



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

# C PROGRAMMING INTRODUCTION

## TUẦN 5: BIỂU THỨC

# Nội dung

- Biểu thức
  - Ôn tập
    - Biểu thức toán học
    - Biểu thức nhị phân
    - Biểu thức điều kiện
  - Thực hành

# Biểu thức và toán tử

- Toán tử số học

- Cộng  $+$
- Trừ  $-$
- Nhân  $*$
- Chia  $/$
- Chia lấy dư  $\%$

- Ví dụ

- **`fag = x % y;`**
- **`c = a - (a/b)*b;`**
- **`sum = var1 + var2 + var3;`**

# Biểu thức và toán tử

- Thứ tự ưu tiên
  - Toán tử có thứ tự ưu tiên (vd “nhân chia trước cộng trừ sau”)
    - Sử dụng dấu ngoặc khi cần thiết
  - Ví dụ: Tính giá trị trung bình của ba số **a**, **b** và **c**
    - Sai:  $a + b + c / 3$
    - Đúng:  $(a + b + c) / 3$

# Biểu thức và toán tử

- Thứ tự ưu tiên:

Operator(s)	Operation(s)	Order of evaluation (precedence)
()	Parentheses	Evaluated first. If the parentheses are nested, the expression in the innermost pair is evaluated first. If there are several pairs of parentheses “on the same level” (i.e., not nested), they are evaluated left to right.
*, /, or %	Multiplication Division Modulus	Evaluated second. If there are several, they re evaluated left to right.
+ or -	Addition Subtraction	Evaluated last. If there are several, they are evaluated left to right.

# Ra quyết định: Toán tử so sánh và toán tử quan hệ

- Các câu lệnh thực thi được:
  - Thực hiện hành động (tính toán, vào/ra dữ liệu)
  - Ra quyết định
    - "Qua" hoặc "trượt" khi nhận điểm kiểm tra
- Cấu trúc điều khiển **if**
  - Nếu điều kiện đúng, khối lệnh theo sau **if** được thực hiện
    - 0 tương ứng với sai, khác 0 tương ứng với đúng
  - Luôn thoát ra khỏi cấu trúc **if**
- Từ khóa
  - Các từ dành riêng cho C
  - Không thể được sử dụng làm định danh hoặc tên biến

# Ra quyết định: Toán tử so sánh và toán tử quan hệ

- Toán tử quan hệ

- Nhỏ hơn  $<$   $a < 5$
- Nhỏ hơn hoặc bằng  $<=$   $a <= b$
- Lớn hơn  $>$   $a > b + c$
- Lớn hơn hoặc bằng  $>=$   $a >= b + c$
- Bằng  $==$   $a == -6$
- Khác  $!=$   $a != 0$

# Ra quyết định: Toán tử so sánh và toán tử quan hệ

Keywords			
auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while



```
2#include <stdio.h>
3
4int main()
5{
6    int num1, num2;
7
8    printf( "Enter two integers, and I will tell you\n" );
9    printf( "the relationships they satisfy: " );
10    scanf( "%d%d", &num1, &num2 );    /* read two integers */
11
12    if ( num1 == num2 )
13        printf( "%d is equal to %d\n", num1, num2 );
14
15    if ( num1 != num2 )
16        printf( "%d is not equal to %d\n", num1, num2 );
17
18    if ( num1 < num2 )
19        printf( "%d is less than %d\n", num1, num2 );
20
21    if ( num1 > num2 )
22        printf( "%d is greater than %d\n", num1, num2 );
23
24    if ( num1 <= num2 )
25        printf( "%d is less than or equal to %d\n",
26                num1, num2 );
```

```
27
28     if ( num1 >= num2 )
29         printf( "%d is greater than or equal to %d\n",
30                num1, num2 );
31
32     return 0;    /* indicate program ended successfully */
33 }
```

Enter two integers, and I will tell you  
the relationships they satisfy: 3 7  
3 is not equal to 7  
3 is less than 7  
3 is less than or equal to 7

Enter two integers, and I will tell you  
the relationships they satisfy: 22 12  
22 is not equal to 12  
22 is greater than 12  
22 is greater than or equal to 12

# Biểu thức và toán tử

- Toán tử logic

- AND                      **&&**                       $(a > 0) \ \&\& \ (b > 0)$
- OR                      **||**                       $(a \leq 0) \ || \ (b \leq 0)$
- Negation                      **!**                       $!(a \ \&\& \ c)$

# Biểu thức và toán tử

- Toán tử bit

- AND

&

- OR (Inclusive OR)

|

- XOR (Exclusive OR)

^

- Dịch trái

<<

- Dịch phải

>>

- Đảo bit

~

- Ví dụ

- $x = 01001011$        $y = 00101100$        $\sim x = 10110100$

- $x \& y = 00001000$      $x | y = 01101111$

- $x \wedge y = 01100111$      $x \ll 2 = 00101100$

# Biểu thức và toán tử

- Phép gán và biểu thức

- **op** : +   -   \*   /   %   <<   >>   &   ^   |
- Nếu **expr1** và **expr2** là biểu thức, thì

**expr1 op= expr2**

- Tương đương với

**expr1 = (expr1) op (expr2)**

- Ví dụ

- **x += 1;**
  - **x = x + 1;**
- } tương đương

# Biểu thức và toán tử

- Biểu thức điều kiện

**`expr1 ? expr2 : expr3`**

- Nếu **`expr1`** đúng thì thực hiện **`expr2`**
  - Nếu **`expr1`** sai thì thực hiện **`expr3`**
- Ví dụ:
    - **`a = 5;`**
    - `b = 10;`**
    - `min = a < b ? a : b;`**

# Biểu thức và toán tử

- Toán tử tăng và giảm
  - Toán tử tăng tiền tố **`++variable`**
  - Toán tử tăng hậu tố **`variable++`**
  - Toán tử giảm tiền tố **`--variable`**
  - Toán tử giảm hậu tố **`variable--`**
- Ví dụ:
  - **`x = 4;`**  
**`y = x++ + 5;`**                      **`// x = 5, y = 9`**
  - **`x = 4;`**  
**`y = ++x + 5;`**                      **`// x = 5, y = 10`**

# Biểu thức và toán tử

- Ép kiểu

`(type-specifier) expression;`

- Ví dụ:

- `(double) date;`

- `fload var1 = 2.7;`

`int var2 = (int) var1;                    // var2 = 7`

- `(char) x;`

- `(int) d1 + d2;`



# Bài tập 5.1

- Viết chương trình chuyển đổi từ ki-lô-mét sang dặm
- Yêu cầu người dùng nhập vào giá trị theo ki-lô-mét và in ra màn hình giá trị theo dặm tương ứng

# Lời giải

```
#include <stdio.h>
/* printf, scanf definitions */

int main(void)
{
    double  miles,      /* distance in miles */
           kms;  /* equivalent distance in kilometers */

    /* Get the distance in kilometers. */
    printf("Enter the distance in kilometers > ");
    scanf("%lf", & kms);

    /* Convert the distance to miles. */
    miles = 1000 * kms;

    /* Display the distance in miles. */
    printf("That equals %f miles.\n", miles);
    return (0);
}
```

# Bài tập 5.2

- Chạy chương trình exercise5\_2.c để minh họa hoạt động của biểu thức logic và biểu thức điều kiện
- Thay thế  $\mathbf{b - a == b - c}$  bởi  $\mathbf{a = b - c}$  và giải thích kết quả

# exercise5\_2.c

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

```
main()
```

```
{
```

```
    int a = 5, b = 6, c = 7;
```

```
    puts("int a = 5, b = 6, c = 7;\n");
```

```
    printf("The value of a > b is      \t%i\n\n", a > b);
```

```
    printf("The value of b < c is      \t%i\n\n", b < c);
```

```
    printf("The value of a + b >= c is  \t%i\n\n", a + b >= c);
```

```
    printf("The value of a - b <= b-c is\t%i\n\n", a - b <= b - c);
```

```
    printf("The value of b - a == b - c is\t%i\n\n", b - a == b - c);
```

```
    printf("The value of a * b != c * c is\t%i\n\n", a * b < c * c);
```

```
}
```

# Bài tập 5.3

- Gõ và biên dịch chương trình exercise5\_3.c dưới đây, chương trình minh họa hoạt động của biểu thức điều kiện
- Thay đổi chương trình bằng cách xóa biến abs và max

# exercise5\_3.c

```
#include <stdio.h>
main()
{
    int n, m, abs, max;

    printf("Enter a positive or negative integer: ");
    scanf("%i", &n);

    printf("\nYou entered %i.\n", n);
    abs = n < 0 ? -n : n;
    printf("Its absolute value is %i.\n", abs);

    printf("\nEnter two integers (e.g. 1 2): ");
    scanf("%i %i", &n, &m);

    printf("\nYou entered %i and %i.\n", n, m);
    max = n > m ? n : m;
    printf("%i is the larger value.\n", max);
}
```

# Bài tập 5.4

- Ví dụ này minh họa hiện tượng **tràn số** nguyên xuất hiện khi một phép toán số học cố gắng tạo ra một giá trị số lớn hơn khả năng biểu diễn
- Gõ và biên dịch chương trình để xem kết quả

# exercise5\_4.c

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

```
void main(void)
{
    unsigned int x = UINT_MAX - 1;
    signed int y = INT_MAX - 1;

    printf("x is an unsigned int, occupying %i bytes.\n\n", sizeof(x));

    printf("The initial value of x is %u\n", x);
    x++;
    printf("Add 1; the new value of x is %u\n", x);
    x++;
```



# exercise5\_4.c

```
printf("Add 1; the new value of x is %u\n", x);  
x++;  
printf("Add 1; the new value of x is %u\n", x);  
  
printf("\ny is a signed int, occupying %i bytes.\n\n", sizeof(y));  
  
printf("The initial value of y is %i\n", y);  
y++;  
printf("Add 1; the new value of y is %i\n", y);  
y++;  
printf("Add 1; the new value of y is %i\n", y);  
y++;  
printf("Add 1; the new value of y is %i\n", y);  
  
return;  
}
```

# Bài tập 5.5

- Viết chương trình yêu cầu người dùng nhập vào hai số thực và lưu vào hai biến  $x$ ,  $y$
- Sử dụng cấu trúc điều khiển **if** để kiểm tra quan hệ giữa  $x$  và  $y$

# Lời giải

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

```
main()
```

```
{
```

```
    double num1, num2;
```

```
    printf( "Enter two doubles, and I will tell you\n" );
```

```
    printf( "the relationships they satisfy: " );
```

```
    scanf( "%f%f", &num1, &num2 ); /* read two integers */
```

```
    if ( num1 == num2 )
```

```
        printf( " %f is equal to %f\n", num1, num2 );
```

```
    if ( num1 != num2 )
```

```
        printf( " %f is not equal to %f\n ", num1, num2 );
```

# Lời giải

```
if ( num1 < num2 )  
    printf( "%f is less than %f\n", num1, num2 );  
  
if ( num1 > num2 )  
    printf( "%f is greater than %f\n", num1, num2 );  
  
if ( num1 <= num2 )  
    printf( "%f is less than or equal to %f\n",  
           num1, num2 );  
  
if ( num1 >= num2 )  
    printf( "%f is greater than or equal to %f\n",  
           num1, num2 );  
  
return 0;    /* indicate program ended successfully */  
}
```



25 YEARS ANNIVERSARY  
**SOICT**

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you  
for your  
attentions!**



[soict.hust.edu.vn/](http://soict.hust.edu.vn/)



[fb.com/groups/soict](https://fb.com/groups/soict)

