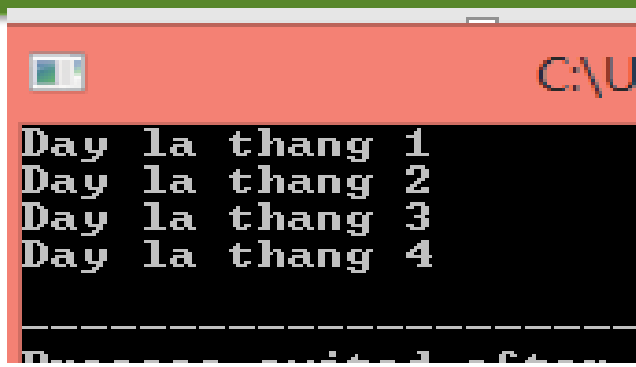
Câu lệnh continue

☐ Ý nghĩa

Lệnh **break** cho phép thoát ra khỏi vòng lặp chứa nó ngay lập tức.

Ví dụ 1:

```
for (int i=1; i < =12;i++ )  
{   if (i==5) break;  
    printf("Day la thang %d\n", i);  
}
```



```
C:\U  
Day la thang 1  
Day la thang 2  
Day la thang 3  
Day la thang 4  
-----  
Program terminated after
```

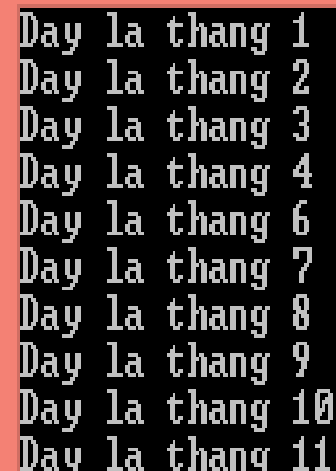
□ Ý nghĩa

khi gặp lệnh **continue**, chương trình sẽ bỏ qua các lệnh còn lại của lần lặp hiện tại, và quay lại thực hiện lần lặp tiếp theo của vòng lặp

*Lệnh **continue** chỉ áp dụng cho cấu trúc lặp, không áp dụng cho cấu trúc switch.*

Ví dụ 2:

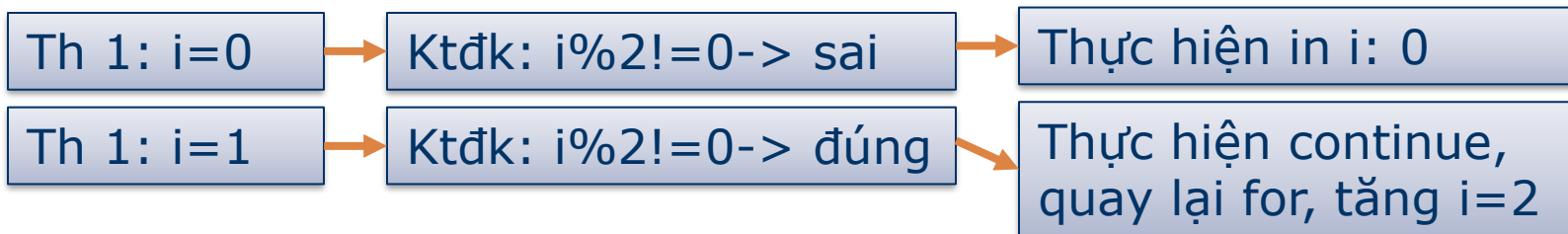
```
for (int i=1; i < =12;i++ )  
{   if (i==5) continue;  
    printf("Day la thang %d\n", i);  
}
```



```
Day la thang 1  
Day la thang 2  
Day la thang 3  
Day la thang 4  
Day la thang 6  
Day la thang 7  
Day la thang 8  
Day la thang 9  
Day la thang 10  
Day la thang 11
```

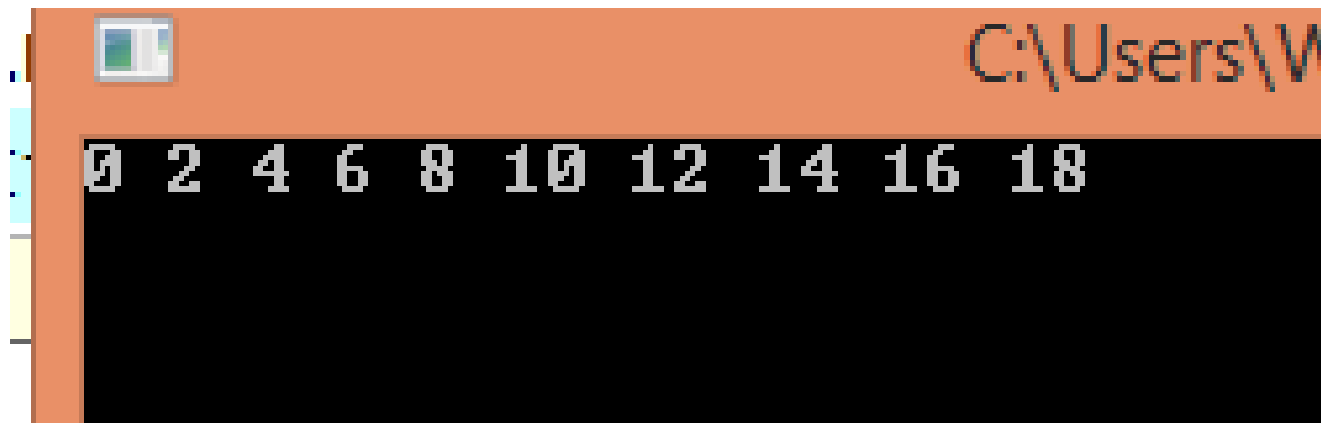
Ví dụ 3

```
5   for(int i=0;i<20;i++)  
6   {  
7       if (i%2!=0)//gap lenh continue chương trình  
8       //bo qua lenh dong 10 va tiep tục quay lại for  
9       continue;  
10      printf("i= %d\n",i);  
11  }
```



Ví dụ 3

```
5   for(int i=0;i<20;i++)  
6   {  
7       if (i%2!=0)//gap lenh continue chuong trinh  
8           //bo qua lenh dong 10 va tiep tục quay len for  
9       continue;  
10      printf("i= %d\n",i);  
11  }
```



```
C:\Users\W  
0  
2  
4  
6  
8  
10  
12  
14  
16  
18
```

Ví dụ 4:

Giả sử mật khẩu là 1. Đọc vào một mật khẩu người dung tối đa n lần. Nếu nhập đúng số 1 thì thoát không phải kiểm tra hết n lần. Nếu nhập không đúng số 1 báo nhập sai

```
#include<stdio.h>
#include<conio.h>

main()
{
    int pass, n;
    printf("Nhap n= "); scanf("%d",&n);
    for (int i=0;i<n;i++)
    {
        printf("Nhap pass= "); scanf("%d",&pass);
        if (pass==1) break;
        printf("Ban nhap sai\n");
    }
    getch();
}
```

Cho biết kết quả sau khi thực hiện đoạn code sau

```
int i, a=5;
for(i=0;i<=3;i++)
{
    if (i==2)
        continue;
    a+=1;
    if(i>1)
        break;
    printf("%d",a);
}
```

Bài tập về nhà

- ❖ Sinh viên làm các bài tập có mã dưới đây trên cổng thực hành.
- ❖ Mỗi sinh viên cần có ít nhất 5 bài đạt AC.
- ❖ Sinh viên không đạt sẽ bị trừ 1 điểm chuyên cần.
- ❖ SV không tiếp cận được cổng thực hành thì gửi mail cho GV để lấy đề bài và nộp bài qua mail.
- ❖ Deadline: 9/3

Mã bài tập	
C01003	C01015
C01004	C01048
C01005	C01050
C01006	C01066
C01014	C02001