**Break Và Continue**

**1.Câu Lệnh Break**

Break được sử dụng khi bạn muốn dừng vòng lặp (for, while, do-while) ngay lập tức, thông thường câu lệnh break thường đi kèm với 1 điều kiện kích hoạt, ví dụ như sau khi tìm được đáp án trong vòng lặp.

Khi câu lệnh break được thực thi thì các câu lệnh bên dưới break trong vòng lặp sẽ không được thực hiện nốt, vòng lặp sẽ kết thúc ngay tại ví trí của câu lệnh break

Với vòng lặp lồng nhau thì câu lệnh break có ý nghĩa với vòng lặp gần nhất chứa nó.

Xem xét ví dụ sau :

#include <stdio.h>

int main(){

int n = 5;

for(int i = 1; i <= 5; i++){

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

if(i == 3){

break;

}

printf("C programming !\n");

}

return 0;

}

Output :

1 BCN

C programming !

2 BCN

C programming !

3 BCN

Giải thích :

1. Vòng lặp 1 : i = 1, in ra 1 BCN, break chưa thực hiện nên tiếp tục in ra C programming !
2. Vòng lặp 2 : i = 2, in ra 2 BCN, break chưa thực hiện nên tiếp tục in ra C programming !
3. Vòng lặp 3 : i = 3, in ra 3 BCN, do if đúng nên câu lệnh break được thực thi, vòng lặp kết thúc ngay mà không thực hiện câu lệnh in ra C programming !

Ví dụ 1 : Tìm Ước Chung Lớn Nhất của 2 số a và b

Ý tưởng : Duyệt từ số nhỏ hơn trong 2 số a và b về 1, nếu gặp số đầu tiên cả a và b chia hết thì in ra và dừng vòng lặp

#include <stdio.h>

int main(){

int a = 18, b = 12;

int min = a < b ? a : b;

for(int i = min; i >= 1; i--){

if((a % i == 0) && (b % i == 0)){

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

break;

}

}

return 0;

}

Output :

6

Ví dụ 2 : In ra chữ số chẵn đầu tiên tính từ bên phải của số tự nhiên n

Ý tưởng : Tách từng chữ số của n bằng vòng lặp while, khi gặp chữ số chẵn thì dừng vòng lặp

#include <stdio.h>

int main(){

int n = 12689791;

while(n != 0){

if(n % 2 == 0){ // n % 10 % 2 cũng được

printf("%d\n", n % 10);

break;

}

n /= 10;

}

return 0;

}

Output :

8

Ví dụ 3 : Nhập 1 số nguyên từ bàn phím nếu nhập số âm thì yêu cầu nhập lại, nhập số không âm thì cho dừng

Ý tưởng : Dùng một vòng lặp lặp vĩnh viễn và kích hoạt câu lệnh break khi người dùng nhập số không âm

#include <stdio.h>

int main(){

int n;

while(1){

printf("Nhap n : ");

scanf("%d", &n);

if(n >= 0){

break;

}

else{

printf("Nhap lai\n");

}

}

return 0;

}

Ví dụ 4 : Câu lệnh break với vòng for lồng nhau

#include <stdio.h>

int main(){

for(int i = 1; i <= 2; i++){

for(int j = 1; j <= 5; j++){

printf("BCN\n");

if(j == 2){

break; // có tác dụng dừng vòng for j

}

printf("C programming !\n");

}

}

return 0;

}

Output :

BCN

C programming !

BCN

BCN

C programming !

BCN

**2.Câu Lệnh Continue**

Câu lệnh continue khi được thực hiện nó sẽ bỏ qua các câu lệnh bên dưới nó trong vòng lặp và quay trở lại vòng lặp mới ngay.

Với vòng lặp lồng nhau thì câu lệnh continue có ý nghĩa với vòng lặp gần nhất chứa nó.

Xem xét ví dụ sau :

#include <stdio.h>

int main(){

for(int i = 1; i <= 4; i++){

printf("BCN\n");

if(i % 2 == 0){

continue;

}

printf("C\n");

}

return 0;

}

Output :

BCN

C

BCN

BCN

C

BCN

Giải thích :

1. Vòng lặp i = 1, in ra BCN, kiểm tra if không đúng nên tiếp tục thực hiện câu lệnh in ra C
2. Vòng lặp i = 2, in ra BCN, kiểm tra if đúng nên bỏ qua câu lệnh in ra C bên dưới mà quay trở lại luôn cập nhật i
3. Vòng lặp i = 3, in ra BCN, kiểm tra if không đúng nên tiếp tục thực hiện câu lệnh in ra C
4. Vòng lặp i = 4, in ra BCN, kiểm tra if đúng nên bỏ qua câu lệnh in ra C bên dưới mà quay trở lại luôn cập nhật i

Ví dụ 1 : Nhập vào các số tự nguyên, kết thúc nhập bằng cách nhập số 0 và chỉ tính tổng các số nguyên dương được nhập

#include <stdio.h>

int main(){

int tong = 0, n;

while(1){

printf("Nhap n : ");

scanf("%d", &n);

if(n < 0){

continue; // bỏ qua các lệnh bên dưới

}

else if(n == 0){

break;

}

tong += n;

}

printf("%d\n", tong);

return 0;

}