

Cơ sở lập trình

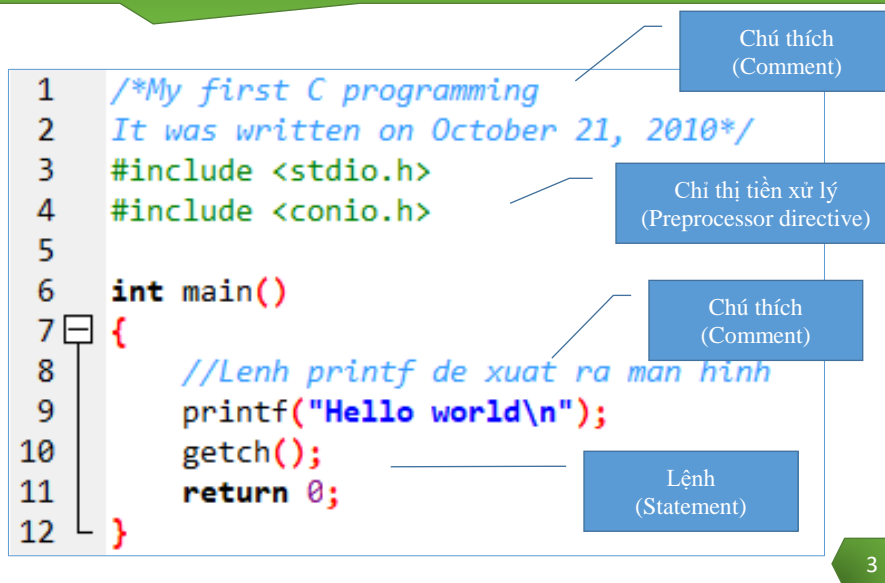
Bài 3. Giới thiệu ngôn ngữ lập trình C (3 tiết)
Bài 4. Kiểu dữ liệu và biểu thức trong C (6 tiết)

Biên soạn TS. Trần Minh Thái
Giảng viên: Lê Thị Minh Nguyễn
Email: nguyentlm@huflit.edu.vn

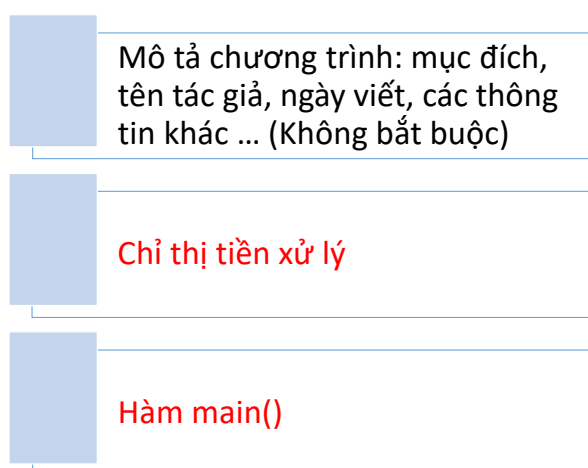
Mục tiêu

1. Cấu trúc cơ bản của một chương trình C
2. Các khái niệm, ký hiệu, quy tắc trong C
3. Kiểu dữ liệu cơ sở
4. Các toán tử
5. Các hàm thư viện C cơ bản
6. Bài tập

Cấu trúc cơ bản của chương trình C

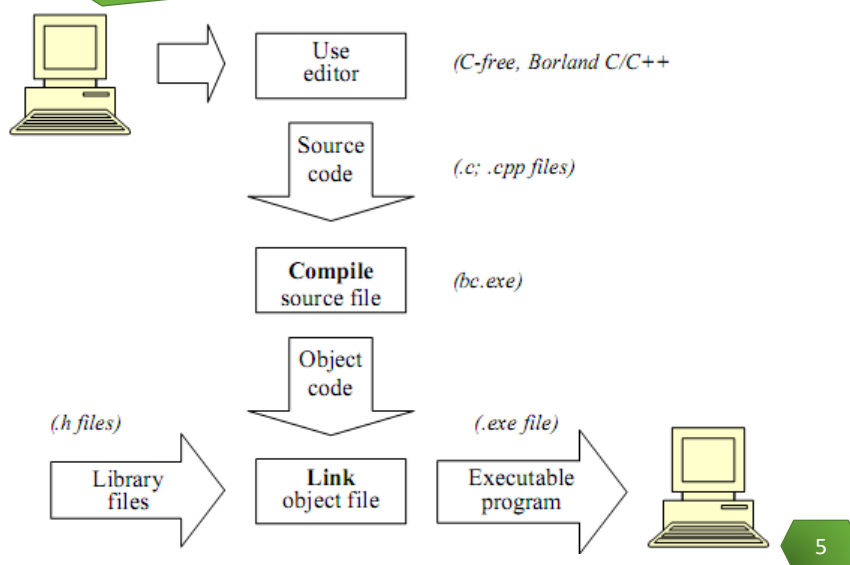


Cấu trúc cơ bản của một chương trình C



4

Các bước thực thi chương trình C



Phân tích chương trình ví dụ

#include <stdio.h> thể hiện đoạn chương trình kết hợp với file `stdio.h` (Standard Input/Output header file).

→ Tập tin này cho phép code sử dụng các lệnh có sẵn trong C để đọc dữ liệu từ bàn phím và xuất kết quả ra màn hình (**printf**) → **Chỉ thị tiền xử lý**

```

1  /*My first C programming
2  It was written on October 21, 2010*/
3  #include <stdio.h>
4  #include <conio.h>
5
6  int main()
7  {
8      //Lệnh printf để xuất ra màn hình
9      printf("Hello world\n");
10     getch();
11     return 0;
12 }
```

6

Hàm int main()

- Hàm main() là bắt buộc và được thực hiện đầu tiên khi thực thi chương trình C
- Các lệnh trong hàm main() được đặt trong cặp dấu { }
- Chương trình sẽ thực hiện những lệnh theo thứ tự trong hàm main()

7

Hàm int main()

- Lệnh

Lệnh thực hiện một chức năng nào đó (khai báo, gán, xuất, nhập, ...) và được kết thúc bằng dấu chấm phẩy (;)

- Khối lệnh

Khối lệnh gồm nhiều lệnh và được đặt trong cặp dấu ngoặc { }

8

Các lệnh trong hàm main

```

2  It was written on October 21, 2010*/
3  #include <stdio.h>
4  #include <conio.h>
5
6  int main()
7  {
8      //Lệnh printf để xuất ra màn hình
9      printf("Hello world\n");
10     getch();
11     return 0;
12 }

```

- Lệnh `printf("Hello world\n");` dùng để xuất ra màn hình dòng chữ “Hello World”
- Lệnh `getch();` dùng để chờ nhấn một phím bất kỳ trên bàn phím để tiếp tục thực hiện tiếp lệnh kế
- Lệnh `return 0;` dùng để kết thúc hàm `main()` – Kết thúc chương trình và trả về giá trị mã là 0

!!! Mọi lệnh đều được kết thúc bằng dấu ;

9

Chú thích (comment)

```

/*My first C programming
It was written on October 21, 2010*/
//Lệnh printf để xuất ra màn hình

```

- Được lập trình viên ghi chú hay diễn giải trong chương trình
- Đây **không phải là lệnh**
- Chú thích một dòng: dùng `//` trước chú thích
- Chú thích cho nhiều dòng: dùng cặp dấu `/*` và `*/` để bao nội dung chú thích

10

Một số tập tin thư viện thường dùng

- ***stdio.h***: định nghĩa các hàm vào ra chuẩn như các hàm xuất dữ liệu (printf()), nhập giá trị cho biến (scanf()), nhận ký tự từ bàn phím (getc()), in ký tự ra màn hình (putc()), nhập một chuỗi ký tự từ bàn phím (gets()), xuất chuỗi ký tự ra màn hình (puts())
- ***conio.h***: định nghĩa các hàm vào ra trong chế độ DOS, như clrscr(), getch(), ...

11

Một số tập tin thư viện thường dùng

- ***math.h***: Định nghĩa các hàm toán học như: abs(), sqrt(), log(), log10(), sin(), cos(), tan(), acos(), asin(), atan(), pow(), exp(), ...
- ***alloc.h***: định nghĩa các hàm vào ra cấp thấp gồm các hàm open(), _open(), read(), _read(), close(), _close(), creat(), _creat(), creatnew(), eof(), filelength(), lock(), ...

12

Chỉ thị tiền xử lý (*Preprocessor directive*)

- Các chỉ thị tiền xử lý là những dòng được đưa vào trong mã của chương trình phía sau dấu **#**
- Những dòng này không phải là lệnh của chương trình nhưng chỉ thị cho tiền xử lý
- Tiền xử lý kiểm tra mã lệnh trước khi biên dịch thực sự và ***thực hiện tất cả các chỉ thị trước*** khi thực thi mã lệnh của các câu lệnh thông thường

13

Chỉ thị tiền xử lý (*Preprocessor directive*)

Đặc điểm:

1. Mô tả trên một dòng, không có dấu ;
2. Trường hợp cần mô tả trên nhiều dòng dùng dấu \ ở cuối mỗi dòng

14

Chỉ thị tiền xử lý (*Preprocessor directive*)

1. Macro definitions: *#define*

#define “định danh” “thay thế”

Mục đích: Thay thế bất kỳ sự xuất hiện của “định danh” trong phần còn lại của các mã lệnh bằng “thay thế”.

“Thay thế”: có thể là một biểu thức hoặc một lệnh

15

Chỉ thị tiền xử lý (*Preprocessor directive*)

1. Macro definitions – Ví dụ

```
#define MAX_SIZE 100
int table1[MAX_SIZE];
int table2[MAX_SIZE];
```

→

```
int table1[100];
int table2[100];
```

16

Chỉ thị tiền xử lý (*Preprocessor directive*)

1. Macro definitions – Ví dụ

Định nghĩa thay thế một hàm có tham số

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

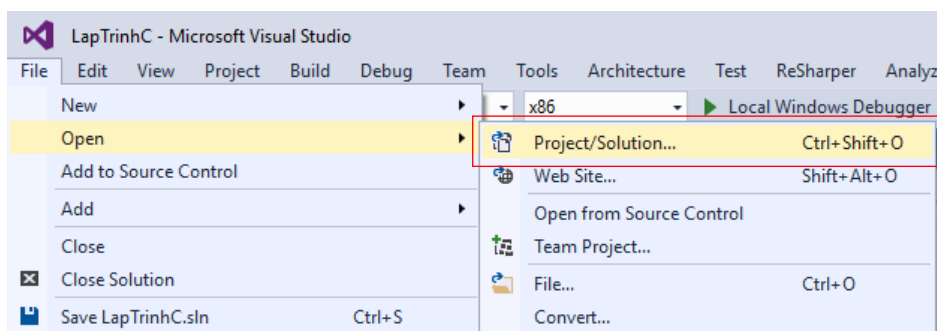
```
#define getmax(a, b) ((a)>(b)?(a):(b))
```

```
int main()
{
    int x = 5, y;
    y = getmax(x,2);
    printf(y);
    return 0;
}
```

17

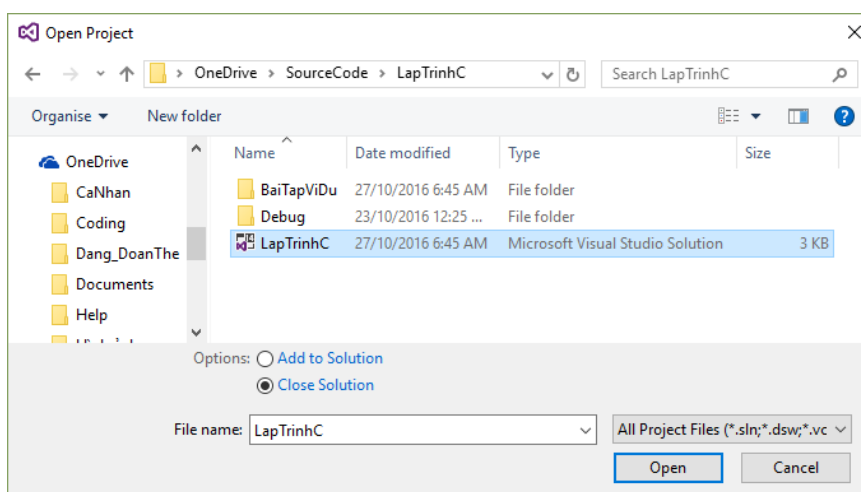
Đối với Dev-C chỉ cần mở file source code bằng cách Chọn File\ Open
 Đối với Microsoft Visual Studio .Net 2015 thì phải mở thông qua Project

Mở Project có sẵn trong MS VS 2015



19

Mở Project có sẵn trong MS VS 2015



20

Tập các ký tự thường dùng trong C

- Chữ cái hoa: A, B, ..., Z
- Chữ cái thường: a, b, c, ..., z
- Chữ số: 0, 1, ..., 9
- Các ký hiệu toán học: + - * / = () %
- Ký hiệu gạch nối: _
- Các ký hiệu đặc biệt như: . , ; [] { } ? ! \ & | % # ...
- Không được dùng các ký hiệu như: α , φ , Ω , π , ... hoặc tiếng việt có dấu: â, ã, ô...

21

Từ khóa (keyword)

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

22

Định danh (identifier Name)

Dùng để đặt tên biến, tên hằng, tên hàm, ...

- Bắt đầu bằng một ký tự
- Các ký tự trong tên biến chỉ có thể là các ký tự chữ, số hoặc dấu gạch dưới (_)
- Không được trùng với các từ khoá
- Không được trùng với phạm vi khai báo
- Tên dễ hiểu, súc tích và gợi nhớ
- **Phân biệt chữ HOA và thường**

23

Các định danh nào sau đây là không hợp lệ?

- *Tinh Tong*
- *Tinh-Tong*
- *Tinh_Tong*
- *x_Mu_2*
- *2_Mu_2*
- *Tien\$*
- *default*
- *yahoo.com*

24

Các kiểu dữ liệu (Data type)

- Kiểu cơ sở: tích hợp sẵn trong ngôn ngữ
 - Ký tự
 - Số nguyên
 - Số thực
 - Kiểu void
- Kiểu tự định nghĩa: dùng những kiểu cơ sở để xây dựng thành những kiểu dữ liệu mới cho phù hợp với bài toán
 - Mảng, cấu trúc ký tự, danh sách liên kết, cây, ...

25

Kiểu số nguyên (Integer type)

Data type	Size	Value range
char	1 byte	-128 đến 127 hoặc 0 đến 255 (Ký tự dạng mã ASCII)
unsigned char	1 byte	0 đến 255
signed char	1 byte	-128 đến 127
int	2 hoặc 4 bytes	-32,768 đến 32,767 hoặc -2,147,483,648 đến 2,147,483,647
unsigned int	2 hoặc 4 bytes	0 đến 65,535 hoặc 0 đến 4,294,967,295
short	2 bytes	-32,768 đến 32,767
unsigned short	2 bytes	0 đến 65,535
long	4 bytes	-2,147,483,648 đến 2,147,483,647
unsigned long	4 bytes	0 đến 4,294,967,295

26

Kiểu số thực (Floating-Point Type)

Type	Size	Value range	Precision
float	4 bytes	1.2E-38 to 3.4E+38	6 chữ số phần thập phân
double	8 bytes	2.3E-308 to 1.7E+308	15 chữ số phần thập phân
long double	10 bytes	3.4E-4932 to 1.1E+4932	19 chữ số phần thập phân

27

Biến (Variable)

- Một biến đại diện cho một vùng nhớ hay tập các vùng nhớ trên bộ nhớ chính của máy tính
- Biến để lưu trữ các giá trị do người dùng nhập vào hoặc các giá trị tạm thời trong quá trình tính toán
- Mỗi biến sẽ có tên và kiểu dữ liệu tương ứng
- **Phải khai báo biến trước khi sử dụng**

28

Khai báo biến

Cú pháp <Kiểu dữ liệu> tên biến;

Ví dụ:

```
int a; //Khai báo biến để lưu số nguyên tên a
```

```
float c; //Khai báo biến để lưu số thực tên c
```

Khai báo nhiều biến cùng kiểu

<Kiểu dữ liệu> tên biến1, tên biến2, tên biến3;

Ví dụ: int a, x, y;

29

Lấy kích thước của biến/ kiểu dữ liệu

- Kích thước của một số kiểu dữ liệu phụ thuộc vào hệ điều hành cụ thể của mỗi máy
- Để biết chính xác kích thước kiểu dữ liệu hay biến:

sizeof(tên kiểu/ biến)

→ Toán tử sizeof trả về kích thước theo đơn vị byte

30

Khai báo & gán giá trị ban đầu cho biến

<Kiểu dữ liệu> tên biến = giá trị hằng;

Ví dụ:

```
int a = 5;
```

```
float b = 5.4f, c = 9.2f;
```

```
char ch = 'n';
```

31

Các loại giá trị hằng (literal)

- **Hằng số:** Đó là các giá trị xác định, một hằng số có thể là *nguyên* (có kiểu dữ liệu int, hay long int) hay *thực* (có kiểu dữ liệu là float, double, long double).
- **Hằng ký tự:** Được đặt trong dấu nháy đơn. Ví dụ: 'A', 'a' tương ứng với giá trị nguyên 65, 97 trong bảng mã ASCII.
- **Hằng chuỗi:** Là tập hợp các ký tự được đặt trong cặp dấu nháy kép " ". Ví dụ: "Lập trình C"

32

Qui định viết giá trị hằng (literal)

Kiểu	Qui định suffix	Ví dụ
int		123
long	Có ký tự l hoặc L ở cuối	123L
unsigned int	Có ký tự u hoặc U ở cuối	123U
unsigned long	Có ký tự ul hoặc UL ở cuối	123UL
float	Có ký tự f hoặc F ở cuối	123.45F
double		123.45
long double	Có ký tự l hoặc L ở cuối	123.45L

33

Qui định viết giá trị hằng (literal)

- Một hằng số nguyên có thể thể hiện theo dạng số hệ thập phân (decimal), hệ bát phân (octal) hay hệ thập lục phân (hexadecimal)
- Tiền tố (prefix) là 0x hay 0X thể hiện cho hexadecimal và tiền tố 0 cho octal
- Ví dụ: 0x39F (hexadecimal), 056 (octal)

34

Định nghĩa biến hằng (constant)

- Không thay đổi giá trị trong khi thực thi chương trình
- Dùng toán tử **#define**

Cú pháp: `#define <tên_hằng> <giá_trị_hằng>`

Ví dụ: `#define MAX 100`

- Dùng từ khoá **const**

`const <kiểu_dữ_liệu> <tên_biến> = <giá_trị>;`

Ví dụ: `const int MAX = 100;`

Tên hằng số nên viết bằng chữ in HOA

35

Biểu thức (Expression)

- Biểu thức là một sự kết hợp giữa các toán tử (operator) và các toán hạng (operand) theo đúng một trật tự nhất định.
- Mỗi toán hạng có thể là một hằng, một biến hoặc một biểu thức khác.
- Trong trường hợp, biểu thức có nhiều toán tử, ta dùng cặp dấu ngoặc đơn () để chỉ định toán tử nào được thực hiện trước.

36

Cho biết những lỗi sai và sửa lại cho đúng

1. `int n = -100;`
2. `int x = 089, int a = 0x137E`
3. `unsigned int i = -100;`
4. `int = 2.9, b = 0x34G;`
5. `long m = 2, p = 4;`
6. `int 2k;`
7. `float y = y * 2;`
8. `char ch = "b";`

37

Chuyển đổi kiểu trong câu lệnh gán

- Giá trị của biểu thức bên phải dấu = được tự động chuyển thành kiểu dữ liệu của biến bên trái dấu =

Ví dụ:

```
int i=100;
```

```
double d = 123.456;
```

- Nếu thực thi lệnh `i = d;` thì `i = 123` (chuyển đổi kiểu mất mát thông tin)
- Nếu thực thi lệnh `d = i;` thì `d = 100.0` (chuyển đổi kiểu không mất mát thông tin)

38

Chuyển đổi kiểu trong câu lệnh gán

- Chuyển đổi từ KDL có miền giá trị nhỏ sang KDL có miền giá trị lớn hơn: `char`→`int`→`long`→`float`→`double` thì không mất thông tin
- Khi chuyển đổi từ KDL có miền giá trị lớn sang KDL có miền giá trị nhỏ hơn: `double`→`float`→`long`→`int`→`char` thì mất thông tin

39

Chuyển kiểu trong biểu thức

- Khi các hằng và biến có kiểu khác nhau trong một biểu thức thì giá trị của chúng phải được chuyển thành cùng kiểu trước khi thực hiện các phép toán
- Trình biên dịch sẽ thực hiện việc chuyển kiểu (convert) tự động đến kiểu của toán hạng có kiểu lớn nhất. Việc chuyển kiểu này gọi là thăng cấp kiểu (type promotion)

40

Chuyển kiểu trong biểu thức

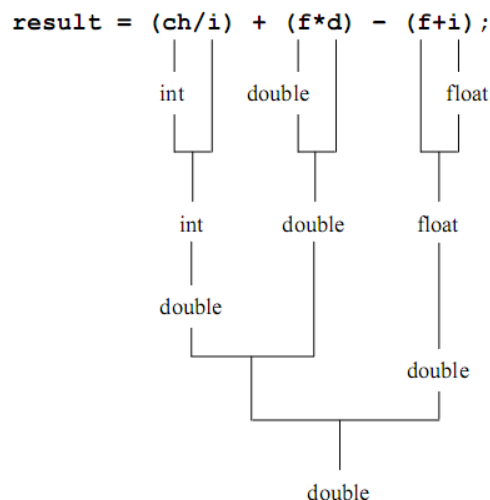
- Ví dụ:

char ch;

int i;

float f;

double d;



41

Ép kiểu (casting)

- Dùng để ép kiểu của một biểu thức thành một kiểu theo ý muốn của lập trình viên

- Cú pháp

(kiểu dữ liệu) biểu thức

- Ví dụ:

`float kq = (float)5/2;`

42

Toán tử số học (arithmetic operator)

Ký hiệu	Ý nghĩa	Ghi chú
+	Cộng	
-	Trừ	
*	Nhân	
/	Chia	Đối với 2 số nguyên thì kết quả là chia lấy phần nguyên
%	Chia lấy phần dư	Chỉ áp dụng cho 2 số nguyên

43

Toán tử gán phức hợp

Ký hiệu	Ví dụ	Ý nghĩa
+=	$n += 25$	$n = n + 25$
-=	$n -= 25$	$n = n - 25$
*=	$n *= 25$	$n = n * 25$
/=	$n /= 25$	$n = n / 25$
%=	$n \% = 25$	$n = n \% 25$

44

Toán tử so sánh

Phép toán so sánh

Stt	Ký hiệu	Ý nghĩa
1	>	Lớn hơn
2	<	Nhỏ hơn
3	>=	Lớn hơn hoặc bằng
4	<=	Nhỏ hơn hoặc bằng
5	= =	Bằng nhau
6	!=	Khác nhau

45

Các toán tử khác

PHÉP TOÁN LOGIC			
1	!	NOT	
2	&&	AND	
3		OR	
TOÁN TỬ TĂNG GIẢM			
1	++	Tăng 1	Nếu toán tử tăng giảm đặt trước thì tăng giảm trước rồi tính biểu thức hoặc ngược lại.
2	--	Giảm 1	
PHÉP TOÁN THAO TÁC TRÊN BIT			
1	&	AND	
2		OR	
3	^	XOR	
4	<<	Dịch trái	
5	>>	Dịch phải	
6	~	Lấy phần bù theo bit	

46

Cho biết kết quả của đoạn chương trình sau:

```
int a, b;
b=a%2 + a/2 + --a;
Với a = 17 Kết quả: a = ?; b = ?
Với a = 3 Kết quả: a = ?; b = ?
```

```
-----
int a, b;
b=a/3 + a--;
Với a = 8 Kết quả: a = ?; b = ?
Với a = 21 Kết quả: a = ?; b = ?
```

47

Toán tử điều kiện

(ĐK)?<BT trường hợp đúng>:<BT trường hợp sai>

Ví dụ:

```
int n;
```

```
(n%2==0)? n++ : n--;
```

n = 10 → (n%2==0): **đúng**
→ thực hiện lệnh n++ = 10++ = 11

n = 21 → (n%2==0): **sai**
→ thực hiện lệnh n-- = 21-- = 20

→ nếu n = 10 thì giá trị n = 11

→ nếu n = 21 thì giá trị n = 20

48

Toán tử điều kiện

Cho biết kết quả đoạn chương trình sau:

int k;

m = (k%3==0)?k++: k--;

Với k=10 Kết quả: m = ?

Với k=15 Kết quả: m = ?

49

Thứ tự ưu tiên toán tử

Toán tử	Độ ưu tiên	Trình tự kết hợp
() [] ->	1	Từ trái qua phải
! ~ ++ -- - + * & sizeof	2	Từ phải qua trái
* / %	3	Từ trái qua phải
+ -	4	Từ trái qua phải
<< >>	5	Từ trái qua phải
< <= >= >	6	Từ trái qua phải
== !=	7	Từ trái qua phải
&	8	Từ trái qua phải
	9	Từ trái qua phải
^	10	Từ trái qua phải
&&	11	Từ trái qua phải
	12	Từ trái qua phải
? :	13	Từ phải qua trái
= += -= *= /= %=	14	Từ phải qua trái

50

Hàm xuất chuỗi/ giá trị ra màn hình

- Thư viện `<stdio.h>`

- Xuất:

`printf("hàng chuỗi");`

Vd: `printf("Xin chao cac ban");`

`printf("chuỗi định dạng", đối số 1, đối số 2);`

Vd: `int a=5;`

`float b=2.7;`

`printf("Gia tri cua bien a=%d, b=%f", a, b);`

Có thể có nhiều đối số theo thứ tự của chuỗi định dạng tương ứng

51

Hàm nhập giá trị cho biến từ bàn phím

- Thư viện `<stdio.h>`

- Cú pháp

`scanf("chuỗi định dạng", &tên biến);`

Vd: `int x;`

`scanf("%d", &x);`

Địa chỉ của biến
(**&**tên biến)

52

Chuỗi định dạng

Stt	Tên kiểu	Ý nghĩa	Chuỗi định dạng
Kiểu số thực			
1	float		%f
2	double		%lf
3	long double		%lf
Kiểu số nguyên/ chuỗi ký tự			
1	char	Ký tự	%c
		Số nguyên	%d
2	unsigned char	Số nguyên dương	%d
3	int	Số nguyên	%d
4	unsigned int	Số nguyên dương	%u
5	long	Số nguyên	%ld
6	unsigned long	Số nguyên dương	%lu
7	char *	Chuỗi	%s

53

Xuất những ký tự đặc biệt

Ký tự	Ý nghĩa	Ví dụ
\'	Xuất dấu nháy đơn	printf("\'"); Kết quả: '
\"	Xuất dấu nháy đôi	printf("\'"); Kết quả: "
\\	Xuất dấu chéo ngược “\”	printf("\\"); Kết quả: \
\0	Ký tự Null	Dùng để gán ký tự kết thúc của chuỗi
\a	Alert : Tiếng bíp	

54

Xuất ký tự điều khiển

Ký tự	Ý nghĩa	Ví dụ
\t	Tab vào một đoạn ký tự trắng	printf(("xyz\tzyx")); Kết quả: xyz zyx
\b	Xuất lùi về trước	printf("xyz\t\bzyx"); Kết quả: xyzzyx
\n hoặc endl	Xuống dòng	printf("xyz\nzyx"); Kết quả: xyz zyx
\r	Về đầu dòng	printf("xyz\rzyx"); Kết quả: zyx

55

Ví dụ hàm nhập xuất – Cách 1

```
#include <stdio.h>

int main()
{
    int a, b, tong;
    float thuong;
    printf("Nhap vao so nguyen a: ");
    scanf("%d", &a);
    printf("Nhap vao so nguyen b: ");
    scanf("%d", &b);

    tong = a + b;
    thuong = (float)a / b;
    printf("Tong %d + %d = %d", a, b, tong);
    printf("\nThuong %d : %d = %f", a, b, thuong);

    return 0;
}
```

56

Ví dụ hàm nhập xuất – Cách 2

```
#include <stdio.h>

int main()
{
    int a, b;

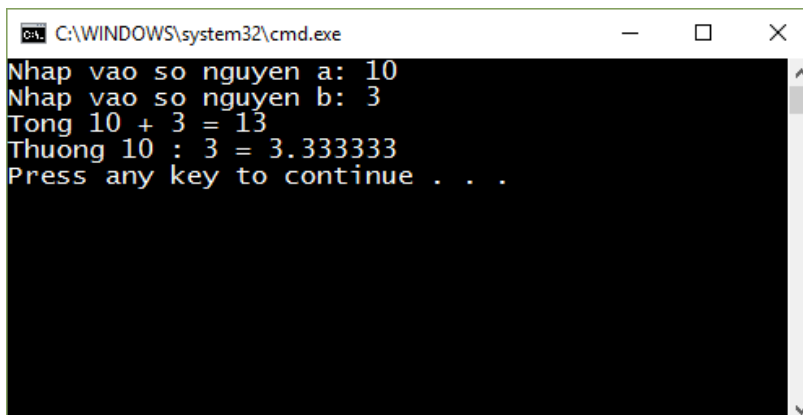
    printf("Nhap vao so nguyen a: ");
    scanf("%d", &a);
    printf("Nhap vao so nguyen b: ");
    scanf("%d", &b);

    printf("Tong %d + %d = %d", a, b, a + b);
    printf("\nThuong %d : %d = %f", a, b, (float)a/b);

    return 0;
}
```

57

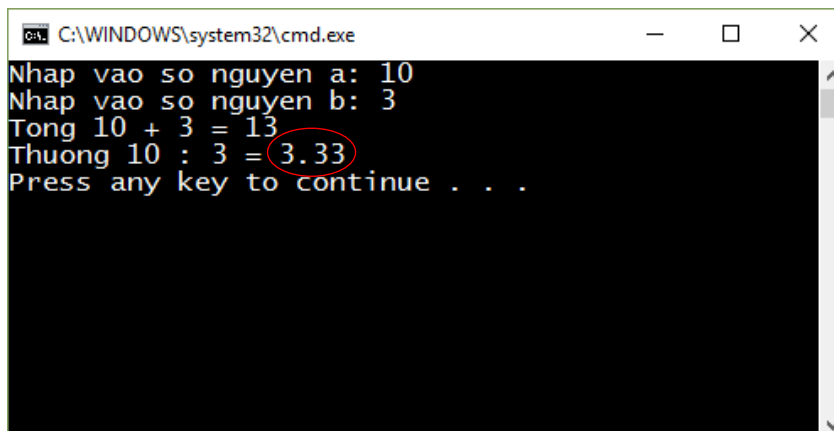
Kết quả của chương trình



```
C:\WINDOWS\system32\cmd.exe
Nhap vao so nguyen a: 10
Nhap vao so nguyen b: 3
Tong 10 + 3 = 13
Thuong 10 : 3 = 3.333333
Press any key to continue . . .
```

58

Cần hiển thị kết quả chia gồm 2 số phần lẻ



```

C:\WINDOWS\system32\cmd.exe
Nhap vao so nguyen a: 10
Nhap vao so nguyen b: 3
Tong 10 + 3 = 13
Thuong 10 : 3 = 3.33
Press any key to continue . . .
  
```

59

Thay đổi tham số của printf()

```


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

int main()
{
    int a, b;

    printf("Nhap vao so nguyen a: ");
    scanf("%d", &a);
    printf("Nhap vao so nguyen b: ");
    scanf("%d", &b);

    printf("Tong %d + %d = %d\n", a, b, a + b);
    printf("Thuong %d : %d = %.2f\n", a, b, (float)a / b);

    getch();
    return 0;
}
  
```



60

Có thể áp dụng cho số nguyên

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

```
int main()
{
    printf("Thoi gian: %d:%d:%d\n", 2, 5, 30);
    getch();
    return 0;
}
```

Thoi gian: 2:5:30

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

```
int main()
{
    printf("Thoi gian: %02d:%02d:%02d\n", 2, 5, 30);
    getch();
    return 0;
}
```

Thoi gian: 02:05:30

61

Bài tập 1

- Viết chương trình in ra màn hình thông tin cá nhân theo mẫu sau:

* *Thông tin cá nhân* *

Họ tên:

Lớp:

Thông tin liên lạc:

- *Địa chỉ:*

- *Số điện thoại:*

62

Cách trình bày chương trình

- Mỗi lệnh nằm trên một dòng. Cuối dòng lệnh **PHẢI** có dấu chấm phẩy (;)
- Lệnh quá dài có thể được viết thành nhiều dòng sao cho mỗi lệnh phải được quan sát trọn vẹn trong phạm vi của số lệnh

63

Cách trình bày chương trình (tt)

- Không nên đặt nhiều lệnh trên cùng một dòng, ngay cả các khai báo biến, nếu các biến có khác kiểu cũng nên đặt trên các dòng khác nhau.
- Có các chú thích, ghi chú đầy đủ
- Chương trình phân cấp các khối lệnh con theo từng cột

64

Bài tập 2 – Viết các chương trình sau

1. Nhập vào 3 số nguyên a, b và c → tính giá trị trung bình cộng của 3 số trên và xuất kết quả ra màn hình
2. Nhập vào giờ, phút và giây, đổi ra giây và xuất kết quả ra màn hình
3. Nhập vào 1 số nguyên dương n gồm 3 chữ số, hãy in ra chữ số ở hàng trăm, hàng chục và hàng đơn vị
4. Nhập vào 2 số thực x và y, in ra màn hình số lớn nhất
(nếu $x = y$ thì xuất giá trị x)

65

Bài tập 2 (tt)

5. Làm tròn một số thực x được nhập từ bàn phím thành số nguyên (Nếu phần lẻ < 5 thì làm tròn xuống, ngược lại làm tròn lên. VD: $3.4 \rightarrow 3$; $3.6 \rightarrow 4$)
6. Nhập vào một số đo nhiệt độ (độ Fahrenheit) và xuất ra nhiệt độ tương đương (độ Celsius) theo công thức:

$$^{\circ}C = \frac{5}{9}(^{\circ}F - 32)$$
7. Viết chương trình nhập vào chiều dài và chiều rộng (số nguyên) của hình chữ nhật. Tính diện tích và chu vi của hình chữ nhật

66

Các lỗi thường gặp khi viết chương trình

- Sử dụng lệnh nhưng **KHÔNG** khai báo thư viện hàm
- **KHÔNG** khai báo các biến sử dụng trong chương trình
- Lưu một giá trị vào một biến nhưng **KHÔNG** cùng kiểu dữ liệu với biến
- Sử dụng biến trong một biểu thức khi **CHƯA** có giá trị

67

Các lỗi thường gặp khi viết chương trình

- Dùng phép chia nhưng **KHÔNG** ép kiểu cho số nguyên

Ví dụ:

double kq = 3.2 + 2/3 + 1.5;

thì kq sẽ bằng 4.7

thay vì

double kq = 3.2 + (double)2/3 + 1.5;

thì kq sẽ bằng 5.366666666666667

68

Bài tập về nhà

(2.1) Viết chương trình nhập vào bán kính hình tròn (*số thực*). Tính diện tích và chu vi của hình tròn

(2.2) Nhập vào 3 số nguyên a, b và c, cho biết 3 số vừa nhập có thứ tự tăng dần ($a < b < c$) không? Và xuất kết quả

69

Bài tập về nhà

(2.3) Viết chương trình in kích thước và miền giá trị [*min...max*] của tất cả kiểu dữ liệu cơ bản, riêng kiểu số thực in thêm thông tin về độ chính xác

• Hướng dẫn: Toán tử sizeof(), <limits.h> và <float.h>

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

int main()
{
    printf("* int: size = %d byte(s); value range = [%d...%d]",
           sizeof(int), INT_MIN, INT_MAX);
    //Tuong tu cho nhung kieu khac
    getch();
    return 0;
}
```

70

Q&A



71