

CHƯƠNG 4

CÁC CẤU TRÚC ĐIỀU KHIỂN

Nội Dung

- Giới thiệu
- Phát biểu
- Cấu trúc chọn
 - if - if else
 - switch/case
- Cấu trúc lặp
 - for
 - while - do
 - do - while

Giới thiệu

- *Cấu trúc điều khiển* qui định thứ tự thực hiện thao tác hay tính toán trong chương trình.
- Có ba cấu trúc điều khiển cơ bản là cấu trúc tuần tự, cấu trúc chọn, cấu trúc lặp.
- *Cấu trúc tuần tự* là cấu trúc mặc nhiên.
- *Cấu trúc chọn* biểu diễn các quyết định.
- *Cấu trúc lặp* cho phép lặp lại nhiều lần một số thao tác.

Phát biểu

- Một *phát biểu đơn* trong C là một biểu thức bất kỳ kết thúc bởi dấu chấm phẩy (;).
- Các dấu { và } được dùng để gom nhóm các khai báo và phát biểu thành một *phát biểu ghép* hay một *khối*.
- Một khối, về mặt cú pháp, tương đương với một phát biểu đơn.

If-Else

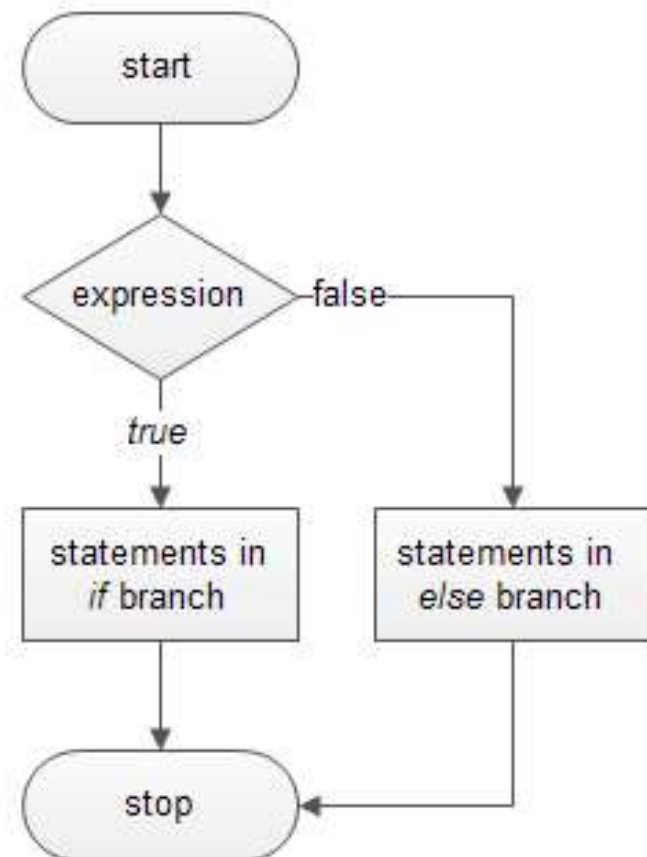
- *if – else*: Phát biểu if – else biểu diễn quyết định

if (expression)

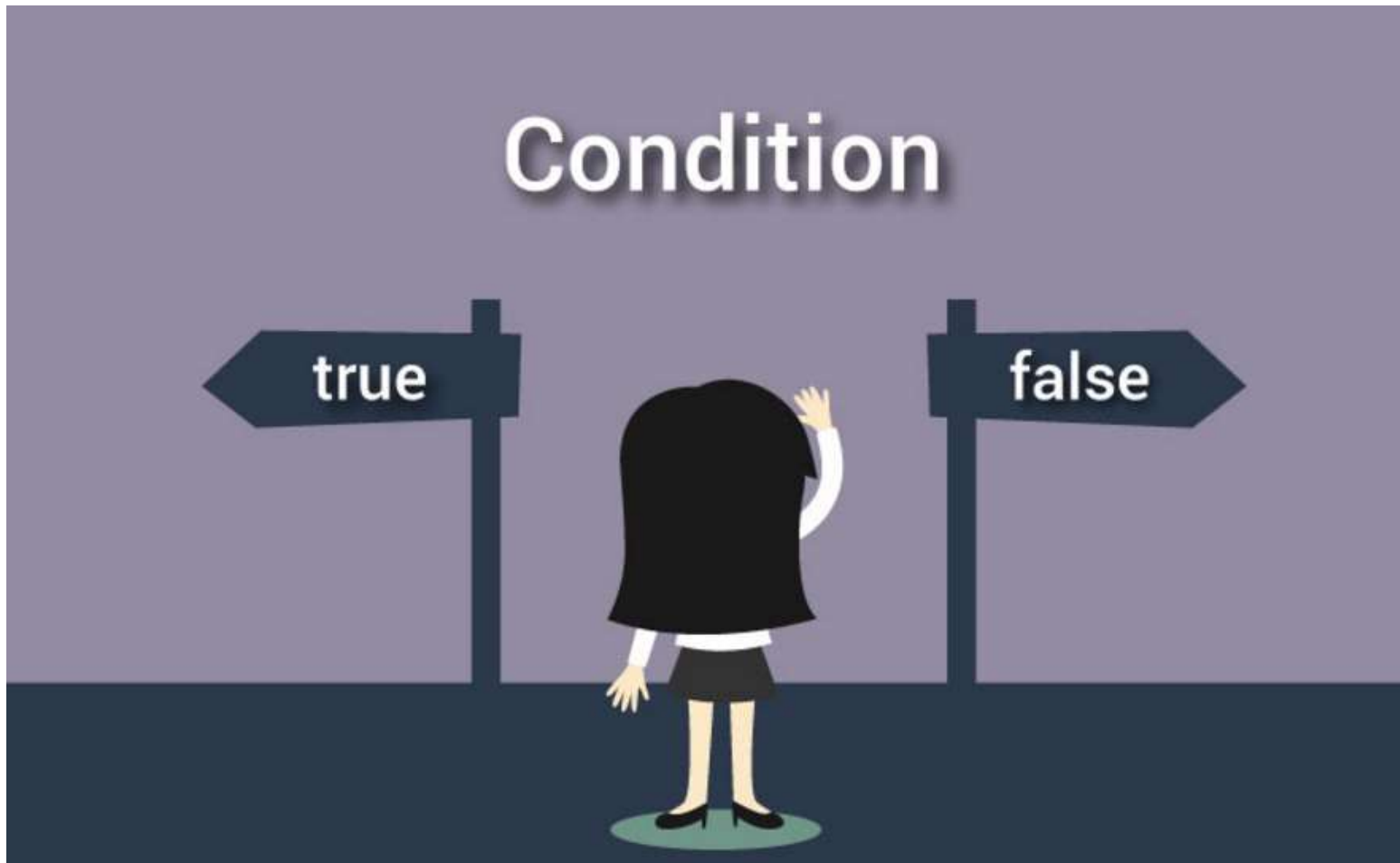
statement¹

[else

statement²]



If-Else



If-Else

- Phần else có thể được bỏ qua

```
if (dtb >= 9)
    printf("Xuat sac");
```

- Các phát biểu if có thể lồng nhau

```
if (a+b > c)
    if (b+c > a)
        if (c+a > b)
            printf("a,b,c la 3 canh cua
mot tam giac");
```

If-Else

- if lồng nhau có thể viết gần như tương đương bằng cách sử dụng phép toán **&&**

```
if (a+b > c && b+c > a && c+a > b)
    printf("a,b,c la 3 canh cua mot
    tam giac");
```

- Ví dụ: Viết phát biểu tương đương pb sau:

```
if (x != 0)
    if (1/x < 1)
        y = asin(sqrt(1-1/x/x));
```


If-Else

- if lồng nhau có thể viết gần như tương đương bằng cách sử dụng phép toán **&&**

```
if (a+b > c && b+c > a && c+a > b)
    printf("a,b,c la 3 canh cua mot
    tam giac");
```

- Ví dụ: Viết phát biểu tương đương pb sau:

```
if (x != 0)
    if (1/x < 1)
        y = asin(sqrt(1-1/x/x));
```

If-Else

- *If* có cả phần *else* cho phép chọn lựa một trong 2 nhánh của quyết định.

```
if (dtb >= 5)
```

```
    printf("Dat");
```

```
else
```

```
    printf("May man lan sau");
```

If-Else lồng nhau

```
if (dtb >= 8)
    printf("Gioi");
else
    if (dtb >= 7)
        printf("Kha");
    else
        if (dtb >= 5)
            printf("Trung binh");
        else
            printf("Hen gap lai");
```

If-Else lồng nhau

```
if (dtb >= 8)
    printf("Gioi");
else if (dtb >= 7)
    printf("Kha");
else if (dtb >= 5)
    printf("Trung binh");
else
    printf("Hen gap lai");
```

If-Else

```
if (th < 1 || th > 12)
{
    printf("Thang khong hop le");
    return 1;
}
if (th == 2)
{
    if (nam % 400 == 0 || nam % 4 == 0 && nam % 100 != 0)
        songay = 29;
    else
        songay = 28;
}
if (th == 4 || th == 6 || th == 9 || th == 11)
    songay = 30;
else
    songay = 31;
```

Switch - case

- Phát biểu switch biểu diễn một quyết định nhiều nhánh. Giá trị kiểm tra được so sánh với các mẫu để xác định nhánh nào được chọn.

```
switch (expression) {  
    case const-expr: statements  
    case const-expr: statements  
    default: statements  
}
```

Switch - case

- Mỗi case là một biểu thức hằng thuộc kiểu đếm được.
- Phát biểu `break` được dùng để thoát tức thời khỏi switch.

switch-case – Ví dụ

```
switch (th) {  
case 2:  
    songay = 28 + (nam % 400 == 0 || nam % 4 == 0 && nam % 100  
        != 0);  
    break;  
case 4: case 6: case 9: case 11:  
    songay = 30;  
    break;  
case 1: case 3: case 5: case 7: case 8: case 10: case 12:  
    songay = 30;  
    break;  
default:  
    printf("Thang khong hop le");  
    return 1;  
}  
printf("So ngay trong thang %d nam %d la: %d\n", th, nam,  
songay);
```

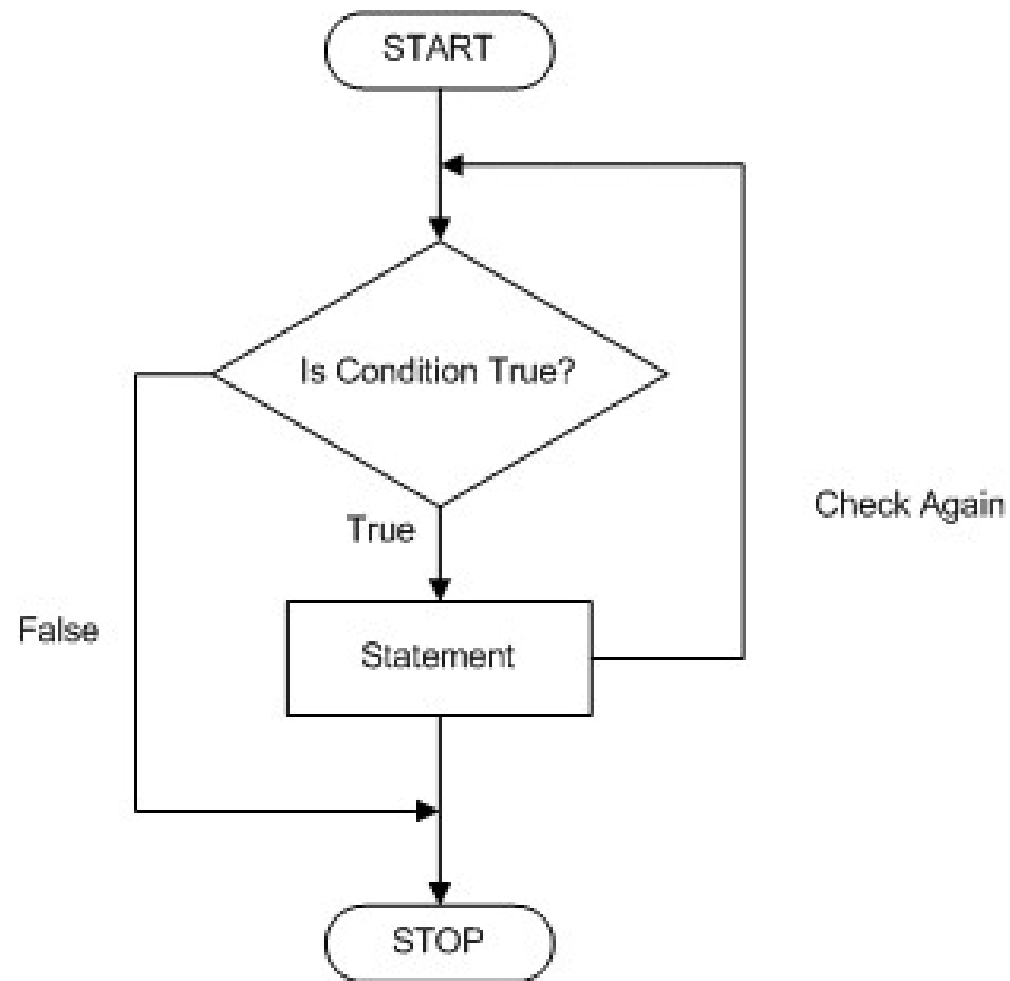

Lặp – phát biểu while

- Cú pháp:

```
while (expression)  
statement
```

- Phát biểu while lặp lại việc thực hiện *statement* cho đến khi biểu thức điều kiện (*expression*) có giá trị sai.

Lặp – phát biểu while



Phát biểu while – Ví dụ

```
void main()
{
    int n, m, tong = 0;
    printf("Nhap vao so n: "); scanf("%d", &n);
    m = n;
    while(m) {
        tong += m%10;
        m /= 10;
    }
    printf("Tong cac chu so cua %d la %d\n", n, tong);
}
```

The diagram illustrates the while loop structure. A box labeled "Test condition" points to the `while(m)` part of the code. A box labeled "Loop body" points to the block containing `tong += m%10;` and `m /= 10;`.

Phát biểu while – Ví dụ

```
void main()
{
    int n, i, S;
    printf("Tính tổng n số nguyên đầu tiên.\n");
    printf("Nhập n: "); scanf("%d", &n);
    S = 0; i = 1;
    while(i <= n)
    {
        S += i;
        i++;
    }
    printf("Tổng của %d số nguyên đầu tiên là: %d\n",
n, S);
}
```

The diagram illustrates the components of the while loop in the provided code. A box labeled "Test condition" has an arrow pointing to the expression `i <= n` within the `while` statement. Another box labeled "Loop body" has an arrow pointing to the block of code inside the curly braces, which contains `S += i;` and `i++;`. The `i++;` line is also circled in blue.

Phát biểu while – Ví dụ

```
void main()
{
    int n, i, S;
    printf("Tính tổng n số nguyên đầu tiên.\n");
    printf("Nhập n: "); scanf("%d", &n);
    S = 0; i = 1;
    while(i <= n)
    {
        S += i;
        i++;
    }
    printf("Tổng của %d số nguyên đầu tiên là: %d\n",
n, S);
}
```

The diagram illustrates the components of the while loop in the provided code. A box labeled "Test condition" points to the expression `i <= n` within the `while` statement. Another box labeled "Loop body" points to the block of code inside the curly braces, which contains `S += i;` and `i++;`. The `i++;` statement is circled in blue.

Phát biểu while – Bài tập

1. Viết chương trình đếm các chữ số của một số nguyên.
2. Viết chương trình tính ước số chung lớn nhất của 2 số.
3. Viết chương trình kiểm tra một số nguyên có đối xứng không.

Phát biểu while

Lưu ý:

- Phát biểu while có thể không lặp lần nào.
- Phát biểu while phải kết thúc sau một số hữu hạn lần lặp. Muốn vậy phải có một trong các điều kiện sau:
 - *Có ít nhất một phát biểu trong phần thân của phát biểu while làm thay đổi giá trị của điều kiện.*
 - *Có phát biểu break hoặc return bên trong phần thân của phát biểu while.*

Phát biểu while – Ví dụ

```
void main()
{
    int n, i, S;
    printf("Tính tong n so nguyen dau tien.\n");
    printf("Nhap n: "); scanf("%d", &n);
    S = 0; i = 1;
    while(i <= n)
    {
        S += i;
    }
    printf("Tong cua %d so nguyen dau tien la: %d\n", n, S);
}
```

Test condition

Loop body

Lặp thiên thu!!!

Lặp – phát biểu for

- Cú pháp:

for (*expr*₁; *expr*₂; *expr*₃)
statement

- Phát biểu **for** lặp lại việc thực hiện *statement* cho đến khi biểu thức điều kiện (*expr*₂) có giá trị sai.
- Thông thường *expr*₁, *expr*₃ là phép gán hoặc gọi hàm *expr*₂ là biểu thức điều kiện.

Lặp – phát biểu for

- Phát biểu for tương đương với:

```
expr1;  
while (expr2) {  
    statement  
    expr3;  
}
```

- Biểu thức *expr*₃ *thường* có tác dụng làm thay đổi *expr*₂.

Lặp – phát biểu for

- $expr_1$, $expr_2$, $expr_3$ đều không bắt buộc phải có (optional).

```
for (;;) {  
    ...  
}
```

sẽ lặp vô tận, trừ khi có *break* hoặc *return* trong phần thân của vòng lặp.

- Phát biểu for rất thuận tiện khi muốn cài đặt vòng lặp với số lần lặp được biết trước.

Phát biểu for – Ví dụ

```
void main()
{
    int n, i, S;
    printf("Tính tổng n số nguyên đầu tiên.\n");
    printf("Nhập n: "); scanf("%d", &n);
    S = 0;
    for(i = 1; i <= n; i++)
    {
        S += i;
    }
    printf("Tổng của %d số nguyên đầu tiên là: %d\n", n, S);
}
```

Diagram illustrating the components of a for loop:

- Khởi động** (Initialization): `i = 1`
- Biểu thức điều kiện (Test condition)**: `i <= n`
- Thường để thay đổi biểu thức điều kiện** (Usually to change the test condition): `i++`
- Thân vòng lặp (Loop body)**: `S += i;`

Phát biểu for – Ví dụ

```
void main()
{
    int n, m, tong = 0;
    printf("Nhập vào số n: "); scanf("%d",
    &n);
    for (m = n; m; m /= 10)
        tong += m%10;
    printf("Tổng các chữ số của %d là %d\n",
    n, tong);
}
```

The diagram illustrates the components of the `for` loop in the provided code snippet. It features four callout boxes with arrows pointing to specific parts of the `for` loop syntax:

- Khởi động** (Initialization): Points to the `m = n` part of the loop header.
- Biểu thức điều kiện (Test condition)**: Points to the `m` part of the loop header, which acts as the condition.
- Thường để thay đổi biểu thức điều kiện** (Usually to change the test condition): Points to the `m /= 10` part of the loop header.
- Thân vòng lặp (Loop body)**: Points to the `tong += m%10;` statement inside the loop.

Lặp – phát biểu do - while

- Cú pháp:

do

statement

while (expression) ;

- Phát biểu *while* kiểm tra điều kiện ở đầu vòng lặp. Phát biểu do-while ngược lại, kiểm tra ở cuối vòng lặp. Vì vậy thân vòng lặp *do – while* luôn luôn được thực hiện *ít nhất một lần*.

Phát biểu do – while

Ví dụ:

- Viết chương trình đếm số chữ số của một số nguyên dương bằng 2 cách:
 1. Dùng vòng lặp while
 2. Dùng vòng lặp do - while

Phát biểu do – while – Ví dụ

```
void main()
{
    int n, m, tong = 0;
    printf("Nhap vao so n: ");
    scanf("%d", &n);
    m = n;
    int dem = 0;
    while (m)
    {
        dem++;
        m /= 10;
    }
    printf("So chu so cua %d la %d\n", n, dem);
}
```


Phát biểu do – while – Ví dụ

Output:

Nhap vao so n: 1243

So chu so cua 1243 la 4

Press any key to continue . . .

Output:

Nhap vao so n: 0

So chu so cua 0 la 0

Press any key to continue . . .



Phát biểu do – while – Ví dụ

```
void main()
{
    int n, m, tong = 0;
    printf("Nhap vao so n: ");
    scanf("%d", &n);
    m = n;
    int dem = 0;
    do {
        dem++;
        m /= 10;
    } while(m);
    printf("So chu so cua %d la %d\n", n, dem);
}
```

Phát biểu do – while – Ví dụ

Output:

Nhap vao so n: 1243

So chu so cua 1243 la 4

Press any key to continue . . .

Output:

Nhap vao so n: 0

So chu so cua 0 la 1

Press any key to continue . . .

Break và Continue

- Phát biểu `break` cho phép thoát khỏi vòng lặp ở giữa vòng lặp thay vì ở đầu hoặc ở cuối.
- Phát biểu `break` thoát khỏi vòng lặp ngay lập tức mà không cần kiểm tra điều kiện thoát. Thường được dùng với phát biểu `if`.
- Phát biểu `break` có thể được dùng cho `switch` và các vòng lặp.

Break và Continue

- Phát biểu `continue` bỏ qua phần còn lại của lần lặp đang thực thi và bắt đầu lần lặp mới.
- Với các phát biểu `while` và do `while` phát biểu `continue` chuyển điều khiển đến phần kiểm tra điều kiện.

Break và Continue

Ví dụ:

- Viết chương trình cho phép nhập vào một số nguyên dương n , tính và xuất ra số nguyên tố đầu tiên đứng sau n .

Break và Continue – Ví dụ

```
// Ver 1: not so good
void main()
{
    int n,m,a;
    printf("Tim so nguyen to ngay sau so nguyen duong n.\n");
    printf("Nhap n: ");
    scanf("%d", &n);
    for(m = n+1; ; m++)
    {
        if (m == 2 || m == 3) break;
        a = 2;
        if (m % a == 0) continue;
        while (m % a != 0)
            a++;
        if (a >= m) break;
    }
    printf("So nguyen to ke tiep cua %d la: %d\n", n, m);
}
```

Break và Continue – Ví dụ

```
// Ver 2: much better
void main()
{
    int n,m,a;
    printf("Tim so nguyen to ngay sau so nguyen duong n.\n");
    printf("Nhap n: ");
    scanf("%d", &n);
    for(m = n+1; ; m++)
    {
        if (m == 2 || m == 3) break;
        a = 2;
        if (m % a == 0) continue;
        while (m % a != 0 && a <= m/a)
            a++;
        if (a > m/a) break;
    }
    printf("So nguyen to ke tiep cua %d la: %d\n", n, m);
}
```


Break và Continue

Bài tập:

- Viết lại chương trình kể trên, cải tiến cho hiệu quả thêm.