Chào mừng các bạn trở lại với Series Lập Trình Căn Bản C. Hôm nay chúng ta sẽ tiếp tục bài học về vòng lặp trong ngôn ngữ C.

Đầu tiên, mình sẽ giới thiệu về 2 câu lệnh thần thánh không thể bỏ qua khi học vòng lặp.

1. Break – câu lệnh thoát vòng lặp

Khi chương trình chạy và gặp câu lệnh **break**, nó sẽ tự động thoát ra khỏi vòng lặp đang có.

Ví dụ 1: In các số từ 1 tới 30

#include "stdio.h"

int main(){

int i = 1; *// khởi tạo i bắt đầu từ 1*

while(1){ *// while(1) là chạy vòng lặp vô hạn*

if(i > 30) *// nếu i > 30*

break; *//thoát khỏi vòng lặp*

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

i++;

}

return 0;

1. Continue – Bỏ qua giá trị hiện tại

Khi chương trình đang chạy và gặp câu lệnh **continue**, nó sẽ bỏ qua giá trị hiện tại và đi đến giá trị tiếp theo.

Ví dụ: In các số chẵn từ 1 tới 30

#include "stdio.h"

int main(){

int i = 1;

while(i <= 30){

i++;

if(i % 2 == 1) *// nếu giá trị đang kiểm tra là số lẻ*

continue; *// bỏ qua giá trị hiện tại*

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

}

return 0;

}

Chỉ với 2 câu lệnh này và 3 vòng lặp trong ngôn ngữ C, các bạn đã có thể làm ra những thứ rất tuyệt vời.

**TÓM TẮT VỀ VÒNG LẶP**

Các bạn đã được học 3 câu lệnh lặp trong ngôn ngữ C. Nhưng việc có đến 3 câu lệnh hỗ trợ cho một việc ( lặp ) thì sẽ khiến các bạn khó xử và không biết sử dụng câu lệnh nào lúc nào. Nên mình viết bài viết này tổng hợp lại về 3 vòng lặp và có sự so sánh giữa chúng để các bạn có thể dễ hình dung về nó, sử dụng thuận tiện hơn.

**Cú pháp**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên vòng lặp | cú pháp | | ghi chú |
| while | while(điều kiện)  {  *//lệnh hoặc cụm lệnh;*  } | | Điều kiện còn đúng --> còn thực hiện |
| Do...while | Do  {  *//lệnh hoặc cụm lệnh;*  }while(điều kiện); | | Điều kiện còn đúng --> còn thực hiện |
| For | for ( giá - trị - bắt - đầu; điều - kiện - dừng; bước - nhảy)  {  *//lệnh hoặc cụm lệnh;*  } | | Điều kiện dừng ( ở giữa) còn đúng thì còn chạy. |
| BREAK | | Thoát khỏi vòng lặp | |
| CONTINUE | | Bỏ qua giá trị hiện tại | |

Chú ý:

+ Vòng while và for có thể kết thúc mà không lặp lại lần nào ( ít nhất 0 lần) . Vòng do...while thực hiện ít nhất 1 lần.

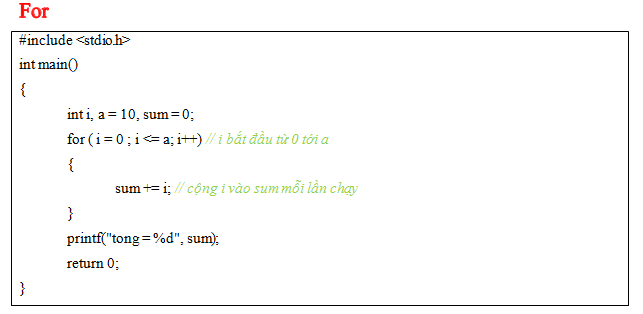
+ Sau vòng lặp do...while cần có dấu chấm phẩy. Với while và for thì không cần

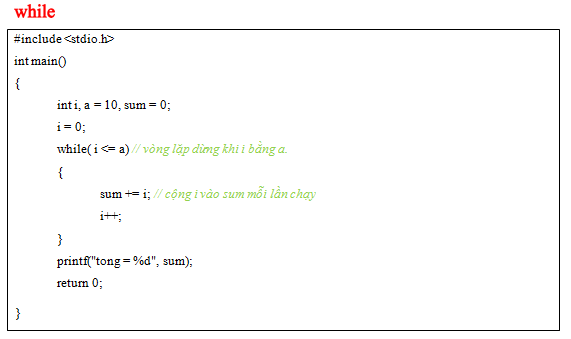
+ Vòng for: Mỗi lần chạy nó sẽ kiểm tra điều - kiện - dừng, nếu đúng thì chạy, sai thì không chạy. Và chạy theo cụm.

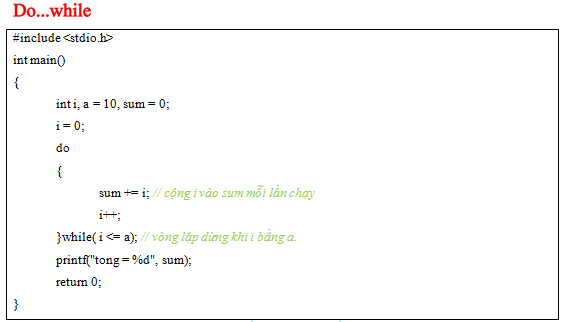
* + Giá - trị - bắt - đầu --> Điều - kiện - dừng --> đúng thực hiện lệnh, sai thì thoát ra.
  + Sau khi thực hiện cái ở trên xong thì...
  + Bước - nhảy --> Điều - kiện - dừng --> đúng thực hiện lệnh, sai thì thoát ra.

+ Các vòng lặp khi gặp lệnh break; thì sẽ thoát khỏi vòng lặp. Lệnh continue thì sẽ bỏ qua giá trị tại đó.

VD: Tỉnh tổng theo 3 cách ( 3 vòng lặp)







## Bài 1. Nhập vào một số n, hãy tính tổng từ 3 đến n .

Làm bài này bằng 3 cách ( tương ứng 3 loại vòng lặp).

## Bài 2. Tính 2 mũ n, không sử dụng hàm pow.

Gợi ý: Sử dụng vòng lặp for. Giá trị i bắt đầu từ 1 tới n

mỗi lần vậy ta cho p nhân 2. p=p\*2 hoặc p\*=2; Chú ý: ban đầu phải khai báo p=1.

Vì nếu ta cho p=0 ==> kq sẽ là 0, vì 0\* bất cứ số nào = 0.

## Bài 3. Tính giá trị của công thức sau:

a, S(n) = 1! + 2! + 3! +...+ n!

Gợi ý:

+ Cho một giá trị gt = gt\*i để tính giai thừa

+ Một giá trị S=S+gt để tính tổng. Khai báo ban đầu S=0 và gt = 1; b, S(n)= 1/2 + 1/3 + 1/4 +....+1/n.

Bài này nhớ ép kiểu float.

## Bài 4: Sử dụng vòng lặp để kiểm tra xem số n có phải là:

a, số nguyên tố hay không.

b, số chính phương hay không. c, số hoàn hảo hay không.

Giải thích:

* + Số nguyên tố là số có các đặc tính:
    - Chỉ chia hết cho 1 và chính nó.
    - Toàn bộ đều là số lẽ, trừ số 2. Và số 2 cũng là số nguyên tố nhỏ nhất.
    - Bất cứ số nào <2 đều không phải là SNT. Các bạn có thể tra bảng SNT từ 1 -1000 trong Google.
  + Số chính phương là số có đặc tính: Căn bậc 2 của nó là một số nguyên VD: Số 9 có căn bậc 2 là 3. ==> số chính phương.

số 8 có căn bậc 2 là 2.115642 ( phẩy tào lao đấy :v ) ==> không phải.

* + Số hoàn hào là số mà có ước số bé hơn nó cộng lại bằng chính nó VD: 6 có ước bé hơn nó là 1 2 3. Và tổng 1+2+3=6 ==> số hoàn hảo

8 có ước bé hơn nó là 1 + 2 + 4. Và tổng 1+2+4=7 khác 8 ==> không phải.

**Bài 5:**

**a, Nhập vào 2 số a, b. Tìm các ước chung của 2 số.**

Gợi ý: Cho chạy vòng lặp. Tìm ước của a và ước của b. Sau đó so sánh nếu bằng nhau ==> in ra.

**b, Tìm Ước chung lớn nhất (UCLN) và Bội chung nhỏ nhất (BCNN) của 2 số.**