

Chapter 1. Introduction to C programming language

NỘI DUNG

1. Một số khái niệm cơ bản
2. Lệnh nhât/xuất, lệnh gán
3. Các phép toán cơ sở
4. Giới thiệu một số môi trường lập trình
5. Ví dụ & câu hỏi thảo luận

1. MỘT SỐ KHÁI NIỆM CĂN BẢN

Ngôn ngữ lập trình

- Ngôn ngữ lập trình được dùng để lập trình máy tính, tạo ra các chương trình máy tính phục vụ cho một mục đích cụ thể nào đó.
- Các ngôn ngữ lập trình minh họa cho học phần này (C-Free 5, Dev-C++, CodeBlocks, python,...)
- Các ngôn ngữ lập trình sinh viên có thể nghiên cứu để đáp ứng chương trình đào tạo đại học ngành công nghệ thông tin như Java, Python, C#,...

Cấu trúc chung của một chương trình

1. Khai báo sử dụng các thư viện chuẩn, các thư viện do người dùng định nghĩa.
2. Khai báo các hằng, biến, kiểu dữ liệu mới,...
3. Khai báo tiêu đề các hàm
4. Chương trình chính
5. Viết chi tiết cho các hàm

```
#include<iostream.h>
#define maxn 10000
int tong(int a, int b); // tiêu đề hàm
int main()
{
    int a,b;
    cout<<"Nhập a = ";cin>>a;
    cout<<"Nhập b = ";cin>>b;
    cout<<"Tổng của 2 số là : "<<tong(a,b);

    cout<<endl; // "\n"
    return 0;
}
int tong(int a, int b)
{
    return a+b;
}
```

Cách ghi chú thích trong chương trình

Khi viết chương trình nên đưa vào các dòng chú thích để chương trình dễ đọc, dễ hiểu. Điều này cũng giúp cho việc sửa đổi nâng cấp chương trình về sau được thuận lợi. Có hai cách sau để ghi lời chú thích:

Cách 1: đặt dấu // trước dòng muốn chú thích

Cách 2: bao đoạn lệnh cần chú thích giữa hai dấu /* ... */

Lời chú thích có thể đặt ở bất kỳ vị trí nào trong chương trình. Nếu chương trình dài phức tạp nên đưa chú thích vào trước mỗi đoạn lệnh hoặc trước mỗi chương trình con.

Kiểu dữ liệu

- Một kiểu dữ liệu là một tập hợp các giá trị mà một dữ liệu thuộc kiểu dữ liệu đó có thể nhận được.
Ví dụ các kiểu dữ liệu số nguyên, số thực, ký tự, chuỗi,...
- Trên một kiểu dữ liệu, xác định một số phép toán đối với các dữ liệu thuộc kiểu dữ liệu tương ứng.
- Người lập trình có thể tạo ra một kiểu dữ liệu mới; và khi đó cũng cần tạo ra các phép toán trên kiểu dữ liệu đó.

Dữ liệu kiểu số nguyên

| Từ khóa | Kích thước | Miền giá trị |
|---------------|------------|---------------------------|
| unsigned int | 16 bits | 0 .. 65535 |
| int | 16 bits | -32768 .. 32767 |
| unsigned long | 32 bits | 0 .. 4294967295 |
| long | 32bits | -2147483648 .. 2147483647 |

Ghi chú:

Thông số này từ môi trường BC3.1. Để biết mỗi biến chiếm bao nhiêu byte bộ nhớ sử dụng hàm sizeof(<kiểu dữ liệu>)

Các phép toán số học đối với số nguyên

| Toán tử | Ý nghĩa | Ví dụ |
|---------|------------------------|--------------|
| + | cộng | |
| - | trừ | |
| * | nhân | |
| / | chia lấy thương nguyên | $5 / 2 = 2$ |
| % | chia lấy số dư | $6 \% 4 = 2$ |

(các phép toán này đã được liệt kê theo độ ưu tiên)

Dữ liệu kiểu số thực

| Từ khoá | Kích thước | Miền giá trị của trị tuyệt đối |
|-------------|------------|--|
| float | 32 bits | $3.4 * (10^{-38})$ đến $3.4 * (10^{38})$ |
| double | 64 bits | $1.7 * (10^{-308})$ đến $1.7 * (10^{308})$ |
| long double | 80 bits | $3.4 * (10^{-4932})$ đến $1.1 * (10^{4932})$ |

Trên miền số thực không có phép toán chia lấy dư

Chuyển đổi kiểu dữ liệu

- Chẳng hạn với a, b là hai số nguyên. Cần xuất giá trị của phân số a / b thì cần phải chuyển đổi dữ liệu của kết quả để có kết đúng.
- Có bốn cách sau đây để thực hiện: **(float)a / b, float(a) / b, a / float(b);**
- Riêng trường hợp nếu a hoặc b là hằng số thì việc chuyển a hoặc b thành số thực có thể thực hiện bằng cách ghi thêm .0 phía sau hằng đó (ví dụ **1 / b** ghi thành **1.0 / b**).

Hằng, biến, biểu thức, hàm

- Khái niệm:
 - Hằng (*constant*) là đại lượng có *giá trị không đổi trong chương trình*.
- Khai báo hằng:
 - `#define <tên hằng> <giá trị hằng>` hoặc là:
 - `const <kiểu dữ liệu> <tên hằng> = giá trị hằng;`
- Một số hằng thường dùng:
“\n”, “\t”, “\0”, endl, INT_MIN, INT_MAX, M_PI,...

biến

- **Biến:**
 - Biến là đại lượng mà *giá trị có thể thay đổi trong chương trình.*
- **Khai báo biến:**
 - Kiểu dữ liệu <danh sách các biến>;

biểu thức

- Biểu thức là sự kết hợp các các toán hạng (*operand*) và các toán tử (*operator*) theo một quy tắc xác định.
- Các toán hạng có thể là biến, hằng, hàm...
- Các toán tử rất đa dạng: cộng, trừ, nhân, chia,...

Hàm/hàm chuẩn/hàm tự tạo

- Hàm có chức năng
 - Nhận dữ liệu đầu vào (*các tham số vào*)
 - Thực hiện một công việc nào đó
 - Trả về kết quả ứng với tham số truyền vào
- Hàm không có giá trị trả về (trả về void)
- Hàm có giá trị trả về (trả về khác void)
- Hàm chuẩn và hàm tự tạo
- Một số hàm chuẩn

2. LỆNH NHẬP/XUẤT, LỆNH GÁN

+Lệnh nhập dữ liệu với scanf ()

scanf (<các hằng chuỗi định dạng>, <địa chỉ biến 1>, <địa chỉ biến 2>, . . .);

+Lệnh xuất dữ liệu với printf()

printf (<các hằng chuỗi định dạng>, <biểu thức 1>, <biểu thức 2>, . . .);

+Lệnh gán

tên biến = <biểu thức>;

Hằng chuỗi định dạng chứa mã định dạng sau đây:

d : số nguyên hệ 10

ld : số long

c : ký tự

s : chuỗi ký tự

f : số thực float

lf : số thực double

+Lệnh nhập dữ liệu với cin

cin >> <biến 1> [>> <biến 2>] [...];

+Lệnh xuất dữ liệu với cout

cout << <biểu thức1> [<<biểu thức2>] [...]

Ghi chú liên quan câu lệnh cout

- int setw(int n);

Quy định độ rộng trên màn hình để xuất giá trị là n cột.

- int setprecision(int n);

Quy định số lượng chữ số sau dấu chấm thập phân cần xuất ra màn hình là n

3. CÁC PHÉP TOÁN CƠ SỞ

Toán tử so sánh

| toán tử | định nghĩa |
|---------|-------------------|
| < | nhỏ hơn |
| > | lớn hơn |
| == | bằng |
| <= | nhỏ hơn hoặc bằng |
| >= | lớn hơn hoặc bằng |
| != | khác nhau |

Toán tử logic

| a | b | !a | a && b | a b |
|---|---|----|--------|-------|
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |

Toán tử tăng, giảm

- Toán tử tăng `++` sẽ cộng 1 vào toán hạng của nó, toán tử giảm `--` sẽ trừ đi 1.
- Dấu phép toán `++` và `--` có thể đứng trước hoặc đứng sau toán hạng, như vậy có thể có tất cả bốn cách viết sau: `++n`, `n++`, `--n`, `n--`.
- Sự khác nhau giữa `++ a` và `a ++` ở chỗ: trong khi phép `n ++` thì `n` được tăng sau khi giá trị của nó đã được tính trong biểu thức, còn trong phép `++n` thì `n` được tăng trước khi giá trị của nó được tính trong biểu thức.
- Sự khác nhau giữa `--n` và `n--` cũng tương ứng như vậy.

Toán tử điều kiện

(<biểu thức điều kiện> ? <biểu thức 1> : <biểu thức 2>)

Ý nghĩa:

Nếu <biểu thức điều kiện> là đúng thì trả về giá trị của <biểu thức 1> ngược lại trả về giá trị của <biểu thức 2>.

VD: Để tìm giá trị lớn nhất của hai số a, b thì có thể viết đoạn lệnh như sau: `max = a > b ? a : b ;`

Thứ tự ưu tiên của các toán tử phép gọi hàm

!

/ , * , \ , %, &&

+, -, ||

< , >, ==, <=, >=, !=

4. MÔI TRƯỜNG LẬP TRÌNH

Introduction to a
programming
environment

- CFREE
- DEV C (<iostream>/ chỉ dẫn using namespace std;)
- CodeBlocks (#include<bits/stdc++.h>/chỉ dẫn using namespace std;)
- Python
- Java

Một số hàm chuẩn về kiểu dữ liệu số

- o double sqrt(double x) // can bac 2
- o double pow(double x, double y) // luy thua x^y
- o int abs(int x) // lay tri tuyet doi so nguyen
- o double fabs(double x) // so thuc
- o long int labs(long int x) // so nguyen long
- o void srand(time(NULL)) // khai tao bo so ngau nhien
- o unsigned int rand() // sinh so ngau nhien
- o log2(n); ví dụ log2(32)=5

5. VÍ DỤ & CÂU HỎI THẢO LUẬN

Ví dụ 1

Cho biết kết quả của các đoạn lệnh sau:

1)

```
int a =2, b = 3;
```

```
int m = a++ > b-- ? --a : b++;
```

```
cout << m+a+b;
```

2)

```
int a = 5, b = 9;
```

```
int m = a>b ? a++ : b++;
```

```
int n = a<b ? --a : --b;
```

```
cout << --n + (++m);
```

3)

```
#include<iostream.h>
int main()
{
    int a=5,b=4;
    cout<<a+++b<<endl; // equivalent: (a++)+b
    cout<<a+b<<endl;
    int c=5,d=4;
    cout<<++c+(++d)++<<endl;
    cout<<c+d<<endl;
    return 0;
}
```

Ví dụ 2

Nhập một số nguyên dương n có ba chữ số. Hãy tính tổng các chữ số của số đó.

Hướng dẫn:

Tìm cách tách các chữ số của số n, sau đó tính tổng các chữ số đó: chữ số đơn vị = $n \% 10$, chữ số hàng chục = $n \% 100 / 10$, chữ số hàng trăm = $n / 100$.

Một số bộ test:

| | test1 | test2 | test3 | test4 |
|---------|-------|-------|-------|-------|
| n | 197 | 135 | 111 | 100 |
| Kết quả | 17 | 9 | 3 | 1 |

Q.Nhập vào t giây, hãy đổi t giây ra dạng giờ – phút – giây. Ví dụ: t=3700 thì kết quả là : 1h 1m 40s

Ví dụ 3

Trong mặt phẳng tọa độ OXY cho ba điểm A, B, C lần lượt có tọa độ là $(x_a, y_a), (x_b, y_b), (x_c, y_c)$. Hãy tìm tọa độ trọng tâm, diện tích đường tròn nội tiếp, diện tích đường tròn ngoại tiếp của tam giác ABC .

Note: $S = p.r$; trong đó P là nửa chu vi, S là diện tích tam giác ABC ; $R = (a \times b \times c) : 4S; \Rightarrow r, R; \Rightarrow \dots$