



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN
MÔN: **NHẬP MÔN LẬP TRÌNH**

HƯỚNG DẪN THỰC HÀNH

TUẦN 04

HÀM (CƠ BẢN)

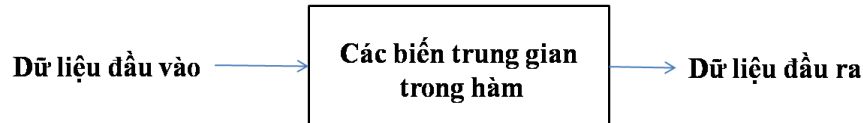
TP.HCM, ngày 09 tháng 09 năm 2017

MỤC LỤC

1	Khái niệm	3
2	Các loại tham số dùng cho hàm.....	3
2.1	Tham số giá trị (tham trị)	4
2.2	Tham số dạng tham chiếu (tham chiếu).....	5
3	Biến tĩnh cục bộ.....	5
4	Bài tập.....	6

1 Khái niệm

Trong thực tế, có nhiều vấn đề đòi hỏi cần phân rã thành các mục tiêu nhỏ hơn. Sau khi hoàn thành tất cả các mục tiêu đó thì ta có thể hoàn thành công việc. Đây là một trong những phong cách lập trình ban đầu của ngành công nghệ phần mềm và vẫn còn tồn tại cho tới ngày nay. Trong ngôn ngữ C, có khái niệm hàm nhằm hiện thực hóa nhu cầu này của chúng ta. Nhìn chung ta có thể hình dung hàm là một hình chữ nhật như sau



Ta thấy thường ta sẽ cung cấp dữ liệu đầu vào cho hàm, sau đó hàm sẽ tính toán theo yêu cầu. Trong quá trình này, hàm có thể tự định nghĩa thêm các biến phụ trợ bên trong nếu cần thiết. Cuối cùng hàm sẽ cho ra một kết quả như mong đợi.

Có nhiều trường hợp hàm không có đầu vào hoặc không có đầu ra, nhưng không có trường hợp không có đầu vào lẫn đầu ra. Ngoài ra các biến phụ trợ cũng có thể không cần trong những trường hợp tính toán đơn giản.

2 Các loại tham số dùng cho hàm

Phần này trình bày ví dụ đơn giản về việc làm tròn số thực:

Dòng	
1	<code>#include <stdio.h></code>
2	<code>#include <math.h></code>
3	
4	<code>double LamTron(double);</code> // Khai bao ham
5	
6	<code>double LamTron(double x){</code> // Dinh nghia ham
7	<code>double kq;</code>
8	<code>if(x >= 0) kq = floor(x + 0.5);</code>
9	<code>else kq = -floor(-x + 0.5);</code>
10	<code>return kq;</code>
11	<code>}</code>
12	
13	<code>void main(){</code>
14	<code>double a, y;</code>
15	<code>printf("Nhap so can lam tron: ");</code>
16	<code>scanf("%lf", &a);</code>

17	y = LamTron(a); // Goi ham
18	printf(“Ket qua sau khi lam tron: %lf\n” , y);
19	}

Trong ví dụ này ta thấy việc làm tròn số không thực hiện trong hàm main, thay vào đó công việc này đã được tách ra và làm trong một hàm có tên là ‘LamTron’. Dòng 4 gọi là khai báo hàm và dòng 6 gọi là định nghĩa hàm. Trước khi định nghĩa hàm ta cần phải khai báo hàm trước. Sau khi ta đã khai báo và định nghĩa hàm cần thiết, thì ở dòng 16 ta chỉ cần gọi lại hàm đã định nghĩa để sử dụng.

Ta phân tích kĩ hơn về hàm ‘LamTron’ dòng số 6. Hàm này tên là ‘LamTron’, việc đặt tên hàm tuân thủ giống như qui tắc đặt tên biến. Hàm này có một tham số đầu vào kiểu double tên là x, và x này chỉ là tên đại diện dùng trong hàm hay còn gọi là tham số hình thức (đây cũng chính là dữ liệu đầu vào của hàm này). Bên trong hàm ‘LamTron’, ta có khai báo thêm biến trung gian kq để chứa kết quả tính toán (đây cũng chính là các biến trung gian do hàm tự khai báo định nghĩa riêng), và cuối cùng ở dòng lệnh số 10 là nơi ta trả về kết quả sau quá trình tính toán của hàm ‘LamTron’.

Tiếp theo ta phân tích hàm main. Trong hàm này ta thấy khai báo hai biến a và y. Biến a là biến chứa dữ liệu do người dùng nhập vào thông qua hàm scanf. Biến này được dùng để truyền vào hàm ‘LamTron’ khi ta gọi ở dòng 17. Ta gọi biến a lúc này là tham số thực, còn x mà ta đã làm trong định nghĩa hàm ‘LamTron’ là tham số hình thức. Hai tham số này là một, ta có thể hiểu giá trị x trong ‘LamTron’ chính là giá trị a lúc ta gọi hàm. Đặc biệt ta dùng biến y để chứa kết quả trả về của dòng mã số 10.

Thông qua ví dụ này ta có thể biết cách tách vấn đề lớn ra thành các bài toán nhỏ và dùng các hàm để giải quyết. Sau đó trong hàm main ta sẽ tập hợp lại các hàm con này để cùng nhau giải quyết bài toán lớn hơn.

2.1 Tham số giá trị (tham trị)

Trong ví dụ trên, biến x trong hàm ‘LamTron’ được gọi là tham trị. Ta chú ý dòng mã số 17 khi gọi hàm ‘LamTron’, ta gọi là truyền tham trị (vì biến x là tham trị trong hàm ‘LamTron’). Khi truyền tham trị thì biến x và biến a có những vùng nhớ khác nhau, điểm chung duy nhất của hai biến này là giá trị. Hay nói cách khác khi truyền tham trị thì giá trị của biến a sẽ được sao chép qua biến x, mọi thay đổi của biến x trong hàm ‘LamTron’ không gây ảnh hưởng gì tới biến a bên trong hàm ‘main’. Ta có thể thử nghiệm bằng cách trong hàm main ta in thử giá trị biến a sẽ thấy không hề thay đổi (Sinh viên tự thử nghiệm).

2.2 Tham số dạng tham chiếu (tham chiếu)

Ta sẽ dùng ví dụ trên với một ít hiệu chỉnh để thấy sự khác biệt.

Dòng	
1	<code>#include <stdio.h></code>
2	<code>#include <math.h></code>
3	
4	<code>double LamTron(double&); // Khai báo hàm</code>
5	
6	<code>void LamTron(double& x){ // Định nghĩa hàm</code>
7	<code>if(x >= 0) x = floor(x + 0.5);</code>
8	<code>else x = -floor(-x + 0.5);</code>
9	<code>}</code>
10	
11	<code>void main(){</code>
12	<code>double a;</code>
13	<code>printf("Nhập số cần làm tròn: ");</code>
14	<code>scanf("%lf", &a);</code>
15	<code>LamTron(a); // Gọi hàm</code>
16	<code>printf("Kết quả sau khi làm tròn: %lf\n", a);</code>
17	<code>}</code>

Hoàn toàn giống nhau về chức năng, nhưng lúc nào biến x trong hàm ‘LamTron’ là tham chiếu. Vì vậy mọi sự thay đổi của biến x trong hàm này sẽ ảnh hưởng tới biến a trong hàm main vì lúc này a và x có cùng chung một vùng nhớ. Do đó lúc này vai trò của đầu ra hàm không cần thiết nữa nên ta chuyển thành kiểu void, và dĩ nhiên cũng sẽ loại bỏ biến y trong hàm ‘main’.

3 Biến tĩnh cục bộ

Trong phần này, ta sẽ tìm hiểu khái niệm và cách dùng biến tĩnh cục bộ. Giống như biến cục bộ, biến tĩnh cục bộ cũng được khai báo và định nghĩa trong một hàm nào đó. Tuy nhiên với biến cục bộ thì giá trị và vùng nhớ của nó sẽ bị hủy khi hàm thực thi dòng lệnh cuối cùng. Trong ví dụ về hàm ‘LamTron’ phiên bản đầu tiên, biến x sẽ bị hủy giá trị và vùng nhớ khi dòng lệnh số 11 thực thi xong (Dĩ nhiên hàm ‘LamTron’ phải được gọi). Với biến tĩnh cục bộ thì nó sẽ duy trì vùng nhớ và giá trị cho tới khi **chương trình kết thúc**, như vậy thời gian sống của nó dài hơn biến cục bộ thông thường. Biến cục bộ tĩnh được định nghĩa chỉ một lần duy nhất khi nó được khai báo trong hàm, những lần sau khi hàm được gọi thì dòng định nghĩa sẽ được bỏ qua.

Dòng	
------	--

1	#include <stdio.h>
2	
3	void TichLuy(double n){
4	//Khai bao bien tinh cuc bo sum
5	static double sum = 0;
6	sum += n;
7	return sum;
8	}
9	
10	void main(){
11	TichLuy(2);
12	TichLuy(3.5);
13	double kq = TichLuy(4.1);
14	printf("Ket qua sau qua trinh tich luy: %lf", kq);
15	}

Trong đoạn mã trên, ta thấy hàm TichLuy được gọi ba lần nhưng dòng mã số 5 chỉ được thực thi một lần duy nhất. Như vậy dòng lệnh số 14 sẽ in ra giá trị của tổng $2 + 3.5 + 4.1 = 9.6$

4 Bài tập

Sinh viên làm các bài tập sau:

- 4.1 Viết chương trình in ra tất cả các ước số của n (n do người dùng nhập vào)
- 4.2 Viết chương trình in ra tất cả các ước số nguyên tố của n (n do người dùng nhập vào)
- 4.3 Viết chương trình in ra tất cả các số nguyên tố nhỏ hơn n (n do người dùng nhập vào)
- 4.4 Viết chương trình in ra tất cả các số đối xứng nhỏ hơn n (n do người dùng nhập vào)
- 4.5 Viết một hàm tính k! với k nguyên dương bất kỳ, nhập n, k ($n \geq k \geq 0$) từ bàn phím, sử dụng hàm đó tính số tổ hợp chập k của n theo công thức:

$$C_n^k = \frac{n!}{k!(n-k)!}$$

- 4.6 Viết chương trình, thực hiện những công việc sau: Xây dựng hàm Power(x, n) để tính lũy thừa x^n của số thực x bất kỳ với n nguyên dương. Sử dụng hàm Power(x, n) để tính:

$$S = 1.5^8 - \frac{(x+1)}{1^2} + \frac{(x+2)^2}{2^2} - \frac{(x+3)^3}{3^2} + \dots + (-1)^n \frac{(x+n)^n}{n^2}$$

- 4.7 Sử dụng hàm tính giai thừa và lũy thừa để tính gần đúng

$$e^x \approx S = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

4.8 Viết chương trình giải phương trình bậc hai, trong đó có sử dụng hàm tính delta.

4.9 Viết chương trình giải hệ phương trình hai ẩn (các hệ số a, b, c, d, e, f được nhập vào) có sử dụng hàm tính định thức.

4.10 Viết chương trình thực hiện các công việc sau:

- Viết hàm đọc một số (nhận tham số là số có một chữ số, in ra màn hình chuỗi tương ứng với số đó, ví dụ: 1 → “một”)
- Nhập vào một số nguyên có 3 chữ số, xuất ra màn hình chuỗi của việc đọc số này.
- Nhập vào một số nguyên n có giá trị nhỏ hơn 1000000, xuất ra màn hình chuỗi của việc đọc số này.

4.11 Nhiệt độ F(Fahrenheit), và nhiệt độ C(Celcius) liên hệ với nhau theo công thức:

$$C = \frac{5(F - 32)}{9}$$

Hãy viết chương trình cho phép người dùng nhập vào độ F hay độ C và đổi sang độ còn lại.