

Chương IV

CÁC THÀNH PHẦN CƠ BẢN VÀ CÁC KIỂU DỮ LIỆU CỦA C

CBGD: ThS. Trần Anh Dũng

cuu duong than cong. com

DANH HIỆU

- Danh hiệu là tên của hằng, biến, hàm... hoặc các ký hiệu đã được quy định đặc trưng cho một thao tác nào đó.
- Danh hiệu có hai loại:
 - ❑ Ký hiệu.
 - ❑ Danh hiệu: Từ khóa và danh hiệu.

CBGD: ThS. Trần Anh Dũng

2

DANH HIỆU

- **Ký hiệu (symbol):** là các dấu đã được C quy định để biểu diễn cho một thao tác nào đó.

ký hiệu đơn → Một dấu biểu diễn một thao tác

dấu + đặc trưng cho phép cộng

dấu - đặc trưng cho phép trừ

dấu * đặc trưng cho phép nhân

dấu / đặc trưng cho phép chia

dấu % đặc trưng cho phép lấy số dư của phép chia nguyên

dấu = đặc trưng cho phép gán

dấu > đặc trưng cho phép so sánh lớn hơn

...

CBGD: ThS. Trần Anh Dũng

3

cuu duong than cong. com

DANH HIỆU

- **Ký hiệu (symbol):** là các dấu đã được C quy định để biểu diễn cho một thao tác nào đó.

ký hiệu kép → Hai dấu biểu diễn một thao tác

dấu == đặc trưng cho phép so sánh bằng

dấu /* bắt đầu cho một ghi chú

dấu */ kết thúc ghi chú

dấu >= đặc trưng cho phép so sánh lớn hơn hay bằng

dấu ++ đặc trưng cho phép tăng trị lên một

dấu && đặc trưng cho phép and luận lý

...

CBGD: ThS. Trần Anh Dũng

4

cuu duong than cong. com

DANH HIỆU

- **Danh hiệu (Identifier):** là các từ khóa của ngôn ngữ hoặc tên của các hằng, biến, hàm trong C.

Từ khóa (keyword) là các danh hiệu mà C đã định nghĩa sẵn cho lập trình viên sử dụng để thiết kế chương trình, tập các từ khóa của C sẽ được liệt kê trong phần phụ lục. **VD:** *if, for, while, ...*

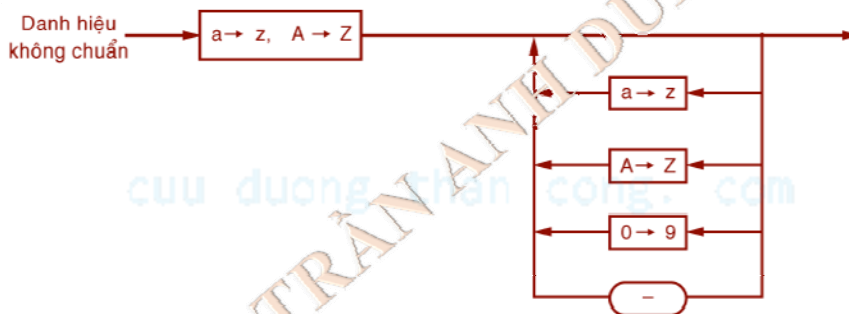
Danh hiệu là tên của các hằng, biến, hàm... Nếu các hằng, biến, hàm ... này do C đã khai báo và thiết kế sẵn thì các danh hiệu có được gọi là các danh hiệu chuẩn.

Nếu các hằng, biến, hàm ... này do lập trình viên khai báo và định nghĩa trong quá trình thiết kế chương trình thì các danh hiệu đó được gọi là các danh hiệu không chuẩn.

CBGD: ThS. Trần Anh Dũng

5

DANH HIỆU



Chú ý: Một danh hiệu có thể bắt đầu bằng dấu gạch dưới

CBGD: ThS. Trần Anh Dũng

6

DANH HIỆU

Ví dụ 4.6:

Xét các danh hiệu sau:

Main là danh hiệu không chuẩn hợp lệ

- bắt đầu là danh hiệu không hợp lệ vì có dấu trừ

2 tháng 9 là danh hiệu không hợp lệ vì có số bắt đầu

kết thúc là danh hiệu không hợp lệ vì có khoảng trắng

CBGD: ThS. Trần Anh Dũng

Ví dụ 4.7:

Xét hai danh hiệu sau

kết_thúc_vòng_lập_in_ra_ký_tự_khoảng_trắng

kết_thúc_vòng_lập_in_ra_ký_tự_k

7

cuu duong than cong. com

DANH HIỆU

Chiều dài một danh hiệu không bị hạn chế, mỗi bộ dịch C sẽ có quy định về chiều dài danh hiệu khác nhau, đối với các bộ dịch C/C++ thì danh hiệu có thể dài tùy ý, tuy nhiên trong các bộ dịch Borland C/C++ có quy định một giá trị xác định số ký tự đầu có nghĩa để phân biệt sự giống nhau và khác nhau giữa hai danh hiệu. Trong Turbo C 2.0, giá trị này là 31, trong Borland C++ 5.02, giá trị này là 55.

CBGD: ThS. Trần Anh Dũng

Ví dụ 4.7:

Xét hai danh hiệu sau

kết_thúc_vòng_lập_in_ra_ký_tự_khoảng_trắng

kết_thúc_vòng_lập_in_ra_ký_tự_k

8

cuu duong than cong. com

CÁC KIỂU DỮ LIỆU CHUẨN CỦA C

KIỂU	KÍCH THƯỚC	TẦM TRỊ BIỂU DIỄN
char	8 bit	128 .. + 127
int	16 bit	32768 .. + 32767
float	32 bit	3,4E37 .. 3,4E+38
double	64 bit	1.7E307.. 1.7E+308

CBGD: ThS. Trần Anh Dũng

9

CÁC KIỂU DỮ LIỆU CHUẨN CỦA C

1- Kiểu char

char là kiểu nguyên một byte, kiểu này có thể được sử dụng để khai báo biến, biến đó sẽ chiếm kích thước trong bộ nhớ là 1 byte và có thể giữ một ký tự hoặc một giá trị 8 bit. Mỗi bộ dịch C sẽ có quy định khác nhau về tầm trị của kiểu **char**, đối với bộ dịch TURBO C VERSION 2.0 kiểu **char** là kiểu có dấu.

Ví dụ 4.8:

```
#include <stdio.h>

main()
{
    char c;
    c = 'a';
    printf ("Ký tự trong biến c là %c ", c);
}
```

Ví dụ 4.9:

```
#include <stdio.h>

main()
{
    char c;
    c = 89;
    printf ("Trị trong biến c là %d ", c);
}
```

CBGD: ThS. Trần Anh Dũng

10

CÁC KIỂU DỮ LIỆU CHUẨN CỦA C

2- Kiểu int

Kiểu **int** là một kiểu số nguyên, kiểu này có thể được sử dụng để khai báo biến, biến đó có kích thước trong bộ nhớ là kích thước của số nguyên mà máy quy định, đối với máy PC, là các máy vi tính đang phổ biến hiện nay, và bộ dịch Borland C/C++ thì chiều dài của kiểu **int** là 16 bit có dấu, như vậy một biến hay hằng thuộc kiểu này có tầm trị biểu diễn từ -32768 đến 32767 (tức từ -2^{15} đến $2^{15} - 1$).

Xét chương trình ví dụ sau

Ví dụ 4.10:

```
#include <stdio.h>
main()
{
    int i;
    i = 1234;
    i = i + 123;
    printf ("Trị trong biến i là %d ", i);
}
```

Có thể dùng chuỗi định dạng xuất nhập "%d" để xuất hay nhập trị cho hằng, biến hoặc biểu thức kiểu **int**.

CBGD: ThS. Trần Anh Dũng

11

CÁC KIỂU DỮ LIỆU CHUẨN CỦA C

3- Kiểu float và double

float là kiểu số thực dấu chấm động, có độ chính xác đơn (7 ký số sau dấu chấm thập phân), **double** là kiểu số thực, dấu chấm động, có độ chính xác kép (15 ký số sau dấu chấm thập phân). Khi dùng các kiểu này khai báo cho biến thì biến có thể lưu trị là số thực mà tầm trị đã được cho trong bảng trên.

Kiểu **double** còn có thể được khai báo là **long float**, do đó khi khai báo **double b**; thì cũng hoàn toàn tương đương với **long float b**;

Để xuất nhập cho hằng, biến, biểu thức **float** chuỗi định dạng được sử dụng là "%f" đối với kiểu **double** thì chuỗi định dạng là "%lf" cho các hàm **printf** và **scanf**.

CBGD: ThS. Trần Anh Dũng

12

CÁC KIỂU DỮ LIỆU CHUẨN CỦA C

Ví dụ 4.12: Viết chương trình nhập hai số thực lớn hơn 0 bất kỳ, sau đó tính lũy thừa của số đầu đối với số sau. Yêu cầu có kiểm tra và báo lỗi.

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

main()
{
    double x, y, luy_thua;
    clrscr();
    printf("Moi nhap 2 so:");
    scanf ("%lf %lf" &x, &y);
    if (x < 0 && (y - (int)y != 0))
        printf("Ban da nhap sai tri");
    else
    {
        luy_thua = pow (x, y);
        printf ("Luy thua cua %5.2lf voi %5.2lf la %5.2lf", x, y, luy_thua);
    }
}
```

CBGD: ThS. Trần Anh Dũng

13

cuu duong than cong. com

CÁC KIỂU DỮ LIỆU CHUẨN CỦA C

Kiểu \ Dạng	signed	unsigned	short	long
Char	signed char → char	unsigned char	x	x
Int	signed int → int	unsigned int → unsigned	short int → int/short	long int → long
Float	x	x	x	long float → double
Double	x	x	x	long double

signed short int → short int
 signed long int → long
 unsigned short int → unsigned short
 unsigned long int → unsigned long

CBGD: ThS. Trần Anh Dũng

14

CÁC KIỂU DỮ LIỆU CHUẨN CỦA C

Bảng 4.2: Các kiểu dữ liệu, tầm trị và kích thước

Kiểu	Kích thước	Tầm trị biểu diễn
unsigned char	8 bit	0..255
Char	8 bit	128..+ 127
unsigned short	16 bit	0..65535
Short	16 bit	32768..+ 32767
Unsigned	16 bit	0..65535
Int	16 bit	32768..+ 32767
unsigned long	32 bit	0.. 4294967295
Long	32 bit	2147483648..+ 2147483647
Float	32 bit	3.4 E 37..3.4 E + 38
Double	64 bit	1.7 E 307..1.7 E + 308
long double	80 bit	3.4 E 4932..1.1 E + 4932

CBGD: ThS. Trần Anh Dũng

15

cuu duong than cong. com

HẲNG

Hằng là những giá trị cố định có trị hoàn toàn xác định và không thể thay đổi được chúng trong quá trình thực thi chương trình.

- Hằng số
- Hằng ký tự
- Chuỗi ký tự
- Biểu thức hằng

CBGD: ThS. Trần Anh Dũng

16

ThS.

ANH DUNG

cuu duong than cong. com

HẲNG

1. Hằng số:

- Hằng số nguyên: integer, long integer
- Hằng số thực.

Lưu ý 1: Khi sử dụng hằng số nguyên vượt quá tầm quy định.

có giá trị đi từ -32768 đến 32767

Ví dụ 4.17: Xét chương trình sau:

```
#include <stdio.h>
main()
{
    printf ("%d %d %d", 32767, 32767 + 1, 32767 + 2);
}
```

Chương trình này sẽ in ra màn hình ba trị sau:

32767 -32768 -32767

CBGD: ThS. Trần Anh Dũng

17

cuu duong than cong. com

HẲNG

Ví dụ 4.18:

```
#include <stdio.h>
#include <conio.h>
main()
{
    clrscr();
    printf ("%ld %ld %ld", 32767L, 32768L, 32769L);
    getch();
}
```

Biến kiểu long integer

CBGD: ThS. Trần Anh Dũng

18

HÀNG

Lưu ý 2:

- Các hằng số được viết không có dấu thập phân hoặc số mũ, sẽ được hiểu là nguyên và được lưu trữ theo kiểu int, ngược lại sẽ được lưu trữ theo kiểu double.
- Các hằng số nguyên lớn hơn khả năng một int được tự động lưu trữ theo kiểu long.
- Các hằng số nguyên lớn hơn một long được lưu trữ theo kiểu double.

CBGD: ThS. Trần Anh Dũng

19

cuu duong than cong. com

HÀNG

- 2. Hằng ký tự:** Hằng ký tự biểu diễn một giá trị ký tự đơn, ký tự này phải được viết giữa cặp dấu nháy đơn ("), mỗi ký tự có một mã số tương ứng trong bảng mã ký tự của máy, bình thường là mã ASCII.

Ví dụ 4.24: 'a' / '9' là những hằng ký tự

Ví dụ 4.25: 'A' có mã là 65 trong bảng mã ASCII.

'0' có mã là 48 (0×30) trong bảng mã ASCII.

CBGD: ThS. Trần Anh Dũng

20

ThS.

cuu duong than cong. com

HÀNG

Bảng 4.3: Chuỗi thoát của các ký tự không hiển thị được

CHUỖI	GIÁ TRỊ	KÝ TỰ ĐƯỢC HIỂU	TÁC DỤNG
\a	0 x 07	BEL	Phát tiếng còi
\b	0 x 08	BS	Xóa ký tự bên trái
\f	0 x 0C	FF	Sang trang
\n	0 x 0A	LF	Xuống dòng
\r	0 x 0D	CR	Enter
\t	0 x 09	HT	Tab theo cột
\v	0 x 0B	VT	Tab theo hàng
\\	0 x 5C	\	Dấu backslash
\'	0 x 2C	'	Dấu nháy đơn
\"	0 x 22	"	Dấu nháy kép
\?	0 x 3F	?	Dấu chấm hỏi
\ddd	ddd	Ký tự có mã theo có số 3 là ddd trong bảng mã ASCII	
\x IIII *	0 x IIII	Ký tự có mã theo có số 16 là IIII trong bảng mã ASCII	

CBGD: ThS. Trần Anh Dũng

21

cuu duong than cong. com

HÀNG

- 3. Chuỗi ký tự:** Trong ngôn ngữ C, một chuỗi ký tự là một loạt các ký tự nằm trong cặp dấu nháy kép (" "); các ký tự này có thể là ký tự được biểu diễn bằng chuỗi thoát.

Ví dụ 4.28:

"Một chuỗi ký tự"

"Chuỗi ký tự có chuỗi thoát: i can't go to school \n"

CBGD: ThS. Trần Anh Dũng

22

ThS. Trần Anh Dũng

HẰNG

Trong bộ nhớ, chuỗi ký tự đang được sử dụng được lưu trữ tại địa chỉ bắt đầu xác định như một dãy ký tự liên tiếp tận cùng bằng một ký tự kết thúc chuỗi, ký tự NUL (tức ký tự '\0', có mã số là 0 trong bảng mã ASCII).

Ví dụ 4.29: chuỗi String được lưu trữ trong bộ nhớ như sau:

S	t	r	i	n	g	\0
---	---	---	---	---	---	----

CBGD: ThS. Trần Anh Dũng

23

cuu duong than cong. com

HẰNG

4. Biểu thức hằng:

- Một biểu thức được xem là một biểu thức hằng nếu giá trị của biểu thức hoàn toàn xác định, như vậy một biểu thức toán học là một biểu thức hằng khi trong biểu thức đó các toán hạng đều là những hằng số hoặc hằng ký tự.
- Khi đó biểu thức hằng sẽ được chương trình biên dịch tính trước ra một trị bằng số xác định và trị này được ghi vào chương trình đã dịch từ chương trình nguồn.

$1 < 8$ sẽ được tính trước và được ghi là 1 (true)

CBGD: ThS. Trần Anh Dũng

24

BIẾN

1. Khai báo biến:

- Tất cả các biến được sử dụng trong một chương trình C đều phải được khai báo trước.
 - Việc khai báo này giúp cho chương trình biên dịch có thể biết được kích thước của biến đó, vị trí của chúng trong bộ nhớ và sự tồn tại của chúng trong chương trình, khi muốn sử dụng biến ta chỉ cần gọi tên biến
- Lưu ý:** tên biến phải là một danh hiệu không chuẩn hợp lệ

CBGD: ThS. Trần Anh Dũng

25

cuu duong than cong. com

BIẾN

1. Khai báo biến:

Ví dụ 4.33:

bat_dau	}	đều là các danh hiệu hợp lệ
if		
_hoten		
Thu_thang_1_68		
printf	}	đều là các danh hiệu không hợp lệ
pow		
31_thang_12		
ket thuc		

CBGD: ThS. Trần Anh Dũng

26

ThS.

cuu duong than cong. com

BIẾN

1. Khai báo biến:

- C là ngôn ngữ nhạy cảm với chữ hoa và chữ thường, do đó nếu hai tên biến hợp lệ khác nhau ở kiểu chữ hoa hoặc thường thì hai biến đó là khác nhau.

BIEN	}	<i>đều là những biến khác nhau</i>
BiEN		
Bien		
biEN		
bien		

CBGD: ThS. Trần Anh Dũng

27

cuu duong than cong. com

BIẾN

1. Khai báo biến:

Cú pháp khai báo biến

`kiểu dsach_tenbien:`

Giải thích:

<i>kiểu</i>	kiểu của các biến cần khai báo
<i>dsach_tenbien</i>	danh sách liệt kê các tên biến cần khai báo, các biến cách nhau bằng dấu ","

Ví dụ 4.35:

```
int lap, count, max;
double he_so_1, he_so_2, delta;
```

CBGD: ThS. Trần Anh Dũng

28

BIẾN

1. Khai báo biến:

- Biến của một chương trình C có thể được khai báo ở một trong ba vị trí sau:

- Ngoài tất cả các hàm (gọi là khai báo biến ngoài), khi đó ta có biến toàn cục.

Ví dụ 4.37:

```
#include <stdio.h>
int a, b;           ← khai báo biến ngoài
main()
{ ...
    a = 10;
    b = a + 24;
    ...
}
```

CBGD: ThS. Trần Anh Dũng

29

cuu duong than cong. com

BIẾN

1. Khai báo biến:

- Biến của một chương trình C có thể được khai báo ở một trong ba vị trí sau:

- Đầu phần thân của một hàm hoặc một khối lệnh (gọi là khai báo biến trong), khi đó ta có biến cục bộ.

Ví dụ 4.38:

```
int tong()
{
    int i, tam;      ← khai báo biến trong
    ...
    for (i = 10; ...
    ...
}
```

CBGD: ThS. Trần Anh Dũng

30

cuu duong than cong. com

BIẾN

1. Khai báo biến:

- Biến của một chương trình C có thể được khai báo ở một trong ba vị trí sau:

- Trong phần định nghĩa đối số của hàm (gọi là đối số hàm hoặc tham số hàm).

Ví dụ 4.39:

```
int luy_thua (int n, char ket_qua)
{
    ...
    for (i = 1; ...
    ...
}
```

↑ ↑
đối số hàm

CBGD: ThS. Trần Anh Dũng

31

cuuduongthancong.com

BIẾN

1. Khai báo biến:

Một biến có thể được khởi động trị ngay sau khi khai báo bằng phép gán với giá trị tương ứng, khi đó cú pháp khai báo biến như sau:

kiểu biến1 = tri1, biến2 = tri2;

với các tri1 và tri2 phải là những giá trị đã xác định.

Ví dụ 4.40:

```
double he_so_1 = 20.37;
char ky_tu = 'A', ky_tu_moi = ky_tu;
```

CBGD: ThS. Trần Anh Dũng

32

THA

cuuduongthancong.com

BIẾN

2. Các kiểu bỏ túc kiểu const và volatile:

a. Từ khóa const:

Khi được khai báo cho biến thì nó xác định rằng biến sẽ không bị thay đổi trị trong suốt quá trình thực thi chương trình, mọi sự thay đổi trị đều gây ra lỗi, biến đó ta gọi là biến hằng.

Cú pháp:

```
const    kiểu tên_biến [= trị được thay thế];
```

Ví dụ 4.41:

```
const double bat_dau = 3.1415;
const int max = 100;
```

CBGD: ThS. Trần Anh Dũng

33

cuu duong than cong. com

BIẾN

2. Các kiểu bỏ túc kiểu const và volatile:

a. Từ khóa const:

Nếu kiểu của biến hằng không nêu cụ thể thì biến hằng đó sẽ thuộc loại int, ngay cả nếu trị được thay thế là một trị khác int thì chỉ phần nguyên được sử dụng và lưu vào biến hằng mà thôi.

Ví dụ 4.42:

```
const max = 100;
const pi = 3.14; khi đó biến pi chỉ là 3 mà thôi
```

CBGD: ThS. Trần Anh Dũng

34

ThS

cuu duong than cong. com

BIẾN

2. Các kiểu bỏ túc kiểu const và volatile:

b. Từ khóa volatile:

Từ khóa volatile: chỉ ra rằng một biến có thể bị thay đổi trị từ một tác nhân không nằm trong chương trình, Từ khóa này làm cho biến của C có một tính linh động rất cao, ví dụ như biến của C có thể thay đổi theo đồng hồ hệ thống hay theo một chương trình nền nào đó.

Cú pháp:

```
volatile <tên_biến>;
```

CBGD: ThS. Trần Anh Dũng

35

BIỂU THỨC

Biểu thức là một sự kết hợp của các toán hạng là các biến, hằng hoặc phép gọi hàm bằng các toán tử xác định của C để tạo ra được một trị, trị này có thể được sử dụng hoặc không được sử dụng tùy nhu cầu của lập trình viên.

Ví dụ 4.43: Đối với C, một biểu thức như sau là hợp lệ:

```
a = (x = 10) - (y = a + 1) * ((b += 1) > 12);
```

CBGD: ThS. Trần Anh Dũng

36

CÁC PHÉP TOÁN CỦA C

1 Toán tử số học

C có các toán tử số học bình thường giữa 2 toán hạng, đó là

- Toán tử cộng (+) : thực hiện phép toán cộng
- Toán tử trừ (-) : thực hiện phép toán trừ
- Toán tử nhân (*) : thực hiện phép nhân
- Toán tử chia (/) : thực hiện phép chia
- Toán tử modulo (%): thực hiện phép toán lấy số dư của phép chia nguyên

Ví dụ

float a, b, c;

double y; y = a * b - c;

Ví dụ

b = - b;

a = + b;

Ví dụ

int a = 10, b = 3, c;

c = a/b;

c = a % b;

c = 3

c = 1

double x = 10., y = 3, z;

z = x/y;

z = 3.33333...

Phép nhân và chia các số nguyên sẽ chỉ cho kết quả nguyên (C tự động cắt bỏ phần thập phân nếu có).

CBGD: ThS. Trần Anh Dũng

37

cuuduongthancong.com

CÁC PHÉP TOÁN CỦA C

Khi thực hiện các phép toán số học, một vấn đề đặt ra là nếu có nhiều toán hạng khác kiểu nhau thì C sẽ thực hiện việc tính toán biểu thức ra sao?

→ C sẽ thực hiện việc chuyển kiểu tự động theo quy luật sau: toán hạng thuộc kiểu có trị nhỏ hơn sẽ được chuyển sang kiểu có trị lớn hơn.

Ví dụ

double a = 10.;

int b = 3;

thì phép toán a/b sẽ đổi giá trị lấy từ b (3) thành số double (3.00...) trước, rồi thực hiện phép toán 10./3.0, kết quả (3.33 ...) sẽ là số double.

CBGD: ThS. Trần Anh Dũng

38

CÁC PHÉP TOÁN CỦA C

2 Toán tử quan hệ

- Toán tử bằng (==) : chỉ ra mối quan hệ bằng giữa hai toán hạng
- Toán tử khác (!=) : chỉ ra mối quan hệ khác giữa hai toán hạng
- Toán tử lớn hơn (>) : chỉ ra mối quan hệ lớn hơn giữa hai toán hạng
- Toán tử nhỏ hơn (<) : chỉ ra mối quan hệ nhỏ hơn giữa hai toán hạng
- Toán tử lớn hơn hoặc bằng (>=) : chỉ ra mối quan hệ lớn hơn hoặc bằng giữa hai toán hạng
- Toán tử nhỏ hơn hoặc bằng (<=) : chỉ ra mối quan hệ nhỏ hơn hoặc bằng giữa hai toán hạng

CBGD: ThS. Trần Anh Dũng

39

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

Khi mỗi quan hệ giữa hai toán hạng theo toán tử quan hệ trong biểu thức là **ĐÚNG** → biểu thức đó sẽ trả về một trị nguyên là **1**

Ngược lại mỗi quan hệ đó là **SAI** → biểu thức đó sẽ trả về một trị nguyên là **0**

Ví dụ

```
...
if (delta > 0)
{
    x1 = ...;
    x2 = ...;
    printf ("Nghiem cua phuong trinh la ...");
}
else
...
```

Ví dụ Xét các khai báo biến và biểu thức sau

```
int a, b, c;
char kt;

a = 1;
b = 2;
c = -3;

kt = (a >= 4) * 2 + (b < 3) - c;
```

4

CBGD: ThS. Trần Anh Dũng

40

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

Cần lưu ý phép toán so sánh Bằng == thường bị quen đánh là "=", tức lệnh gán, điều này không bị chương trình biên dịch báo lỗi vì C vẫn hiểu được, nhưng khi đó biểu thức lại có một ý nghĩa khác, để hiểu rõ ta hãy xét ví dụ sau.

Ví dụ Xét hai trường hợp sau:

1/ $a = (b == 4);$

2/ $a = (b = 4);$

Nếu b bằng 4, thì gán

Gán $b = 4$ (như vậy b đã có trị là 4),

$a = 1$, còn không $a = 0$

lấy trị đang có của b gán vào cho a ,

như vậy a luôn bằng 4

CBGD: ThS. Trần Anh Dũng

41

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

Ví dụ

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

```
#include <conio.h>
```

```
main()
```

```
{
```

```
int x, y;
```

```
clrscr();
```

```
x = 10;
```

```
y = 24;
```

```
printf ("Trị của biểu thức  $x == y$  là %d \n", x == y);
```

```
printf ("Trị của biểu thức  $x <= y$  là %d \n", x <= y);
```

```
printf ("Trị của biểu thức  $x >= y$  là %d \n", x >= y);
```

```
printf ("Trị của biểu thức  $x != y$  là %d \n", x != y);
```

```
if (x == y)
```

```
printf ("Trị của biểu thức  $x == y$  là true \n");
```

```
else
```

```
printf ("Trị của biểu thức  $x == y$  là false \n");
```

```
if (x = y)
```

```
printf ("Trị của biểu thức  $x = y$  là true \n");
```

```
else
```

```
printf ("Trị của biểu thức  $x = y$  là false \n");
```

```
printf ("Voi x là %d va y là %d \n", x, y);
```

```
getch();
```

```
}
```

CBGD: ThS. Trần Anh Dũng

42

cuu duong than cong. com

ThS.

CÁC PHÉP TOÁN CỦA C

Tri của bieu thuc $x == y$ la 0
 Tri của bieu thuc $x <= y$ la 1
 Tri của bieu thuc $x >= y$ la 0
 Tri của bieu thuc $x != y$ la 1
 Tri của bieu thuc $x == y$ la false
 Tri của bieu thuc $x = y$ la true
 Voi x la 24 va y la 24

CBGD: ThS. Trần Anh Dũng

43

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

3 Toán tử logic

C có các toán tử logic not (!), and (&&), or (||).

not !
and &&
or ||

Toán hạng 1	Toán hạng 2	Kết quả		
A	B	! A	A && B	A B
bằng 0	bằng 0	1	0	0
bằng 0	khác 0	1	0	1
khác 0	bằng 0	0	0	1
khác 0	khác 0	0	1	1

CBGD: ThS. Trần Anh Dũng

Các toán tử này có độ ưu tiên là not, and, or; nếu trong biểu thức các toán tử đều ngang cấp nhau thì thứ tự tính toán từ trái sang phải.

44

CÁC PHÉP TOÁN CỦA C

Cần lưu ý rằng toán hạng của các toán tử này có thể là một biểu thức hoặc giá trị bất kỳ nào và ở một trong hai trạng thái: "bằng 0" (false) hoặc "khác 0" (true). Và kết quả của các biểu thức có các toán tử logic này cũng là một giá trị int: 1 hoặc 0, các kết quả này cũng có thể được dùng để tính toán tiếp theo.

Ví dụ 1.54: trong mệnh đề if sau đây thay vì viết

```
if (delta == 0)
{
    ...
}
```

ta có thể viết gọn hơn như sau:

```
if (!delta)
{
    ...
}
```

thì nếu delta là một số bằng 0 thì biểu thức (!delta) sẽ có giá trị bằng 1, khi đó lệnh if sẽ được thực hiện, còn ngược lại, nếu delta khác 0, thì (!delta) có trị = 0, lệnh if sẽ không được thi hành.

CBGD: ThS. Trần Anh Dũng

45

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

Điều cần lưu ý là các phép toán and (&&) và or () được thực hiện từ trái sang phải và việc tính toán sẽ dừng lại ngay khi giá trị của cả biểu thức logic đã xác định trị mà không cần tính tiếp các phần còn lại của biểu thức nữa.

Ví dụ :

$(5 < 4) \&\& (5 < 10) \&\& (6 < 10) \&\& (6 > 5)$

CBGD: ThS. Trần Anh Dũng

46

cuu duong than cong. com
ThS. TRẦN

CÁC PHÉP TOÁN CỦA C

4 Toán tử trên bit

- NOT: \sim , thực hiện việc đảo bit, từ bit 0 qua 1 và ngược lại
- AND: $\&$, thực hiện việc and bit
- OR: $|$, thực hiện việc or bit
- XOR: \wedge , thực hiện việc xor bit
- Dịch trái: \ll , dịch các bit sang trái
- Dịch phải: \gg , dịch các bit sang phải

CBGD: ThS. Trần Anh Dũng

47

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

Bit	Bit	Phép toán			
		$\sim A$	$A \& B$	$A B$	$A \wedge B$
0	0	1	0	0	0
0	1	1	0	1	1
1	0	0	0	1	1
1	1	0	1	1	0

Ví dụ 1.58:

$a = 1030$ có mã nhị phân là 0000 0100 0000 0110

$b = 224$ có mã nhị phân là 0000 0000 1110 0000

thì

$\sim a$	1111	1011	1111	1001
$a \& b$	0000	0000	0000	0000
$a b$	0000	0100	1110	0110
$a \wedge b$	0000	0100	1110	0110

CBGD: ThS. Trần Anh Dũng

48

CÁC PHÉP TOÁN CỦA C

Phép toán AND (&) sẽ kết hợp với một trị mà ta gọi là mặt nạ che các bit dữ liệu không cần quan tâm, như vậy trong giá trị mặt nạ tại những vị trí cần che, bit sẽ có trị là 0, các bit còn lại là 1, và qua đó ta có thể đánh giá được dữ liệu đang làm việc một cách chính xác.

```
#define MASK 0x0200
```

9 ← vị trí

0000 0010 0000 0000

0 2 0 0

bit 9 không bị che

CBGD: ThS. Trần Anh Dũng

49

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

Ví dụ: Nhập vào một số nguyên, xét xem bit có vị trí là 9 có bằng 1 hay không.

```
#include <stdio.h>
#include <conio.h>
#define MASK 0x0200
main()
{ int a;
  clrscr();
  printf ("Moi nhap mot so: ");
  scanf ("%d", &a);
  if (a & MASK)
    printf ("Bit 9 bang 1 \n");
  else
    printf ("Bit 9 bang 0 \n");
  getch();
}
```

CBGD: ThS. Trần Anh Dũng

50



cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

Ví dụ : Nhập vào một số nguyên, xét xem bit có vị trí là 9 có bằng 1 hay không, nếu nó bằng 0 thì bật bit đó lên 1 và in ra kết quả.

```
#include <stdio.h>
#include <conio.h>
#define MASK 0x0200
main()
{ int a;
  clrscr();
  printf ("Moi nhap mot so: ");
  scanf ("%d", &a);
  if (a & MASK)
  {
    printf ("Bit 9 bang 1 \n");
    printf ("Tri cua bien a la %d \n", a);
  }
  else
  {
    a = a | MASK;
    printf ("Bit 9 bang 0 va da duoc bat len 1 \n");
    printf ("Tri cua bien a la %d \n", a);
  }
  getch();
}
```

CBGD: ThS. Trần Anh Dũng

51

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

Ví dụ : Nhập vào một số nguyên, xét xem bit có vị trí là 9 có bằng 1 hay không, nếu nó bằng 0 thì bật đó lên 1, nếu bằng 1 thì đưa về 0 và in ra kết quả.

```
#include <stdio.h>
#include <conio.h>
#define MASK 0x0200
main()
{ int a;
  clrscr();
  printf ("Moi nhap mot so: ");
  scanf ("%d", &a);
  if (a & MASK)
    printf ("Bit 9 bang 1 \n");
  else
    printf ("Bit 9 bang 0 \n");
  printf ("Tri cua bien a truoc khi dao bit 9 la %d \n", a);
  a = a ^ MASK;
  if (a & MASK)
    printf ("Bit 9 sau khi dao bang 1 \n");
  else
    printf ("Bit 9 sau khi dao bang 0 \n");
  printf ("Tri cua bien a sau khi dao bit 9 la %d \n", a);
  getch();
}
```

CBGD: ThS. Trần Anh Dũng

52

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

Toán tử dịch trái (<<) và dịch phải (>>) cho phép thực hiện việc dời các bit của toán hạng sang bên trái hoặc sang phải. Cú pháp như sau:

`biểu_thức_nguyên << số_bit_dời`
`biểu_thức_nguyên >> số_bit_dời`

CBGD: ThS. Trần Anh Dũng

53

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

Trong phép dịch trái, các bit ở bên phải của toán hạng sẽ được ghi vào các giá trị là 0.

Ví dụ :

Xét biến `n` được khai báo

`int n;`

đang có trị `n = 469` (tức `0x01d5`), biểu diễn trong bộ nhớ dưới dạng nhị phân là

0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Khi dịch trái 2 bit, ta có:

`n << 2`

0	0	0	0	0	1	1	1	0	1	0	1	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

CBGD: ThS. Trần Anh Dũng

54

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

Còn trong phép dịch phải, thì tùy theo kiểu dữ liệu của toán hạng bên trái mà ta có hai trường hợp sau:

- Nếu toán hạng bên trái có dữ liệu thuộc dạng **unsigned** (unsigned int, unsigned long, unsigned char) thì phép dịch phải sẽ ghi 0 vào các bit bên trái của kết quả.
- Còn nếu toán hạng bên trái có dữ liệu thuộc dạng **signed** (int, long, char) thì phép dịch phải sẽ ghi bit dấu vào các bit bên trái của kết quả.

CBGD: ThS. Trần Anh Dũng

55

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

Ví dụ 1.64: Nếu biến n được khai báo là

unsigned n;

Giả sử n đang lưu giá trị $n = 42569$ (tức $0xA649$), tức trong bộ nhớ ta có n

1	0	1	0	0	1	1	0	0	1	0	0	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Khi phép dịch phải 2 bit xảy ra ($n \gg 2$), kết quả như sau:

0	0	1	0	1	0	0	1	1	0	0	1	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

0 →

CBGD: ThS. Trần Anh Dũng

56

ThS.

CÁC PHÉP TOÁN CỦA C

Ví dụ 1.65: Nếu biến n được khai báo là

`int n;`

Giả sử n đang lưu giá trị $n = -32766$ (tức $0x8002$), tức trong bộ nhớ ta có n

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

bit dấu

Khi phép dịch phải 2 bit xảy ra ($n \gg 2$), kết quả như sau:

1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

1 →

CBGD: ThS. Trần Anh Dũng

57

CÁC PHÉP TOÁN CỦA C

5 Toán tử tăng giảm

Trong C có hai toán tử đặc biệt gọi là toán tử tăng ($++$) và toán tử giảm ($--$) dùng để tăng hoặc giảm một biến nào đó đi 1. Việc tăng giảm này theo kiểu mà biến được khai báo.

Cú pháp sử dụng:

$++$ biến

biến $++$

$--$ biến

biến $--$

Ví dụ

$++a;$ $a = a + 1;$
 $a++;$

CBGD: ThS. Trần Anh Dũng

58

CÁC PHÉP TOÁN CỦA C

```
int i = 12;
```

```
double b;
```

```
b = i++; (1)
```

```
b = i;
```

```
i = i + 1;
```

```
b = ++i;
```

```
i = i + 1;
```

```
b = i;
```

CBGD: ThS. Trần Anh Dũng

59

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

```
s = 0;
```

```
so = 1;
```

```
while (so <= n)
```

```
{
```

```
    s = s + so;
```

```
    so++;
```

```
}
```

```
while (so <= n)
```

```
    s = s + so++;
```

CBGD: ThS. Trần Anh Dũng

60

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

6 Toán tử gán

Gán đơn giản

biến = trị

Gán phức tạp

biến op = trị

biến = biến op trị

op có thể là * / % + - hoặc << >> & ^ |

Ví dụ

int a, b = 2;

a = 4; → a = 4

b *= a * 3; → b = 24

CBGD: ThS. Trần Anh Dũng

61

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

Kiểu toán hạng trái	Kiểu toán hạng phải	Trị có thể mất sau khi gán
signed char	unsigned char	Giá trị > 127, thành số âm
char	short int	Mất trị từ bit 8 trở đi
char	int	Mất trị từ bit 8 trở đi
char	long	Mất trị từ bit 8 trở đi
short int	long int	Mất 16 bit cao (một int)
int	float	Mất phần thập phân và phần trị lớn hơn một int
float	double	Độ chính xác đo làm tròn

CBGD: ThS. Trần Anh Dũng

Ví dụ 1.72:

int a = 4, b = 1;

b += (a = 2 * b) + (a *= b);

đầu tiên a được gán bằng 2 * b, như vậy a = 2, sau đó a tiếp tục được gán là a * b, vậy a = 2, do đó b = 5.

62

CÁC PHÉP TOÁN CỦA C

7 Toán tử phẩy - Biểu thức phẩy

việc tính toán biểu thức phẩy sẽ được thực hiện từ trái sang phải, kết quả của cả biểu thức phẩy sẽ là giá trị và kiểu của kết quả của biểu thức bên phải.

Cú pháp:

`biểu_thức_1, biểu_thức_kết_quả`

Với `biểu_thức_1` và `biểu_thức_kết_quả` là hai biểu thức bất kỳ.

Ví dụ

`m = (a = 2, t = a + 3);` sẽ cho `a = 2, t = 5` và `m = t = 5`

hoặc `x = (t = 1, t + 4);` sẽ cho `t = 1` và `x = 5`

CBGD: ThS. Trần Anh Dũng

63

CÁC PHÉP TOÁN CỦA C

8 Toán tử điều kiện - biểu thức điều kiện

`điều-kiện ? biểu-thức1 : biểu-thức2`

Ta gọi một biểu thức như vậy là biểu thức điều kiện. Để tính toán biểu thức điều kiện, điều kiện sẽ được tính toán trước, nếu khác 0, `biểu-thức1` sẽ được tính và được lấy làm kết quả, còn nếu bằng 0 thì kết quả sẽ là kết quả của `biểu-thức2`.

Ví dụ

`if (i > 0)`

`n = 1;`

`else n = 0;`

`n = (i > 0) ? 1 : 0;`

CBGD: ThS. Trần Anh Dũng

64

CÁC PHÉP TOÁN CỦA C

Ví dụ Viết chương trình nhập một ký tự, đổi ký tự đó sang ký tự hoa nếu đó là ký tự thường.

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

main()
{
    char c;
    clrscr();
    printf ("Nhap mot ky tu: ");
    c = getchar();
    c = ( c >= 'a' && c <= 'z' ) ? c - 32 : c;
    printf ("Ky tu da duoc doi la: ");
    putchar (c);
    getch();
}
```

CBGD: ThS. Trần Anh Dũng

65

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

9 Toán tử sizeof

Cho ta kích thước của một biến hoặc một kiểu dữ liệu nào đó.

→ không phải quan tâm đến chiều dài cụ thể của các biến.

```
sizeof (biến)
sizeof biến
sizeof (kiểu)
```

Ví dụ Nếu có một biến đã khai báo:

```
double f;

thì:    sizeof (f)
hoặc:   sizeof f
        sizeof (double)
```

sẽ cho ta kích thước của biến f hoặc của một số double nói chung, trong trường hợp này trị sẽ là 8 (byte).

CBGD: ThS. Trần Anh Dũng

66

cuu duong than cong. com

CÁC PHÉP TOÁN CỦA C

10 Độ ưu tiên của các toán tử

Độ ưu tiên	Phép toán	Thứ tự kết hợp
1	() [] ->	Trái qua phải
2	! ~ ++ -- ++ (type) * & size of	Phải qua trái *
3	* / %	Trái qua phải
4	+ -	Trái qua phải
5	<< >>	Trái qua phải
6	< <= > >=	Trái qua phải
7	== !=	Trái qua phải
8	&	Trái qua phải
9	^^	Trái qua phải
10		Trái qua phải
11	&&	Trái qua phải
12		Trái qua phải
13	?:	Phải qua trái *
14	= += -= *= /= %= <<= >>= &= = ^=	Phải qua trái *
15	,	Trái qua phải

CBGD: ThS. Trần Anh Dũng

67

cuu duong than cong. com

CẤU TRÚC TỔNG QUÁT CỦA MỘT CHƯƠNG TRÌNH C

Ví dụ Nhập một số kiểm tra số đó là chẵn hay lẻ.

```
#include <stdio.h>
#include <conio.h>
int kiem_tra (int so);

/* ham kiem_tra nhan vao doi so
   la mot so nguyen, tra ve tri
   - 0 la so chan
   - 1 la so le
*/

main()
{
    int n;
    clrscr();
    printf ("Nhap mot so: ");
    scanf ("%d", &n);
    if (kiem_tra(n))
        printf ("So da nhap la so le \n");
    else
        printf ("So da nhap la so chan \n");
    getch();
}

int kiem_tra (int so)
{
    return (so % 2 == 0)? 0:1;
}
```

→ phần khai báo

→ phần hàm

CBGD: ThS. Trần Anh Dũng

68

cuu duong than cong. com

BÀI TẬP

1. Viết chương trình in ra màn hình trị thập phân của các hằng sau đây

067 01234 0x1a1 0x89ad

0xfb 'h' 022 02365

2. Nhập ba số, tìm số lớn nhất và nhỏ nhất trong ba số đó.

3. Nhập bốn số, sắp xếp theo thứ tự từ lớn tới nhỏ và từ nhỏ tới lớn theo menu sau:

1. Từ lớn tới nhỏ

2. Từ nhỏ tới lớn

3. Kết thúc

Mời bạn chọn thao tác (1...3):

CBGD: ThS. Trần Anh Dũng

69

cuu duong than cong. com

BÀI TẬP

4. Nhập ba cạnh tam giác, kiểm tra ba cạnh đó có thỏa điều kiện hình thành tam giác không, in kết quả kiểm tra.

5. Nhập ba cạnh tam giác, kiểm tra ba cạnh đó có thỏa điều kiện hình thành tam giác không, nếu thỏa in ra kết quả xem tam giác đó là tam giác gì (vuông thì vuông tại đâu, cân thì cân tại đâu ...)?

6. Viết chương trình nhập một ký tự và xử lý theo yêu cầu sau:

- Nếu ký tự là hoa thì đổi sang thường, in kết quả đổi
- Nếu ký tự là thường thì không làm gì cả, in kết quả
- Nếu ký tự là ký số thì in ra màn hình câu: "Day la mot ky so".

7. Dùng hàm pow () để tính bình phương và lũy thừa ba của một số nhập từ bàn phím.

8. Viết chương trình đổi từ độ Fahrenheit (F) sang độ Celcius (C) theo công thức sau:

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

CBGD: ThS. Trần Anh Dũng

70