



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

## TIN HỌC ĐẠI CƯƠNG

### Bài 8. Kiểu dữ liệu và biểu thức trong C

#### Nội dung

- 1. Các kiểu dữ liệu chuẩn trong C
2. Khai báo và khởi tạo biến, hằng
3. Biểu thức trong C
4. Các phép toán trong C
5. Một số toán tử đặc trưng
6. Các lệnh vào ra dữ liệu với các biến

## 8.1. Các kiểu dữ liệu chuẩn trong C

Kiểu dữ liệu	Ý nghĩa	Kích thước	Miền dữ liệu
<b>unsigned char</b>	Kí tự không dấu	1 byte	0 ÷ 255
<b>char</b>	Kí tự có dấu	1 byte	-128 ÷ 127
<b>unsigned int</b>	Số nguyên không dấu	2 byte	0 ÷ 65.535
<b>int</b>	Số nguyên có dấu	2 byte	-32.768 ÷ 32.767

3

## 8.1. Các kiểu dữ liệu chuẩn trong C

Kiểu dữ liệu	Ý nghĩa	Kích thước	Miền dữ liệu
<b>unsigned long</b>	Số nguyên không dấu	4 byte	0 ÷ 4,294,967,295
<b>long</b>	Số nguyên có dấu	4 byte	-2,147,483,648 ÷ 2,147,483,647
<b>float</b>	Số thực dấu phẩy động, độ chính xác đơn	4 byte	± 3.4E-38 ÷ ± 3.4E+38
<b>double</b>	Số thực dấu phẩy động, độ chính xác kép	8 byte	± 1.7E-308 ÷ ± 1.7E+308

4

## Nội dung

1. Các kiểu dữ liệu chuẩn trong C
- 2. Khai báo và khởi tạo biến, hằng
3. Biểu thức trong C
4. Các phép toán trong C
5. Một số toán tử đặc trưng
6. Các lệnh vào ra dữ liệu với các biến

5

## 2.1. Khai báo và khởi tạo biến

- Một biến trước khi sử dụng phải được khai báo
- Cú pháp khai báo:  
**KieuDuLieu tenBien;**  
Hoặc:  
**KieuDuLieu tenBien<sub>1</sub>, ..., tenBien<sub>N</sub>;**
- Ví dụ: Khai báo một biến x thuộc kiểu số nguyên 2 byte có dấu (int), biến y, z, t thuộc kiểu thực 4 byte (float) như sau:

```
int      x;  
float    y, z, t;  
x = 3; y = x + 1;
```

6



## 2.2. Khai báo hằng

- Chú ý:
  - Giá trị của các hằng phải được xác định ngay khi khai báo.
  - Trong chương trình, **KHÔNG thể thay đổi** được giá trị của hằng.
  - #define là chỉ thị tiền xử lý (preprocessing directive)
    - Dễ đọc, dễ thay đổi
    - Dễ chuyển đổi giữa các nền tảng phần cứng hơn
    - Tốc độ nhanh hơn

9

## Nội dung

1. Các kiểu dữ liệu chuẩn trong C
2. Khai báo và khởi tạo biến, hằng
- ⇒ 3. Biểu thức trong C
4. Các phép toán trong C
5. Một số toán tử đặc trưng
6. Các lệnh vào ra dữ liệu với các biến

10

### 3. Biểu thức trong C

- **Biểu thức số học:**
  - Là biểu thức mà giá trị của nó là các đại lượng số học (số nguyên, số thực).
  - Các toán tử là các phép toán số học (cộng, trừ, nhân, chia...), các toán hạng là các đại lượng số học (số, biến, hằng).
  - Ví dụ:
    - $a, b, c$  là các biến thuộc một kiểu dữ liệu số nào đó.
    - $3 * 3.7$
    - $8 + 6/3$
    - $a + b - c...$

11

### 3. Biểu thức trong C

- **Biểu thức logic:**
  - Là biểu thức mà giá trị của nó là các giá trị logic, tức là một trong hai giá trị: Đúng (*TRUE*) hoặc Sai (*FALSE*).
    - Giá trị nguyên khác 0: Đúng (*TRUE*),
    - Giá trị 0: Sai (*FALSE*).
  - Các phép toán logic gồm có
    - AND: VÀ logic, kí hiệu là `&&`
    - OR: HOẶC logic, kí hiệu là `||`
    - NOT: PHỦ ĐỊNH, kí hiệu là `!`

12



### 3. Biểu thức trong C

- Ví dụ về biểu thức logic:

- `(5 > 7) && (9!=10)`      • // có giá trị logic là sai, FALSE
- `0 || 1`                      • // có giá trị logic là đúng, TRUE
- `(5 > 7) || (9!=10)`      • // có giá trị logic là đúng, TRUE
- `0`                              • // có giá trị logic là sai, FALSE
- `!0`                             • // phủ định của 0, có giá trị logic là đúng, TRUE
- `3`                              • // có giá trị logic là đúng, TRUE
- `!3`                             • // phủ định của 3, có giá trị logic là sai, FALSE
- `(a > b) && ( a < b)`      • // Có giá trị sai, FALSE. Giả sử a, b là 2 biến kiểu int

### 3. Biểu thức trong C

- Làm vế phải của lệnh gán.
- Làm toán hạng trong các biểu thức khác.
- Làm tham số thực trong lời gọi hàm.
- Làm chỉ số trong các cấu trúc lặp **for**, **while**, **do while**.
- Làm biểu thức kiểm tra trong các cấu trúc rẽ nhánh **if**, **switch**.



## Nội dung

1. Các kiểu dữ liệu chuẩn trong C
2. Khai báo và khởi tạo biến, hằng
3. Biểu thức trong C
- ⇒ 4. Các phép toán trong C
5. Một số toán tử đặc trưng
6. Các lệnh vào ra dữ liệu với các biến

17

## 4. Các phép toán trong C

- Bao gồm:
  - Nhóm các phép toán số học
  - Nhóm các phép toán thao tác trên bit
  - Nhóm các phép toán quan hệ
  - Nhóm các phép toán logic
  - Ngoài ra C còn cung cấp một số phép toán khác nữa như phép gán, phép lấy địa chỉ...

18

## 4.1. Phép toán số học

Toán tử	Ý nghĩa	Kiểu dữ liệu của toán hạng	Ví dụ
-	Phép đổi dấu	Số thực hoặc số nguyên	int a, b; -12; -a; -25.6;
+	Phép toán cộng	Số thực hoặc số nguyên	float x, y; 5 + 8; a + x; 3.6 + 2.9;
-	Phép toán trừ	Số thực hoặc số nguyên	3 - 1.6; a - 5;
*	Phép toán nhân	Số thực hoặc số nguyên	a * b; b * y; 2.6 * 1.7;
/	Phép toán chia	Số thực hoặc số nguyên	10.0/3.0; (bằng 3.33...) 10/3.0; (bằng 3.33...) 10.0/3; (bằng 3.33...)
/	Phép chia lấy phần nguyên	Giữa 2 số nguyên	10/3; (bằng 3)
%	Phép chia lấy phần dư	Giữa 2 số nguyên	10%3; (bằng 1)

## 4.1. Phép toán trên bit

Toán tử	Ý nghĩa	Kiểu dữ liệu của toán hạng	Ví dụ
&	Phép VÀ nhị phân	2 số nhị phân	0 & 0 (có giá trị 0) 0 & 1 (có giá trị 0) 1 & 0 (có giá trị 0) 1 & 1 (có giá trị 1) 101 & 110 (có giá trị 100)
	Phép HOẶC nhị phân	2 số nhị phân	0   0 (có giá trị 0) 0   1 (có giá trị 1) 1   0 (có giá trị 1) 1   1 (có giá trị 1) 101   110 (có giá trị 111)

## 4.2. Phép toán trên bit

^	Phép HOẶC CÓ LOẠI 2 số nhị phân TRỪ nhị phân	$0 \wedge 0$	(có giá trị 0)
		$0 \wedge 1$	(có giá trị 1)
		$1 \wedge 0$	(có giá trị 1)
		$1 \wedge 1$	(có giá trị 0)
		$101 \wedge 110$	(có giá trị 011)
<<	Phép DỊCH TRÁI nhị phân Số nhị phân	$a \ll n$	(có giá trị $a \cdot 2^n$ )
		$101 \ll 2$	(có giá trị 10100)
>>	Phép DỊCH PHẢI nhị phân Số nhị phân	$a \gg n$	(có giá trị $a/2^n$ )
		$101 \gg 2$	(có giá trị 1)
~	Phép ĐẢO BIT nhị phân (lấy Bù 1)	$\sim 0$	(có giá trị 1)
		$\sim 1$	(có giá trị 0)
		$\sim 110$	(có giá trị 001)

21

## 4.3. Phép toán quan hệ

Toán tử	Ý nghĩa	Ví dụ
>	So sánh lớn hơn giữa 2 số nguyên hoặc thực.	$2 > 3$ (có giá trị 0) $6 > 4$ (có giá trị 1) $a > b$
>=	So sánh lớn hơn hoặc bằng giữa 2 số nguyên hoặc thực.	$6 \geq 4$ (có giá trị 1) $x \geq a$
<	So sánh nhỏ hơn giữa 2 số nguyên hoặc thực.	$5 < 3$ (có giá trị 0),
<=	So sánh nhỏ hơn hoặc bằng giữa 2 số nguyên hoặc thực.	$5 \leq 5$ (có giá trị 1) $2 \leq 9$ (có giá trị 1)
==	So sánh bằng nhau giữa 2 số nguyên hoặc thực.	$3 == 4$ (có giá trị 0) $a == b$
!=	So sánh không bằng (so sánh khác) giữa 2 số nguyên hoặc thực.	$5 != 6$ (có giá trị 1) $6 != 6$ (có giá trị 0)

22

## 4.4. Phép toán logic

Toán tử	Ý nghĩa	Kiểu dữ liệu của toán hạng	Ví dụ
&&	Phép VÀ LOGIC. Biểu thức VÀ LOGIC bằng 1 khi và chỉ khi cả 2 toán hạng đều bằng 1	Hai biểu thức logic	$3 < 5 \ \&\& \ 4 < 6$ (có giá trị 1) $2 < 1 \ \&\& \ 2 < 3$ (có giá trị 0) $a > b \ \&\& \ c < d$
	Phép HOẶC LOGIC. Biểu thức HOẶC LOGIC bằng 0 khi và chỉ khi cả 2 toán hạng bằng 0.	Hai biểu thức logic	$6 \    \ 0$ (có giá trị 1) $3 < 2 \    \ 3 < 3$ (có giá trị 0) $x \geq a \    \ x == 0$
!	Phép PHỦ ĐỊNH LOGIC một ngôi. Biểu thức PHỦ ĐỊNH LOGIC có giá trị bằng 1 nếu toán hạng bằng 0 và có giá trị bằng 0 nếu toán hạng bằng 1	Biểu thức logic	$!3$ (có giá trị 0) $!(2 > 5)$ (có giá trị 1)

23

## 4.5. Phép toán gán

- Cú pháp  
***tenBien = biểu\_thức;***
- Lấy giá trị của *biểu\_thức* gán cho *tenBien*
- Giá trị của *biểu\_thức* phải có kiểu phù hợp với *tenBien*
- Ví dụ:  

```
int a, b, c;  
a = 3;  
b = a + 5;  
c = a * b;
```

24

## 4.5. Phép toán gán

- Biểu thức gán là biểu thức nên nó cũng có giá trị.
- Giá trị của biểu thức gán bằng giá trị của biểu\_thức:  
→ Có thể gán giá trị của biểu thức gán cho một biến khác hoặc sử dụng như một biểu thức bình thường
- Ví dụ:  

```
int a, b, c;  
a = b = 2007;  
c = (a = 20) * (b = 30);
```

25

## 4.5. Phép toán gán

- Phép toán gán thu gọn:  

```
x = x + y; giống như  
x += y;
```
- Dạng lệnh gán thu gọn này còn áp dụng được với các phép toán khác: +, -, \*, /, %, >>, <<, &, |, ^

26

## 4.6. Thứ tự ưu tiên các phép toán

Mức	Các toán tử	Trật tự kết hợp
1	() [] . -> ++ (hậu tố) -- hậu tố	----->
2	! ~ ++ (tiền tố) -- (tiền tố) - *	<-----
	& sizeof	
3	* / %	----->
4	+ -	----->
5	<< >>	----->
6	< <= > >=	----->
7	== !=	----->
8	&	----->
9	^	----->
10		----->
11	&&	----->
12		----->
13	?:	<-----
14	= += -=	<-----

## Nội dung

1. Các kiểu dữ liệu chuẩn trong C
2. Khai báo và khởi tạo biến, hằng
3. Biểu thức trong C
4. Các phép toán trong C
- ➔ 5. Một số toán tử đặc trưng
6. Các lệnh vào ra dữ liệu với các biến

## 5.1. Các phép toán tăng giảm một đơn vị

- Tăng 1 đơn vị:

```
tenBien++;
```

- Giảm 1 đơn vị

```
tenBien--;
```

- Ví dụ:

```
int a = 5;
```

```
float x = 10;
```

```
a ++; // tương đương với a = a + 1;
```

```
x --; // tương đương với x = x - 1;
```

29

## Tiền tố và hậu tố

- Tiền tố: Thay đổi giá trị của biến trước khi sử dụng
- Hậu tố: Tính toán giá trị của biểu thức bằng giá trị ban đầu của biến, sau đó mới thay đổi giá trị của biến
- Ví dụ:

```
int a, b, c;
```

```
a = 3; // a bằng 3
```

```
b = a++; // Dạng hậu tố
```

```
// b bằng 3; a bằng 4
```

```
c = ++a; // Dạng tiền tố
```

```
// a bằng 5, c bằng 5;
```

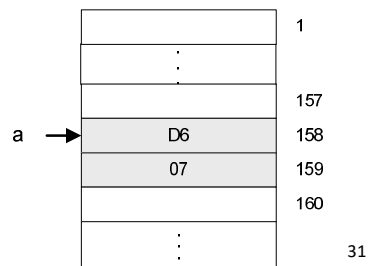
30

## 5.2. Phép toán lấy địa chỉ biến (&)

- Biến thực chất là một vùng nhớ được đặt tên (là tên của biến) trên bộ nhớ của máy tính.
- Mọi ô nhớ trên bộ nhớ máy tính đều được đánh địa chỉ. Do đó mọi biến đều có địa chỉ
- Cú pháp: **&tenBiến;**

– Ví dụ: `int a = 2006;`

→ `&a;` // có giá trị là 158 hay **9E**



## 5.3. Phép toán chuyển đổi kiểu bắt buộc

- Trong biểu thức chứa các toán hạng có kiểu dữ liệu khác nhau, chương trình dịch chuyển đổi kiểu dữ liệu theo nguyên tắc kiểu dữ liệu “**thấp hơn**” (dùng ít bit biểu diễn hơn) sang kiểu dữ liệu “**cao hơn**” (dùng nhiều bit biểu diễn hơn).
- Ép kiểu bắt buộc:

**(KiểuDữLiệu) tenBiến;**

- Tránh ép kiểu “cao” xuống “thấp”

```
int a = 1, b = 3;  
float c ;  
c = a/b; // c = ?
```

32



## 5.4. Biểu thức điều kiện

- Cú pháp

***biểu\_thức\_1 ? biểu\_thức\_2 : biểu\_thức\_3***

–Giá trị của biểu thức điều kiện

- Giá trị của biểu\_thức\_2 nếu biểu\_thức\_1 có giá trị khác 0 (tương ứng với giá trị logic ĐÚNG),
- Ngược lại: Giá trị của biểu\_thức\_3 nếu biểu\_thức\_1 có giá trị bằng 0 (tương ứng với giá trị logic SAI).

- Ví dụ:

```
float x, y, z; // khai báo biến
x = 3.8; y = 7.6; // gán giá trị cho các biến x, y
z = (x < y) ? x : y; // z sẽ có giá trị bằng giá trị
// nhỏ nhất trong 2 số x và y
```