



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# TIN HỌC ĐẠI CƯƠNG

Phần II: LẬP TRÌNH C

# Nội dung chính

- Chương 1: Tổng quan về ngôn ngữ C
- Chương 2: Kiểu dữ liệu và biểu thức trong C
- Chương 3: Vào ra dữ liệu
- **Chương 4: Cấu trúc điều khiển**
- Chương 5: Mảng, con trỏ và chuỗi ký tự
- Chương 6: Cấu trúc
- Chương 7: Hàm
- Chương 8: Tập dữ liệu

# Chương 4: Cấu trúc điều khiển

## 4.1. Lệnh khối

## 4.2. Lệnh rẽ nhánh

- Cấu trúc if, if ... else

## 4.3. Lệnh lựa chọn

## 4.4. Các lệnh lặp

- Vòng lặp for
- Vòng lặp while và do while

## 4.5. Các lệnh nhảy

- Câu lệnh continue
- Câu lệnh break

# Chương 4: Cấu trúc điều khiển

## 4.1. Lệnh khối

## 4.2. Lệnh rẽ nhánh

- Cấu trúc if, if ... else

## 4.3. Lệnh lựa chọn

## 4.4. Các lệnh lặp

- Vòng lặp for
- Vòng lặp while và do while

## 4.5. Các lệnh nhảy

- Câu lệnh continue
- Câu lệnh break

# Lệnh đơn và lệnh khối (compound)

- Lệnh đơn:
  - Là biểu thức theo sau bởi dấu ‘;’
  - Ví dụ: `x = 0; i++; printf("Hello");`
- Lệnh khối (lệnh ghép):
  - Là tập hợp các câu lệnh (đơn và ghép) được đặt trong cặp ngoặc nhọn { }
  - C cho phép khai báo biến trong một khối lệnh
    - Phần khai báo phải nằm trước các câu lệnh
  - Chú ý:
    - Lệnh ghép có thể đặt tại bất cứ chỗ nào mà cú pháp cho phép đặt 1 câu lệnh đơn
    - Không đặt dấu ‘;’ sau một khối lệnh đặt trong ngoặc nhọn

# Cấu trúc lồng nhau

- Trong lệnh ghép có thể chứa lệnh ghép khác
- Có thể khai báo biến trong khối lệnh

```
{//Khai báo đối tượng cục bộ trong khối
```

```
lệnh;
```

```
{//Khai báo đối tượng cục bộ trong khối
```

```
lệnh;
```

```
...
```

```
}
```

```
...
```

```
}
```

Nếu các đối tượng được khai báo trùng tên nhau ?

# Ví dụ

```
#include <stdio.h>
int main()
{ // ham main() cung la mot khoi lenh
    int c = 10, d = 20;
    printf(" Bien ngoai khoi c = %d; d = %d", c, d);
    {
        int c = 10;
        printf("\n Bien trong khoi c = %d; d = %d", c, d);
        printf("\n Gia tri cua cac bien duoc them 10 don vi");
        c = c + 10; d = d + 10;
        printf("\n Bien trong khoi c = %d; d = %d", c, d);
    }
    printf("\n Bien ra ngoai khoi c = %d; d = %d \n", c, d);
    return 0;
} // ket thuc khoi lenh cua ham main()
```

# Ví dụ → Kết quả thực hiện

```
C:\Users\lebavui\source\THDC\vd1.exe
Bien ngoai khoi c = 10; d = 20
Bien trong khoi c = 10; d = 20
Gia tri cua cac bien duoc them 10 don vi
Bien trong khoi c = 20; d = 30
Bien ra ngoai khoi c = 10; d = 30

-----
Process exited after 0.02203 seconds with return value 0
Press any key to continue . . .
```

## Biến địa phương / Biến toàn cục



# Chương 4: Cấu trúc điều khiển

## 4.1. Lệnh khối

## 4.2. Lệnh rẽ nhánh

- Cấu trúc if, if ... else

## 4.3. Lệnh lựa chọn

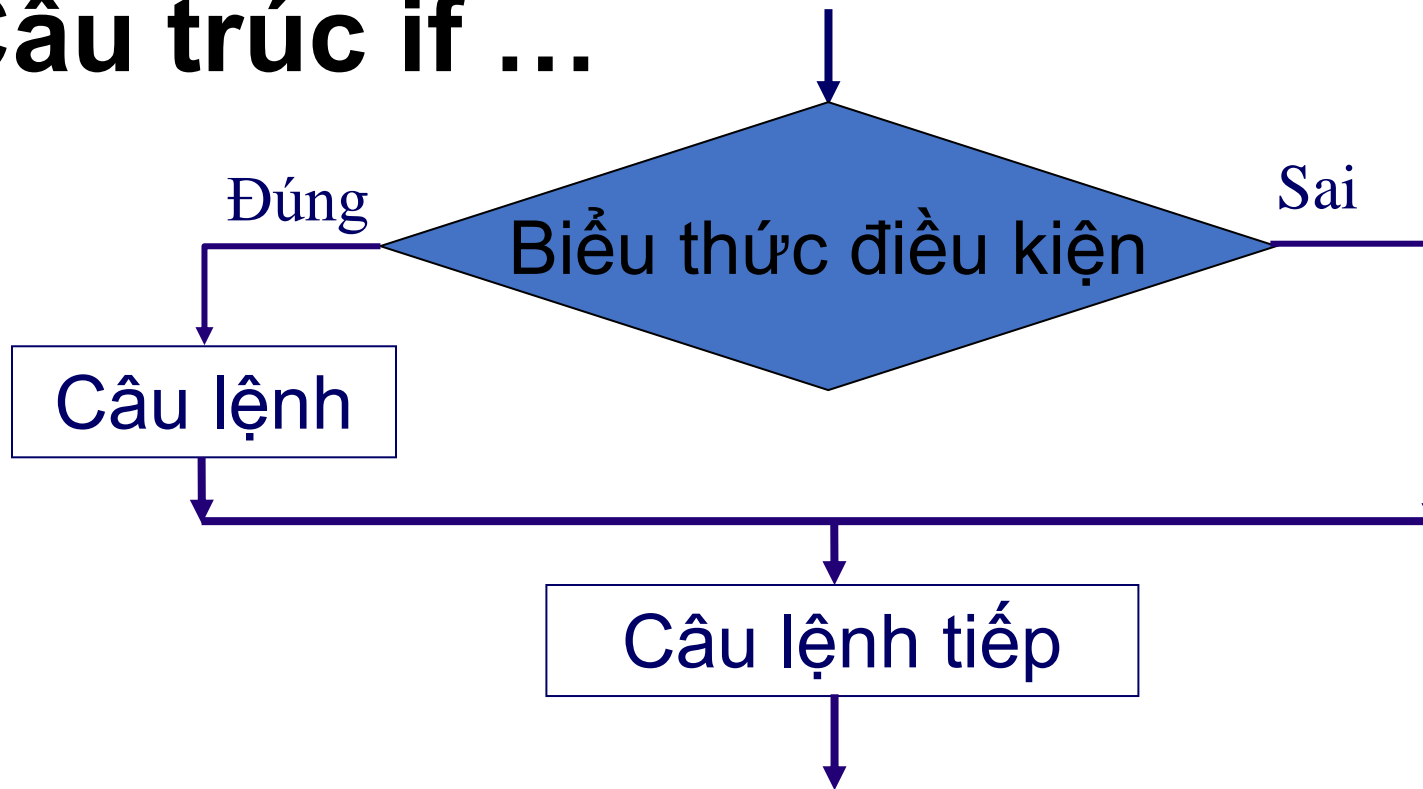
## 4.4. Các lệnh lặp

- Vòng lặp for
- Vòng lặp while và do while

## 4.5. Các lệnh nhảy

- Câu lệnh continue
- Câu lệnh break

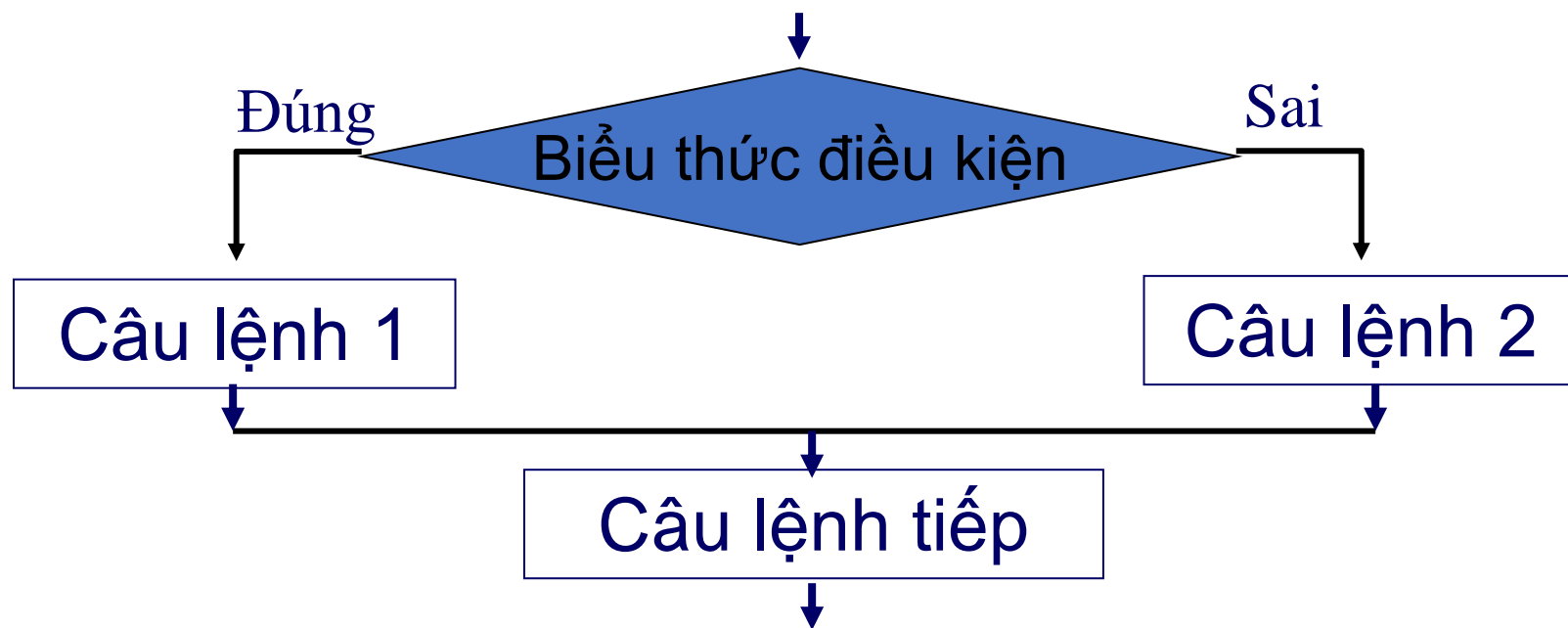
# Cấu trúc if ...



if (Biểu thức điều kiện)  
    Câu lệnh;  
    Câu lệnh kế tiếp;

if (n % 2 == 0)  
    printf("so chan");

# Cấu trúc if....else....



```
if (Điều kiện)
    Câu lệnh 1;
else
    Câu lệnh 2;
Câu lệnh kế tiếp;
```

```
if (x > y)
    z = x;
else
    z = y;
printf("max: %d", z);
```

# Lưu ý

## Biểu thức điều kiện:

- Là biểu thức trả về giá trị logic đúng/sai
  - Giá trị logic đúng/True : khác 0
  - Giá trị logic sai/False: bằng 0

## Ví dụ

`if (2+5) printf("Hello world!");` → Chấp nhận

Câu lệnh: có thể là một lệnh đơn hoặc lệnh khối (đặt trong cặp { } )

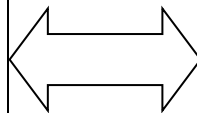
# Ví dụ: So sánh 2 số thực

```
#include <stdio.h>
int main()
{
    float a, b; float max; // khai bao bien
    printf("Nhap gia tri a va b: ");
    scanf("%f %f", &a, &b);
    if(a < b)
        max = b;
    else
        max = a;
    printf("So lon nhat trong 2 so %f va %f la %f ", a, b, max);
    return 0;
} //ket thuc ham main()
```

# Ví dụ: So sánh 2 số thực

```
C:\Users\lebavui\source\THDC\vd1.exe
Nhap gia tri a va b: 4 6
So lon nhat trong 2 so 4.000000 va 6.000000 la 6.000000
-----
Process exited after 3.221 seconds with return value 0
Press any key to continue . . .
```

```
if (a < b)
    max = b;
else
    max = a;
```



```
max = a < b ? b : a;
```

# Ví dụ: Giải phương trình $ax + b = 0$

```
#include <stdio.h>
int main()
{
    float a, b;
    printf("\nGiai phuong trinh bac nhat ax + b = 0");
    printf("\nCho biet he so a b : ");
    scanf("%f%f", &a, &b);
    if (a == 0)
        if (b != 0)
            printf("Phuong trinh vo nghiem");
        else
            printf("Phuong trinh vo so nghiem");
    else
        printf("Phuong trinh co nghiem x = %f", -b / a);
    return 0;
} //ket thuc ham main()
```

# Giải phương trình $ax + b = 0 \rightarrow$ Thực hiện

```
C:\Users\lebvui\source\THDC\vd1.exe

Giai phuong trinh bac nhat  $ax + b = 0$ 
Cho biet he so a b : 4 6
Phuong trinh co nghiem  $x = -1.500000$ 
-----
Process exited after 11.17 seconds with return value 0
Press any key to continue . . .
```

```
Select C:\Users\lebvui\source\THDC\vd1.exe

Giai phuong trinh bac nhat  $ax + b = 0$ 
Cho biet he so a b : 0 0
Phuong trinh vo so nghiem
-----
Process exited after 2.396 seconds with return value 0
Press any key to continue . . .
```



# Ví dụ: Nhập x và tính hàm

```
#include <stdio.h>
#include <math.h>
void main()
{
    float x, fx;
    printf("Nhap x: ");
    scanf("%f", &x);
    if (x < 3)
        fx = x * x + pow(sin(2 * M_PI * x), 4) + 1;
    else if (x == 3)
        fx = 5;
    else
        fx = sqrt(x - 3) + log10(x * x - 3);
    printf("\n Ket qua: %.4f", fx);
}
```

$$f(x) = \begin{cases} x^2 + \sin^4 2\pi x + 1 & \text{khi } x < 3 \\ 5 & \text{khi } x = 3 \\ \sqrt{x-3} + \log_{10}(x^2 - 3) & \text{khi } x > 3 \end{cases}$$

Nhap x: 1.0

Ket qua: 2.0000

Nhap x: 3

Ket qua: 5.0000

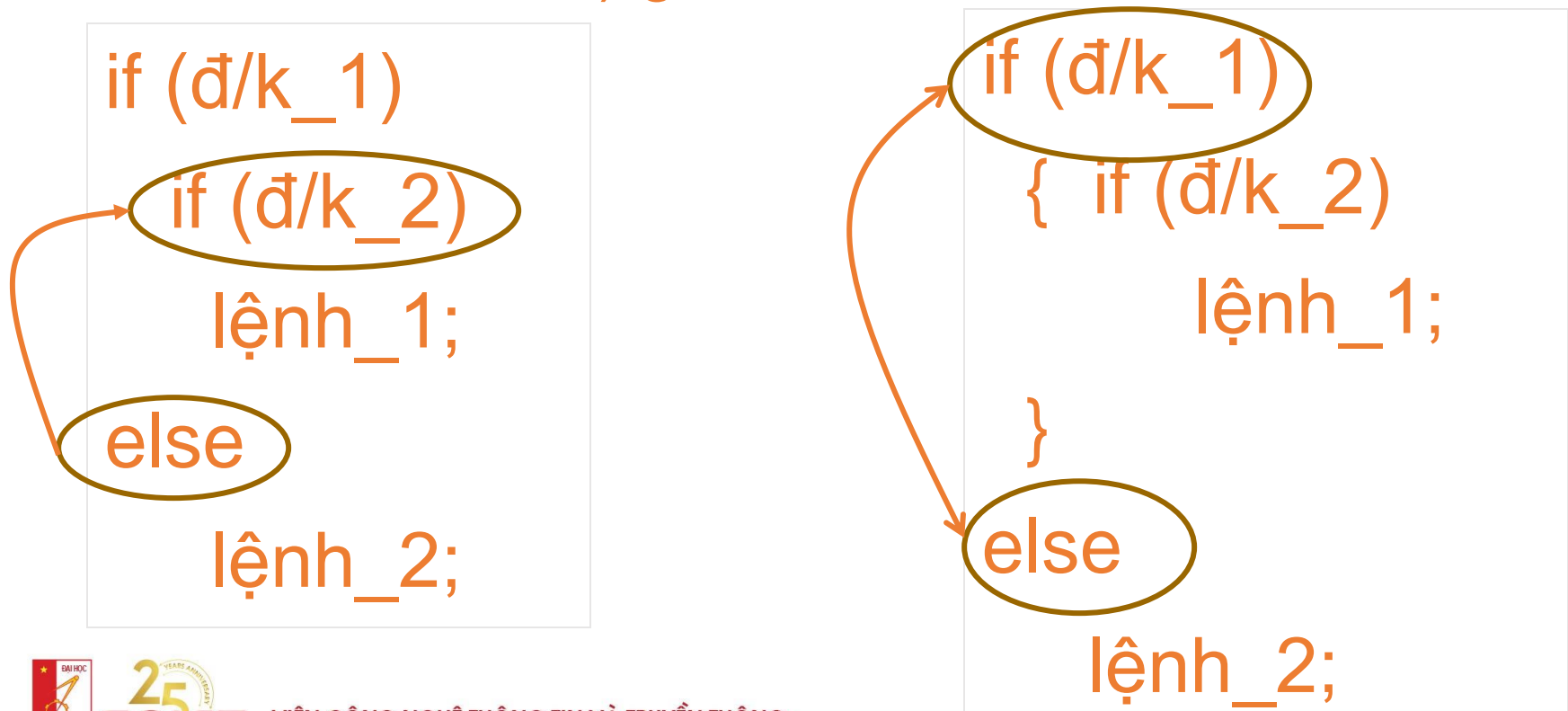
Nhap x: 5.0

Ket qua: 2.7566

# Cấu trúc if / if... else lồng nhau

Cấu trúc if.. và if ...else có thể lồng nhau

- Khi đó else sẽ tương ứng với if (*phía trên, chưa có else*) gần nhất



# Cấu trúc if / if... else lồng nhau → Ví dụ

```
int a, b, c = 10;
```

```
if (a==0)
    if (b==0)
        c = 20;
    else
        c = 30;
```

$a \neq 0 \rightarrow c = 10$

$a=0, b=0 \rightarrow c = 20$

$a=0, b \neq 0 \rightarrow c = 30$

```
if (a==0) {
    if (b==0)
        c = 20;
} else
    c = 30;
```

$a \neq 0 \rightarrow c = 30$

$a=0, b=0 \rightarrow c = 20$

$a=0, b \neq 0 \rightarrow c = 10$

# Chương 4: Cấu trúc điều khiển

## 4.1. Lệnh khối

## 4.2. Lệnh rẽ nhánh

- Cấu trúc if, if ... else

## 4.3. Lệnh lựa chọn

## 4.4. Các lệnh lặp

- Vòng lặp for
- Vòng lặp while và do while

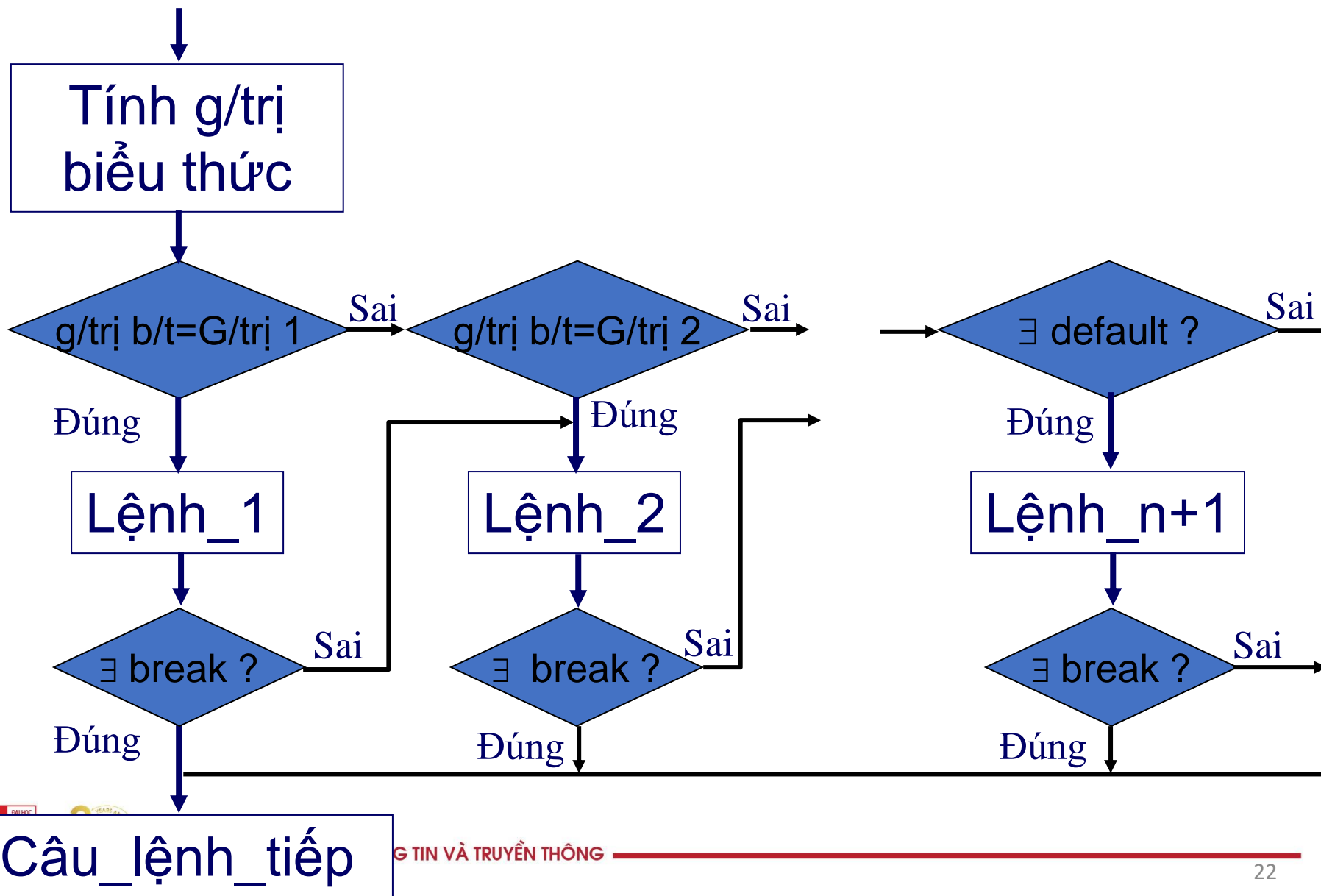
## 4.5. Các lệnh nhảy

- Câu lệnh continue
- Câu lệnh break

# Cấu trúc lựa chọn switch

```
switch (bieu_thuc)
{
    case gia_tri_1: lenh_1; [break];
    case gia_tri_2: lenh_2; [break];
    ...
    [default: lenh_n+1; [break];]
}
Câu_lệnh_tiếp
```

# Cấu trúc lựa chọn switch



# Cấu trúc lựa chọn switch

- Cơ chế hoạt động
  - Tính giá trị của biểu\_thức,
  - So sánh giá trị của biểu\_thức với các giá\_trị\_k (với  $k = 1, 2, \dots, n$ ) nằm sau các từ khóa case
- Xảy ra 2 khả năng:

# Cấu trúc lựa chọn switch → cơ chế hoạt động

1. Tồn tại **giá\_trị\_i** bằng giá trị biểu thức.

- Thực hiện **lệnh\_i**
  - Nếu tồn tại lệnh **break**,
    - Nhảy tới tiếp tục thực hiện **Câu\_lệnh\_tiếp** nằm sau cấu trúc **switch**
  - Nếu không tồn tại lệnh **break**
    - Thực hiện các lệnh sau **lệnh\_i** cho tới khi gặp **break** hoặc tới khi thoát khỏi cấu trúc **switch**
    - Thực hiện **Câu\_lệnh\_tiếp**



# Cấu trúc lựa chọn switch → cơ chế hoạt động

2. Không tồn tại **giá\_trị\_i** ( $i = 1, 2, \dots, n$ ) nào bằng giá trị biểu thức

- Nếu có nhãn **default**:
  - Chương trình sẽ thực hiện **lệnh\_n+1**
  - Thực hiện **Câu\_lệnh\_tiếp** nằm ngay sau cấu trúc **switch**.
- Nếu không có nhãn **default**:
  - Chương trình chuyển sang thực hiện lệnh tiếp theo nằm ngay sau cấu trúc **switch**: **Câu\_Lệnh\_tiếp**

# Cấu trúc lựa chọn switch→Ví dụ 1

Lập trình đọc vào từ bàn phím một số nguyên  $1 \leq N \leq 10$  và đưa ra từ tiếng Anh tương ứng.

# Cấu trúc lựa chọn switch→Ví dụ 1

```
#include <stdio.h>
int main()
{
    int N;
    printf("\nNhập một giá trị số nguyên không âm: ");
    scanf("%d", &N);
    switch (N)
    {
        case 1: printf("%d -> One", N); break;
        case 2: printf("%d -> Two", N); break;
        case 3: printf("%d -> Three", N); break;
        case 4: printf("%d -> Four", N); break;
        case 5: printf("%d -> Five", N); break;
        case 6: printf("%d -> Six", N); break;
        case 7: printf("%d -> Seven", N); break;
        case 8: printf("%d -> Eight", N); break;
        case 9: printf("%d -> Nine", N); break;
        case 10: printf("%d -> Ten", N); break;
        default: printf("Không thỏa mãn điều kiện [1..10]");
    }
    return 0;
}
```

# Cấu trúc lựa chọn switch→ Thực hiện

Nhap mot gia tri so nguyen khong am: 7  
7 -> Seven

Nhap mot gia tri so nguyen khong am: 3  
3 -> Three

Nhap mot gia tri so nguyen khong am: -6  
Khong thoa man dieu kien [1..10]

# Cấu trúc lựa chọn switch→Ví dụ 2

Nhập vào số nguyên không âm, đưa ra ngày trong tuần tương ứng (*theo số dư khi chia cho 7*).

# Cấu trúc lựa chọn switch→Ví dụ 2

```
#include <stdio.h>
int main()
{
    int a;
    printf("Nhap mot gia tri so nguyen khong am: ");
    scanf("%d", &a);
    switch(a % 7) {
        case 0: printf("Chu nhat"); break;
        case 1: printf("Thu Hai"); break;
        case 2: printf("Thu Ba"); break;
        case 3: printf("Thu Tu"); break;
        case 4: printf("Thu Nam"); break;
        case 5: printf("Thu Sau"); break;
        case 6: printf("Thu Bay"); break;
    }
    return 0;
}
```

# Cấu trúc lựa chọn switch → Thực hiện

```
Nhap mot gia tri so nguyen khong am: 123  
Thu Nam
```

```
Nhap mot gia tri so nguyen khong am: 5  
Thu Sau
```

```
Nhap mot gia tri so nguyen khong am: 88  
Thu Nam
```

```
Nhap mot gia tri so nguyen khong am: 52  
Thu Tu
```

# Cấu trúc lựa chọn switch

*Có thể sử dụng đặc điểm: Không có lệnh break chương trình sẽ tự động chuyển xuống thực hiện các câu lệnh tiếp sau để viết chung mã lệnh cho các trường hợp khác nhau nhưng được xử lý như nhau*

**Ví dụ:** Trong một năm các tháng có 30 ngày là 4, 6, 9, 11 còn các tháng có 31 ngày là 1, 3, 5, 7, 8, 10, 12. Riêng tháng hai có thể có 28 hoặc 29 ngày. Hãy viết chương trình nhập vào 1 tháng, sau đó đưa ra kết luận tháng đó có bao nhiêu ngày



# Cấu trúc lựa chọn switch→ Ví dụ

```
#include <stdio.h>
int main()
{
    int thang;
    printf("\nNhập vào tháng trong năm ");
    scanf("%d", &thang);
    switch (thang)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            printf("\n Tháng %d có 31 ngày ", thang);
            break;
```

# Cấu trúc lựa chọn switch

```
case 4:
case 6:
case 9:
case 11:
    printf("\n Tháng %d có 30 ngày ", thang);
    break;
case 2:
    printf("\ Tháng 2 có 28 hoặc 29 ngày");
    break;
default:
    printf("\n Không có tháng %d", thang);
    break;
}
return 0;
}
```

# Cấu trúc lựa chọn switch→Lưu ý

- Giá trị của biểu thức trong cấu trúc switch phải là số nguyên (*kiểu đếm được*)
  - Phải có kiểu dữ liệu là ***char, int, long***
- Các giá trị sau từ khóa case (*gia\_tri\_1, gia\_tri\_2,..* ) cũng phải là số nguyên

Điều kiện trong cấu trúc **if / if..else** cho phép làm việc với các kiểu dữ liệu khác số nguyên

# Các ví dụ

1. Viết chương trình tính cước Taxi theo công thức:

- 1 km đầu tiên có cước là 10000đ,
- 30 km tiếp theo có giá là 8000đ/1km
- Các km sau đó có giá là 6000đ/1km.

2. Viết chương trình giải phương trình bậc hai  
 $ax^2 + bx + c = 0$

3. Viết chương trình giải hệ phương trình bậc nhất

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$$

# Ví dụ 1: Tính cước taxi

```
#include <stdio.h>
#include <math.h>
int main()
{
    unsigned long sotien;
    float sokm;
    printf("\nBan hay cho biet so km da di duoc : ");
    scanf("%f", &sokm);
    if (sokm <= 1.0)
        sotien = 10000;
    else if (sokm <= 31.0)
        sotien = 10000 + (ceil(sokm) - 1.0) * 8000;
    else
        sotien = 250000 + (ceil(sokm) - 31) * 6000;
    printf("\nSo tien can tra = %lud", sotien);
    return 0;
}
```

# Ví dụ 1 → Thực hiện chương trình

Ban hay cho biet so km da di duoc : 0.68

So tien can tra = 10000d

Ban hay cho biet so km da di duoc : 12.45

So tien can tra = 106000d

Ban hay cho biet so km da di duoc : 35.67

So tien can tra = 280000d

Ban hay cho biet so km da di duoc : 24.33

So tien can tra = 202000d

$$\text{sotien} = \text{sokm} \leq 1.0 ? 10000 : \text{sokm} \leq 31 ? 10000 + (\text{ceil}(\text{sokm}) - 1.0) * 8000 : 250000 + (\text{ceil}(\text{sokm}) - 31) * 6000;$$

# Ví dụ 2: Giải phương trình bậc 2

```
#include <stdio.h>
#include <math.h> //Để sử dụng hàm toán học sqrt
int main()
{
    float a, b, c, delta;
    printf("\n\nNhap he so a b c : ");
    scanf("%f%f%f", &a, &b, &c);
    delta = b * b - 4 * a * c;
    if (a == 0)
        printf("P/trinh suy bien thanh p/trinh bac 1 %fx+%f=0", b, c);
    else if (delta < 0)
        printf("Phuong trinh vo nghiem");
    else if (delta == 0)
        printf("Phuong trinh co nghiem kep x1 = x2 = %f", -b / (2 * a));
    else
        printf("Phuong trinh co hai nghiem phan biet\n  x1=%f \n  x2=%f",
            (-b + sqrt(delta)) / (2 * a), (-b - sqrt(delta)) / (2 * a));
    return 0;
}
```

# Ví dụ 2→ Thực hiện chương trình

Nhap he so a b c : 0 3 2

P/trình suy bien thanh p/trình\_bac 1  $3.000000x+2.000000=0$

Nhap he so a b c : 1 2 3

Phuong trinh vo nghiem

Nhap he so a b c : 1 4 4

Phuong trinh co nghiem kep  $x1 = x2 = -2.000000$

Nhap he so a b c : 1 -3 2

Phuong trinh co hai nghiem phan biet

$x1=2.000000$

$x2=1.000000$



# Ví dụ 3: Giải hệ phương trình

```
#include <stdio.h>
int main()
{
    float a1, b1, c1, a2, b2, c2, x, y, dx, dy, d;
    printf("\n\nNhap cac so:\n");
    printf("a1,b1,c1="); scanf("%f%f%f", &a1, &b1, &c1);
    printf("a2,b2,c2="); scanf("%f%f%f", &a2, &b2, &c2);
    d = a1 * b2 - a2 * b1;
    dx = c1 * b2 - c2 * b1;
    dy = a1 * c2 - a2 * c1;
    if (d != 0)
    {
        x = dx / d;
        y = dy / d;
        printf("He PT co nghiem x=%f, y=%f\n", x, y);
    }
    else if (dx == 0)
        printf("He PT co vo so nghiem!\n");
    else
        printf("He phuong trinh vo nghiem!");
    return 0;
}
```

# Ví dụ 3 → Thực hiện chương trình

Nhap cac so:

a1, b1, c1 = 3 5 8

a2, b2, c2 = 2 1 9

He PT co nghiem  $x = 5.285714$ ,  $y = -1.571429$

Nhap cac so:

a1, b1, c1 = 1 2 3

a2, b2, c2 = 1 2 4

He phuong trinh vo nghiem!

Nhap cac so:

a1, b1, c1 = 1 2 3

a2, b2, c2 = 2 4 6

He PT co vo so nghiem!

# Bài tập

1. Lập trình nhập vào một ký tự hệ hexa và đưa ra giá trị hệ 10 tương ứng.
2. Lập trình đọc tọa độ 4 điểm A,B,C,M rồi kiểm tra xem điểm M nằm trong, nằm trên cạnh hay nằm ngoài tam giác ABC.
3. Lập trình nhập 3 số a, b, c. Kiểm tra xem a, b, c có phải là 3 cạnh của tam giác không? Nếu đúng thì là tam giác gì?
4. Lập trình nhập vào một năm bất kỳ, kiểm tra năm đó có phải là năm nhuận không?
5. Lập trình đọc vào từ bàn phím 2 giá trị a, b rồi tính  $y = 15x^2 + x + 7.2$  trong đó

$$x = \begin{cases} \frac{a+b}{3} & \text{if } a < b \\ 1.5172 & \text{if } a = b \\ \frac{a-b}{a^2 + b^2} & \text{if } a > b \end{cases}$$

# Chương 4: Cấu trúc điều khiển

## 4.1. Lệnh khối

## 4.2. Lệnh rẽ nhánh

- Cấu trúc if, if ... else

## 4.3. Lệnh lựa chọn

## 4.4. Các lệnh lặp

- Vòng lặp for
- Vòng lặp while và do while

## 4.5. Các lệnh nhảy

- Câu lệnh continue
- Câu lệnh break

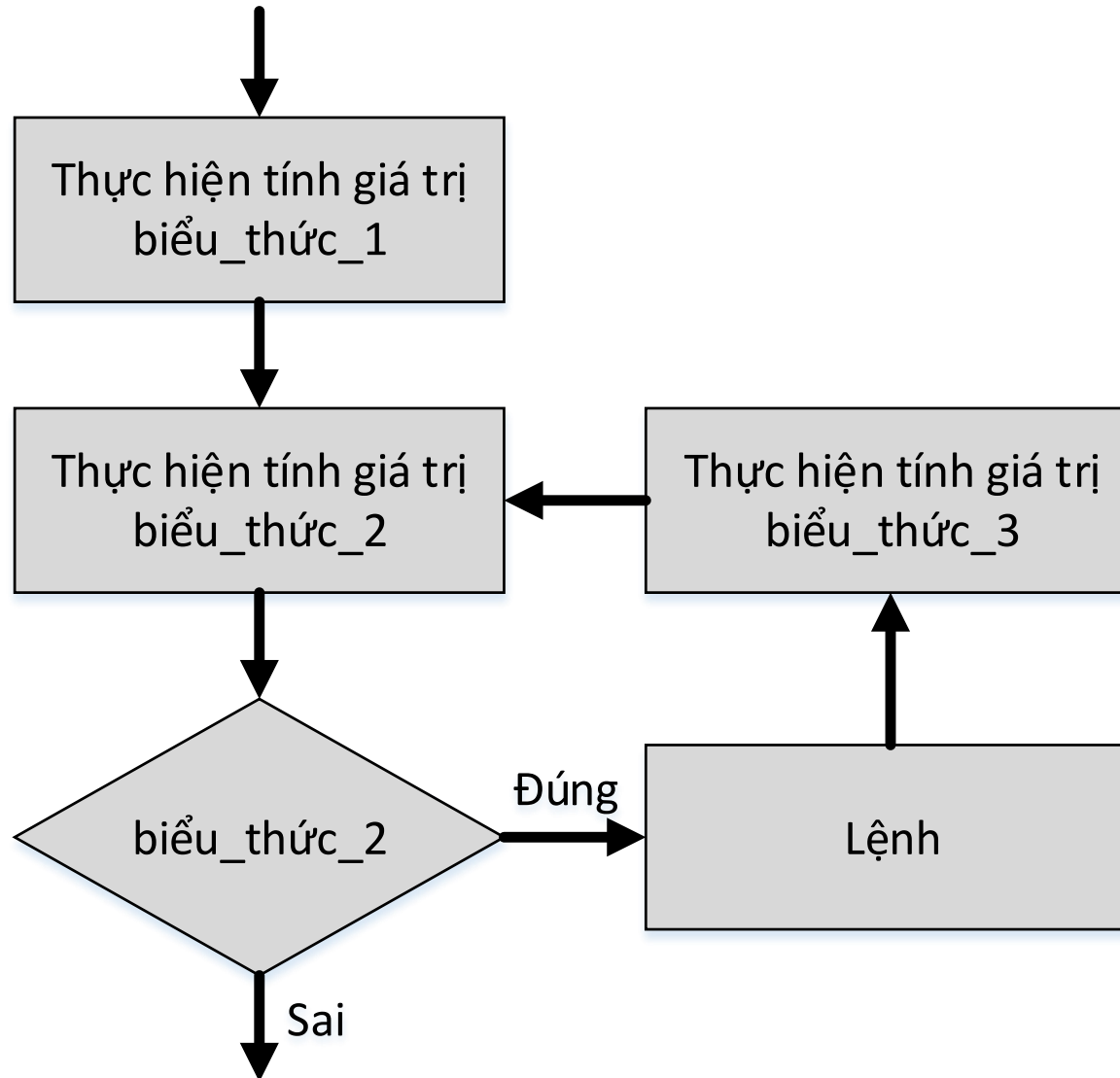
# Cấu trúc lặp *for*

Dùng để thực hiện nhiều lần một công việc

```
for([b.thuc_1];[b.thuc_2];[b.thuc_3]) Lệnh;
```

- b.thuc\_1: Khởi tạo giá trị ban đầu cho vòng lặp
- b.thuc\_2: Điều kiện tiếp tục vòng lặp
- b.thuc\_3: Thay đổi biến điều khiển của vòng lặp
- Lệnh: Có thể là lệnh đơn lệnh kép hoặc lệnh rỗng

# Lưu đồ



# Sử dụng

```
int i;  
for(i = 0; i < 100; i++) Câu_lệnh;
```

```
int i;  
for(i = 0; i < 100; i+=2) Câu_lệnh;
```

```
int i;  
for(i = 100; i > 0; i--) Câu_lệnh;
```

# Tìm các số nguyên lẻ nhỏ hơn 100

```
#include <stdio.h>
int main()
{
    int i;
    for (i = 1; i < 100; i++)
    {
        if (i % 2 == 1) printf("%5d", i);
        if ((i + 1) % 20 == 0) printf("\n");
    }
    return 0;
}
```

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 1  | 3  | 5  | 7  | 9  | 11 | 13 | 15 | 17 | 19 |
| 21 | 23 | 25 | 27 | 29 | 31 | 33 | 35 | 37 | 39 |
| 41 | 43 | 45 | 47 | 49 | 51 | 53 | 55 | 57 | 59 |
| 61 | 63 | 65 | 67 | 69 | 71 | 73 | 75 | 77 | 79 |
| 81 | 83 | 85 | 87 | 89 | 91 | 93 | 95 | 97 | 99 |



# Tìm các số nguyên lẻ nhỏ hơn 100

```
#include <stdio.h>
int main()
{
    int i;
    for (i = 1; i < 100; i += 2)
    {
        printf("%5d", i);
        if ((i + 1) % 20 == 0) printf("\n");
    }
    return 0;
}
```

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 1  | 3  | 5  | 7  | 9  | 11 | 13 | 15 | 17 | 19 |
| 21 | 23 | 25 | 27 | 29 | 31 | 33 | 35 | 37 | 39 |
| 41 | 43 | 45 | 47 | 49 | 51 | 53 | 55 | 57 | 59 |
| 61 | 63 | 65 | 67 | 69 | 71 | 73 | 75 | 77 | 79 |
| 81 | 83 | 85 | 87 | 89 | 91 | 93 | 95 | 97 | 99 |

# Tìm các số nguyên lẻ nhỏ hơn 100

```
#include <stdio.h>
int main()
{
    int i;
    for (i = 99; i > 0; i -= 2)
    {
        printf("%5d", i);
        if ((i - 1) % 20 == 0) printf("\n");
    }
    return 0;
}
```

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 99 | 97 | 95 | 93 | 91 | 89 | 87 | 85 | 83 | 81 |
| 79 | 77 | 75 | 73 | 71 | 69 | 67 | 65 | 63 | 61 |
| 59 | 57 | 55 | 53 | 51 | 49 | 47 | 45 | 43 | 41 |
| 39 | 37 | 35 | 33 | 31 | 29 | 27 | 25 | 23 | 21 |
| 19 | 17 | 15 | 13 | 11 | 9  | 7  | 5  | 3  | 1  |

# Nhập n và tính n!

```
#include <stdio.h>
int main()
{
    int P = 1;
    int i, n;
    printf("Nhap n: "); scanf("%d", &n);
    for (i = 1; i <= n; i++)
        P *= i;
    printf("Ket qua la: %d\n", P);
    return 0;
}
```

Nhap n: 6  
Ket qua la: 720

Nhap n: 9  
Ket qua la: 362880

# Nhập n và tính tổng $S = 1 + \frac{1}{2} + \dots + \frac{1}{n}$

```
#include <stdio.h>
int main()
{
    float S = 0;
    int i, n;
    printf("Nhap n: "); scanf("%d", &n);
    for (i = 1; i <= n; i++)
        S += 1.0/i;
    printf("Ket qua la: %f\n", S);
    return 0;
}
```

Nhap n: 10  
Ket qua la: 2.928968

Nhap n: 5  
Ket qua la: 2.283334

# Tìm số 3 chữ số thỏa mãn $abc = a^3 + b^3 + c^3$

```
#include <stdio.h>
int main()
{
    int i, a, b, c;
    for (i = 100; i < 1000; i++)
    {
        a = i / 100;
        b = i % 100 / 10;
        c = i % 10;
        if (a * a * a + b * b * b + c * c * c == i)
            printf("%d\n", i);
    }
    return 0;
}
```

153

370

371

407

# Tìm số 3 chữ số thỏa mãn $abc = a^3 + b^3 + c^3$

```
#include <stdio.h>
int main()
{
    int a, b, c;
    for (a = 1; a <= 9; a++)
        for (b = 0; b <= 9; b++)
            for (c = 0; c <= 9; c++)
                if (a*a*a+b*b*b+c*c*c == a*100+b*10+c)
                    printf("%d\n", a*100+b*10+c);

    return 0;
}
```

153

370

371

407

# Chú ý

Không nhất thiết phải có đầy đủ các thành phần trong vòng lặp **for**

***int getchar()***: đọc ký tự từ vùng đệm bàn phím. Nếu vùng đệm rỗng, đợi người dùng gõ dãy ký tự (*cho tới khi ấn phím Enter*), sẽ trả về ký tự đầu.

***putchar(int c)***: đưa ký tự ra màn hình.

# Chú ý

## 1. Biểu thức khởi tạo

```
char c; int i=0;
for(; (c=getchar()) != '\n' ; i++)
    putchar(c);
printf("\nSo ky tu: %d", i);
```

Hello world  
Hello world  
So ky tu: 11

## 2. Biểu thức điều khiển

```
char c; int i=0;
for(i = 0 ; ; c = getchar(), i++)
    if(c == '\n') break;
printf("\nSo ky tu: %d", i);
```

Hello world  
  
So ky tu: 12

## 3. Thân vòng lặp

```
char c; int i=0;
for(i = 0; getchar() != '\n', i++; );
printf("\nSo ky tu: %d", i);
```

Hello world  
  
So ky tu: 1



# Bài tập

Lập trình in ra màn hình kết quả như sau  
(chiều cao n được nhập từ bàn phím)

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
*****
```

Hình 1

```
      *
     **
    ***
   ****
  *****
 *****
*****
*****
*****
*****
*****
```

Hình 2

```
          *
         ***
        *****
       ********
      **********
     **********
    **********
   **********
  **********
 **********
 **********
 **********
```

Hình 3

# Cấu trúc lặp while

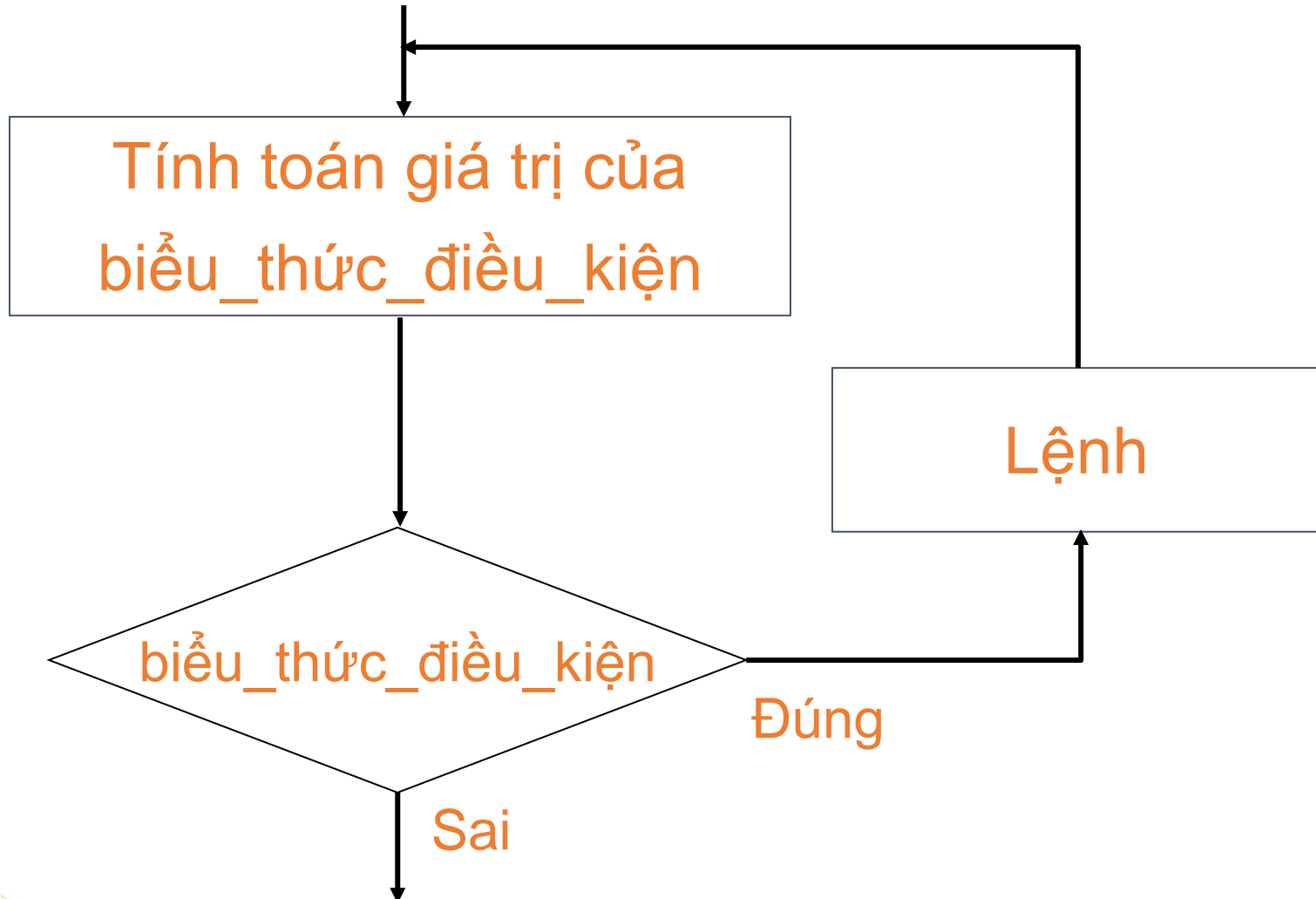
Dùng để thực hiện lặp đi lặp lại một công việc nào đó tới khi điều kiện không thỏa mãn.

Cú pháp:

```
while(bieu_thuc_dieu_kien)  
    lenh;
```

- Chương trình kiểm tra điều kiện **trước** khi lặp
  - Giá trị của biểu thức điều kiện là đúng  $\Rightarrow$  thực hiện lệnh
- Các lệnh của vòng lặp có thể không được thực hiện lần nào nếu **biểu\_thức\_điều\_kiện** sai ngay từ đầu
- Nếu **biểu\_thức\_điều\_kiện** luôn đúng có thể gây lặp vô hạn

# Lưu đồ



# Nhập n và in ra tổng của n số nguyên đầu tiên

```
#include <stdio.h>
int main()
{
    long S = 0;
    int n;
    printf("Nhap n: "); scanf("%d", &n);
    while (n > 0)
    {
        S += n;
        n--;
    }
    printf("Ket qua la: %ld\n", S);
    return 0;
}
```

Nhap n: 10

Ket qua la: 55

Nhap n: 96

Ket qua la: 4656

# Tìm số nguyên lớn nhất thỏa mãn

$$3n^5 - 317n < 5$$

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

int main()
{
    int n;
    while (3*pow(n,5)-317*n<5) n++;
    printf("Ket qua la: %d\n", n-1);
    return 0;
}
```

Ket qua la: 3

# Cho biết kết quả thực hiện chương trình

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

```
int main()
{
    int i = 3;
    while (i > 1)
    {
        if (i % 2 == 0) i = i / 2;
        else i = i * 3 + 1;
        printf("%5d\n", i);
    }
    return 0;
}
```

10

5

16

8

4

2

1

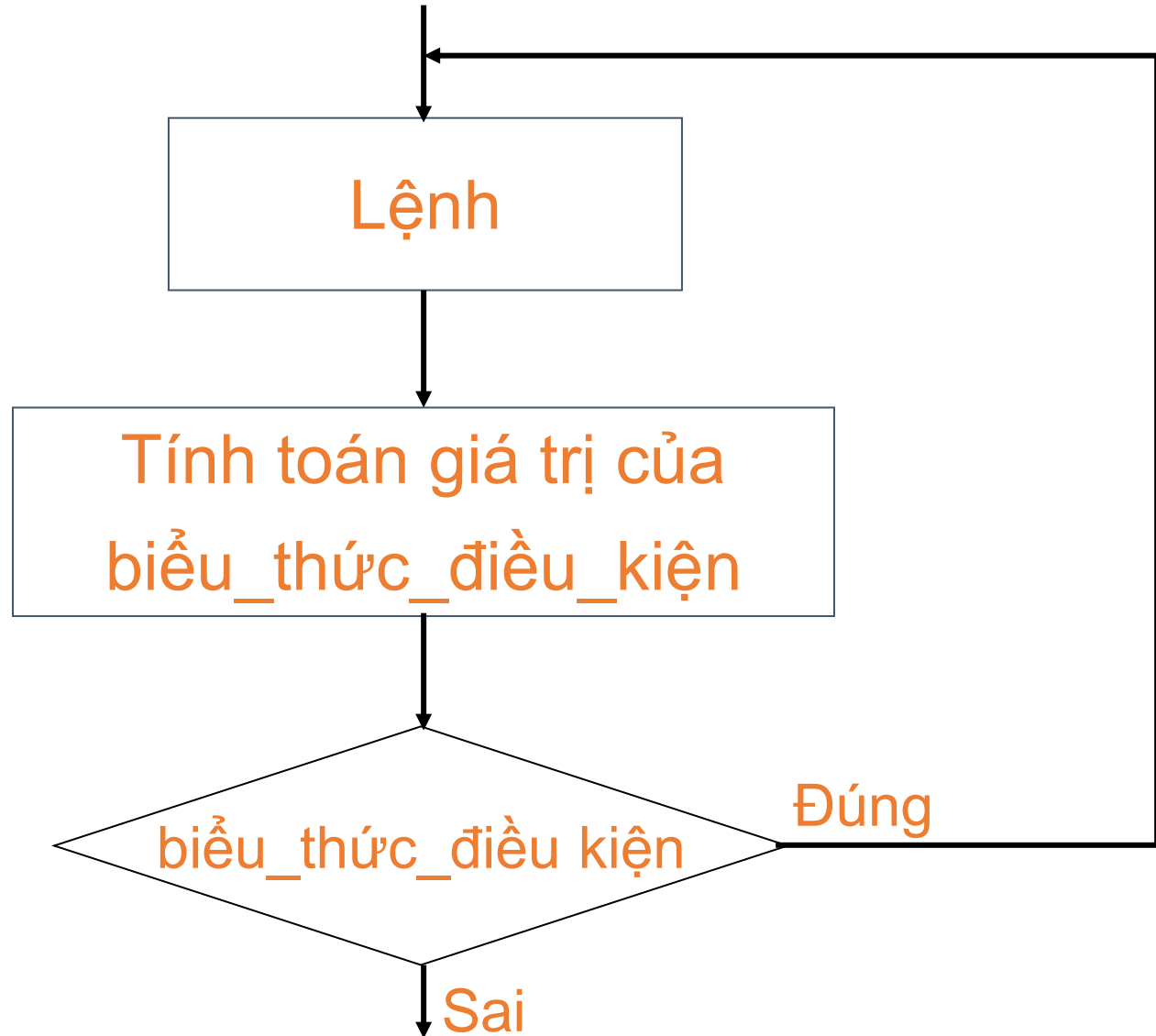
# Cấu trúc lặp do ... while

- Dùng để thực hiện lặp đi lặp lại một công việc nào đó.
- Cú pháp:

```
do {  
    lenh;  
} while (bieu_thuc_dieu_kien) ;
```

- Chương trình kiểm tra điều kiện **sau** khi lặp
- Các `lenh` được thực hiện ít nhất một lần
- Nếu biểu thức luôn đúng, lặp vô hạn

# Lưu đồ





# Nhập n và đưa tổng của n số nguyên đầu tiên

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

```
int main()
```

```
{
```

```
    long S = 0;
```

```
    int n;
```

```
    printf("Nhap n : "); scanf("%d", &n);
```

```
    do {
```

```
        S = S + n;
```

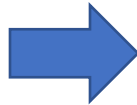
```
        n = n - 1;
```

```
    } while (n > 0);
```

```
    printf("Ket qua la %ld ", S);
```

```
    return 0;
```

```
}
```



```
do
```

```
    S += n--;
```

```
while (n > 0);
```

Nhap n : 96

Ket qua la 4656

# Ví dụ

- Nhập vào điểm của một sinh viên, nếu điểm đó không  $\in [0, 10]$  thì thông báo cho người dùng nhập lại.
- Thực hiện:
  - Nếu dùng lệnh **if**
    - Chỉ kiểm tra được 1 lần
  - Sử dụng **for**
    - Chưa biết trước số lần lặp.
    - Sử dụng vòng lặp không cần xác định trước số lần lặp: **while** / **do while**

# Dòng vòng lặp while

```
#include <stdio.h>
int main()
{
    float diem;
    printf("Chương trình nhập điểm sinh viên\n");
    printf("Nhập điểm (0<=diem<=10):"); scanf("%f", &diem);
    while (diem < 0 || diem > 10)
    {
        printf("\nBạn nhập không đúng!\n");
        printf("Bạn hãy nhập lại (0<=diem<=10):");
        scanf("%f", &diem);
    }
    printf("\nĐiểm bạn vừa nhập là: %.2f", diem);
    return 0;
}
```

# Dùng vòng lặp while → Kết quả

Chương trình nhập điểm sinh viên

Nhập điểm ( $0 \leq \text{diem} \leq 10$ ): 11

Bạn nhập không đúng!

Bạn hãy nhập lại ( $0 \leq \text{diem} \leq 10$ ): -2

Bạn nhập không đúng!

Bạn hãy nhập lại ( $0 \leq \text{diem} \leq 10$ ): 9

Điểm bạn vừa nhập là: 9.00

# Dùng vòng lặp do...while

```
#include <stdio.h>
int main()
{
    float diem;
    printf("Chương trình nhập điểm sinh viên\n");
    do
    {
        printf("Nhập điểm (0<=diem<=10):");
        scanf("%f", &diem);
        if (diem < 0 || diem > 10)
            printf("\nBạn nhập không đúng!\n");
    } while (diem < 0 || diem > 10);
    printf("\nĐiểm bạn vừa nhập là: %.2f", diem);
    return 0;
}
```

# Nhập số và phân tích số nguyên ra thừa số nguyên tố

```
#include <stdio.h>
int main() {
    int N, i;
    do {
        printf("Nhap vao so nguyen duong (0 de ket thuc): "); scanf("%d", &N);
        if (N == 0) break;

        printf("%d = ", N);
        i = 2;
        while (i < N) {
            if (N % i == 0) {
                printf("%d x ", i);
                N = N / i;
            } else i++;
        }
        printf("%d \n", N);
    } while (1);
    return 0;
}
```

# Kết quả

Nhap vao so nguyen duong (0 de ket thuc): 24

$$24 = 2 \times 2 \times 2 \times 3$$

Nhap vao so nguyen duong (0 de ket thuc): 1001

$$1001 = 7 \times 11 \times 13$$

Nhap vao so nguyen duong (0 de ket thuc): 42

$$42 = 2 \times 3 \times 7$$

Nhap vao so nguyen duong (0 de ket thuc): 73

$$73 = 73$$

Nhap vao so nguyen duong (0 de ket thuc): 0

# Ví dụ

Viết chương trình thực hiện công việc

- Nhập vào từ bàn phím 2 số nguyên
- Nhập vào từ bàn phím một ký tự bất kỳ:
  - Nếu đây là một toán tử số học thì đưa ra giá trị tương ứng với toán tử.
  - Nếu không phải thì đưa ra thông báo sai
- Chương trình thực hiện cho tới khi ký tự nhập vào là 'q' hoặc 'Q'



# Mã nguồn (1)

```
#include <stdio.h>
int main()
{
    int a, b;
    char ch;
    int exit = 0;
    printf("Nhap 2 so nguyen a va b: ");
    scanf("%d%d", &a, &b);
    do
    {
        printf("Chon toan tu (+, -, *, /, %% , q de ket thuc): ");
        while (getchar() != '\n'); // Xoa bo dem ban phim
        ch = getchar();
        switch(ch)
        {
            case '+': printf("Ket qua la: %d\n", a + b); break;
            case '-': printf("Ket qua la: %d\n", a - b); break;
            case '*': printf("Ket qua la: %d\n", a * b); break;
```

# Mã nguồn (2)

```
        case '/':
            if (b == 0)
                printf("Khong thuc hien duoc phep chia\n");
            else
                printf("Ket qua la: %d\n", a / b); break;
        case '%':
            if (b == 0)
                printf("Khong thuc hien duoc phep chia\n");
            else
                printf("Ket qua la: %d\n", a % b); break;
        case 'q':
        case 'Q': exit = 1; break;
        default: printf("Khong thuc hien toan tu nay!\n");
    }
} while (exit == 0);
return 0;
}
```

# Kết quả

Nhap 2 so nguyen a va b: 6

4

Chon toan tu (+, -, \*, /, %, q de ket thuc): +

Ket qua la: 10

Chon toan tu (+, -, \*, /, %, q de ket thuc): -

Ket qua la: 2

Chon toan tu (+, -, \*, /, %, q de ket thuc): \*

Ket qua la: 24

Chon toan tu (+, -, \*, /, %, q de ket thuc): /

Ket qua la: 1

Chon toan tu (+, -, \*, /, %, q de ket thuc): %

Ket qua la: 2

Chon toan tu (+, -, \*, /, %, q de ket thuc): ?

Khong thuc hien toan tu nay!

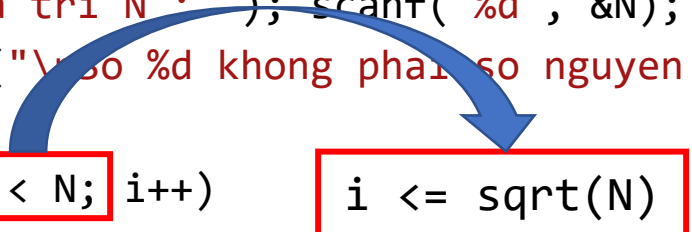
Chon toan tu (+, -, \*, /, %, q de ket thuc): q

# Ví dụ: Nhập một số nguyên, kiểm tra là số nguyên tố không?

```
#include <stdio.h>
#include <math.h>
int main() {
    int N, i, OK = 1;
    printf("\nNhap gia tri N : "); scanf("%d", &N);
    if (N < 2) printf("\nSo %d khong phai so nguyen to", N);
    else {
        i = 2;
        while (N % i != 0) i++;
        if (i == N) printf("\nSo %d la so nguyen to.", N);
        else printf("\nSo %d la hop so.", N);
    }
    return 0;
}
```

# Ví dụ: Nhập một số nguyên, kiểm tra là số nguyên tố không?

```
#include <stdio.h>
#include <math.h>
int main() {
    int N, i, OK = 1;
    printf("\nNhap gia tri N : "); scanf("%d", &N);
    if (N < 2) printf("\nSo %d khong phai so nguyen to", N);
    else {
        for (i = 2; i < N; i++)
            if (N % i == 0) {
                OK = 0;
                break;
            }
        if (OK) printf("\nSo %d la so nguyen to.", N);
        else printf("\nSo %d la hop so.", N);
    }
    return 0;
}
```



# Chương 4: Cấu trúc điều khiển

## 4.1. Lệnh khối

## 4.2. Lệnh rẽ nhánh

- Cấu trúc if, if ... else

## 4.3. Lệnh lựa chọn

## 4.4. Các lệnh lặp

- Vòng lặp for
- Vòng lặp while và do while

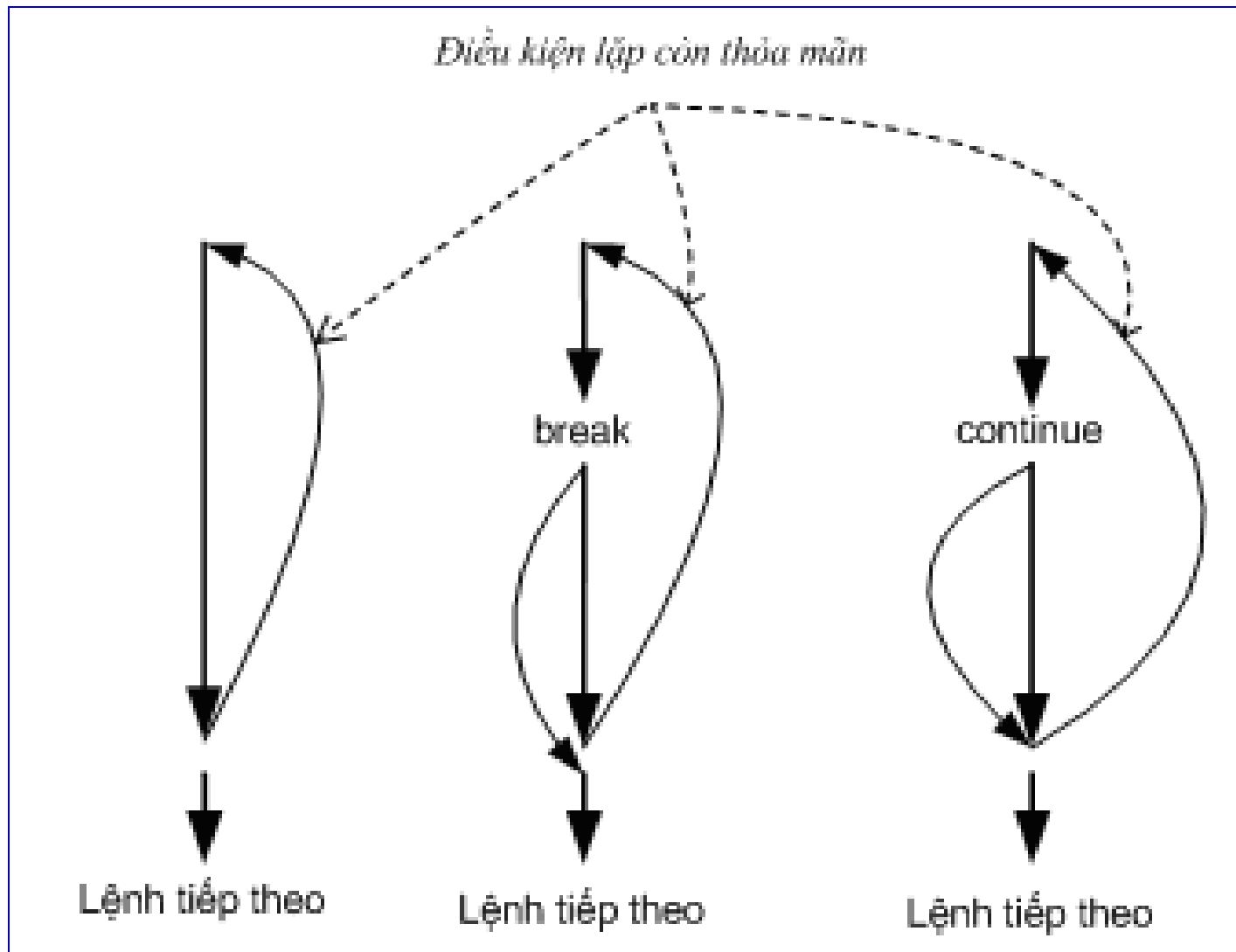
## 4.5. Các lệnh nhảy

- Câu lệnh continue
- Câu lệnh break

# Mục đích

- Các vòng lặp **while/ do ... while/ for** sẽ kết thúc quá trình lặp khi biểu thức điều kiện của vòng lặp không còn được thỏa mãn.
- Tuy nhiên trong lập trình đôi khi ta cũng cần thoát khỏi vòng lặp ngay cả khi biểu thức điều kiện của vòng lặp vẫn còn được thỏa mãn.
- Để hỗ trợ người lập trình làm việc đó, ngôn ngữ C cung cấp 2 câu lệnh là **continue** và **break**

# continue >< break





# continue

- Bỏ qua việc thực hiện các câu lệnh nằm sau lệnh continue trong thân vòng lặp.
- Chuyển sang thực hiện một vòng lặp mới

# Ví dụ 1: In ra 100 số nguyên đầu tiên ngoại trừ các số chia hết cho 5.

```
#include <stdio.h>
int main()
{
    int i;
    for (i = 1; i <= 100; i++)
    {
        if (i % 5 == 0) continue;
        printf("%5d", i);
        if ((i + 1) % 20 == 0) printf("\n");
    }
    return 0;
}
```

# Kết quả

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1  | 2  | 3  | 4  | 6  | 7  | 8  | 9  | 11 | 12 | 13 | 14 | 16 | 17 | 18 | 19 |
| 21 | 22 | 23 | 24 | 26 | 27 | 28 | 29 | 31 | 32 | 33 | 34 | 36 | 37 | 38 | 39 |
| 41 | 42 | 43 | 44 | 46 | 47 | 48 | 49 | 51 | 52 | 53 | 54 | 56 | 57 | 58 | 59 |
| 61 | 62 | 63 | 64 | 66 | 67 | 68 | 69 | 71 | 72 | 73 | 74 | 76 | 77 | 78 | 79 |
| 81 | 82 | 83 | 84 | 86 | 87 | 88 | 89 | 91 | 92 | 93 | 94 | 96 | 97 | 98 | 99 |

# break

- Thoát khỏi vòng lặp ngay cả khi biểu thức điều kiện của vòng lặp vẫn còn được thỏa mãn.
- Chú ý:
  - break dùng để thoát ra khỏi khối lặp hiện tại
  - break cũng dùng để thoát ra khỏi lệnh rẽ nhánh switch

# Ví dụ 2

```
#include <stdio.h>

int main() {
    int n;
    do {
        printf("Nhap n: "); scanf("%d", &n);
        if (n < 0) break;
        if (n > 10) {
            printf("Bo qua so nay!\n");
            continue;
        }
        printf("So vua nhap: %d\n", n);
    } while (n != 0);
    return 0;
}
```

Nhap n: 8  
So vua nhap: 8  
Nhap n: 9  
So vua nhap: 9  
Nhap n: 11  
Bo qua so nay!  
Nhap n: 12  
Bo qua so nay!  
Nhap n: -1

# Ví dụ 3

```
#include <stdio.h>
int main()
{
    int i;
    for (i = 1; i <= 10; i++)
    {
        if (i == 5) continue;
        printf("%5d", i);
        if (i == 7) break;
    }
    return 0;
}
```

1 2 3 4 6 7

# Ví dụ 4

```
#include <stdio.h>
int main()
{
    int i, j;
    for (i = 0; i < 10; i++)
    {
        for (j = 0; j < 10; j++)
            if (j > i) break;
        printf("i:%d j:%d\n", i, j);
    }
    return 0;
}
```

i:0 j:1  
i:1 j:2  
i:2 j:3  
i:3 j:4  
i:4 j:5  
i:5 j:6  
i:6 j:7  
i:7 j:8  
i:8 j:9  
i:9 j:10

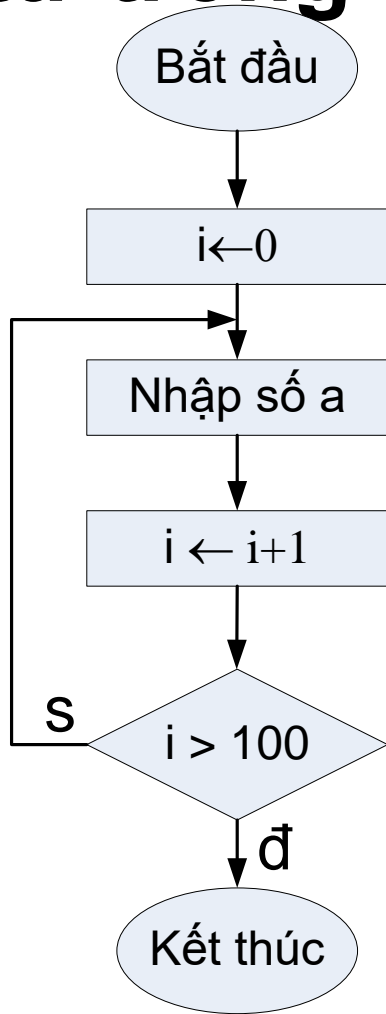
# Ví dụ tổng hợp

Viết chương trình thực hiện các công việc sau

- Nhập vào một dãy số cho tới khi
  - Tổng của dãy lớn hơn 1550 hoặc là
  - Số phần tử trong dãy lớn hơn 100
- Đưa ra số lượng phần tử nằm trong khoảng (35, 70)
- Đưa ra trung bình cộng của các phần tử chia hết cho 7

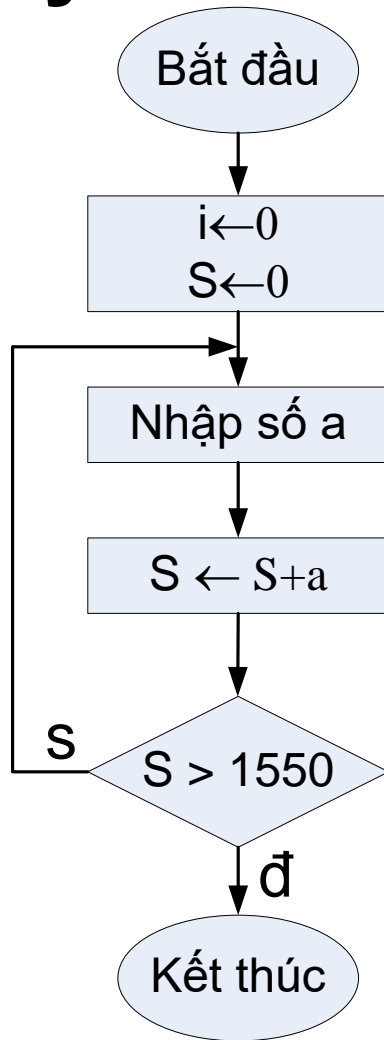


# Nhập một dãy số cho tới khi số phần tử trong dãy lớn hơn 100



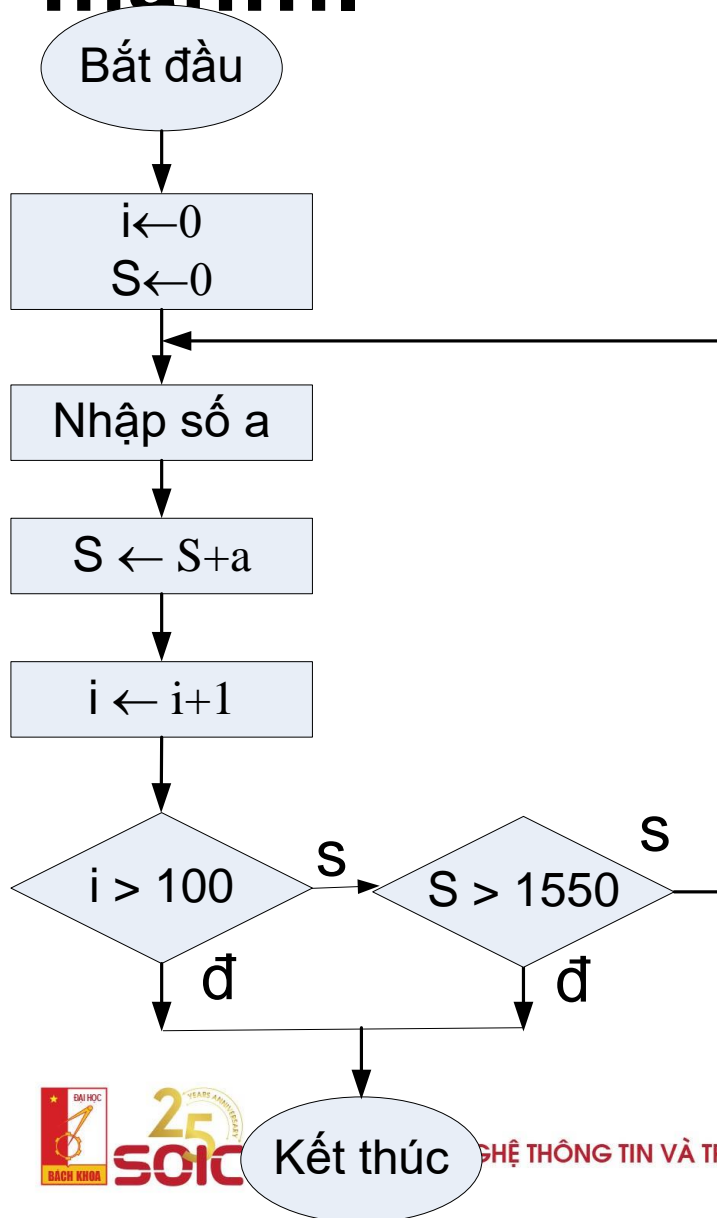
```
#include <stdio.h>
int main()
{
    int a, i = 0;
    do
    {
        printf("Nhap vao so nguyen:");
        scanf("%d", &a);
        i++;
    } while (i <= 100);
}
```

# Nhập một dãy số cho tới khi tổng của dãy lớn hơn 1550



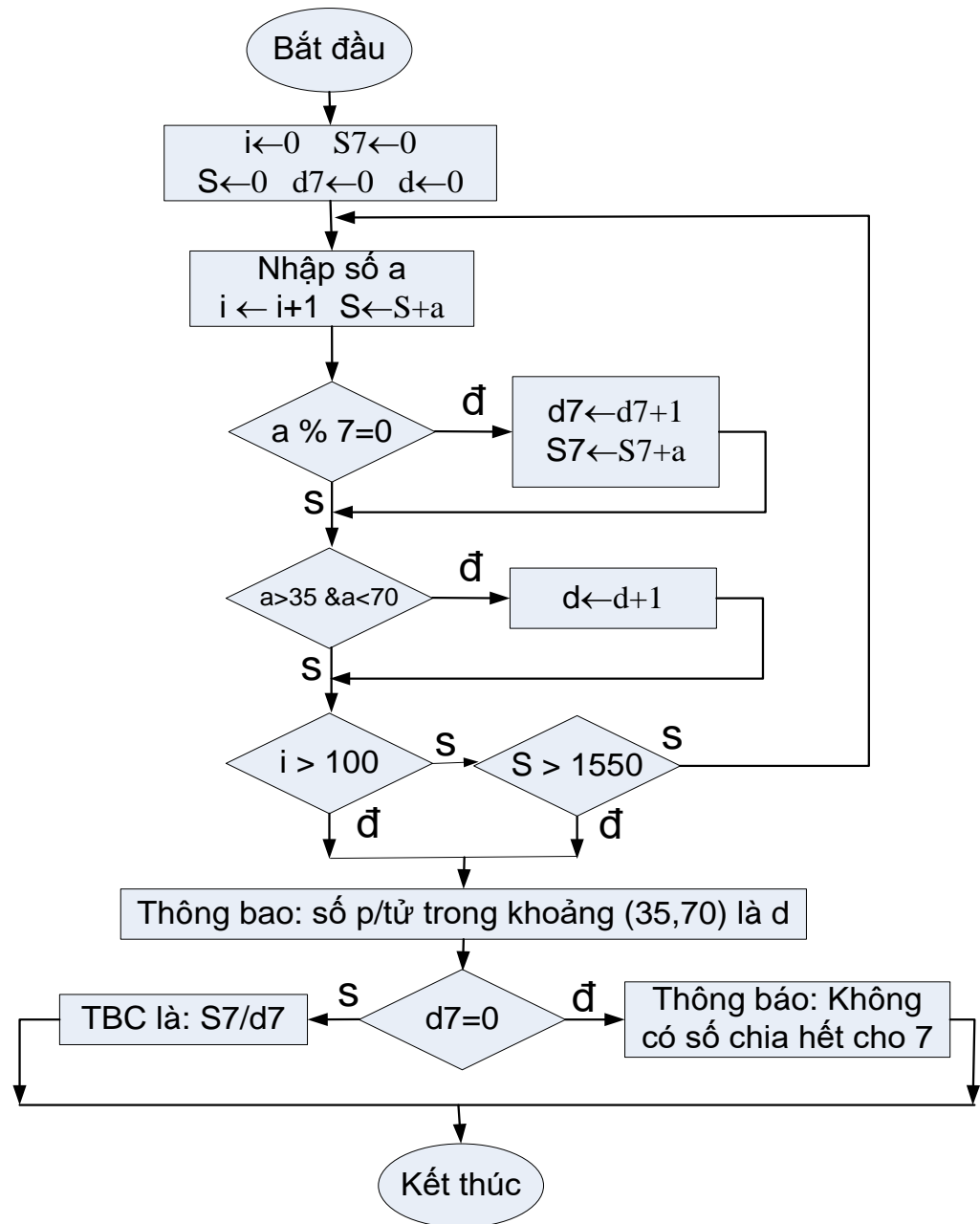
```
#include <stdio.h>
int main()
{
    int a, S = 0;
    do
    {
        printf("Nhap vao so nguyen:");
        scanf("%d", &a);
        S += a;
    } while (S <= 1550);
}
```

# Nhập một dãy số cho tới khi thỏa mãn....



```
#include <stdio.h>
int main()
{
    int a, i = 0, S = 0;
    do
    {
        printf("Nhap vao so nguyen:");
        scanf("%d", &a);
        S += a;
        i++;
    } while ((i <= 100) && (S <= 1550));
}
```

# Ví dụ tổng hợp



# Ví dụ tổng hợp

```
#include <stdio.h>
int main() {
    int a, i = 1, S7 = 0, d7 = 0, S = 0, d = 0;
    do {
        printf("Nhap vao so nguyen:"); scanf("%d", &a);
        i++;
        S += a;
        if (a % 7 == 0) {
            d7++;
            S7 += a;
        }
        if ((a > 35) && (a < 70)) d++;
    } while ((i <= 100) && (S <= 1550));
    printf("So phan tu trong khoang (35, 70) la %d\n", d);
    if (d7 == 0) printf("Khong co so nao chia het cho 7");
    else printf("TBC cac so chia het cho 7 %.4f", (float)S7/d7);
    return 0;
}
```

# Bài tập

1. Lập trình nhập vào số nguyên dương  $N$ . In các số nguyên tố nhỏ hơn  $N$ .
2. Lập trình nhập vào 2 số nguyên dương  $a, b$ . In ra ước số chung lớn nhất và bội số chung nhỏ nhất của  $a$  và  $b$ .
3. Lập trình nhập vào số nguyên dương  $N$ . In ra dãy số Fibonacci nhỏ hơn  $N$ .
4. Lập trình nhập vào số nguyên dương  $N$ . Kiểm tra xem  $N$  có phải là số hoàn hảo hay không?

# Bài tập

5. Viết chương trình yêu cầu người sử dụng nhập vào các số nguyên từ bàn phím cho đến khi nhập số 0 thì kết thúc. Tính trung bình cộng các số chẵn đã được nhập.
6. Đọc vào dãy số cho tới khi gặp một số dương chia hết cho 5; Tìm số lớn nhất của dãy và số lần xuất hiện các giá trị đó.

# Bài tập

7. Viết chương trình đọc x và n vào từ bàn phím rồi tính:

$$S_1 = \sqrt{x + \sqrt{x + \sqrt{x + \cdots + \sqrt{x}}}} \text{ n dấu căn}$$

$$S_2 = 1 + x + \frac{x^2}{2^2} + \frac{x^3}{3^3} + \cdots + \frac{x^n}{n^n}$$

$$S_3 = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

$$S_4 = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \cdots + \frac{(-1)^n x^n}{n!}$$



# Viết chương trình đọc x và n vào từ bàn phím rồi tính

$$S = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

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

```
int main()
```

```
{
```

```
    int n, i;
```

```
    float x, u = 1.0, S = 1.0;
```

```
    printf("Nhap vao so nguyen n : "); scanf("%d", &n);
```

```
    printf("Nhap vao so thuc x : "); scanf("%f", &x);
```

```
    for (i = 1; i <= n; i++) {
```

```
        u *= x / i;
```

```
        S += u;
```

```
    }
```

```
    printf("Ket qua la %.8f", S);
```

```
}
```

Nhap vao so nguyen n : 20

Nhap vao so thuc x : 1

Ket qua la 2.71828198

Nhap vao so nguyen n : 50

Nhap vao so thuc x : 2.31

Ket qua la 10.07442379

# Bài tập (Tính tổng vô hạn)

Tính xấp xỉ số PI theo công thức

$$\frac{\pi}{4} = 1 + \sum_{i=1}^{\infty} \frac{(-1)^i}{2i+1}$$

Với sai số eps (epsilon) nhập từ bàn phím (số hạng cuối cùng có giá trị tuyệt đối không lớn hơn eps)

# Bài tập (Tính tổng vô hạn)

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

int main() {
    float eps, S = 0, u = 1;
    int i = 0, p = 1;
    printf("Nhap do chinh xac: "); scanf("%f", &eps);
    while (fabs(u) > eps) {
        S += u;
        i++;
        p = -p;
        u = p / (float)(2 * i + 1);
    }
    printf("PI = %.9f", S * 4);
    return 0;
}
```

# Bài tập (Tính tổng vô hạn)

8. Đọc  $x$  và  $\text{eps}$  từ bàn phím và tính biểu thức sau với độ chính xác nhỏ hơn  $\text{eps}$ :

$$S_1 = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots + \frac{(-1)^n x^n}{n!} + \dots$$

$$S_2 = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + \frac{(-1)^n}{(2n+1)!} x^{2n+1} + \dots // \sin(x)$$

$$S_3 = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + \frac{(-1)^n}{(2n)!} x^{2n} + \dots // \cos(x)$$

# Tổng kết chương 4

## 1. Câu lệnh khối

Đặt trong cặp ngoặc nhọn { }

## 2. Cấu trúc rẽ nhánh

– if (biểu\_thức), if (biểu\_thức) ... else

## 3. Cấu trúc lệnh lựa chọn

– switch (biểu\_thức) {(case/break/default)}

## 4. Cấu trúc lặp

– for(biểu\_thức\_1; biểu\_thức\_2; biểu\_thức\_3) CâuLệnh;

– while (biểu\_thức) CâuLệnh;

– do Câu\_Lệnh while (biểu\_thức);

## 5. Các lệnh thay đổi cấu trúc lập trình

– continue/ break