



TRƯỜNG ĐẠI HỌC ĐỒNG THÁP
Khoa sư phạm Toán – Tin



Bài giảng

NGÔN NGỮ LẬP TRÌNH C

ThS. GV. Huỳnh Khải Vinh

Học phần: Lập trình căn bản



NỘI DUNG MÔN HỌC



01

Một số khái niệm cơ bản (4)

02

Các kiểu dữ liệu và phép toán (4)

03

Các lệnh cấu trúc (6 + 6)

04

Các thao tác trên dữ liệu kiểu mảng (4 + 4)

05

HÀM VÀ PHẠM VI CỦA BIẾN (3 + 4)



Đặt vấn đề



Bài toán: Để mô phỏng các phép toán cơ bản trên máy tính bỏ túi, ta phân chia công việc:

- Việc 1: **Thực hiện cộng 2 số a , b = tổng**
- Việc 2: **Thực hiện trừ 2 số a , b = hiệu**
- Việc 3: **Thực hiện nhân 2 số a , b = Tích**
- Việc 4: **Thực hiện chia 2 số a , b = Thương**
- Việc 5: **Xem kết quả việc thực hiện phân công**



=> Để thấy rõ hiệu quả của việc phân công công việc như mô tả, C mô tả bằng cách cài đặt HÀM



HÀM & PHẠM VI CỦA BIẾN



- I. Hàm**
- II. Tham số hình thức và tham số thực**
- III. Truyền tham trị và truyền tham biến**
- V. Bài tập**



1. Hàm

- **Hàm**
- Tham số
- Truyền tham trị &
- Truyền tham biến
- Bài tập

➤ Khái niệm

Hàm là **một đoạn chương trình** thực hiện một tác vụ được định nghĩa cụ thể. Các hàm được sử dụng để rút gọn cho một chuỗi các công việc **được thực hiện lặp đi lặp lại** nhiều lần.

```
#include <stdio.h>

void functionName()
{
    ... ..
    ... ..
}

int main()
{
    ... ..
    ... ..
    functionName();
    ... ..
    ... ..
}
```

The diagram illustrates the flow of control between the `main()` function and the `functionName()` function. An arrow originates from the `functionName();` line in the `main()` function, points to the opening curly brace of the `functionName()` function, and then returns to the line immediately following the function call in `main()`.



Hàm

- **Hàm**
- Tham số
- Truyền tham trị &
- Truyền tham biến
- Bài tập

➤ Khai báo

Cú pháp chung:

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

```
<Kiểu_dữ_liệu> <Tên_hàm> ([<ds_tham_số>]);
```

```
int main(){
```

```
    --Lời gọi hàm
```

```
    return 0;
```

```
}
```

```
//Định nghĩa hàm
```



```
//Định nghĩa hàm
```

```
Kiểu_dữ_liệu> <Tên_hàm> ([<ds_tham_số>]){
```

```
    <Khai_báo_biến_cục_bộ>;
```

```
    <Nội_dung_hàm>;
```

```
    return [(<biểu_thức>)];
```

```
}
```



Hàm (tt)

- **Hàm**
- Tham số
- Truyền tham trị &
- Truyền tham biến
- Bài tập

➤ Diễn giải:

- **Kiểu_dữ_liệu**: là kiểu mà giá trị của hàm trả về, các kiểu trả về có thể: **char, int, long, float, double, void**,...
- **Tên_hàm**: đặt bất kỳ tên hợp lệ nào (nên đặt tên liên quan đến nội dung bên trong hàm).
- **ds_tham_số**: khai báo tương tự như biến, các tham số cách nhau bởi dấu phẩy “,”.

Các tham số bên trong cặp () là các tham số hình thức.



Nếu hàm được gọi trước khi nó được định nghĩa thì bắt buộc phải khai báo trước hàm main().



Hàm (tt)

- Hàm
- Tham số
- Truyền tham trị &
- Truyền tham biến
- Bài tập

➤ Ví dụ: Hàm in một số

```
#include <stdio.h>
void xuat(int); // Khai báo hàm
int main() {
    int a, b;
    printf ("Nhap so: ");
    scanf ("%d, %d", &a, &b);
    //Goi hàm
    xuat (a);
    xuat (b);
    return 0;
}

//Định nghĩa hàm
void xuat (int t){
    printf ("\nGia tri nhap %d", t);}
```





2. Tham số

- Hàm
- **Tham số**
- Truyền tham trị &
- Truyền tham biến
- Bài tập

➤ Khái niệm

Khi có thể truyền thông tin đến một hàm được gọi, để hàm đó có dữ liệu xử lý. Chúng ta làm điều đó thông qua các **tham số của hàm** hay **đối số**.

- ✓ Một tham số của hàm *là một biến* được sử dụng trong một hàm.
- ✓ Các tham số của hàm **hoạt động gần như giống hệt với các biến** được định nghĩa bên trong hàm.



Lưu ý:

Điểm khác biệt: được khởi tạo với một giá trị được cung cấp bởi người gọi của hàm.



Tham số

- Hàm
- **Tham số**
- Truyền tham trị &
- Truyền tham biến
- Bài tập

Các tham số của hàm được định nghĩa trong khai báo hàm bằng cách **đặt chúng ở giữa dấu ngoặc đơn sau tên hàm**, với nhiều tham số được **cách nhau bằng dấu phẩy**.

Có 2 loại tham số:

- ✓ Tham số hình thức.
- ✓ Tham số thực.



Tham số

- Hàm
- **Tham số**
- Truyền tham trị &
- Truyền tham biến
- Bài tập

// Ví dụ tham số hình thức:

```
int cong2so(int a, int b) {  
    int tong = a + b;  
    return tong;  
}
```



// Ví dụ tham số thực

```
cong2so(2, 3);
```

```
// Tham số trong lời gọi hàm
```



Phạm vi của biến (tt)

- Hàm
- Tham số
- Truyền tham trị &
- Truyền tham biến
- Bài tập

Trong C, ta có thể khai báo một biến và **chỉ định** phạm vi tiếp cận nó bằng cách xác định phạm vi của nó. Các biến với phạm vi cục bộ chỉ có thể tiếp cận được trong khối mà nó được khai báo.

Ta phân biệt các phạm vi: **phạm vi khối, phạm vi hàm, phạm vi chương trình, phạm vi tập tin.**

❖ Phạm vi khối

Một khối được giới hạn bởi ngoặc {}.

Biến khai báo trong khối đó có phạm vi khối, nghĩa là nó chỉ hoạt động trong khối đó mà thôi.

Phạm vi này còn gọi là **cục bộ**, và biến được gọi là **biến cục bộ**.



Phạm vi của biến (tt)

- Hàm
- Tham số
- Truyền tham trị &
- Truyền tham biến
- Bài tập

❖ Phạm vi hàm

Phạm vi hàm chỉ định **một biến có phạm vi hoạt động từ đầu đến cuối một hàm** (không nhầm lẫn với biến có phạm vi khối). Trong C, chỉ có nhãn (label) đối với lệnh goto là có phạm vi hàm.

❖ Phạm vi chương trình

Biến có phạm vi chương trình khi nó **được khai báo bên ngoài các hàm**. Biến này còn gọi là **biến toàn cục**.

❖ Phạm vi tập tin

Biến được **khai báo là toàn cục** và khai báo với từ khóa **static** được gọi là có phạm vi tập tin.



Phạm vi của biến (tt)

- Hàm
- Tham số
- Truyền tham trị &
- Truyền tham biến
- Bài tập

//VD:

```
#include <stdio.h>
#include <conio.h>
static int i = 1;
long luythua(int a, int x){
    long lt=1;
    if (x<=0)
        lt = 1;
    else
        for (i=1;i<=x; i++)
            lt *= a;
    return lt;
}
```

```
void main() {
    int a = 2;
    int x = 3;
    for (; i<=a; i++);
    printf ("%d", i*a);
    printf ("\n%ld",
        luythua(a,x) );
    return 0;
}
```



Bài tập tạo hàm

- Hàm
- **Tham số**
- Truyền tham trị &
- Truyền tham biến
- Bài tập

Viết chương trình thực hiện các phép toán cơ bản đối với 2 số nguyên:

Yêu cầu:

- Menu lựa chọn phép toán
- Cộng hai số
- Trừ hai số
- Tích 2 số
- Thương 2 số



3. Truyền tham trị và tham biến

- Hàm
- Tham số
- Truyền tham trị &
- Truyền tham biến
- Bài tập

➤ Truyền tham trị

Khi các tham số được truyền theo trị, một bản sao giá trị của các tham số thực được tạo ra và gán cho các tham số hình thức của hàm.

Vì vậy, mọi sự thay đổi trong hàm trên bản sao sẽ không ảnh hưởng đến giá trị ban đầu của biến (nếu có) nằm trong lời gọi hàm.



Truyền tham trị và tham biến

- Hàm
- Tham số
- Truyền tham trị &
- Truyền tham biến
- Bài tập

```
//Tham trị
#include <stdio.h>
#include <conio.h>
void hoanvi(a,b){
    int tam;
    tam = a;
    a = b;
    b = tam;
}
```



```
int main(){
    int a, b;
    printf("\nNhap so thu nhat = ");
    scanf("%d",&a);
    printf("\nNhap so thu hai = ");
    scanf("%d",&b);
    printf("\nTruoc khi hoan vi a
    =%d, b = %d", a, b);
    hoanvi(a,b);
    printf("\nSau khi hoan vi a
    =%d, b = %d", a, b);
    return 0;
}
```



Truyền tham trị và tham biến

- Hàm
- Tham số
- Truyền tham trị &
- Truyền tham biến
- Bài tập

➤ Truyền tham biến

Hàm gọi sẽ **truyền trực tiếp tham số** đó cho hàm được gọi.

Trong trường hợp này tham số hình thức là tham số thực.

Như vậy, nếu bên trong hàm bị gọi có thay đổi đến tham số hình thức thì những thay đổi đó cũng sẽ có tác dụng trên tham số thực tương ứng.

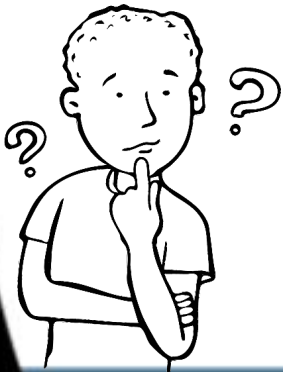


Truyền tham trị và tham biến

- Hàm
- Tham số
- Truyền tham trị &
- Truyền tham biến
- Bài tập

```
//Tham biến
#include <stdio.h>
#include <conio.h>
void hoanvi (*a, *b) {
    int tam;
    tam = *a;
    *a = *b;
    *b = tam;
}
```

```
int main() {
    int a, b;
    printf("\nNhap so thu nhat= ");
    scanf("%d", &a);
    printf("\nNhap so thu hai = ");
    scanf("%d", &b);
    printf("\nTruoc khi hoan vi a
    =%d, b = %d", a, b);
    hoanvi (&a, &b);
    printf("\nSau khi hoan vi a
    =%d, b = %d", a, b);
    return 0;
}
```





Bài tập

- Hàm
- Tham số
- Truyền tham trị &
- Truyền tham biến
- **Bài tập**

5.1. Viết chương trình in dãy Fibonanci từ 1 đến n, với n nhập từ bàn phím. Yêu cầu:

- Viết hàm tính giá trị Fibo của một số n.
- In dãy Fibo từ 1 -> n bằng cách gọi hàm đã xây dựng (định nghĩa).

5.2. Viết chương trình tính tổ hợp C_n^k với k, n nhập từ bàn phím. Yêu cầu:

- Viết hàm tính giai thừa của số n.
- Gọi hàm giai thừa đã định nghĩa để tính tổ hợp theo công thức:



Bài tập

- Hàm
- Tham số
- Truyền tham trị &
- Truyền tham biến
- **Bài tập**

5.3. Viết chương trình in tam giác PASCAL với chiều cao nhập vào VD: $h = 5 \Rightarrow$ in tam giác pascal như sau:

1						C_0^0					
1	1					C_1^0	C_1^1				
1	2	1				C_2^0	C_2^1	C_2^2			
1	3	3	1			C_3^0	C_3^1	C_3^2	C_3^3		
1	4	6	4	1		C_4^0	C_4^1	C_4^2	C_4^3	C_4^4	
1	5	10	10	5	1	C_5^0	C_5^1	C_5^2	C_5^3	C_5^4	C_5^5

5.4. Viết chương trình in ra dãy số nguyên tố bé hơn n. Chú ý: viết hàm xét số nguyên tố.



Bài tập

- Hàm
- Tham số
- Truyền tham trị &
- Truyền tham biến
- **Bài tập**

5.5. Viết chương trình nhập vào 2 số nguyên: a: tử số, b: mẫu số. Yêu cầu:

- Định nghĩa hàm tìm ước số chung lớn nhất của 2 số a, b.
- In phân số tối giản của phân số đã nhập.

5.6. Viết chương trình nhập vào 2 số a, b và một phép toán. Yêu cầu:

- Định nghĩa hàm cộng, trừ, nhân, chia 2 số.
- Tùy theo phép toán đã nhập gọi các hàm tương ứng. Ngoài ra, cho phép người sử dụng lựa chọn để tiếp tục hay dừng chương trình.



Bài tập

- Hàm
- Tham số
- Truyền tham trị &
- Truyền tham biến
- **Bài tập**

Đệ quy

Hàm gọi lại chính hàm của nó gọi là đệ quy.

Hàm đệ quy có thuận lợi là cho thấy thuật toán rõ ràng, cách viết đơn giản. Tuy nhiên thường là chạy chậm vì nó đòi hỏi nhiều bộ nhớ để lưu trữ các biến hàm trung gian (stacks). Cho nên cũng không nên sử dụng nhiều nếu không cần thiết.

5.7. Viết và gọi hàm đệ quy cộng số nguyên từ 1 đến n nhập từ bàn phím.

5.8. Viết lại hàm Fibo theo đệ quy.

5.9. Viết lại hàm giai thừa theo đệ quy.

5.10. Viết hàm tổ hợp theo đệ quy.

Nếu: $k = 0$ hoặc $k = n$ thì $\text{tohop}(k, n) = 1$

Ngược lại: $\text{tohop}(k, n) = \text{tohop}(k-1, n-1) + \text{tohop}(k, n-1)$



Bài tập

- Hàm
- Tham số
- Truyền tham trị &
- Truyền tham biến
- **Bài tập**

Định nghĩa các hàm sau đây theo 2 cách đệ quy và không đệ quy

5.11. Tính tổng dãy số:

$$1/2 + 2/3 + 3/4 + \dots + n/(n+1).$$

5.12. Tính giá trị thứ n theo công thức:

- Nếu $n = 0$ thì $F(0) = 3$

- Nếu $n = 1$ thì $F(1) = 2 * F(0) + 3$

....

- Nếu $n = n$ thì $F(n) = 2 * (F(n-1) + 3$

5.13. Tính số mũ a^n .



Bài tập

- Hàm
- Tham số
- Truyền tham trị &
- Truyền tham biến
- **Bài tập**

Sử dụng các yêu cầu của bài tập chương 4, viết thành hàm, nhập, xuất, tính toán, sắp xếp trên ma trận:

Bài tập 4.9 – 4.15



Tổng kết chương



- Khái niệm hàm
- Phương pháp khởi tạo và định nghĩa hàm
- Tham số trong lời gọi hàm
- Phân biệt truyền tham trị và truyền tham biến
- Phân tích và vận dụng thực hành khởi tạo và định nghĩa hàm.

