

# TIN HỌC CƠ SỞ 2

# CÁC KIỂU DỮ LIỆU CƠ SỞ TRONG NGÔN NGỮ C

Đặng Ngọc Hùng

Khoa: Công nghệ thông tin 1



### Nội dung

- 1 Các kiểu dữ liệu cơ sở
- Biến, Hằng, Câu lệnh & Biểu thức
- 3 Các lệnh nhập xuất
- 4 Một số ví dụ minh họa



# Các kiểu dữ liệu cơ sở

- ❖ Ngôn ngữ C có 4 kiểu cơ sở như sau:
  - Kiểu số nguyên: giá trị của nó là các số nguyên như 2912, -1706, ...
  - Kiểu số thực: giá trị của nó là các số thực như
     3.1415, 29.12, -17.06, ...
  - Kiểu logic: giá trị đúng hoặc sai.
  - Kiểu ký tự: 256 ký tự trong bảng mã ASCII.



# Kiểu số nguyên

### ❖ Các kiểu số nguyên (có dấu)

■ n bit có dấu: -2<sup>n-1</sup> ... +2<sup>n-1</sup> - 1

Kiểu (Type)	Độ lớn (Byte)	Miền giá trị (Range)
char	1	-128 <b>+127</b>
int	2	-32.768 +32.767
short	2	-32.768 +32.767
long	4	-2.147.483.648 +2.147.483.647



# Kiểu số nguyên

- ❖ Các kiểu số nguyên (không dấu)
  - n bit không dấu: 0 ... 2<sup>n</sup> 1

Kiểu (Type)	Độ lớn (Byte)	Miền giá trị (Range)
unsigned char	1	0 255
unsigned int	2	0 65.535
unsigned short	2	0 65.535
unsigned long	4	0 4.294.967.295



### Kiểu số thực

### ❖ Các kiểu số thực (floating-point)

- Ví dụ
  - $17.06 = 1.706*10 = 1.706*10^{1}$

Kiểu (Type)	Độ lớn (Byte)	Miền giá trị (Range)
float (*)	4	3.4*10 <sup>-38</sup> 3.4*10 <sup>38</sup>
double (**)	8	1.7*10 <sup>-308</sup> 1.7*10 <sup>308</sup>

- (\*) Độ chính xác đơn (Single-precision) chính xác đến 7 số lẻ.
- (\*\*) Độ chính xác kép (Double-precision) chính xác đến 16 số lẻ.



## Kiểu logic

### ❖Đặc điểm

- C ngầm định một cách không tường minh:
  - false (sai): giá trị 0.
  - true (đúng): giá trị khác 0, thường là 1.
- C++: bool

- 0 (false), 1 (true), 2 (true), 2.5 (true)
- 1 > 2 (0, false), 1 < 2 (1, true)



## Kiểu ký tự

### ❖Đặc điểm

- Tên kiểu: char
- Miền giá trị: 256 ký tự trong bảng mã ASCII.
- Chính là kiểu số nguyên do:
  - Lưu tất cả dữ liệu ở dạng số.
  - Không lưu trực tiếp ký tự mà chỉ lưu mã ASCII của ký tự đó.

- Lưu số 65 tương đương với ký tự 'A'...
- Lưu số 97 tương đương với ký tự 'a'.



# Định nghĩa kiểu dữ liệu

```
Cú pháp
typedef <tên kiểu> <tên kiểu mới>;
```

- ❖ Ví dụ
  - typedef int SoNguyen;
  - → Có thể khai báo: SoNguyen x;

# & Biến

### Biến

```
Ví dụ
int i;
int j, k;
unsigned char dem;
float ketqua, delta;
```

```
Cú pháp
```

```
<kiểu> <tên biến>;
<kiểu> <tên biến 1>, <tên biến 2>;
```

# VC BB Biến

- Phải khai báo biến trước khi sử dụng.
- ❖Trong C/ C++ có thể khai báo biến ở:
  - Ngoài hàm
  - Đầu hàm
  - Tham số hàm
  - Trong chương trình



## Hằng số

### HĂNG

```
Cú pháp
#define <tênhằng> <giá trị>
```

```
Ví dụ
#define MAX 100
#define PI 3.14
```

```
// Không có = hay ;
```



### Biểu thức

#### ❖ Khái niệm

- Tạo thành từ các toán tử (Operator) và các toán hạng (Operand).
- Toán tử tác động lên các giá trị của toán hạng và cho giá trị có kiểu nhất định.
- Toán tử: +, -, \*, /, %....
- Toán hạng: hằng, biến, lời gọi hàm...
- ❖ Ví dụ
  - 2 + 3, a div 5, (a + b) \* 5, ...



### Toán tử gán

- ❖ Khái niệm
  - Thường được sử dụng trong lập trình.
  - Gán giá trị cho biến.
- Cú pháp
  - <bién> = <giá trị>;
  - <bién> = <bién>;
  - <bién> = <biéu thức>;
  - Có thể thực hiện liên tiếp phép gán.



### Toán tử gán

```
void main()
      int a, b, c, d, e, thuong;
      a = 10;
      b = a;
      thuong = a / b;
      a = b = c = d = e = 156;
      e = 156;
      d = e;
      c = d;
      b = c;
      a = b;
```



# Các toán tử toán học

### ❖Toán tử 1 ngôi

- Chỉ có một toán hạng trong biểu thức.
- ++ (tăng 1 đơn vị), -- (giảm 1 đơn vị)
- Đặt trước toán hạng
  - Ví dụ ++x hay --x: thực hiện tăng/giảm trước.
- Đặt sau toán hạng
  - Ví dụ x++ hay x--: thực hiện tăng/giảm sau.

- x = 10; y = x++; // y = 10 và x = 11
- x = 10; y = ++x; // x = 11 và y = 11



# Các toán tử toán học

### ❖Toán tử 2 ngôi

- Có hai toán hạng trong biểu thức.
- +, -, \*, /, % (chia lấy phần dư)
- $\mathbf{x} = \mathbf{x} + \mathbf{y} \Leftrightarrow \mathbf{x} + \mathbf{y}$

- a = 1 + 2; b = 1 2; c = 1 \* 2; d = 1 / 2;
- e = 1\*1.0 / 2; f = float(1) / 2; g = float(1 / 2);
- h = 1 % 2;
- $x = x * (2 + 3*5); \Leftrightarrow x *= 2 + 3*5;$



### Các toán tử trên bit

#### ❖ Các toán tử trên bit

- Tác động lên các bit của toán hạng (nguyên).
- & (and), | (or), ^ (xor), ~ (not hay lấy số bù 1)
- >> (shift right), << (shift left)</p>
- Toán tử gộp: &=, |=, ^=, ~=, >>=, <<=</p>

&	0	1
0	0	0
1	0	1

^	0	1
0	0	1
1	1	0

1	0	1
0	0	1
1	1	1

~	0	1
	1	0



### Các toán tử trên bit

```
void main()
      int a = 5; // 0000 0000 0000 0101
      int b = 6; // 0000 0000 0000 0110
      int z1, z2, z3, z4, z5, z6;
      z1 = a \& b; // 0000 0000 0000 0100
      z2 = a \mid b; // 0000 0000 0000 0111
      z3 = a ^b; // 0000 0000 0000 0011
      z4 = -a; // 1111 1111 1010
      z5 = a >> 2;// 0000 0000 0000 0001
      z6 = a \ll 2;// 0000 0000 0001 0100
```



### Các toán tử quan hệ

### ❖ Các toán tử quan hệ

- So sánh 2 biểu thức với nhau
- Cho ra kết quả 0 (hay false nếu sai) hoặc 1 (hay true nếu đúng)
- **=** ==, >, >=, <, <=, !=

#### ❖ Ví du

- s1 = (1 == 2); s2 = (1 != 2);
- s3 = (1 > 2); s4 = (1 >= 2);
- s5 = (1 < 2); s6 = (1 <= 2);



### Các toán tử logic

### Các toán tử logic

- Tổ hợp nhiều biểu thức quan hệ với nhau.
- && (and), || (or), ! (not)

&&	0	1
0	0	0
1	0	1

- 11	0	1
0	0	1
1	1	1

#### Ví dụ

- s1 = (1 > 2) && (3 > 4);
- s2 = (1 > 2) || (3 > 4);
- s3 = !(1 > 2);



### Toán tử điều kiện

#### ❖ Toán tử điều kiện

- Đây là toán tử 3 ngôi (gồm có 3 toán hạng)
- <biểu thức 1> ? <biểu thức 2> : <biểu thức 3>
  - <biểu thức 1> đúng thì giá trị là <biểu thức 2>.
  - <biểu thức 1> sai thì giá trị là <biểu thức 3>.

- s1 = (1 > 2) ? 2912 : 1706;
- int s2 = 0;
- 1 < 2 ? s2 = 2912 : s2 = 1706;



## Toán tử phẩy

### ❖Toán tử phẩy

- Các biểu thức đặt cách nhau bằng dấu ,
- Các biểu thức con lần lượt được tính từ trái sang phải.
- Biểu thức mới nhận được là giá trị của biểu thức bên phải cùng.

- x = (a++, b = b + 2);
- $\Leftrightarrow$  a++; b = b + 2; x = b;



# Độ ưu tiên của các toán tử

Toán tử	Độ ưu tiên
() [] -> .	$\rightarrow$
! ++ + * (cast) & sizeof	<del>&lt;</del>
* / %	$\rightarrow$
+ -	$\rightarrow$
<< >>	$\rightarrow$
< <= > >=	$\rightarrow$
== !=	$\rightarrow$
8	$\rightarrow$ $\rightarrow$
	$\rightarrow$
&&	$\rightarrow$
	$\rightarrow$
?:	<del>-</del>
= += -= *= /= %= &=	<del>`</del>
,	<del>-</del>



# Độ ưu tiên của các toán tử

#### Quy tắc thực hiện

- Thực hiện biểu thức trong () sâu nhất trước.
- Thực hiện theo thứ tự ưu tiên các toán tử.
- = => Tự chủ động thêm ()

#### ❖ Ví du

- n = 2 + 3 \* 5;
- => n = 2 + (3 \* 5);
- a > 1 && b < 2
- = => (a > 1) && (b < 2)



## Viết biểu thức cho các mệnh đề

x lớn hơn hay bằng 3

$$x >= 3$$

a và b cùng dấu

```
((a>0) && (b>0)) || ((a<0) && (b<0))
(a>0 && b>0) || (a<0 && b<0)
```

❖p bằng q bằng r

$$(p == q) \&\& (q == r) hoặc (p == q \&\& q == r)$$

$$(x > -5) \&\& (x < 5) hoặc (x > -5 \&\& x < 5)$$



### Câu lệnh

#### ❖ Khái niệm

- Là một chỉ thị trực tiếp, hoàn chỉnh nhằm ra lệnh cho máy tính thực hiện một số tác vụ nhất định nào đó.
- Trình biên dịch bỏ qua các khoảng trắng (hay tab hoặc xuống dòng) chen giữa lệnh.

```
a=2912;
a = 2912;
a
=
2912;
```



### Câu lệnh

#### ❖ Phân loại

- Câu lệnh đơn: chỉ gồm một câu lệnh.
- Câu lệnh phức (khối lệnh): gồm nhiều câu lệnh đơn được bao bởi { và }



# Câu lệnh xuất

- ❖Thư viện
  - #include <stdio.h> (standard input/output)
- Cú pháp
  - printf(<chuỗi định dạng>[, <đs1>, <đs1>, ...]);
  - <chuỗi định dạng> là cách trình bày thông tin xuất và được đặt trong cặp nháy kép "".
    - Văn bản thường (literal text)
    - Ký tự điều khiển (escape sequence)
    - Đặc tả (conversion specifier)



- ❖ Văn bản thường (literal text)
  - Được xuất y hệt như lúc gõ trong chuỗi định dạng.
- ❖ Ví dụ
  - Xuất chuỗi Hello World
    - printf("Hello"); printf("World");
    - printf("Hello World");
  - Xuất chuỗi a + b
    - → printf("a + b");



- ❖Ký tự điều khiển (escape sequence)
  - Gồm dấu \ và một ký tự như trong bảng sau:

Ký tự điều khiển	Ý nghĩa
\ <mark>a</mark>	Tiếng chuông
\ <b>b</b>	Lùi lại một bước
\ <u>n</u>	Xuống dòng
\t	Dấu tab
<b>\\</b>	In dấu ∖
\?	In dấu ?
\ <mark>"</mark>	In dấu "

- ❖ Ví dụ
  - printf("\t"); printf("\n");
  - printf("\t\n");



### ❖Đặc tả (conversion specifier)

- Gồm dấu % và một ký tự.
- Xác định kiểu của biến/giá trị muốn xuất.
- Các đối số chính là các biến/giá trị muốn xuất,
   được liệt kê theo thứ tự cách nhau dấu phẩy.

Đặc tả	Ý nghĩa	
% <b>c</b>	Ký tự	char
%d, %ld	Số nguyên có dấu	int, short, long
%f, %lf	Số thực	float, double
%s	Chuỗi ký tự	char[], char*
%u	Số nguyên không dấu	unsigned int/short/long

# VC & BB

# Chuỗi định dạng

- int a = 10, b = 20;
- printf("%d", a);
- printf("%d", b);
- printf("%d %d", a, b);

- → Xuất ra 10
- → Xuất ra 20
- → Xuất ra 10 20

- float x = 15.06;
- printf("%f", x); → Xuất ra 15.060000
- printf("%f", 1.0/3); → Xuất ra 0.333333



### Định dạng xuất

### Cú pháp

- Định dạng xuất số nguyên: %nd
- Định dạng xuất số thực: %n.kd

```
int a = 1706;
float x = 176.85;
printf("%10d", a);printf("\n");
printf("%10.2f", x);printf("\n");
printf("%.2f", x);printf("\n");
```

```
      1706

      176.85

      176.85
```



- Phối hợp các thành phần
  - int a = 1, b = 2;
  - Xuất 1 cong 2 bang 3 và xuống dòng.
    - printf("%d", a); // Xuất giá trị của biến a
    - printf(" cong "); // Xuất chuỗi " cong "
    - printf("%d", b); // Xuất giá trị của biến b
    - printf(" bang "); // Xuất chuỗi " bang "
    - printf("%d", a + b); // Xuất giá trị của a + b
    - printf("\n");
       // Xuất điều khiển xuống dòng \n
    - printf("%d cong %d bang %d\n", a, b, a+b);



# Câu lệnh nhập

#### ❖Thư viện

#include <stdio.h> (standard input/output)

### Cú pháp

- scanf(<chuỗi định dạng>[, <đs1>, <đs1>, ...]);
- <chuỗi định dạng> giống định dạng xuất nhưng chỉ có các đặc tả.
- Các đối số là tên các biến sẽ chứa giá trị nhập và được đặt trước dấu &



# Câu lệnh nhập

### ❖ Ví dụ, cho a và b kiểu số nguyên

- scanf("%d", &a); // Nhập giá trị cho biến a
- scanf("%d", &b); // Nhập giá trị cho biến b
- scanf("%d%d", &a, &b);
- Các câu lệnh sau đây sai
  - scanf("%d", a); // Thiếu dấu &
  - scanf("%d", &a, &b);// Thiếu %d cho biến b
  - scanf("%f", &a); // a là biến kiểu số nguyên
  - scanf("%9d", &a); // không được định dạng
  - scanf("a = %d, b = %d", &a, &b");



### Câu lệnh nhập xuất khác

- ❖ Các hàm nhập/ xuất trong thư viện iostream.h
  - Hàm nhập giá trị từ bàn phím:
    - cin>>tên biến;
    - Ví dụ:

int a;

cin>>a;//nhập giá trị cho biến a từ bàn phím

- Hàm xuất giá trị ra màn hình:
  - cout<<tên biến hoặc chuỗi ký tự;</li>
  - Ví dụ: int a = 5
  - cout<<"giá trị của a = "<<a<<"\n";</li>
  - Các biến và chuỗi cách nhau bởi dấu << (chuỗi nằm trong cặp dấu nháy kép "")



### Xuất có định dạng

#### Thư viện iomanip.h

- ❖ cout < < setw(n) < < tên biến;
  </p>
  - → chừa một khoảng n ký tự để xuất giá trị.
  - Ví dụ:
  - int a = 3, b = 9;
    - cout<<a<<setw(5)<<b;



### Xuất có định dạng

- cout<<setprecision(n);</pre>
  - → xuất số thập phân gồm n-1 chữ số lẻ.
    - Ví dụ:

```
float a = 3.1234, b = 9;
cout<< setprecision(3)<<a; //\rightarrow 3.12
```



### Một số hàm hữu ích khác

- Các hàm trong thư viện toán học
  - #include <math.h>
  - 1 đầu vào: double, Trả kết quả: double
    - acos, asin, atan, cos, sin, ...
    - exp, log, log10
    - sqrt
    - ceil, floor
    - abs, fabs
  - 2 đầu vào: double, Trả kết quả: double
    - double pow(double x, double y)



# Một số hàm hữu ích khác

- int x = 4, y = 3, z = -5;
- float t = -1.2;
- float kq1 = sqrt(x1);
- int kq2 = pow(x, y);
- float kq3 = pow(x, 1/3);
- float kq4 = pow(x, 1.0/3);
- int kq5 = abs(z);
- float kq6 = fabs(t);