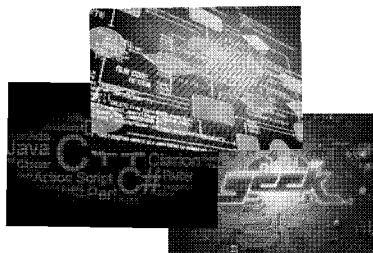




ĐẠI HỌC QUỐC GIA TP HCM  
TRƯỜNG ĐẠI HỌC  
CÔNG NGHỆ THÔNG TIN

# NHẬP MÔN LẬP TRÌNH CHƯƠNG V HÀM VÀ THAM SỐ



Nguyễn Trọng Chính  
chinhnt@uit.edu.vn

## HÀM VÀ THAM SỐ

- ❖ HÀM ĐỆ QUY
- ❖ HÀM TRONG CHƯƠNG TRÌNH NHIỀU TẬP TIN MÃ NGUỒN

## HÀM ĐỆ QUY

### ❖ KHÁI NIỆM

Hàm đệ quy là hàm mà trong phần thân hàm có lời gọi hàm đến chính nó một cách trực tiếp hoặc gián tiếp.

#### Ví dụ 1

```
int giaithua(int n) {  
    if (n > 1)  
        return giaithua(n - 1) * n;  
    else  
        return 1;  
}
```

## HÀM ĐỆ QUY

#### Ví dụ 2

```
int chan(int n);  
int le(int n);  
int chan(int n) {  
    if (n < 2) return !n;  
    return le(n-1);  
}  
int le(int n) {  
    if (n < 2) return n;  
    return chan(n-1);  
}
```

## HÀM ĐỆ QUY

### ❖ CÁC THÀNH PHẦN CỦA MỘT HÀM ĐỆ QUY:

- Điều kiện dừng: là trường hợp suy thoái của biểu thức truy hồi mà hàm thực hiện.
- Phần đệ quy: là một hoặc nhiều lời gọi hàm đến trực tiếp (như trong ví dụ 1) hoặc gián tiếp (như trong ví dụ 2) đến bản thân hàm.

Lưu ý: Phải chắc chắn điều kiện dừng trong hàm đệ quy được thực hiện. Nếu không, hàm sẽ được gọi đệ quy đến khi tràn stack.

## HÀM ĐỆ QUY

### ❖ CÁC DẠNG HÀM ĐỆ QUY:

\* Đệ quy tuyến tính là hàm đệ quy trong phần thân hàm chỉ chứa một lời gọi hàm đến chính nó.

Ví dụ:

```
int tong(int n) {  
    if (n == 1) return n;  
    return tong(n - 1) + n;  
}
```

## HÀM ĐỆ QUY

### ❖ CÁC DẠNG HÀM ĐỆ QUY:

\* **Đệ quy nhị phân** là hàm đệ quy trong phần thân hàm chứa hai lời gọi hàm đến chính nó. Đệ quy nhị phân thường được dùng cho các vấn đề liên quan đến cây nhị phân.

Ví dụ:

```
int fibo(int n) {  
    if (n < 2) return n;  
    return fibo(n - 1) + fibo(n - 2);  
}
```

## HÀM ĐỆ QUY

### ❖ CÁC DẠNG HÀM ĐỆ QUY:

\* **Đệ quy phi tuyến** là hàm đệ quy trong phần thân hàm chứa nhiều hơn hai lời gọi hàm đến chính nó.

\* **Đệ quy tương hỗ** là các hàm mà trong phần thân của nó chứa lời gọi hàm lẫn nhau.

Trường hợp các hàm chan(), le() trong ví dụ 2 là trường hợp đệ quy tương hỗ. Trong đó, phần thân hàm chan() có lời gọi hàm đến hàm le() và trong phần thân hàm le() có lời gọi hàm đến hàm chan().

## HÀM ĐỆ QUY

### ❖ĐẶC ĐIỂM CỦA HÀM ĐỆ QUY:

- Thể hiện giải thuật một cách tự nhiên theo logic.
- Phải thực hiện nhiều lần gọi hàm làm tốn thời gian xử lý và bộ nhớ.
- Trong những trường hợp đơn giản cần tốc độ xử lý cao, các hàm đệ quy cần được chuyển thành các cấu trúc lặp phù hợp.

## BÀI TẬP

Xây dựng các hàm sau:

- 1) Vẽ tam giác Pascal.
- 2) Viết chương trình đổi số thập phân thành số thập lục phân.
- 3) In ra giải pháp cho bài toán tháp Hà Nội với số đĩa  $n$ .

## HÀM TRONG CHƯƠNG TRÌNH NHIỀU TẬP TIN MÃ NGUỒN

### ❖CẤU TRÚC CHƯƠNG TRÌNH

Chương trình được phân chia thành các module và được tổ chức thành các file như sau:

\* **File .h** (file header) chứa các khai báo hàm, kiểu dữ liệu. File .h thường có cấu trúc như sau:

```
#ifndef so_hieu
```

```
#define so_hieu
```

```
các khai báo và các chỉ thị tiền xử lý
```

```
#endif
```

## HÀM TRONG CHƯƠNG TRÌNH NHIỀU TẬP TIN MÃ NGUỒN

### ❖CẤU TRÚC CHƯƠNG TRÌNH

❖ + Chỉ thị **#ifndef** kiểm tra một ký hiệu đã được định nghĩa hay chưa

+ Chỉ thị **#endif** cho biết kết thúc phần xử lý kiểm tra của chỉ thị **#ifndef** tương ứng

+ **so\_hieu** là một ký hiệu đảm bảo chưa được định nghĩa trong bất kỳ file .h trong mã nguồn chương trình.

Mục đích của cấu trúc này để khi biên dịch các file mã nguồn không bị trường hợp khai báo trùng do quá trình include file bị trùng lặp.

## HÀM TRONG CHƯƠNG TRÌNH NHIỀU TẬP TIN MÃ NGUỒN

### ❖CẤU TRÚC CHƯƠNG TRÌNH

\* File .cpp/c (code) : chứa các khai báo biến, các hàm được hiện thực.

Ví dụ:

Viết chương trình cho phép chọn lựa chức năng in bảng cửu chương và vẽ một hình chữ nhật kích thước m dòng n cột lên màn hình.

#### ***file tienich.h***

```
#ifndef _tienich_001
#define _tienich_001
int nhap(char *tb);
#endif
```

#### ***File tienich.cpp***

```
#include <stdio.h>
int nhap(char *tb) {
    int a;
    printf(tb);scanf("%d%c",&a);
    return a;
}
```

**file bang.h**

```
#ifndef _bang_001
#define _bang_001
void bangcuuchuong();
#endif
```

**File bang.cpp**

```
#include <stdio.h>
#include "tienich.h"
void bangcuuchuong() {
    int i,j;
    i = nhap("Bang cuu chuong cho cot i = ");
    for(j = 2; j < 10; printf("%dx%d=%d\n",i,j,i*j), j++);
}
```

**file hinhcn.h**

```
#ifndef _hinh_001
#define _hinh_001
void vehinhcn();
#endif
```

**File hinhcn.cpp**

```
#include <stdio.h>
#include "tienich.h"
void vehinhcn() {
    int i,j, n, m;
    m = nhap("So dong m = "); n = nhap("So cot n = ");
    for (i = 0; i < m; printf("\n"), i++)
        for(j = 0; j < n; printf("*"), j++);
}
```



**File CT.cpp**

```
#include "tienich.h"
#include "bang.h"
#include "hinhcn.h"
void main() {
    int chon;
    do {
        chon = nhap("1-In bang\n2-Ve hinh\n0-Thoat\nChon: ");
        switch(chon) {
            case 1: bangcuuchuong();break;
            case 2: vehinhcn();break;
        }
    } while (chon);
}
```

## **HÀM TRONG CHƯƠNG TRÌNH NHIỀU TẬP TIN MÃ NGUỒN**

### **❖ TRIỂN KHAI TRÊN VC++**

Triển khai chương trình nhiều tập tin mã nguồn trên Visual C++ qua các bước sau:

- Tạo project.
- Add các file .h
- Add các file .cpp
- Build project, VC++ sẽ tự động biên dịch và liên kết các file cần thiết để tạo thành file .exe.