

## HƯỚNG DẪN THỰC HÀNH

# CÂU LỆNH LẶP FOR, WHILE, DO...WHILE

## 1 Lệnh FOR

### 1.1 Cú pháp

```
for(<Khởi tạo>; <Biểu thức điều kiện>; <Bước nhảy>)  
    <Lệnh>;
```

Quy trình thực hiện vòng lặp for:

- **Bước 1:** Khi bắt đầu vòng lặp for, phần <Khởi tạo> được thực hiện, và thực hiện đúng 1 lần. Sau đó chuyển sang bước 2
- **Bước 2:** Kiểm tra <Biểu thức điều kiện>, nếu đúng thì thực hiện <Lệnh>, rồi sang bước 3; nếu sai kết thúc vòng lặp.
- **Bước 3:** Thực hiện <Bước nhảy>, rồi trở lại bước 2

Ví dụ 1: Hiển thị lên màn hình câu “xin chào” 10 lần

```
#include "stdio.h"  
void main()  
{  
    int i;  
    for(i = 0; i < 10; i++)  
    {  
        printf("xin chào\n");  
    }  
}
```

Ví dụ 2: Xuất các số chẵn trong khoảng từ 1 tới 10

```
#include "stdio.h"  
void main()  
{  
    int i;  
    for(i = 2; i <= 10; i+=2)  
    {  
        printf("%d\n", i);  
    }  
}
```

### 1.2 Lệnh continue

Trong <Lệnh>, nếu gặp lệnh continue thì quy trình thực hiện lặp tức chuyển sang bước 2, các câu lệnh bên dưới tại bước lặp hiện tại sẽ không được thực thi

Ví dụ: Xuất các số lẻ trong khoảng từ 1 tới 10

```
#include "stdio.h"
void main()
{
    int i;
    for(i=1; i<=10; i++)
    {
        if (i%2 == 0)
            continue;
        printf("%d\n", i);
    }
}
```

### 1.3 Lưu ý

Câu lệnh for là một câu lệnh đơn và có thể lồng nhau.

Ví dụ:

```
void main()
{
    int n=3,m=3,i,j;
    if (n < 10 && m < 20)
    {
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < m; j++)
            {
                printf("%d", i + j);
            }
            printf("\n");
        }
    }
}
```

Không được thêm dấu ; ngay sau lệnh for, vì nó tương đương câu lệnh rỗng

Ví dụ:

```
void main()
{
    int i;
    for (i = 0; i < 10; i++);
    {
        printf("%d\n", i);
    }
    // Tương đương với
    for (i = 0; i < 10; i++)
    {
    }
    printf("%d\n", i);
}
```

```
}

```

Thông thường, <Bước nhảy> phải làm ảnh hưởng đến <Biểu thức điều kiện>, làm cho <Biểu thức điều kiện> có thể đạt được giá trị sai, nếu không, vòng lặp sẽ lặp vô tận (không dừng lại được)

Ví dụ: (lặp vô tận)

```
void main()
{
    int i;
    for(i=0; i<10; i--)
        printf("Cau lenh lap\n");
}
```

## 2 Lệnh while

### 2.1 Cú pháp

```
while(<Biểu thức điều kiện>)
    <Lệnh>
```

Quy trình thực hiện vòng lặp while:

- **Bước 1:** Kiểm tra <Điều kiện lặp>, nếu đúng thì sang bước 2, nếu sai thì kết thúc vòng lặp
- **Bước 2:** Thực hiện <Câu lệnh>; sau đó quay lại bước 1

Ví dụ: Hiển thị lên màn hình câu “xin chào” 10 lần

```
#include "stdio.h"
void main()
{
    int i=1;
    while(i<=10)
    {
        printf("xin chao\n");
        i++;
    }
}
```

### 2.2 Lưu ý

- Câu lệnh while là một câu lệnh đơn và có thể lồng nhau. Ví dụ:

```
void main()
{
    int m=3;
    int n=3;
    if (n < 10 && m < 20)
    {
```

```

        while (n >= 1)
        {
            while (m >= 1)
            {
                printf("%d", m);
                m--;
            }
            n--;
        }
    }
}

```

- Trong <Lệnh>, phải có câu lệnh ảnh hưởng đến giá trị của <Biểu thức điều kiện>, làm cho <Biểu thức điều kiện> có thể đạt đến giá trị sai. Nếu không đạt đến điều kiện dừng (<Biểu thức điều kiện> sai) thì vòng lặp sẽ lặp vô tận. Ví dụ:

```

#include "stdio.h"
void main()
{
    int i=1;
    while(i<=10)
    {
        printf("Cau lenh lap\n");
    }
}

```

- Không được thêm ; ngay sau lệnh while. Ví dụ:

```

#include "stdio.h"
void main()
{
    int n = 0;
    while (n < 10);
    {
        printf("%d\n", n);
        n++;
    }
    // Tương đương
    while (n < 10)
    {
    }
    printf("%d\n", n);
    n++;
}

```

### 3 Lệnh do ... while

#### 3.1 Cú pháp

```
Do
    <Lệnh>
while(<Biểu thức điều kiện>);
```

Quy trình thực hiện vòng lặp do ... while:

- **Bước 1:** Thực hiện <Lệnh> sau đó sang bước 2
- **Bước 2:** Kiểm tra <Biểu thức điều kiện>, nếu đúng thì trở lại bước 1, nếu sai thì kết thúc vòng lặp

Ví dụ: Hiển thị lên màn hình câu “Cau lenh lap” 10 lần

```
#include "stdio.h"
void main()
{
    int i=1;
    do
    {
        printf("Cau lenh lap\n");
        i++;
    } while(i<=10);
}
```

#### 3.2 3.3.2 Lưu ý

Câu lệnh do... while là một câu lệnh đơn và có thể lồng nhau. Ví dụ:

```
#include "stdio.h"
void main()
{
    int a = 1, b;
    do
    {
        b = 1;
        do
        {
            printf(" %d ", a + b);
            b = b + 2;
        } while (b < 10);
        a++;
        printf("\n");
    } while (a < 10);
}
```

Câu lệnh do... while sẽ được thực hiện ít nhất 1 lần do điều kiện lặp được kiểm tra ở cuối. Ví dụ:

```
#include "stdio.h"
void main()
{
    int n;
    do
    {
        printf("Nhap n: ");
        scanf("%d", &n);
    } while (n < 1 || n > 100);
}
```

- Sau dấu ‘)’ ở dòng **while** có dấu ‘;’, nếu không có sẽ bị sai cú pháp
- Trong <Lệnh>, phải có câu lệnh ảnh hưởng đến giá trị của <Biểu thức điều kiện>, làm cho <Biểu thức điều kiện> có thể đạt đến giá trị sai. Nếu không đạt đến điều kiện dừng (<Biểu thức điều kiện> sai) thì vòng lặp sẽ lặp vô tận

#### 4 Lệnh break

Trong các lệnh lặp, nếu gặp lệnh break, vòng lặp lập tức kết thúc

Ví dụ: Xuất số chẵn nhỏ nhất trong đoạn [a, b].

```
// Viết bằng lệnh for
#include "stdio.h"
void main()
{
    int a, b, i;
    scanf("%d%d", &a, &b);
    for(i=a; i<=b; i++)
    {
        if(i%2 == 0)
        {
            printf("%d", i);
            break;
        }
    }
}
```

Ví dụ: Xuất số chẵn nhỏ nhất trong đoạn [a, b].

// Viết bằng lệnh while

```
#include "stdio.h"
void main()
```

```

{
    int a, b, i;
    scanf("%d%d", &a, &b);
    i = a;
    while(i<=b)
    {
        if(i%2 == 0)
        {
            printf("%d", i);
            break;
        }
        i++;
    }
}

```

## 5 Bài tập

1. Tìm số nguyên dương  $n$  nhỏ nhất sao cho  $1 + 2 + 3 + \dots + n > 1000$ .
2. Nhập số nguyên dương  $n$ , tính:

$$S = \sqrt[n]{n + \sqrt[n-1]{n-1 + \sqrt[n-2]{n-2 + \dots + \sqrt[3]{3 + \sqrt{2}}}}}$$

3. Nhập số nguyên dương  $n$ , tính:

$$S = \sqrt[n+1]{n! + \sqrt[n]{(n-1)! + \sqrt[n-1]{(n-2)! + \dots + \sqrt[3]{2! + \sqrt{1!}}}}}$$

4. Nhập số nguyên dương  $n$ , tính:

$$S = \sqrt{2 * n + \sqrt{2 * (n-1) + \dots + \sqrt{4 + \sqrt{2}}}}$$

5. Nhập số nguyên dương  $n$ , tính:

$$S = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{n!}$$

6. Nhập số thực  $x$ , tính:

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

7. Nhập số thực  $-1 < x < 1$ , tính:

$$\frac{1}{1-x} = 1 + x + x^2 + \dots = \sum_{n=0}^{\infty} x^n$$

8. Nhập số thực  $x$ , tính:

$$\cos(x) = 1 - \frac{1}{2}x^2 + \frac{1}{24}x^4 - \frac{1}{720}x^6 + \dots = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

9. Nhập số thực  $x$ , tính:

$$\operatorname{arccot}(x) = \frac{\pi}{2} - x + \frac{1}{3}x^3 - \frac{1}{5}x^5 + \frac{1}{7}x^7 - \frac{1}{9}x^9 + \dots = \frac{\pi}{2} \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} x^{2n+1}$$

10. Nhập số thực  $-1 < x < 1$ , tính:

$$\arctan(x) = x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \frac{1}{7}x^7 + \dots = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{2n-1} x^{2n-1}$$

11. Nhập số thực  $x$ , tính:

$$e^x = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \dots = \sum_{n=0}^{\infty} \frac{1}{n!} x^n$$

12. Nhập số thực  $x$ , tính:

$$\ln(1+x) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 - \frac{1}{4}x^4 + \dots = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} x^n$$

13. Nhập số thực  $x$ , tính:

$$\ln\left(\frac{1+x}{1-x}\right) = 2x + \frac{2}{3}x^3 + \frac{2}{5}x^5 - \frac{2}{7}x^7 + \dots = \sum_{n=1}^{\infty} \frac{2}{2n-1} x^{2n-1}$$

14. Nhập số thực  $x$ , tính:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

15. Nhập số nguyên dương  $n$ .

- Kiểm tra có phải là số nguyên tố.
- In ra tất cả các số nguyên tố nhỏ hơn hoặc bằng  $n$ .
- Tìm các thừa số nguyên tố của  $n$ .
- In các chữ số trong  $n$  từ phải qua trái.
- In các chữ số trong  $n$  từ trái qua phải.
- Tính tổng các chữ số cho đến khi tổng này nhỏ hơn 10 (Ví dụ :  $n = 9999 \rightarrow 36 \rightarrow 27 \rightarrow 18 \rightarrow 9$ )
- Chứa toàn số lẻ hay toàn số chẵn?
- Số có đối xứng hay không
- Các chữ số xếp tăng dần hay giảm dần (từ trái qua phải) không?
- Chữ số lớn nhất và nhỏ nhất xuất hiện trong  $n$ .
- In ra tất cả các cặp số nguyên dương  $a$  và  $b$ , sao cho  $a^2 + b^2 < n$



16. Nhập 2 số nguyên dương  $a, b$  khác không.
  - a. Tìm USCLN( $a, b$ )
  - b. Tìm hai số nguyên  $x$  và  $y$  sao cho:  $\text{USCLN}(a, b) = a * x + b * y$ .
  - c. Tìm BSCNN( $a, b$ )
17. Nhập 3 số nguyên dương  $a, b$  và  $c$ . Tìm  $\text{USCLN}(a, b, c)$ .
18. Xác định phần tử thứ  $k$  của dãy Fibonacci:  $f_0 = 0, f_1 = 1, f_k = f_{k-1} + f_{k-2}$ , với  $k \geq 2$ .
19. In ra bình phương của  $n$  số nguyên dương đầu tiên, nhưng chỉ dùng phép '+' và/hoặc phép '-'.
20. Nhập hai số nguyên  $a \geq 0$  và  $d > 0$ . Gọi  $q$  là kết quả của phép chia nguyên  $a$  cho  $d$  và  $r$  là số dư. Hãy xác định giá trị của  $q$  và  $r$ , nhưng không sử dụng phép toán '/' và '%'.
  - a. Tính  $a * b$  (không dùng phép \*)
  - b. Tính  $a \% b$  (không dùng phép % và /)
22. Tính  $a + b$ . Yêu cầu: Chỉ sử dụng phép gán có dạng:
  - <biến> = <số>;
  - <biến> = <biến> + 1;
  - <biến> = <biến>;
23. Nhập vào dãy số thực dương. Giả sử rằng quá trình nhập số sẽ dừng khi nhập số 0.
  - a. Tìm và in ra số lớn nhất, số nhỏ nhất trong số các số thực đã nhập.
  - b. Tính tổng các số đã nhập và in ra.
24. Nhập vào một số nguyên dương  $n$ , tính giá trị của vế phải và vế trái của mỗi công thức sau đây, in kết quả ra màn hình. Kiểm tra và in ra kết quả cho biết hai vế có bằng nhau không
  - a.  $1^2 + 2^2 + 3^2 + \dots + n^2 = n(n+1)(2n+1)/6$
  - b.  $1 \times 2 \times 3 + 2 \times 3 \times 4 + 3 \times 4 \times 5 + \dots + n \times (n+1) \times (n+2) = \frac{n(n+1)(n+2)(n+3)}{4}$
  - c.  $1 \times 1! + 2 \times 2! + 3 \times 3! + \dots + n \times n! = (n+1)! - 1$
  - d.  $\frac{1}{2!} + \frac{2}{3!} + \frac{3}{4!} + \dots + \frac{n}{(n-1)!} = 1 - \frac{1}{(n+1)!}$