

CHƯƠNG 2 CẤU TRÚC ĐIỀU KHIỂN

Tìm hiểu và cài đặt các cấu trúc rẽ nhánh, lựa chọn, lặp và các ký hiệu phép toán trong ngôn ngữ C. Mô tả cách hoạt động và hướng dẫn chạy từng bước chương trình.

I. TÓM TẮT LÝ THUYẾT

I.1. Các ký hiệu

STT	KÝ HIỆU	DIỄN GIẢI	VÍ DỤ
1	{ }	Bắt đầu và kết thúc hàm hay khối lệnh.	void main() { }
2	;	Kết thúc khai báo biến, một lệnh, một lời gọi hàm, hay khai báo nguyên mẫu hàm.	int x; void NhapMang(int a[], int &n);
3	//	Chú thích (ghi chú) cho một dòng. Chỉ có tác dụng đối với người đọc chương trình.	//Ham nay dung de nhap mang void NhapMang(int a[], int &n);
4	/* */	Tương tự như ký hiệu //, nhưng cho trường hợp nhiều dòng.	/* Dau tien nhap vao n. Sau do nhap cho tung phan tu */ void NhapMang(int a[], int &n);

I.2. Các kiểu dữ liệu cơ bản trong C

STT	Kiểu	GHI CHÚ	KÍCH THƯỚC	ĐỊNH DẠNG
Kiểu liên tục (số thực)				
1	float		4 bytes	%f
2	double		8 bytes	%lf
3	long double		10 bytes	%lf
Kiểu rời rạc (số nguyên)				
1	char	Ký tự	1 byte	%c
		Số nguyên	1 byte	%d
2	unsigned char	Số nguyên dương	1 byte	%d
3	int	Số nguyên	2 bytes	%d
4	unsigned int	Số nguyên dương	2 bytes	%u
5	long	Số nguyên	4 bytes	%ld
6	unsigned long	Số nguyên dương	4 bytes	%lu
7	char *	Chuỗi		%s

I.3. Bảng ký hiệu các phép toán

STT	PHÉP TOÁN	Ý NGHĨA	GHI CHÚ
PHÉP TOÁN SỐ HỌC			
1	+	Cộng	
2	-	Trừ	
3	*	Nhân	
4	/	Chia lấy phần nguyên	
5	%	Chia lấy phần dư	
PHÉP TOÁN QUAN HỆ			
1	>	Lớn hơn	
2	<	Nhỏ hơn	
3	>=	Lớn hơn hoặc bằng	
4	<=	Nhỏ hơn hoặc bằng	
5	==	Bằng nhau	
6	!=	Khác nhau	
PHÉP TOÁN LOGIC			
1	!	NOT	
2	&&	AND	
3		OR	
TOÁN TỬ TĂNG GIẢM			
1	++	Tăng 1	Nếu toán tử tăng giảm đặt trước thì tăng giảm trước rồi tính biểu thức hoặc ngược lại.
2	--	Giảm 1	
PHÉP TOÁN THAO TÁC TRÊN BIT			
1	&	AND	
2		OR	
3	^	XOR	
4	<<	Dịch trái	
5	>>	Dịch phải	
6	~	Lấy phần bù theo bit	

I.4. Các hàm cơ bản

STT	TÊN HÀM	THƯ VIỆN	DIỄN GIẢI	VÍ DỤ
1	printf	#include<stdio.h>	Xuất ra màn hình.	<pre>#include<stdio.h> #include<conio.h> #include<dos.h> void main() { int c = 1, n; clrscr(); printf("Nhap n:"); scanf("%d", &n); do{ textcolor(c); gotoxy(20, 10); cprintf("%d", n); c++; if (c>15) c = 1; delay(200); } while(!kbhit()); }</pre>
2	scanf	#include<stdio.h>	Lấy dữ liệu từ bàn phím.	
3	gotoxy	#include<conio.h>	Di chuyển dấu nháy đến tọa độ (x, y) trên màn hình văn bản.	
4	textcolor	#include<conio.h>	Đặt màu cho chữ (có giá trị từ 0 đến 15).	
5	cprintf	#include<stdio.h>	Xuất ra màn hình với màu chữ đã định liền trước đó.	
6	delay	#include<dos.h>	Dừng thực hiện lệnh tiếp sau một khoảng thời gian.	
7	kbhit	#include<conio.h>	Kiểm tra xem có nhấn phím.	

I.5. Cấu trúc rẽ nhánh

a. Cấu trúc if

```
if (biểu thức điều kiện)
{
    <khối lệnh> ;
}
```

Nếu biểu thức điều kiện cho kết quả khác không thì thực hiện khối lệnh.

Ví dụ:

```
#include <conio.h>
#include <stdio.h>
void main ()
{
    float number ;

    printf ( "Nhap mot so trong khoang tu 1 den 10 => " );
    scanf ( "%f", &number );
    if (number >5)
        printf ( "So ban nhap lon hon 5. \n" );
    printf ( "%f la so ban nhap. ", number );
}
```

b. Cấu trúc if ... else

```

if (biểu thức điều kiện)
{
    <khối lệnh 1>;
}
else
{
    <khối lệnh 2>;
}

```

Nếu biểu thức điều kiện cho kết quả khác không thì thực hiện khối lệnh 1, ngược lại thì cho thực hiện khối lệnh thứ 2. Biểu thức điều kiện phải đặt trong cặp dấu ngoặc tròn.

Ví dụ: Giải và biện luận phương trình: $ax+b=0$

```

#include <conio.h>
#include <stdio.h>
void main ()
{
    float a, b;
    printf ( "\n Nhập vào a:");
    scanf ( "%f", &a);
    printf ( " Nhập vào b:");
    scanf ( "%f", &b) ;
    if (a== 0)
        if (b== 0)
            printf ( " \n PTVSN");
        else
            printf ( " \n PTVN");
    else
        printf ( " \n Nghiệm x=%f", -b/a);
    getch ();
}

```

I.6. Cấu trúc lựa chọn switch

```

switch (biểu thức)
{
    case n1:
        các câu lệnh ;
        break ;
    case n2:
        các câu lệnh ;
        break ;
    .....
    case nk:
        <các câu lệnh> ;
        break ;
}

```

[**default:** các câu lệnh]

}

- n_i là các **hằng số nguyên hoặc ký tự**.
- Phụ thuộc vào giá trị của biểu thức viết sau **switch**, nếu:
 - Giá trị này = n_i thì thực hiện câu lệnh sau case n_i .
 - Khi giá trị biểu thức không thỏa tất cả các n_i thì thực hiện câu lệnh sau **default** nếu có, hoặc thoát khỏi câu lệnh **switch**.
 - Khi chương trình đã thực hiện xong câu lệnh của **case** n_i nào đó thì nó sẽ thực hiện luôn các lệnh thuộc **case** bên dưới nó mà không xét lại điều kiện (do các n_i được xem như các nhãn) → Vì vậy, để chương trình thoát khỏi lệnh **switch** sau khi thực hiện xong một trường hợp, ta dùng lệnh **break**.

Ví dụ: Tạo menu cấp 1 cho phép chọn menu bằng số nhập từ bàn phím.

```
#include<stdio.h>
#include<conio.h>
int ChonTD ()
{
    int chon ;

    printf ("Thuc Don") ;
    printf ("\n1. Lau thai!") ;
    printf ("\n2. Nuoc ngot!") ;
    printf ("\n3. Ca loc hap bau!") ;
    printf ("\n4. Chuot dong!") ;
    printf ("\n Xin moi ban chon mon an!") ;
    scanf ("%d",&chon) ;
    return chon ;
}

void TDchon(int chon)
{
    switch (chon)
    {
        case 1:
            printf ("\nBan chon lau thai!") ;
            break ;
        case 2:
            printf ("\nBan chon nuoc ngot!") ;
            break ;
        case 3:
            printf ("\nBan chon ca loc hap bau!") ;
            break ;
        case 4:
            printf ("\nBan chon chuot dong!") ;
```

```

        break ;
    default:
        printf ("\nBan chon khong dung!") ;
    }
}

void main()
{
    clrscr() ;
    int c ;
    c=ChonTD() ;
    TDchon(c) ;
    getch() ;
}

```

I.7. Cấu trúc lặp

a. *for*

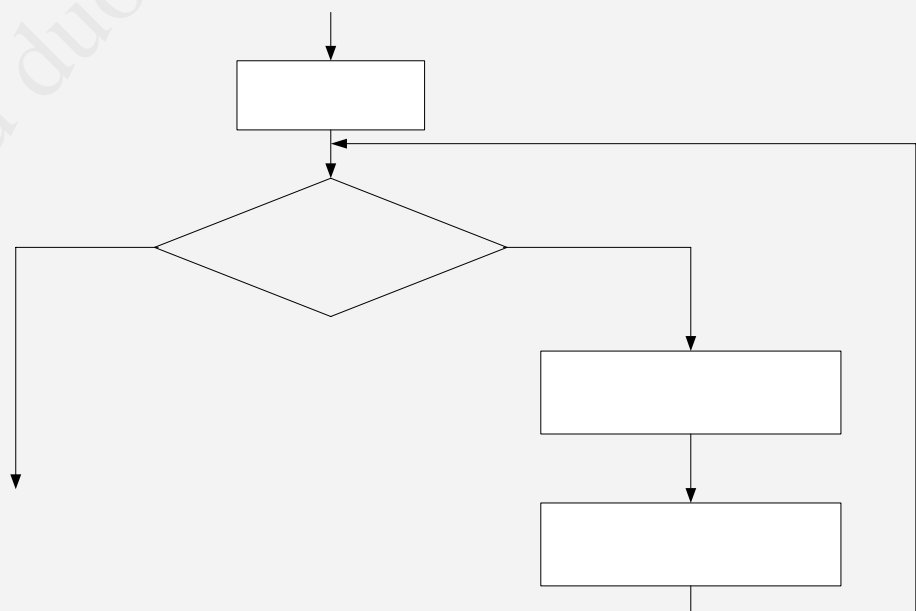
```

for (<biểu thức khởi gán>; <biểu thức điều kiện>; <biểu thức tăng/giảm>)
{
    <khối lệnh>;
}

```

Bất kỳ biểu thức nào trong 3 biểu thức nói trên đều có thể vắng nhưng phải giữ dấu chấm phẩy (;).

Hoạt động của cấu trúc điều khiển for:



Bước 1: Khởi gán cho biểu thức 1

Bước 2: Kiểm tra điều kiện của biểu thức 2.

- Nếu **biểu thức 2** $\neq 0$ thì cho thực hiện các lệnh của vòng lặp, thực hiện biểu thức 3. Quay trở lại bước 2.
- Ngược lại thoát khỏi lặp.

Ví dụ: In ra màn hình bảng mã ASCII từ ký tự số 33 đến 255.

```
#include<conio.h>
#include<stdio.h>
void main()
{
    for (int i=33;i<=255;i++)
        printf("Ma ASCII cua %c: %d\t", i, i) ;
    getch () ;
}
```

b. **while**

```
< Khởi gán>
while ( <biểu thức điều kiện>)
{
    lệnh/ khối lệnh;
    < tăng/giảm chỉ số lặp>;
}
```

🔗 **Lưu ý:** Cách hoạt động của **while** giống **for**

Ví dụ: Tính giá trị trung bình các chữ số của số nguyên n gồm k chữ số.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    long n, tong=0;
    int sochuso=0;
    float tb;

    printf ("Nhap vao gia tri n gom k chu so") ;
    scanf ("%ld",&n) ;
    while(n>0)
    {
        tong=tong+n%10 ;
        sochuso++ ;
        n=n/10 ;
    }

    tb=1.0*tong/sochuso ;
    printf ("Gia tri trung binh la: %f", tb) ;
}
```

```

    getch () ;
}

```

c. *do ... while*

```

do
{
    < khối lệnh> ;
} while (biểu thức điều kiện) ;

```

Thực hiện khối lệnh cho đến khi biểu thức có giá trị bằng 0.

Ví dụ: Nhập ký tự từ bàn phím hiển thị lên màn hình mã ASCII của ký tự đó, thực hiện đến khi nhấn phím ESC (Mã ASCII của phím ESC là 27).

```

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

void main()
{
    int ma ;
    do {
        ma=getch () ;
        if (ma !=27)
            printf ("Mã ASCII %c:%d\t", ma, ma);
    }while (ma!=27) ;

    getch () ;
}

```

⚡ Lặp **while** kiểm tra điều kiện trước khi thực hiện lặp, còn vòng lặp **do...while** thực hiện lệnh lặp rồi mới kiểm tra điều kiện. Do đó vòng lặp **do...while** thực hiện lệnh ít nhất một lần.

I.8. break và continue

a. *break*

Dùng để kết thúc vòng lặp trực tiếp chứa nó khi thỏa điều kiện nào đó.

Ví dụ: Cho phép người dùng nhập liên tục giá trị n cho đến khi nhập âm thì dừng.

```

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

void main()
{

```



```

while (1)
{
    printf("\nNhap n: ");
    scanf("%d", &n);
    if(n<0)
        break;
}

getch ();
}

```

b. *continue*

Dùng để bỏ qua một lần lặp khi thỏa điều kiện nào đó.

Ví dụ: In ra màn hình giá trị từ 10 đến 20 trừ đi số 13 và số 17.

```

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

void main()
{
    for(int i=10 ; i<=20; i++)
    {
        if(i==13||i==17)
            continue;
        printf("%d\t", i);
    }
    getch ();
}

```

II. BÀI TẬP

II.1. Phương pháp chạy tay từng bước để tìm kết quả chương trình

- ❖ Xác định chương trình có sử dụng những biến nào.
- ❖ **Giá trị ban đầu** của mỗi biến.
- ❖ Những **biến nào sẽ bị thay đổi** trong quá trình chạy chương trình thì lập thành bảng có dạng sau:

Bước (Hoặc lần thực hiện)	Biến 1	Biến 2	...	Biến n	Kết quả in ra màn hình
0	Giá trị 0	Giá trị 0	...	Giá trị 0	
1	Giá trị 1	Giá trị 1	...	Giá trị 1	
2	Giá trị 2	Giá trị 2	...	Giá trị 2	
...	
...	

🔗 *Lưu ý từng lệnh và biểu thức điều kiện trong đoạn chương trình*

cuu duong than cong . com

Ví dụ: Cho biết kết quả của đoạn chương trình sau:

```
void main()
{
    int i, a = 4;
    clrscr();
    for(i = 0 ; i<a; i++)
        printf("%d\n", i);
}
```

Chương trình gồm 2 biến *i* và *a*, chỉ có biến *i* có giá trị thay đổi trong quá trình chạy chương trình nên ta lập bảng sau:

a có giá trị là 4

Bước thực hiện	Giá trị của biến <i>i</i>	Kết quả in ra màn hình
0	0	0
1	1	0 1
2	2	0 1 2
3	3	0 1 2 3
4	4	

Tại bước 4, giá trị của *i* = 4 vi phạm điều kiện lặp (*i*<*a*) nên vòng lặp kết thúc. Do đó kết quả in ra màn hình:

0
1
2
3

II.2. Bài tập cơ bản

a. Cấu trúc *if* / *if..else* và *switch*

1. Cho biết kết quả của đoạn chương trình sau:

```
int a=9, b=6;
a++;
a=a+b--;
a=a+(--b);
```

```
if(a%2==0)
```

```
    printf("Gia tri cua a la chan");
```

```
printf("Tong cua a va b la: %d", a+b) ;
```

2. Cho biết kết quả của đoạn chương trình sau:

```
int a=7, b=8;
```

```
a++;
```

```
a=a+(b--);
```

```
--b;
```

```
a--;
```

```
a=(--a)+(--b);
```

```
if(a%2!=0)
```

```
    printf("\n a la so le");
```

```
else
```

```
    printf("\n a la so chan");
```

```
printf("\na = %d",a);
```

3. Cho biết kết quả của đoạn chương trình sau:

```
int x=5, y;
```

```
y=x++ + 5;
```

```
printf("x=%d, y=%d\n", x, y);
```

```
y*=6;
```

```
x=y%7;
```

```
printf("x=%d,y=%d,y/x=%d", x, y, y/x);
```

4. Nhập vào hai số nguyên a, b. In ra màn hình giá trị lớn nhất.
5. Cho ba số a, b, c đọc vào từ bàn phím. Hãy tìm giá trị lớn nhất của ba số trên và in ra kết quả.
6. Cho ba số a, b, c đọc vào từ bàn phím. Hãy in ra màn hình theo thứ tự tăng dần các số. (Chỉ được dùng thêm hai biến phụ).
7. Viết chương trình nhập vào một số nguyên n gồm ba chữ số. Xuất ra màn hình chữ số lớn nhất ở vị trí nào?
Ví dụ: $n=291$. Chữ số lớn nhất nằm ở hàng chục (9).
8. Viết chương trình nhập vào số nguyên n gồm ba chữ số. Xuất ra màn hình theo thứ tự tăng dần của các chữ số.

Ví dụ: $n=291$. Xuất ra 129.

9. Nhập vào ngày, tháng, năm. Kiểm tra xem ngày, tháng, năm đó có hợp lệ hay không? In kết quả ra màn hình.
10. Nhập vào giờ, phút, giây. Kiểm tra xem giờ, phút, giây đó có hợp lệ hay không? In kết quả ra màn hình.
11. Viết chương trình nhập vào ngày, tháng, năm hợp lệ. Cho biết năm này có phải là năm nhuận hay không? In kết quả ra màn hình.
12. Viết chương trình tính diện tích và chu vi các hình: tam giác, hình vuông, hình chữ nhật và hình tròn với những thông tin cần được nhập từ bàn phím.
13. Viết chương trình tính tiền cước TAXI. Biết rằng:
 - KM đầu tiên là 5000^d.
 - 200m tiếp theo là 1000^d.
 - Nếu lớn hơn 30km thì mỗi km thêm sẽ là 3000^d.
 Hãy nhập số km sau đó in ra số tiền phải trả.
14. Nhập vào 3 số nguyên dương a, b, c. Kiểm tra xem 3 số đó có lập thành tam giác không? Nếu có hãy cho biết tam giác đó thuộc loại nào? (Cân, vuông, đều, ...).
15. Viết chương trình nhập vào số nguyên dương n. Kiểm tra xem n có phải là số chính phương hay không? (số chính phương là số khi lấy căn bậc 2 có kết quả là nguyên).

b. Cấu trúc lặp

16. Cho biết kết quả của đoạn chương trình sau:

```
int a=18;
for(int i=1; i<=a; i++)
    if(a%i==0)
        printf("\t %d", i);
```

17. Cho biết kết quả của đoạn chương trình sau:

```
for(int i=0; i<5; i++)
{
    for(int j=0; j<=i; j++)
        printf("%d\t", j);
    printf("\n");
}
```

```
}
```

18. Cho biết kết quả của đoạn chương trình sau:

```
int i=10, s=0;
while(i>0)
{
    if(i%2==0)
        s+=i;
    else
        if(i>5)
            s+=2*i;
    i--;
}
printf("s = %d",s);
```

19. Cho biết kết quả của đoạn chương trình sau:

```
int a=18, i=1;
do{
    if(a%i==0)
        printf("\t %d",i);
    i++;
} while(i<=a);
```

20. Cho biết kết quả của đoạn chương trình sau:

```
int a=11, b=16, i=a;
while( i<b )
{
    if(i%2==0)
    {
        printf("\t %d", i);
        break;
    }
    i++;
}
```

21. Cho biết kết quả của đoạn chương trình sau:

```
int a=10, s=0, i=0;
```

```
while( i<a )
{
    i++;
    if(i%2==0) continue;
    else s=s+i;
}
printf("s=%d",s);
```

22. Cho biết kết quả của đoạn chương trình sau:

```
int i=1,s=0;
while(1)
{
    s=s+i++;
    if(i%2)
        i=i+2;
    else
        i=i+1;
    if(i>20)
        break;
}
printf("%d",s);
```

23. Viết chương trình in ra màn hình hình chữ nhật đặc kích thước $m \times n$ (m, n nhập từ bàn phím).

Ví dụ: Nhập $m=5, n=4$

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

24. Viết chương trình in ra màn hình hình chữ nhật rỗng kích thước $m \times n$ (m, n nhập từ bàn phím).

Ví dụ: Nhập $m=5, n=4$

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

25. Viết chương trình in ra màn hình tam giác vuông cân đặc có độ cao h (h nhập từ bàn phím).

Ví dụ: Nhập $h=4$

```

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

26. Viết chương trình in ra màn hình tam giác cân rỗng có độ cao h (h nhập từ bàn phím).

Ví dụ: Nhập $h=4$

```

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

27. Viết chương trình in ra màn hình tam giác cân đặc có độ cao h (h nhập từ bàn phím).

Ví dụ: Nhập $h=4$

```

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

28. Viết chương trình in ra màn hình tam giác cân rỗng có độ cao h (h nhập từ bàn phím).

Ví dụ: Nhập $h=4$

```

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

29. Viết chương trình nhập số nguyên dương n . Liệt kê n số nguyên tố đầu tiên.
30. Viết chương trình nhập vào hai số nguyên dương a và b . Tìm ước số chung lớn nhất và bội số chung nhỏ nhất của a và b .
31. Viết chương trình nhập vào một số nguyên n gồm tối đa 10 chữ số (4 bytes). In ra màn hình giá trị nhị phân của số trên. (Hướng dẫn: chia lấy dư cho 2 và xuất theo thứ tự ngược lại dùng hàm gotoxy, wherex, wherey).
32. Viết chương trình đếm số ước số của số nguyên dương N .

Ví dụ: $N=12$

số ước số của 12 là 6

33. Một số hoàn thiện là một số có tổng các ước số của nó (không kể nó) bằng chính nó. Hãy liệt kê các số hoàn thiện nhỏ hơn 5000.

Ví dụ: số 6 là số hoàn thiện vì tổng các ước số là $1+2+3=6$.

34. Nhập vào ngày, tháng, năm. Cho biết đó là ngày thứ mấy trong năm.
35. In ra dãy số Fibonacci
- $$f_1 = f_0 = 1 ;$$
- $$f_n = f_{n-1} + f_{n-2} ; \quad (n > 1)$$

II.3. Bài tập luyện tập và nâng cao

36. Cài đặt tất cả các lưu đồ đã vẽ ở chương 1.
37. Nhập vào ngày, tháng, năm. Kiểm tra xem ngày, tháng, năm đó có hợp lệ hay không, nếu hợp lệ cho biết ngày sau đó là bao nhiêu.
Ví dụ: *Nhập 31/12/2003*
Ngày sau đó 01/01/2004
38. Nhập vào ngày, tháng, năm. Kiểm tra xem ngày, tháng, năm đó có hợp lệ hay không, nếu hợp lệ cho biết ngày trước đó là bao nhiêu.
Ví dụ: *Nhập 01/01/2003*
Ngày trước đó 31/12/2002
39. (*) Nhập vào ngày, tháng, năm của năm 2003. Hãy kiểm tra xem dữ liệu có hợp lệ hay không? Nếu hợp lệ hãy cho biết đó là ngày thứ mấy trong tuần. (hai, ba, tư, ..., CN). (Hướng dẫn: lấy ngày 01 tháng 01 năm 2003 là ngày thứ tư làm mốc).
40. Nhập vào giờ, phút, giây. Kiểm tra xem giờ, phút, giây đó có hợp lệ hay không, nếu hợp lệ cho biết giờ sau đó 1 giây là bao nhiêu.
Ví dụ: *Nhập 01:59:59*
Giờ sau đó 1 giây 02:00:00
41. Nhập vào giờ, phút, giây. Kiểm tra xem giờ, phút, giây đó có hợp lệ hay không, nếu hợp lệ cho biết giờ trước đó 1 giây là bao nhiêu.
Ví dụ: *Nhập 02:00:00*
Giờ trước đó 1 giây 01:59:59
42. Viết chương trình in ra bảng cửu chương từ 2 đến 9.
43. (*) Vẽ hình cánh quạt sau:

Sử dụng các hàm `cprintf()`, `textcolor()`, `delay()`, `kbhit()`, ... thay đổi màu để tạo cảm giác cho cánh quạt xoay cho đến khi nhấn một phím bất kỳ.

- Giáo trình
- Bài Tập Kỹ Thuật Lập Trình**
- Trang **30**