

TRƯỜNG ĐÀO TẠO LẬP TRÌNH VIÊN VÀ QUẢN TRỊ MẠNG QUỐC TẾ BACHKHOA-APTECH

---

# BÀI 6: VÒNG LẶP

# MỤC TIÊU BÀI HỌC

Tìm hiểu về vòng lặp for trong C

Làm việc với toán tử dấu phẩy (,)

Tìm hiểu về các vòng lặp lồng nhau

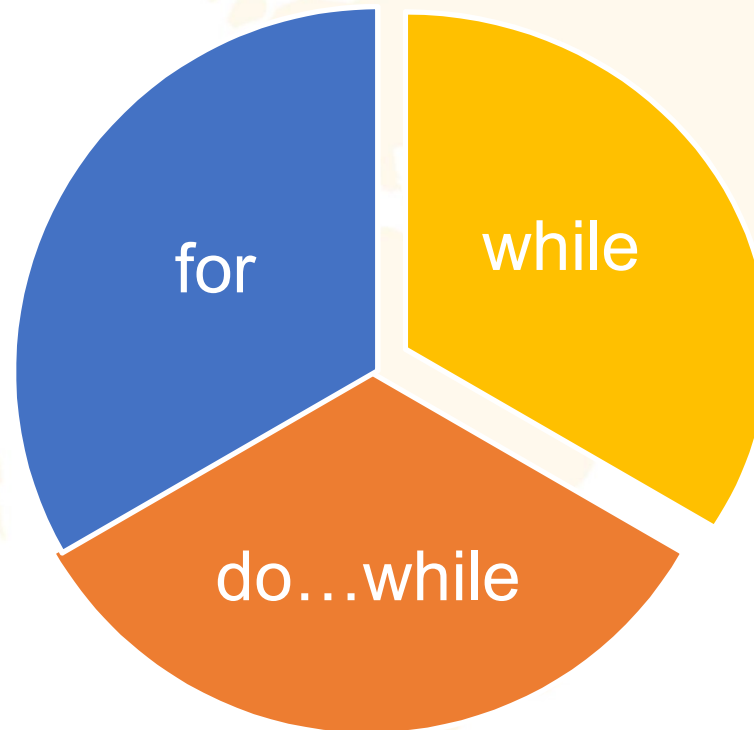
Tìm hiểu về vòng lặp while và do-while

Làm việc với các lệnh break và continue

Tìm hiểu về hàm exit()

# TÌM HIỂU VỀ VÒNG LẶP

- ✓ Vòng lặp là Một đoạn mã lệnh trong chương trình thực hiện lặp đi lặp lại cho đến khi một điều kiện xác định được thỏa mãn
- ✓ Các kiểu cấu trúc vòng lặp



# VÒNG LẶP – for 1-3

## ✓ Cú pháp

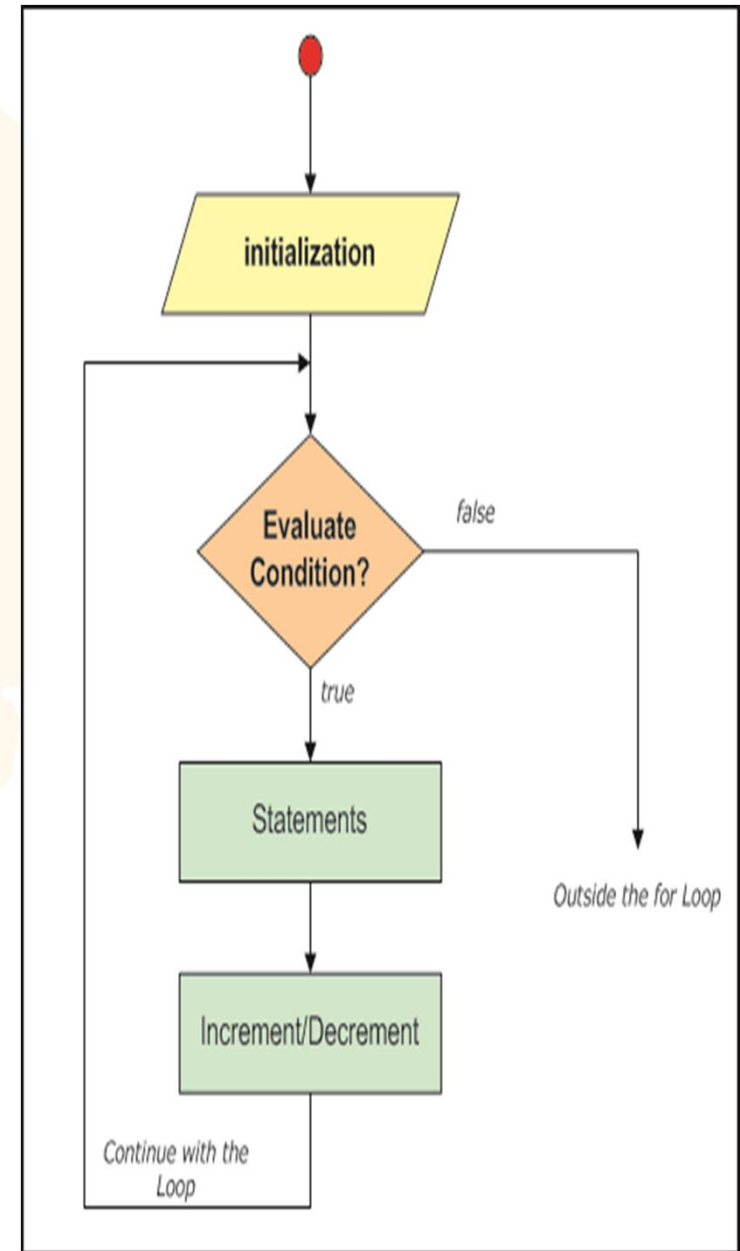
```
for (initialize counter; conditions; increment / decrement) {  
    //One or more Statements;  
}
```

## Trong đó

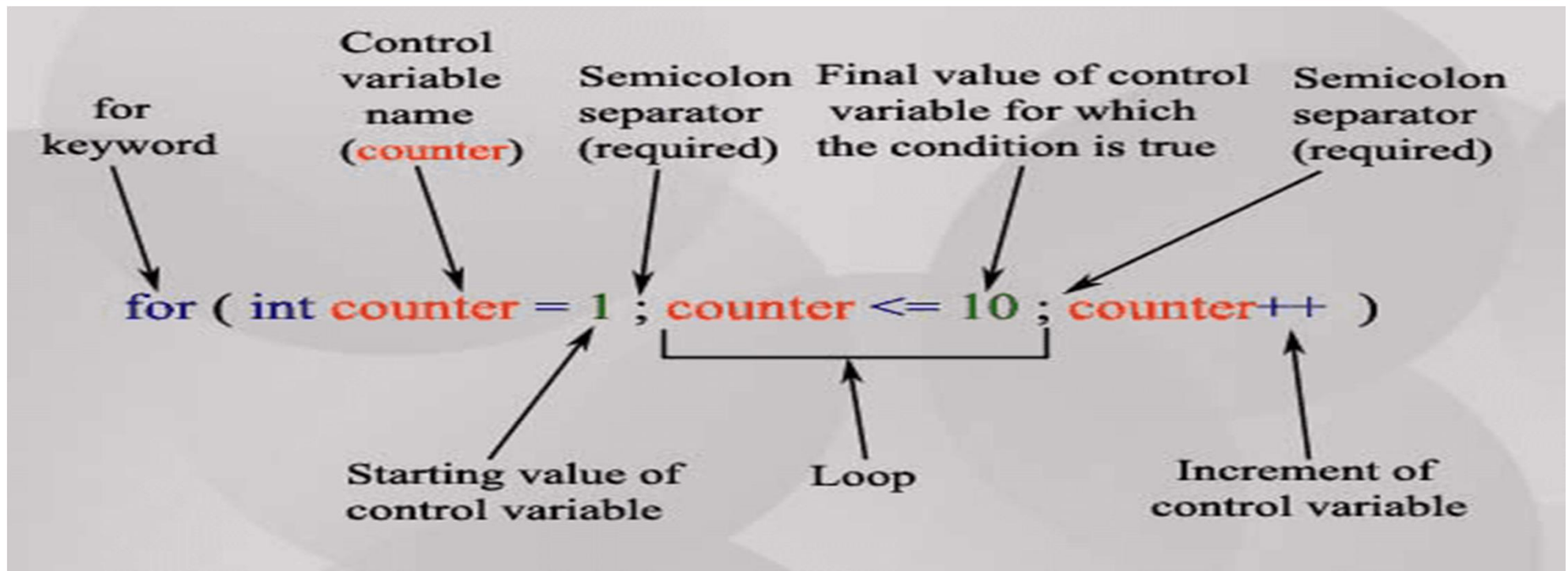
*initialize counter*: là một lệnh gán để khởi tạo biến điều khiển của vòng lặp trước khi đi vào vòng lặp

✓ *conditions* : là các biểu thức quan hệ để chỉ định khi nào vòng lặp sẽ kết thúc

*increment/decrement* : tăng hoặc giảm biến điều khiển



# VÒNG LẶP - for 2-3

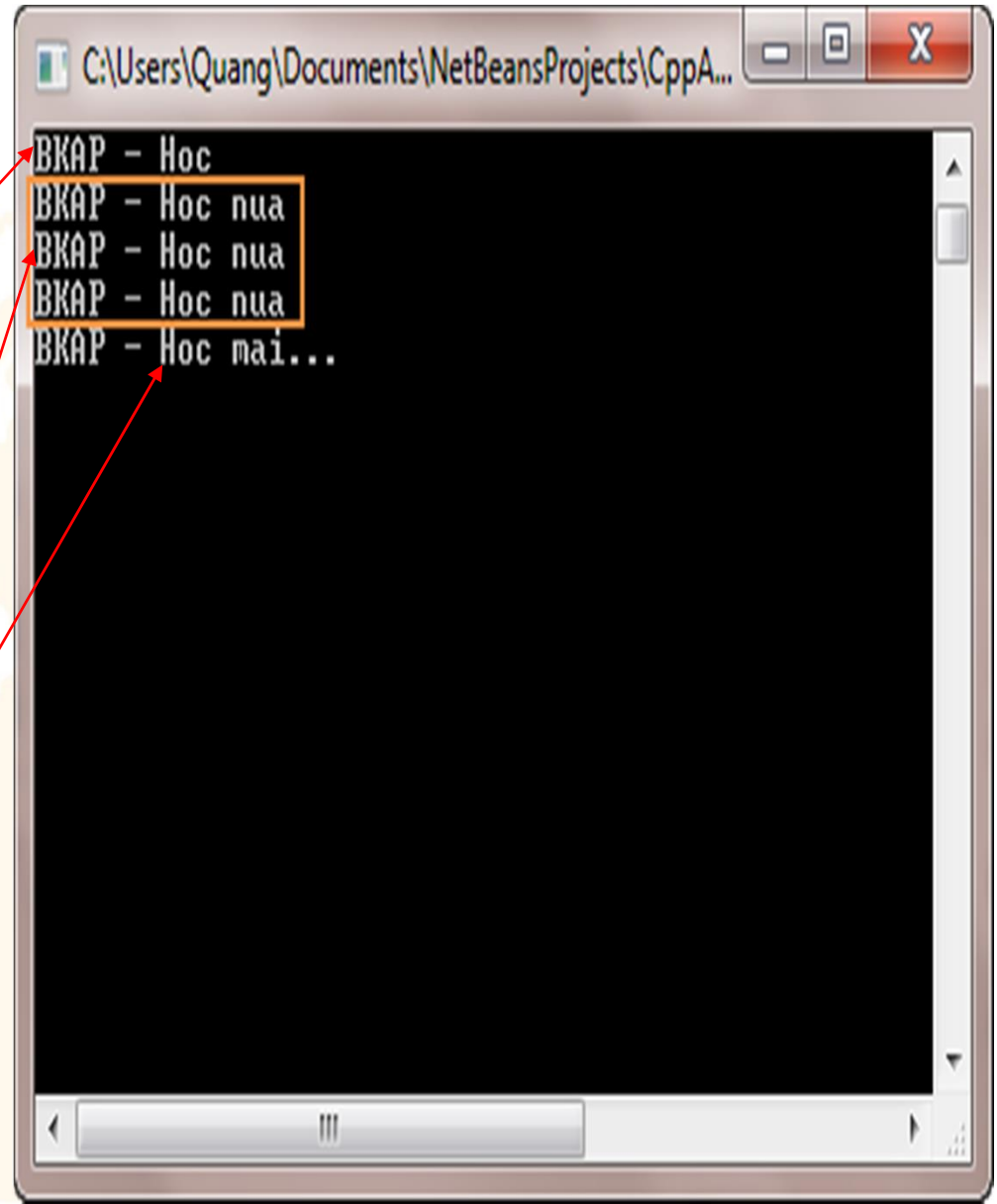


- Ba phần của vòng lặp **for** phải được phân cách bởi dấu chấm phẩy(`;`)
- Phần lệnh tạo nên thân vòng lặp có thể là một lệnh đơn hoặc một lệnh ghép (một tập nhiều lệnh)
- Vòng lặp **for** tiếp tục được thực thi khi biểu thức kiểm tra điều kiện vẫn có giá trị **true**. Khi điều kiện trở thành **false**, chương trình thực hiện lệnh theo sau vòng lặp **for**

# VÒNG LẶP - for 3-3

```
#include <stdio.h>
#include <conio.h>

main() {
    //Khoi tao bien dieu khien
    int i;
    //Cau lenh in 1
    printf("BKAP - Hoc\n");
    //Vong for
    for (i = 0; i < 3; i++) {
        //Khoi lenh than vong for
        printf("BKAP - Hoc nua\n");
    }
    //Cau lenh in 2
    printf("BKAP - Hoc mai...");
    getch();
}
```



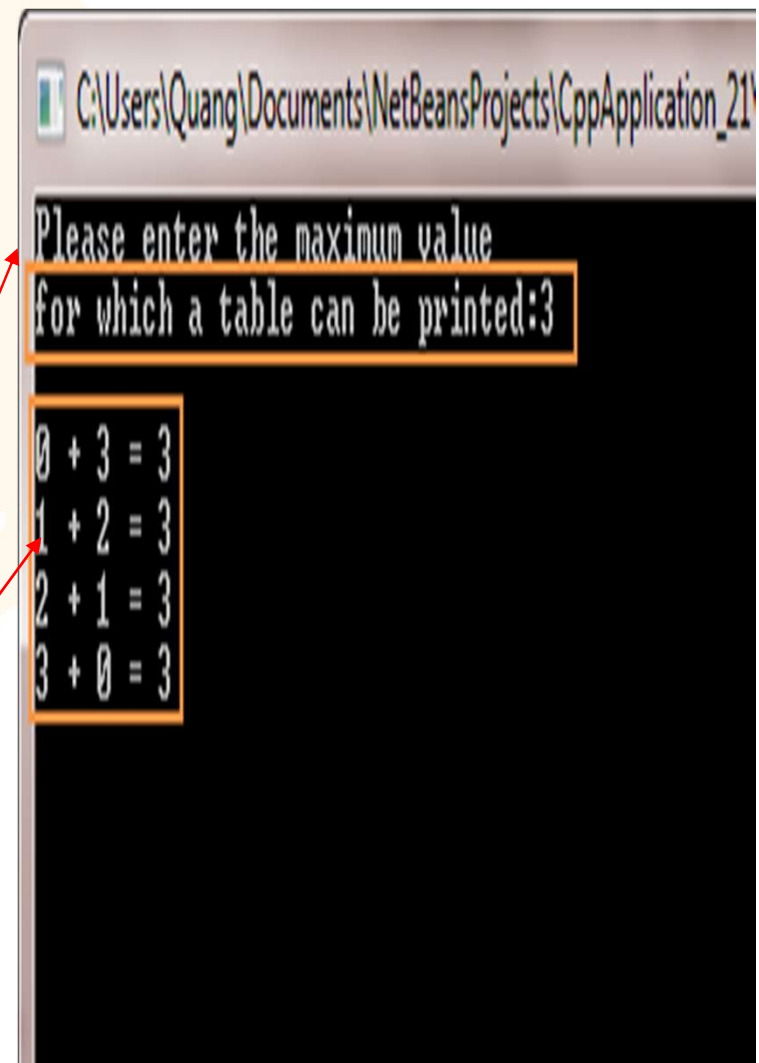
# TOÁN TỬ DẤU PHẪY

Vòng lặp **for** có thể được mở rộng bằng cách chứa nhiều giá trị khởi tạo và nhiều biểu thức tăng trị trong đặc tả của vòng lặp **for**

Cú pháp: **exprn1 , exprn2 ;**

```
#include <stdio.h>
#include <conio.h>

main() {
    int i, j, max;
    printf("Please enter the maximum value \n");
    printf("for which a table can be printed:");
    scanf("%d", &max);
    for (i = 0, j = max; i <= max; i++, j--) {
        printf("\n%d + %d = %d", i, j, i + j);
    }
    getch();
}
```





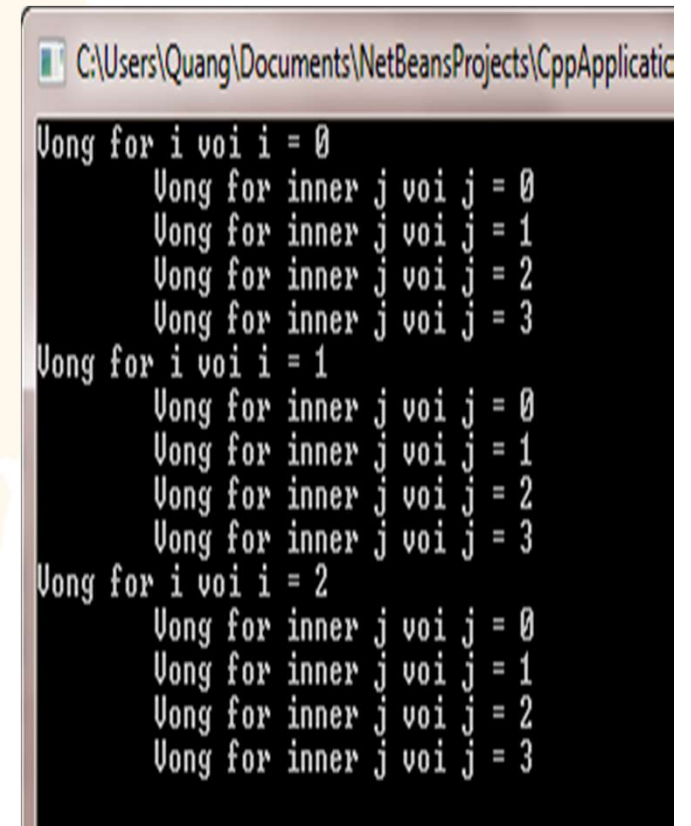
# VÒNG LẶP FOR LỒNG NHAU

- **Các vòng lặp for lồng nhau** khi nó có dạng như sau:

```
for (i = 1; i < max1; i++) {  
    ...  
    for (j = 0; j <= max2; j++) {  
        ...  
    }  
    ...  
}
```

Ví dụ

```
#include <stdio.h>  
#include <conio.h>  
  
main() {  
    int i, j;  
    for (i = 0; i < 3; i++) {  
        printf("Vong for i voi i = %d\n", i);  
        for (j = 0; j < 4; j++) {  
            printf("\tVong for inner j voi j = %d\n", j);  
        }  
    }  
    getch();  
}
```



```
Vong for i voi i = 0  
    Vong for inner j voi j = 0  
    Vong for inner j voi j = 1  
    Vong for inner j voi j = 2  
    Vong for inner j voi j = 3  
Vong for i voi i = 1  
    Vong for inner j voi j = 0  
    Vong for inner j voi j = 1  
    Vong for inner j voi j = 2  
    Vong for inner j voi j = 3  
Vong for i voi i = 2  
    Vong for inner j voi j = 0  
    Vong for inner j voi j = 1  
    Vong for inner j voi j = 2  
    Vong for inner j voi j = 3
```



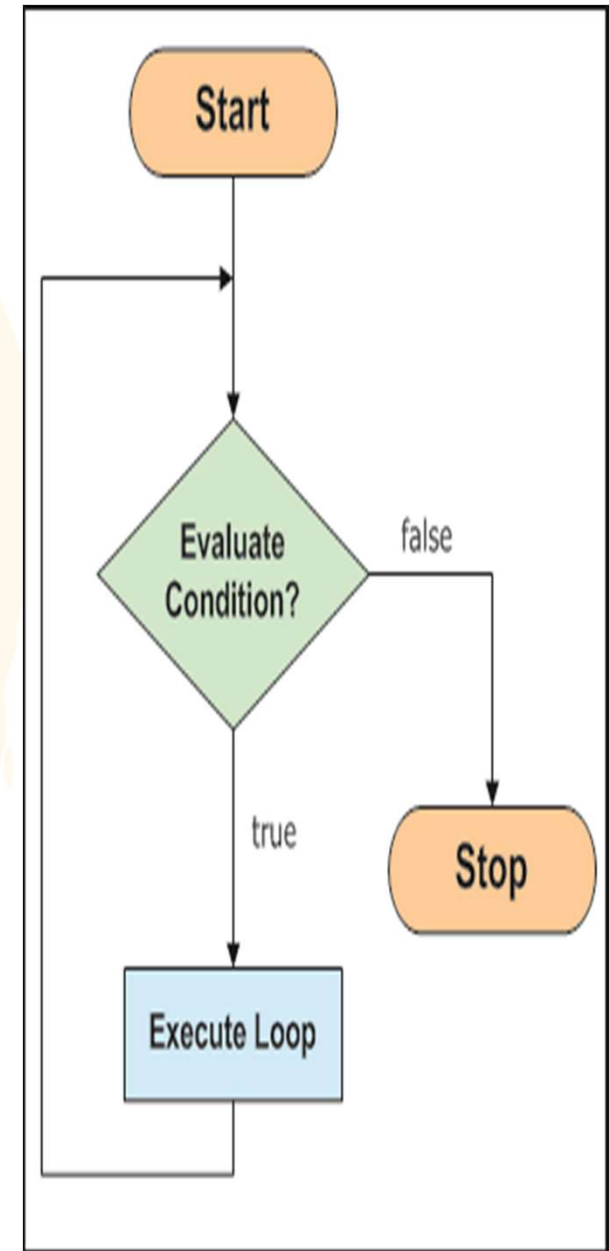
# VÒNG LẶP – while 1-2

## ■ Cú pháp

```
while (conditions) {  
    One or more Statements;  
}
```

Trong đó

- **conditions**: Vòng **while** sẽ thực hiện các lệnh khi **conditions** mang giá trị **true**, kết thúc khi **conditions** mang giá trị **false**



# VÒNG LẶP – while 2-2

- Ví dụ

```
#include <stdio.h>
#include <conio.h>

main() {
    int count = 1;
    while (count <= 10) {
        printf("Day la vong lap lan thu %d\n", count);
        count++;
    }
    printf("\nVong lap ket thuc.\n");
    getch();
}
```

```
C:\Users\Quang\Documents\NetBeansProjects\CppApplicati
Day la vong lap lan thu 1
Day la vong lap lan thu 2
Day la vong lap lan thu 3
Day la vong lap lan thu 4
Day la vong lap lan thu 5
Day la vong lap lan thu 6
Day la vong lap lan thu 7
Day la vong lap lan thu 8
Day la vong lap lan thu 9
Day la vong lap lan thu 10
Vong lap ket thuc.
```

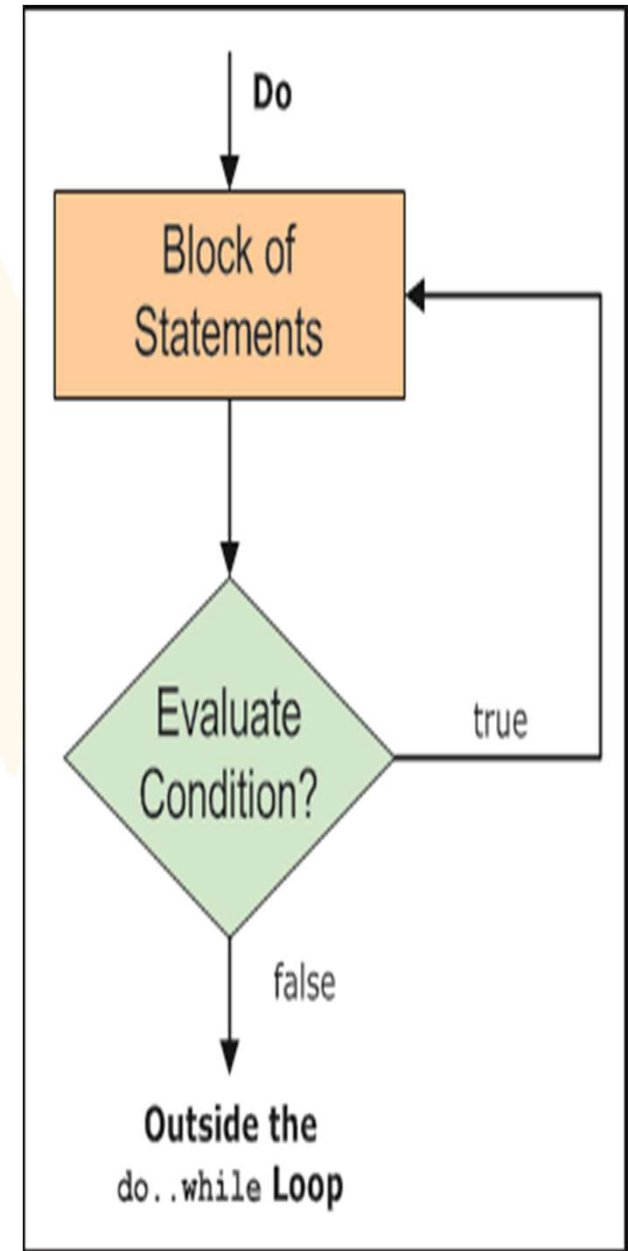
# VÒNG LẶP – do...while 1-2

## ■ Cú pháp

```
do {  
    //One or more Statement;  
} while (conditions);
```

Trong đó

- Các câu lệnh trong phần thân của vòng lặp được thực thi trước khi biểu thức điều kiện được kiểm tra
- **conditions**: các biểu thức điều kiện của vòng lặp. Vòng **do...while** sẽ thực hiện các lệnh khi **conditions** mang giá trị **true**, kết thúc khi **conditions** mang giá trị **false**



# VÒNG LẶP – do...while 2-2

- Ví dụ

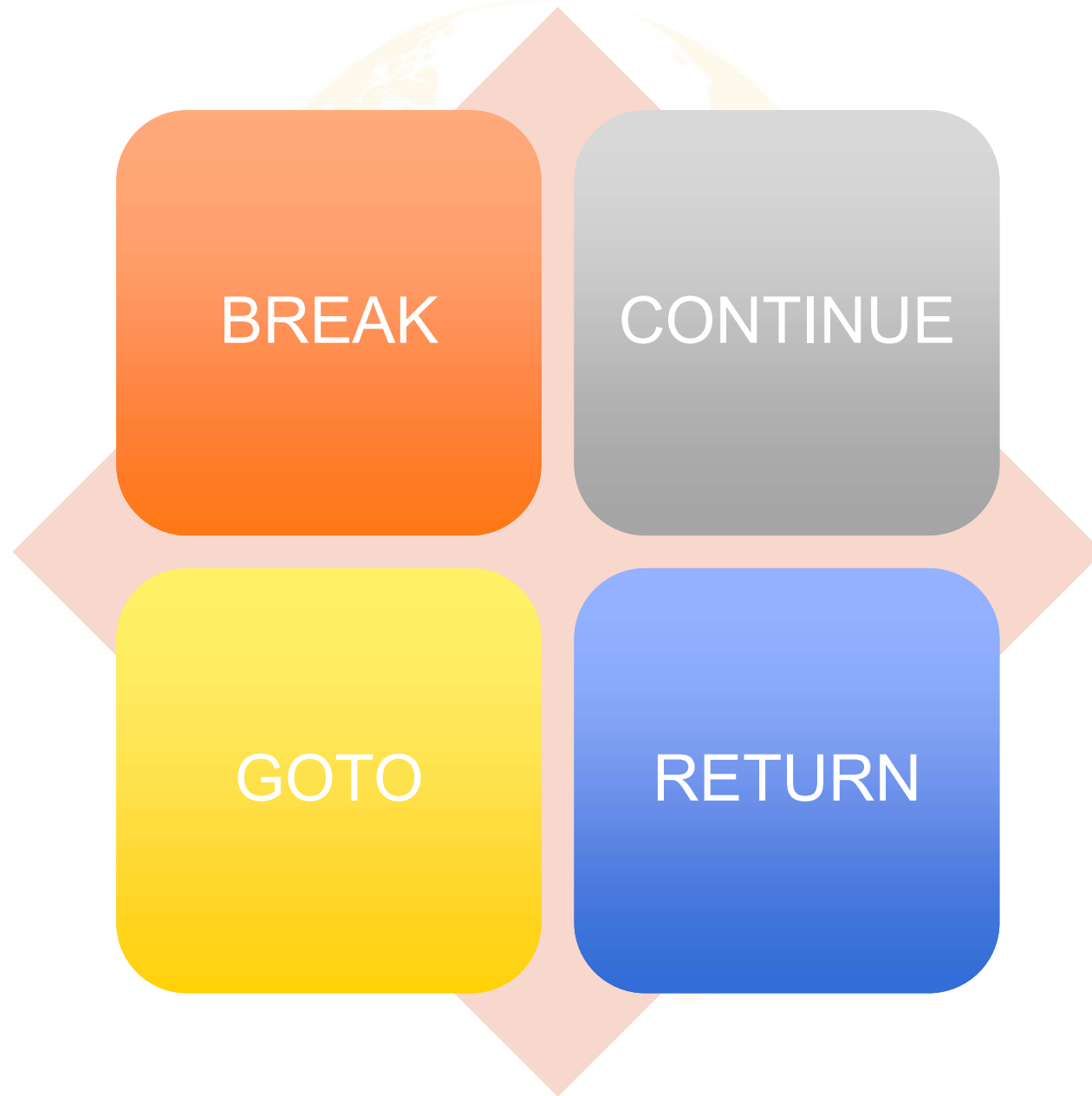
```
#include <stdio.h>
#include <conio.h>
```

```
main() {
    int num1, num2;
    num2 = 0;
    do {
        printf("\nNhap vao mot so : ");
        scanf("%d", &num1);
        printf(" num1 = %d", num1);
        num2++;
    } while (num1 != 0);
    printf("\nTong so lan lap la: %d", num2);
    getch();
}
```

```
C:\Users\Quang\Documents\NetBeansProjects\Cpp

Nhap vao mot so : 1
num1 = 1
Nhap vao mot so : 3
num1 = 3
Nhap vao mot so : 9
num1 = 9
Nhap vao mot so : 0
num1 = 0
Tong so lan lap la: 4
```

# CÁC LỆNH CHUYỂN ĐIỀU KHIỂN

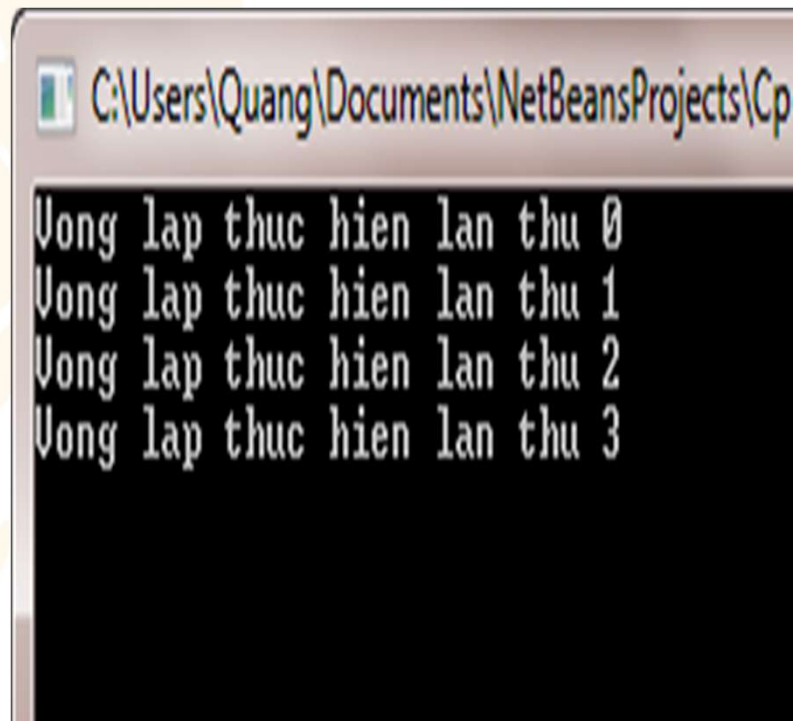


# LỆNH break

- ✓ Lệnh **break** được sử dụng để kết thúc một mệnh đề **case** trong câu lệnh **switch**
- ✓ Nó cũng có thể được sử dụng để kết thúc ngay giữa vòng lặp
- ✓ Khi gặp lệnh **break**, vòng lặp sẽ kết thúc ngay và điều khiển được chuyển đến lệnh kế tiếp bên ngoài vòng lặp

```
#include <stdio.h>
#include <conio.h>

main() {
    int i;
    for (i = 0; i < 5; i++) {
        printf("Vong lap thuc hien lan thu %d\n", i);
        if (i==3) {
            break;
        }
    }
    getch();
}
```



```
C:\Users\Quang\Documents\NetBeansProjects\Cp
Vong lap thuc hien lan thu 0
Vong lap thuc hien lan thu 1
Vong lap thuc hien lan thu 2
Vong lap thuc hien lan thu 3
```



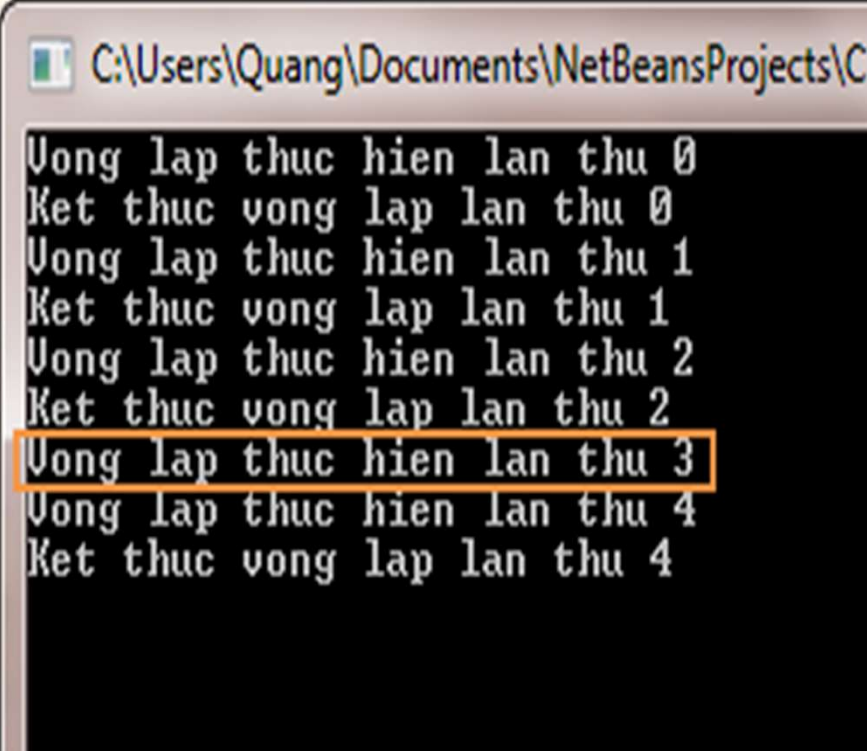
# LỆNH **continue**

- ✓ Lệnh **continue** dùng để bắt đầu thực hiện lần lặp kế tiếp của vòng lặp
- ✓ Khi gặp lệnh **continue**, các câu lệnh còn lại trong thân vòng lặp bị bỏ qua và điều khiển được chuyển đến lần lặp kế tiếp

✓ Ví dụ:

```
#include <stdio.h>
#include <conio.h>

main() {
    int i;
    for (i = 0; i < 5; i++) {
        printf("Vong lap thuc hien lan thu %d\n",i);
        if (i==3) {
            continue;
        }
        printf("Ket thuc vong lap lan thu %d\n",i);
    }
    getch();
}
```



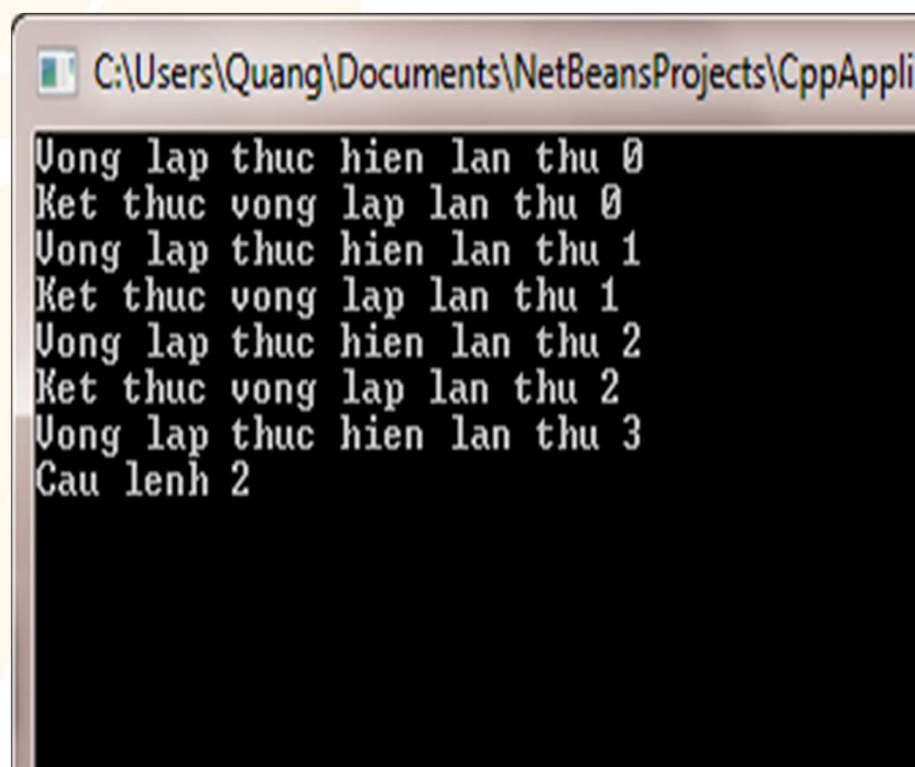
```
C:\Users\Quang\Documents\NetBeansProjects\C
Vong lap thuc hien lan thu 0
Ket thuc vong lap lan thu 0
Vong lap thuc hien lan thu 1
Ket thuc vong lap lan thu 1
Vong lap thuc hien lan thu 2
Ket thuc vong lap lan thu 2
Vong lap thuc hien lan thu 3
Vong lap thuc hien lan thu 4
Ket thuc vong lap lan thu 4
```

# LỆNH goto

- ✓ Lệnh **goto** chuyển điều khiển đến một câu lệnh bất kỳ khác bên trong cùng một hàm trong một chương trình C
- ✓ Điều này thật ra vi phạm đến qui luật của một ngôn ngữ lập trình cấu trúc.
- ✓ Chúng làm giảm độ tin cậy của chương trình và chương trình khó bảo trì

```
#include <stdio.h>
#include <conio.h>

main() {
    int i;
    for (i = 0; i < 5; i++) {
        printf("Vong lap thuc hien lan thu %d\n", i);
        if (i == 3) {
            goto lable;
        }
        printf("Ket thuc vong lap lan thu %d\n", i);
    }
    printf("Cau lenh 1\n");
lable:
    printf("Cau lenh 2");
    getch();
}
```



```
C:\Users\Quang\Documents\NetBeansProjects\CppAppli
Vong lap thuc hien lan thu 0
Ket thuc vong lap lan thu 0
Vong lap thuc hien lan thu 1
Ket thuc vong lap lan thu 1
Vong lap thuc hien lan thu 2
Ket thuc vong lap lan thu 2
Vong lap thuc hien lan thu 3
Cau lenh 2
```

Lệnh return được sử dụng để trở về từ một hàm

Thực hiện lệnh return để trở về vị trí mà tại đó hàm được gọi

Lệnh return có thể có một giá trị đi cùng, giá trị này được trả về cho chương trình gọi

Hàm `exit()` được sử dụng để thoát khỏi chương trình

Sử dụng hàm này sẽ kết thúc ngay chương trình và điều khiển được chuyển về cho hệ điều hành

# TÓM TẮT BÀI HỌC

- Các cấu trúc vòng lặp sẵn có trong C:
  - ✓ Vòng lặp **for**
  - ✓ Vòng lặp **while**
  - ✓ Vòng lặp **do ... while**
- Trong C, vòng lặp **for** cho phép sự thực thi các câu lệnh được lặp lại
  - Toán tử '**dấu phẩy**'
- Phần thân của lệnh **do** được thực hiện ít nhất một lần
- Trong C có bốn lệnh thực hiện sự rẽ nhánh không điều kiện: **return**, **goto**, **break**, và **continue**
- Hàm **exit()** kết thúc ngay chương trình và điều khiển được chuyển trở về cho hệ điều hành



TRƯỜNG ĐÀO TẠO LẬP TRÌNH VIÊN VÀ QUẢN TRỊ MẠNG QUỐC TẾ BACHKHOA-APTECH

# THANK FOR WATCH!

