

# **BÀI 5**

## **CẤU TRÚC VÒNG LẶP**

# MỤC TIÊU

Người học sau khi học xong bài 5 sẽ có các khái niệm cơ bản về các vấn đề sau:

- Vòng lặp for
- Vòng lặp while, do while
- Câu lệnh break
- Câu lệnh continue

# KIẾN THỨC CẦN CÓ

Các kiến thức cần thiết:

- Học xong bài 4;
- Biết về cấu trúc rẽ nhánh, điều kiện lặp và kết thúc vòng lặp;
- Khuyến nghị học môn Tin học cơ bản.

# NỘI DUNG

1. Vòng lặp for
2. Vòng lặp while
3. Vòng lặp do while
4. Câu lệnh break
5. Câu lệnh continue

# 1. VÒNG LẶP FOR

- Vòng lặp for có thể tạo lên chu trình bằng cách dùng thêm lệnh if và go. Nó cho phép xây dựng chu trình hiệu quả hơn.

- Cú pháp:

```
for (biểu thức 1; biểu thức 2; biểu thức 3)  
    khối lệnh;
```

- Các bước hoạt động của câu lệnh for:
  - Xác định biểu thức 1
  - Xác định biểu thức 2
  - Tùy theo tính đúng, sai của biểu thức 2 sẽ thực hiện 1 trong 2 nhánh:
    - Đúng : Thực hiện khối lệnh
    - Sai : Thoát khỏi vòng lặp for.

# 1. CÂU LỆNH FOR (tiếp theo)

## Ví Dụ:

```
/*Chương trình in ra 100 chữ xin chào*/  
for (i = 0, i < 100, i++)  
{  
    printf("Xin chao");  
}  
getch();
```

```
/*Chương trình đổi vị trí 2 kí tự*/  
for ( , , )  
{  
    c = x[i];  
    x[i] = x[j];  
    x[j] = c;  
}  
getch();
```

## 2. Vòng lặp WHILE

- Vòng lặp while dùng để xây dựng chu trình thực hiện lệnh.
- Cú pháp:

```
while (biểu thức)
{
    khối lệnh;
}
```

- Các bước hoạt động của vòng lặp:
  - Xác định giá trị biểu thức
  - Căn cứ giá trị, sẽ thực hiện 1 trong 2 nhánh:
    - Đúng : Thì thoát chu trình và thực hiện lệnh tiếp theo
    - Sai : Thực hiện lại chương trình.
- Vòng lặp while có thể tạo ra vòng lặp vô tận, hoặc không thực hiện khối lệnh nào khi biểu thức sai.

### 3. TOÁN TỬ DO WHILE

- Toán tử do while dùng để xây dựng chu trình thực hiện lệnh trong đó có kiểm tra điều kiện ở cuối chu trình.
- Cú pháp:

```
do  
    Khối lệnh;  
while (biểu thức);
```
- Các bước hoạt động của toán tử:
  - Thực hiện các khối lệnh;
  - Kiểm tra giá trị biểu thức;
  - Căn cứ giá trị, sẽ thực hiện 1 trong 2 nhánh:
    - Đúng: Thì thoát chu trình và thực hiện lệnh tiếp theo;
    - Sai : Thực hiện lại chương trình.
- Câu lệnh do while có thể tạo ra vòng lặp vô tận, hoặc không thực hiện khối lệnh nào khi biểu thức sai.



## 4. CÂU LỆNH BREAK

- Câu lệnh break cho phép thoát ra khỏi các vòng lặp, chu trình như for, switch, while, do while...
- Câu lệnh break có thể thoát ra khỏi chu trình từ vị trí bất kỳ, tại nơi câu lệnh được đặt;
- Câu lệnh break có thể được thay thế bằng câu lệnh goto với nhãn thích hợp.

### Ví Dụ:

```
if((n mod i)==0)
{
s = s+1;
break;
}
```

## 5. CÂU LỆNH CONTINUE

- Câu lệnh continue dùng để bắt đầu một vòng mới của chu trình bên trong nhất chứa nó.
- Khi gặp câu lệnh continue bên trong vòng lặp for, chương trình sẽ quay lại điểm bắt đầu của vòng lặp.
- Khi gặp câu lệnh continue bên trong vòng lặp while hoặc do while, chương trình sẽ quay lại điểm kiểm tra điều kiện kết thúc chương trình.
- Câu lệnh continue không làm việc với lệnh switch

### Ví Dụ :

- nếu  $a[i][j] = 0$  thì thực hiện lại;
- nếu không thì tăng biến s lên 1.

```
for(i =0, i<8,i++) {  
    for(j=0, j<6,j++) {  
        if (a[i][j]==0)  
            continue;  
        s = s+1;  
        -----  
    }  
}
```

# TÓM LƯỢC CUỐI BÀI

- Qua nội dung Bài 5 người học được cung cấp các kiến thức cơ bản để có thể lập trình được các vòng lặp như do while, while, for...
- Cấu trúc do-while gần giống như cấu trúc while, khác nhau ở quá trình kiểm tra điều kiện kết thúc;
- Câu lệnh break cho phép kết thúc vòng lặp ngay bên trong hàm mà không cần đợi kiểm tra điều kiện kết thúc;
- Câu lệnh continue cho phép thực hiện lại hàm mà không cần thực thi hết hàm.