Translated from English to Vietnamese - www.onlinedoctranslator.com

Vòng lặp

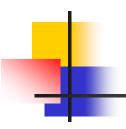
Phiên 6

Chi danh tho



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

Lập trình cơ bản với C/Phần 6/3 trong số 21



3 loại cấu trúc vòng lặp

Cácvìvòng lặp

Cáctrong khivòng lặp

Caclàm...trong khivòng lặp



Cácvìvòng lặp-100

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 tà một biểu thức quan hệ, xác định khi nào vòng lặp sẽ thoát
- Tham số đánh giả xác định cách biến điều khiển vòng lặp thay đổi mỗi khi vòng lặp được thực thi



Cácvìvòng lặp-20

- Ba phần của**vì**vòng lặp phải được phần tách bằng dấu chấm phẩy(;)
- Câu lệnh tạo thành phần thần của vòng lặp có thể là một câu lệnh đơn hoặc một câu lệnh ghép
- Cácvìvòng 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 sauvìvòng lặp



Cácvìvòng lặp-3

```
/*Chương trình này trình bày vòng lặp for trong chương trình C */
   # bao gồm <stdio.h>
                                                        Ví du
   chủ yếu()
   {
                số nguyên;
                printf("\tĐây là \n");
                đối với (số lượng = 1; số lượng <=6; số lượng++)
                         printf("\n\t\t đẹp");
                   intf("\n\t\t thế giới. \n");
```



Toán tử dấu phẩy

Phạm vi của**vì**vòng lặp có thể được mở rộng bàng 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 bảng."); scanf("%d", &max);

đối với (i = 0, j = tối đa; i <=tối đa; i++, j--)
printf("\n%d + %d = %d",i, j, i + j);
```

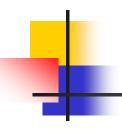


Lồng nhauvì Vòng lặp-1

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

Lồng nhauvì 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("\\\\");
         for (k = 0; k \le j; k++) /*vong lặp for bên trong*/
         printf("*");
```



Cáctrong khiVòng lặp 1

Cú pháp

trong khi (điều kiện là đúng) tuyên bố:

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



Cáctrong khiVòng 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++;
        printi("\n Vòng lặp đã hoàn tất. \n");
```



làm...trong khiVòng lặp-1

Cú pháp

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

- Tronglàm trong khivòng lặp phần thân của mã được thực hiện một lần trước khi thử nghiệm được thực hiện
- Khi điều kiện trở thành Sai trong mộtlàm trong khivòng lặp sẽ kết thúc và quyền điều khiển sẽ chuyển đến câu lệnh xuất hiện ngay sautrong khituyên bố

làm...trong khiVòng lặp 20

```
# bao gồm <stdio.h>
    chủ yếu ()
   int số1, số2;
                  s\tilde{o}2 = 0;
   LÀM
          printf( "\nNhập một số : ");
          scanf("%d",&num);
          printf( "Số là %d",num1); num2++;
   } trong khi (num() != 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 */
                                                        Lập trình cơ bản với C/Phần 6/14 trong số 21
```



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



goto nhãn

- 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



break tuyên bố

- Câu lệnh break được sử dụng để bết 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

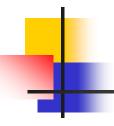


câu lệnh ngắt



continue tuven bó

- 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 bở 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);
```



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