Chương III

HÀM VÀ TỔ CHỨC CHƯƠNG TRÌNH

3.1. Mở đầu

Một chương trình viết trong C là một dãy các hàm trong đó có một hàm chính là hàm **main()**.

Hàm là một đoạn chương trình độc lập, giải quyết một công việc hoàn chỉnh và có thể được sử dụng nhiều lần trong chương trình.

Hàm chia các bài toán lớn thành các công việc nhỏ hơn giúp cho việc thực hiện một công việc lặp lại nào đó một cách nhanh chóng mà không cần viết lại mã lệnh chương trình.

Thứ tự hàm trong chương trình là bất kỳ, song chương trình luôn thực hiện bắt đầu từ hàm **main()**.

3.2. Định nghĩa hàm

Cú pháp tổng quát để định nghĩa hàm như sau:

```
Kiểu_trả_về  tên_hàm (kiểu và danh_sách_tham_số)
{
        /* thân hàm */
        Các_câu_lệnh ;
        return giá_trị ;
}
```

- ***Kiểu_trả_về*** sẽ chỉ ra kiểu của kết quả cần trả về của hàm. Nếu hàm không cần trả về kết quả thì ***kiểu_trả_về*** sẽ là ***void***.

- **Tên_hàm** được đặt tên theo quy tắc định danh.

3.2. Định nghĩa hàm

- Danh_sách_tham_số: Tham số của hàm là phương tiện để truyền dữ liệu cần thiết từ bên ngoài vào trong hàm và từ trong hàm ra bên ngoài. Nếu có nhiều tham số thì chúng phải cách nhau bởi dấu phẩy và phải khai báo riêng biệt nhau.

- Câu lệnh **return** dùng để kết thúc việc thực hiện của một hàm (nếu hàm có giá trị trả về), trả kết quả và chuyển quyền điều khiển về nơi gọi hàm. Giá trị kết quả này phải có kiểu phù hợp với **kiểu_trả_về** đã được khai báo ở dòng tiêu đề. Cú pháp tổng quát của lệnh **return**:

return biểu_thức;

Lưu ý: *C không cho phép các hàm lồng nhau, nghĩa là phần định nghĩa của hàm này phải độc lập hoàn toàn với hàm khác.*

ĐỊNH NGHĨA HÀM DÙNG LỆNH: *#define*

Trong một số trường hợp, định nghĩa hàm dùng lệnh ***#define*** sẽ đơn giản hơn. Cú pháp:

#define Tên_hàm(Các_tham_số) Biểu_thức_Giá_trị

VÍ DỤ:

```
#define  SUM(x, y)  (x + y) //Tổng của hai số
#define  SQR(x)  (x*x) // Bình phương của một số
#define  MAX(x, y)  (x > y) ? x : y //Tìm số lớn nhất của 2 số
```

3.3. Khai báo nguyên mẫu hàm

Một khai báo nguyên mẫu hàm sẽ cung cấp cho trình biên dịch mô tả về một hàm sẽ được định nghĩa ở một vị trí nào đó trong chương trình.

Cú pháp tổng quát của một khai báo nguyên mẫu hàm:

Kiểu_trả_về tên_hàm **(*kiểu và*** danh_sách_tham_số**);**

3.4. Gọi hàm

Cú pháp gọi hàm:

```
Tên_hàm (danh_sách_đối_số) ;
```

Cặp dấu ngoặc () bắt buộc phải có cho dù hàm có đối số hay là không.

Trong **danh_sách_đối_số** không đưa ra kiểu dữ liệu của đối số. Nếu hàm cần truyền nhiều đối số thì chúng phải tách nhau bởi dấu phẩy.

Ví dụ 1: Hàm không có giá trị trả về

```
1  #include<stdio.h>
2  void line();// Khai bao nguyen mau
3  void line()/*Dinh nghia ham line*/
4  {
5      for(int i=0;i<19;i++)
6      printf("*");//In ra 19 dau *
7      printf("\n");
8  }
9  main()
10 {
11     line();//Goi ham line
12     printf("* Minh hoa ve ham *\n");
13     line();// Goi ham line
14 }
```

Kết quả khi chạy chương trình :

```
--------------------------------------------------------------------------------
*******************
* Minh hoa ve ham *
*******************
--------------------------------------------------------------------------------
```

Ví dụ 2: Hàm có giá trị trả về

```
1  #include<stdio.h>
2  unsigned long gt(int n);// Khai bao nguyen mau
3  unsigned long gt(int n)/*Dinh nghia ham line*/
4  {
5      unsigned long giai_thua = 1;
6      for(int i=1;i<=n;i++)
7      giai_thua*= i; return giai_thua;
8  }
9  main()
10 {
11     int x;
12     printf("Nhap vao mot so: ");
13     scanf("%d",&x);
14     if(x>=0)
15     printf("%d! = %lu",x,gt(x));
16 }
```

Kết quả khi chạy chương trình :

```
Nhap vao mot so: 6
6! = 720
```

3.5 Phạm vi của biến

Phạm vi của biến: là những vùng trong chương trình mà biến có thể được sử dụng.

Thời gian sống của biến: là khoảng thời gian biến tồn tại trong bộ nhớ. Mỗi biến sau khi khai báo sẽ được cấp phát một vùng nhớ để lưu trữ nó. Đến một thời điểm nào đó, vùng nhớ này sẽ bị thu hồi lại để lưu trữ biến khác.

Biến sử dụng gồm hai loại là:

- **Biến toàn cục :** khai báo bên ngoài tất cả các hàm và tác dụng trong toàn bộ chương trình.

- Biến cục bộ : được khai báo trong thân hàm hoặc khối lệnh và chỉ tác dụng trong phạm vi khai báo.

3.6 Hàm đệ quy

- Hàm đệ quy là hàm mà từ một điểm trong thân của nó có thể gọi tên hàm của chính nó.
- Khi hàm gọi đệ quy chính nó thì mỗi lần gọi, máy sẽ tạo ra một tập biến cục bộ mới hoàn toàn độc lập với tập biến cục bộ đã được tạo ra từ các lần gọi trước đó.
- Số lần gọi đệ quy phải có giới hạn, tức là việc gọi đệ quy phải có điểm dừng.

Xây dựng hàm đệ quy

Hàm đệ quy thường được viết theo dạng sau:

```
if (trường_hợp_cơ_sở)
{ /*giải quyết bài toán và trả về kết quả*/
}
else /*trường_hợp_tổng_quát */
{
//gọi đệ quy tới hàm với đối số truyền vào có giá trị khác
}
```

- **Phần không đệ quy**: bao gồm các trường hợp cơ sở có thể giải quyết trực tiếp, mà không cần đến bài toán con nào cả. Phần này sẽ quyết định tính dừng của thuật toán.
- **Phần đệ quy**: đây là trường hợp tổng quát. Bài toán được quy về một bài toán cùng dạng nhưng nhỏ hơn và có giá trị tham số thay đổi

Ví dụ: Hàm đệ quy tính n!

```
1  #include<stdio.h>
2  unsigned long gt(int n)
3  {
4      if(n==0||n==1)return 1;
5      else return n*gt(n-1);
6  }
7  main()
8  {
9      int x;
10     printf("Nhap vao mot so nguyen duong: ");
11     scanf("%d",&x);
12     if(x>=0)
13     printf("%d! = %lu",x,gt(x));
14 }
```

Kết quả chạy chương trình:

```
--------------------------------------------------------------------------------
Nhap vao mot so nguyen duong: 7
7! = 5040

--------------------------------------------------------------------------------
```

3.7 Một số hàm toán học sẵn có trong C

Tên hàm	Khai báo	Ý nghĩa
rand()	stdlib.h	Cho 1 giá trị ngẫu nhiên từ 0 đến 32767
random(x)	stdlib.h	Cho 1 giá trị ngẫu nhiên từ 0 đến x
pow(x,y)	math.h	Tính x mũ y
sqrt(x)	math.h	Tính căn bậc 2 của x
sin(x), cos(x), tan(x)	math.h	Tính sin, cosin, tang của góc x có số đo x radian
abs(a)	stdlid.h	Cho giá trị tuyệt đối của số nguyên a
labs(a)	stdlid.h	Cho giá trị tuyệt đối của số nguyên dài a
exp(x)	math.h	Tính e mũ x
log(x)	math.h	Tính logarit cơ số e của x
log10(x)	math.h	Tính logarit cơ số 10 của x
ceil(x)	math.h	Phần nguyên nhỏ nhất không nhỏ hơn x
floor(x)	math.h	Phần nguyên lớn nhất không lớn hơn x

Bài tập luyện tập

Bài 1: Viết hàm kiểm tra một số có phải số chính phương hay không ? hàm trả về giá trị là 1 nếu số được kiểm tra là số chính phương, ngược lại hàm trả về giá trị 0. Từ đó liệt kê các số chính phương trong khoảng (m;n).

Bài 2: Viết hàm kiểm tra một số có phải số nguyên tố hay không ? hàm trả về giá trị là 1 nếu số được kiểm tra là số nguyên tố, ngược lại hàm trả về giá trị 0.Từ đó liệt kê các số nguyên tố trong khoảng (m;n).

Bài 3: Viết chương trình trong đó chứa một hàm để tính giai thừa của một số tự nhiên. Trong hàm main() in ra màn hình hệ số của số hạng thứ k trong khai triển nhị thức Newton : $\mathbf{(a+b)^{100}}$?

Bài tập luyện tập

Bài 4: Viết chương trình xuất ra màn hình các hệ số của các số hạng khi khai triển nhị thức Newton của $(1+x)^n$ với n nguyên dương nhập từ bàn phím.

Bài 5: Viết hàm đệ quy tính các biểu thức sau :

a) S = 1+2+3+…+n

b) P = x*x*x*…*x (n lần số x)

c) Q = 5+10+15+…+5*n

Bài 6: Tính giá trị lớn nhất và nhỏ nhất của 4 số ?

Bài 7: Xuất ra màn hình tam giác **Pascal** với chiều cao n.

Bài 8: Viết hàm in ra số dạng máy tính bỏ túi với các tham số hàm từ 0 đến 9.