

LẬP TRÌNH CĂN BẢN

Phần 2 - Chương 3 CÁC LỆNH CÓ CẤU TRÚC





Nội dung chương này

- Khởi lệnh trong C
- Cấu trúc rẽ nhánh
- Cấu trúc lựa chọn
- Cấu trúc vòng lặp
- Các câu lệnh “đặc biệt”



Khởi lệnh trong C (1)

- Là 1 dãy các khai báo cùng với các câu lệnh nằm trong cặp dấu ngoặc móc { và }.

```
#include <stdio.h>
#include<conio.h>
int main ()
{   /*đây là đầu khối*/
    char ten[50];
    printf("Xin cho biet ten cua ban !");
    scanf("%s",ten);
    getch();
    return 0;
} /*đây là cuối khối*/
```



Khối lệnh trong C (2)

- 1 khối lệnh có thể chứa nhiều khối lệnh khác gọi là **khối lệnh lồng nhau** (không hạn chế).

```
{  
    ... lệnh;  
    {  
        int a,b;      /*biến a, b trong khối lệnh thứ nhất*/  
        ... lệnh;  
    }  
    ...lệnh;  
    {  
        int a,b;      /*biến a,b trong khối lệnh thứ hai*/  
        ... lệnh;  
    }  
}
```



Phạm vi các biến

- Có thể khai báo các biến cùng tên trong các khối.
- Nếu một biến được khai báo bên ngoài khối lệnh và không trùng tên với biến bên trong khối lệnh thì nó cũng dùng được bên trong khối.
- Một khối lệnh con có thể sử dụng các biến bên ngoài, nhưng điều ngược lại không đúng.

```
{  
    int a, b, c;  
    ...lệnh;  
    {  
        int c, d;  
        ...lệnh;  
    }  
}
```



Cấu trúc rẽ nhánh (if)

```
if (boolean_expression) statement;
```

- Statement được thực hiện nếu **boolean_expression** có giá trị đúng (true), !=0.
- Khối **else** là tùy chọn

```
if (boolean_expression) statement1; else statement2;
```

`boolean_expression !=0` => Statement1 được thực hiện

`boolean_expression ==0` => Statement2 được thực hiện



Ví dụ - Lệnh if

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

int main () {
    float a;
    printf("Nhap a = "); scanf("%f",&a);
    if (a !=0 )
        printf("Nghich dao cua %f la %f",a,1/a);
    getch();
    return 0;
}
```



Ví dụ - Lệnh if-else

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

int main () {
    float a;
    printf("Nhap a = "); scanf("%f", &a);
    if (a != 0 )
        printf("Nghich dao cua %f la %f", a, 1/a);
    else
        printf("Khong the tim duoc nghich dao cua a");
    getch();
    return 0;
}
```




Câu lệnh và khối lệnh

- C cho phép nhóm các câu lệnh liên tiếp vào 1 khối.
- 1 khối lệnh có thể được dùng như 1 lệnh đơn.
- Ví dụ:

```
if (a>b)
{
    max = a;
    z++;
}
else
    max = b;
```

} Khối lệnh

} Câu lệnh đơn

Nhầm lẫn khi dùng if



```
#include <stdio.h>
main()
{
    int number;
    printf("enter number: ");
    scanf("%d", &number);

    if (number = 0)
        printf("equal 0\n");
    else
        printf("not equal 0\n");
}
```

- Chương trình trên sai ở đâu?



Chú ý khi dùng if-else

- Câu lệnh **if-else** lồng nhau
- **else** sẽ kết hợp với **if** gần nhất chứa có **else**
- Trong trường **if** bên trong không có **else** thì phải viết nó trong cặp dấu **{ }** để tránh sự kết hợp else if sai.
- Ví dụ

```
if ( so1>0)
    if (so2 > so3)
        a=so2;
    else /*else của if (so2>so3)*/
        a=so3;
```

```
if (so1>0)
{
    if (so2>so3)/*lệnh if này không có else*/
        a=so2;
}
else /*else của if (so1>0)*/
    a=so3;
```

Cấu trúc lựa chọn (switch-case) (1)



- C cung cấp 1 cấu trúc đẹp - dùng 1 dãy các câu lệnh **if**.

```
int day; // day of week 0=Monday, 1=Tuesday, 2=...
if (day == 0) {
    printf("Monday");
} else if (day == 1) {
    printf("Tuesday");
} else if (day == 2) {
    printf("Wednesday");
} else if (day == 3) {
    printf("Thursday");
} else if (day == 4) {
    printf("Friday");
} else if (day == 5) {
    printf("Saturday");
} else if (day == 6) {
    printf("sunday");
} else {
    printf("invalid day of week ");
}
```

Cấu trúc lựa chọn (switch-case) (2)



- **switch-case** có thể đưa đến mã máy (machine code) hiệu quả hơn (vì jump tables có thể được dùng)

```
int day; // day of week 0=Monday, 1=Tuesday, 2=...
switch(day) {
case 0:
    printf("Monday");
    break;
case 1:
    printf("Tuesday");
    break;
...
case 6:
    printf("Sunday");
    break;
default:
    printf("invalid day of week");
}
```

Cấu trúc lựa chọn (switch-case) (3)



```
switch (expr) {  
  case value1:  
    statement_sequence1;  
    break;  
  case value2:  
    statement_sequence2;  
    break;  
  case value3:  
    statement_sequence3;  
    return;  
  .  
  .  
  .  
  [default:  
    default_statements; ]  
}
```

● Cú pháp:

- Tính giá trị của biểu thức **expr** trước.
- Nếu giá trị **expr** bằng **value1** thì thực hiện **statement_sequence1** rồi thoát.
- Nếu giá trị **expr** khác **value1** thì so sánh nó với **value2**, nếu bằng **value2** thì thực hiện **statement_sequence2** rồi thoát.
- Cứ như thế, so sánh tới giá trị **n**.
- Nếu tất cả các phép so sánh trên đều sai thì thực hiện **default_statements** của trường hợp **default**.

Cấu trúc lựa chọn (switch-case) (4)



- **Các chú ý:**

- Kiểu của `expr` và các `valuei` phải là kiểu số nguyên (`int`, `char`, `long`, ...).
- Nếu `break/return` vắng mặt, câu lệnh trong các `case` bên dưới có thể được thực hiện cho đến khi gặp `break/return` hoặc kết thúc lệnh `switch`.



Ví dụ - switch-case (1)

Tính

```
c = compute('+', 46, 53);
```

```
int compute(char op, int a, int b) {  
    switch(op) {  
        case '-':  
            b *= -1;  
        case '+':  
            return a+b;  
        case '*':  
            return a*b;  
        case ':':  
        case '/':  
            return a/b;  
        default:  
            printf("invalid op %c", c);  
            return ERR;  
    }  
}
```

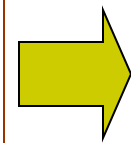
If `op == '-'` the statement for case `+` is also executed

return `a/b` for both cases `:` and `/`



Ví dụ - switch-case (2)

```
#include <stdio.h>
#include <conio.h>
int main ()
{
    int thang;
    clrscr();
    printf("\n Nhap vao thangs trong nam ");
    scanf("%d",&thang);
    switch(thang)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            printf("\n Thang %d co 31 ngay ",thang);
            break;
        case 4:
        case 6:
        case 9:
        case 11:
            printf("\n Thang %d co 30 ngay ",thang);
            break;
        case 2:
            printf ("\ Thang 2 co 28 hoac 29 ngay");
            break;
        default :
            printf("\n Khong co thang %d", thang);
            break;
    }
    getch();
    return 0;
}
```



In ra số ngày
của 1 tháng



Cấu trúc lặp

- Cho phép lặp lại thực hiện 1 công việc nhiều lần.
- Có 2 loại:
 - Lặp với số lần xác định
 - **for**
 - Lặp với số lần không xác định
 - **while**
 - **do-while**



Vòng lặp for (1)

for (Biểu thức 1; biểu thức 2; biểu thức 3) <Công việc>;

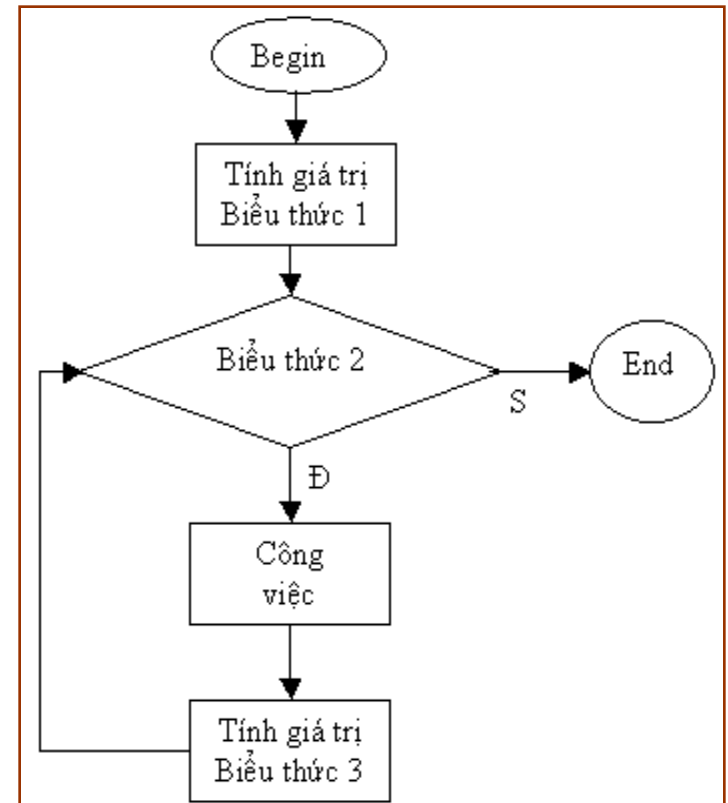
- Thứ tự thực hiện:

B1: Tính giá trị **biểu thức 1**

B2: Tính giá trị **biểu thức 2**

- Nếu giá trị **biểu thức 2** là sai ($\neq 0$) \Rightarrow thoát khỏi **for**
- Nếu giá trị **biểu thức 2** là đúng ($= 0$) \Rightarrow thực hiện <**Công việc**>

B3: Tính giá trị biểu thức 3
rồi quay lại B2





Vòng lặp for (2)

Ví dụ:

```
#include <stdio.h>
main()
{
    int i;          // i is loop variable
    for (① i=1; ② i<=10; ③ i++) {
        printf("%d: %d\n", i, i*i);
    }
}
```

- ① Khởi tạo i giá trị 1
- ② Thoát vòng lặp nếu $i > 10$
- ③ Tăng i sau mỗi lần lặp

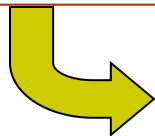


Vòng lặp for (2)

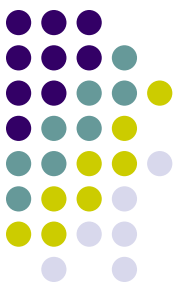
- Chương trình in dãy số nguyên từ 1..10

```
#include <stdio.h>
#include<conio.h>
int main()
{
    int i;
    clrscr();
    printf("\n Day so tu 1 den 10 :");
    for(i=1; i<=10; i++)
        printf("%d ",i);
    getch();
    return 0;
}
```

output



Day so tu 1 den 10 :1 2 3 4 5 6 7 8 9 10



Vòng lặp for (3)

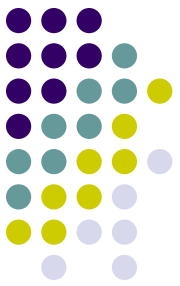
- Nhập số nguyên n. Tính tổng các số nguyên từ 1..n.

```
#include <stdio.h>
#include <conio.h>
int main ()
{
    unsigned int n,i,tong;
    clrscr();
    printf("\nNhập vào số nguyên dương n:");scanf("%d",&n);
    tong=0;
    for (i=1; i<=n; i++)
        tong+=i;
    printf("\n Tổng từ 1 đến %d =%d ",n,tong);
    getch();
    return 0;
}
```

output



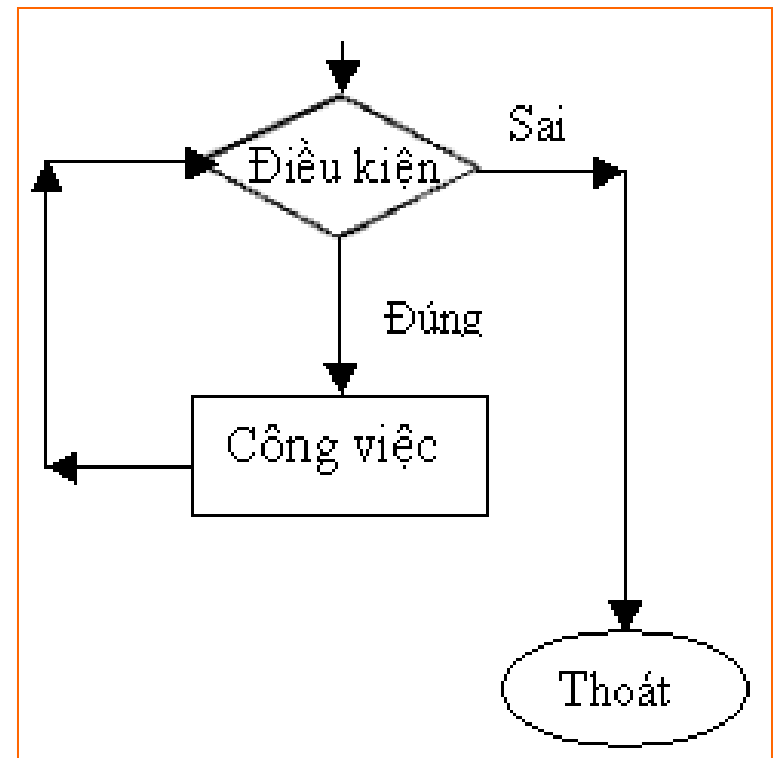
```
Nhập vào số nguyên dương n:9
Tổng từ 1 đến 9 =45 _
```



Vòng lặp while (1)

while (*Biểu thức điều kiện*) <Công việc>;

- <Công việc>: có thể là 1 câu lệnh hay 1 khối lệnh.
- **Các bước thực hiện:**
 - Kiểm tra *Biểu thức điều kiện* trước.
 - Nếu điều kiện sai ($==0$) thì thoát khỏi lệnh while.
 - Nếu điều kiện đúng ($!=0$) thì thực hiện công việc rồi quay lại kiểm tra điều kiện tiếp.



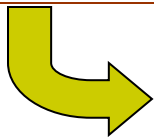


Vòng lặp while (2)

- Chương trình in dãy số nguyên từ 1..10

```
#include <stdio.h>
#include <conio.h>
int main () {
    int i;
    clrscr();
    printf("\n Day so tu 1 den 10 :");
    i=1;
    while(i<=10)
        printf("%d ", i++);
    getch();
    return 0;
}
```

output



Day so tu 1 den 10 :1 2 3 4 5 6 7 8 9 10



Vòng lặp while (3)

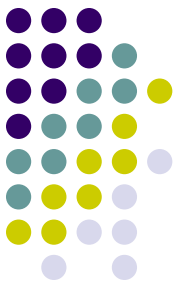
- Nhập số nguyên n. Tính tổng các số nguyên từ 1..n.

```
#include <stdio.h>
#include <conio.h>
int main () {
    unsigned int n,i,tong;
    clrscr();
    printf("\n Nhập vào số nguyên dương n:");
    scanf("%d",&n);
    tong=0;
    i=1;
    while(i<=n){
        tong+=i;
        i++;
    }
    printf("\n Tổng từ 1 đến %d =%d ",n,tong);
    getch();
    return 0;
}
```

output



```
Nhap vào số nguyên dương n:9
Tổng từ 1 đến 9 =45 _
```



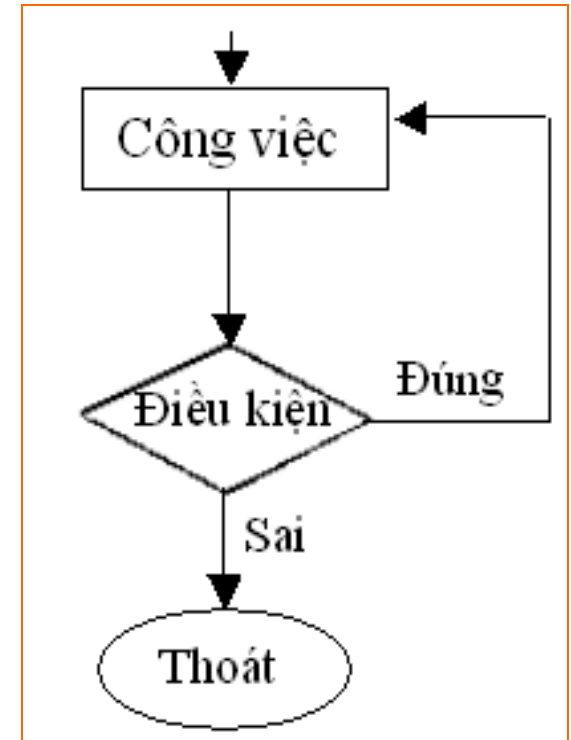
Vòng lặp do-while (1)

do

<Công việc>

while (*<Biểu thức điều kiện>*);

- **<Công việc>**: có thể là 1 câu lệnh hay 1 khối lệnh.
- **Các bước thực hiện:**
 - Công việc được thực hiện **trước**, sau đó mới kiểm tra **điều kiện**.
 - Nếu điều kiện sai thì thoát khỏi lệnh do-while.
 - Nếu điều kiện còn đúng thì thực hiện công việc rồi quay lại kiểm tra điều kiện tiếp.






Vòng lặp do-while (2)

- Chương trình in dãy số nguyên từ 1..10

```
#include <stdio.h>
#include <conio.h>
int main () {
    int i;
    clrscr();
    printf("\n Day so tu 1 den 10 :");
    i=1;
    do
        printf("%d ",i++);
    while (i<=10);
    getch();
    return 0;
}
```

output



Day so tu 1 den 10 :1 2 3 4 5 6 7 8 9 10




Vòng lặp do-while (3)

- Nhập số nguyên n.
Tính tổng các số nguyên từ 1..n.

```
#include <stdio.h>
#include <conio.h>
int main () {
    unsigned int n, i, tong;
    clrscr();
    printf("\n Nhap vao so nguyen duong n: ");
    scanf("%d", &n);
    tong=0;
    i=1;
    do{
        tong+=i;
        i++;
    }while(i<=n);
    printf("\n Tong tu 1 den %d =%d ", n, tong);
    getch();
    return 0;
}
```

output



```
Nhap vao so nguyen duong n:9
Tong tu 1 den 9 =45 _
```



So sánh các vòng lặp

- *Vòng lặp for/while:*
 - Kiểm tra điều kiện trước thực hiện công việc sau.
 - Công việc có thể không được thực hiện lần nào.
 - Vòng lặp kết thúc khi nào điều kiện sai.
- *Vòng lặp do-while*
 - Thực hiện công việc trước kiểm tra điều kiện sau.
 - Công việc được thực hiện ít nhất 1 lần.
 - Vòng lặp kết thúc khi nào điều kiện sai.



Câu lệnh đặc biệt

- Lệnh **break**
 - Dừng để thoát khỏi **vòng lặp** hoặc **switch-case**.
 - Tiếp tục thực hiện lệnh liền sau đó.
- Lệnh **continue**
 - Trong vòng lặp, khi gặp lệnh **continue**, chương trình sẽ bỏ qua các câu lệnh sau **continue**
 - **for**: quay lên tính trị cho biểu thức 3, rồi kiểm tra điều kiện coi có lặp tiếp không.
 - **while/do-while**: kiểm tra điều kiện coi có lặp tiếp không.

Hết chương

