



Vòng lặp

Phiên 6

Chỉ dành cho Trung tâm Aitech sử dụng



Mục tiêu

- Hiểu vòng lặp 'for' trong 'C'
- Làm việc với toán tử dấu phẩy
- Hiểu vòng lặp lồng nhau
- Hiểu vòng lặp 'while' và vòng lặp 'do-while'
- Làm việc với các câu lệnh break và continue
- Hiểu hàm exit()



Vòng lặp là gì?

Đoạn mã trong chương trình được thực thi nhiều lần cho đến khi một điều kiện cụ thể được đáp ứng



3 loại cấu trúc vòng lặp

Các vòng lặp

Các trong khi vòng lặp

Các làm...trong khi vòng lặp



Các vòng lặp-1

Cú pháp

```
for (initialize counter; conditional test; re-evaluation parameter)
{
    statement
}
```

- Bộ đếm khởi tạo là một câu lệnh gán đặt biến điều khiển vòng lặp trước khi vào vòng lặp
- Kiểm tra có điều kiện là một biểu thức quan hệ, xác định khi nào vòng lặp sẽ thoát
- Tham số đánh giá xác định cách biến điều khiển vòng lặp thay đổi mỗi khi vòng lặp được thực thi



Cácvòng lặp-2

- Ba phần củavòng lặp phải được phân tách bằng dấu chấm phẩy(;)
- Câu lệnh tạo thành phần thân của vòng lặp có thể là một câu lệnh đơn hoặc một câu lệnh ghép
- Cácvòng lặp tiếp tục thực hiện miễn là kiểm tra điều kiện đánh giá là đúng. Khi điều kiện trở thành sai, chương trình tiếp tục trên câu lệnh theo sauvòng lặp



Các vòng lặp-3

**/*Chương trình này trình bày vòng lặp for trong chương trình C */
bao gồm <stdio.h>**

chủ yếu()

{

**số nguyên;
printf("\tĐây là \n");**

**đối với (số lượng = 1; số lượng <=6; số lượng++)
printf("\n\t\tđẹp");**

printf("\n\t\tthế giới. \n");

}

Ví dụ



Toán tử dấu phẩy

Phạm vi của vòng lặp có thể được mở rộng bằng cách bao gồm nhiều hơn một biểu thức khởi tạo hoặc tăng trong đặc tả vòng lặp for

Định dạng là: **biểu thức1, biểu thức2;**

bao gồm <stdio.h>

chủ yếu()

{

int i, j, tối đa;

printf("Vui lòng nhập giá trị lớn nhất \n"); printf("có thể in bằng. "); scanf("%d", &max);

đối với (i = 0, j = tối đa; i <= tối đa; i++, j--)

printf("\n%d + %d = %d", i, j, i + j);

}

Ví dụ



Lồng nhau và Vòng lặp-1

Các vòng lặp sẽ được gọi là một lồng nhau khi nó là được viết như sau

```
đối với (i = 1; i < max1; i++)  
{
```

```
    .  
    .  
    đối với (j = 0; j <= max2; j++) {
```

```
        .  
        .  
    }  
    .  
    .  
}
```



Lồng nhau vì Vòng lặp-2

bao gồm <stdio.h>

chủ yếu()

{

số nguyên i, j, k;

tôi = 0;

printf("Nhập số hàng :"); scanf("%d",
&i);

inf("\n");

đối với (j = 0; j < i; j++) {

inf("\n");

for (k = 0; k <= j; k++) /*vòng lặp for bên trong*/

printf("*");

;

}

Ví dụ



Các trường hợp Vòng lặp-1

Cú pháp

trong khi (điều kiện là đúng)
tuyên bố;

Vòng lặp while lặp lại các câu lệnh trong khi một số
điều kiện được chỉ định là Đúng



Các trong khi Vòng lặp-2

/* Một chương trình đơn giản sử dụng vòng lặp while */

bao gồm <stdio.h>

chủ yếu()

{

int đếm = 1;

trong khi(đếm <= 10) {

printf("\n Đây là lần lặp %d\n",count); count++;

}

printf("\n Vòng lặp đã hoàn tất. \n");

}

Ví dụ



làm...trong khi Vòng lặp-1

Cú pháp

```
LÀM{  
    tuyên bố;  
} while (điều kiện);
```

- Trong **làm trong khi** vòng lặp phần thân của mã được thực hiện một lần trước khi thử nghiệm được thực hiện
- Khi điều kiện trở thành Sai trong một **làm trong khi** vòng lặp sẽ kết thúc và quyền điều khiển sẽ chuyển đến câu lệnh xuất hiện ngay sau **trong khi** tuyên bố

làm...trong khi Vòng lặp-2

```
# bao gồm <stdio.h>
```

```
    chủ yếu ()
```

```
{
```

```
    int số1, số2;
```

```
        số2 = 0;
```

```
    LÀM
```

```
{
```

```
        printf( "\nNhập một số : ");
```

```
        scanf("%d",&num1);
```

```
        printf( "Số là %d",num1); num2++;
```

```
    } trong khi (num1 != 0);
```

```
    printf( "\nTổng số đã nhập là %d",--num2);
```

```
/*num2 được giảm trước khi in vì số lượng cho số nguyên cuối cùng (0)
không được xem xét */
```

```
}
```

Ví dụ



Câu lệnh nhảy-1

return sự biểu lộ

- Câu lệnh return được sử dụng để trả về từ một hàm
- Nó khiến cho việc thực thi trở về điểm mà lệnh gọi hàm được thực hiện
- Câu lệnh return có thể có giá trị mà nó trả về cho chương trình



Câu lệnh nhảy-2

goto nhấ

- Câu lệnh goto chuyển quyền điều khiển sang bất kỳ câu lệnh nào khác trong cùng một hàm trong chương trình C
- Trên thực tế, nó vi phạm các quy tắc của một ngôn ngữ lập trình có cấu trúc chặt chẽ
- Chúng làm giảm độ tin cậy của chương trình và khiến chương trình khó bảo trì



Câu lệnh nhảy-3

break tuyên bố

- Câu lệnh break được sử dụng để kết thúc một trường hợp trong câu lệnh switch
- Nó cũng có thể được sử dụng để chấm dứt đột ngột một vòng lặp
- Khi gặp câu lệnh break trong vòng lặp, vòng lặp sẽ bị chấm dứt ngay lập tức và quyền điều khiển được chuyển đến câu lệnh theo sau vòng lặp



câu lệnh ngắt

Ví dụ

bao gồm <stdio.h>

chủ yếu ()

```
{  
    int đếm1, đếm2;  
    đối với (số đếm1 = 1, số đếm2 = 0; số đếm1 <=100; số đếm1++) {  
  
        printf("Nhập %d count2 : ",count1);  
        scanf("%d", &count2);  
        nếu(j==100) phá vỡ;  
    }  
}
```



Câu lệnh nhảy-4

continue tuyên bố

- Câu lệnh continue khiến vòng lặp tiếp theo của vòng lặp bao quanh bắt đầu
- Khi gặp câu lệnh này, các câu lệnh còn lại trong thân vòng lặp sẽ bị bỏ qua và quyền điều khiển được chuyển sang bước khởi tạo lại



tiếp tục tuyên bố

bao gồm <stdio.h>

chủ yếu ()

{

int số;

for(num = 1; num <= 100; num++) {

nếu(số % 9 == 0)

Tiếp tục;

printf("%d\t", số);

}

}

Ví dụ



Câu lệnh nhảy-5

`exit()` chức năng

- `exit()` được sử dụng để thoát khỏi chương trình
- Việc sử dụng chức năng này sẽ khiến chương trình bị chấm dứt ngay lập tức và quyền kiểm soát nằm trong tay hệ điều hành