



TRƯỜNG ĐẠI HỌC ĐỒNG THÁP
Khoa sư phạm Toán – Tin



Bài giảng

NGÔN NGỮ LẬP TRÌNH C

GV: Nguyễn Thị Mỹ Dung

Email: ntmdung@dthu.edu.vn

Học phần: Lập trình căn bản (3tc)

Sinh viên lớp: ĐHCNTT20



NỘI DUNG MÔN HỌC



01

Một số khái niệm cơ bản (4)

02

CÁC KIỂU DỮ LIỆU VÀ PHÉP TOÁN (4)

03

Các lệnh cấu trúc (6 + 6)

04

Các thao tác trên dữ liệu kiểu mảng (4 + 4)

05

Hàm và phạm vi của biến (3 + 4)



Kiểu dữ liệu & phép toán



- I. Từ khóa**
- II. Các phép toán logic, toán tử**
- III. Kiểu dữ liệu**
- IV. Tên và quy cách đặt tên**
- V. Khai báo biến, hằng, biểu thức**
- VI. Câu lệnh, khối lệnh & chú thích**
- VII. Các hàm nhập xuất dữ liệu**
- VIII. Cách chuyển đổi kiểu dữ liệu**
- IX. Bài tập**



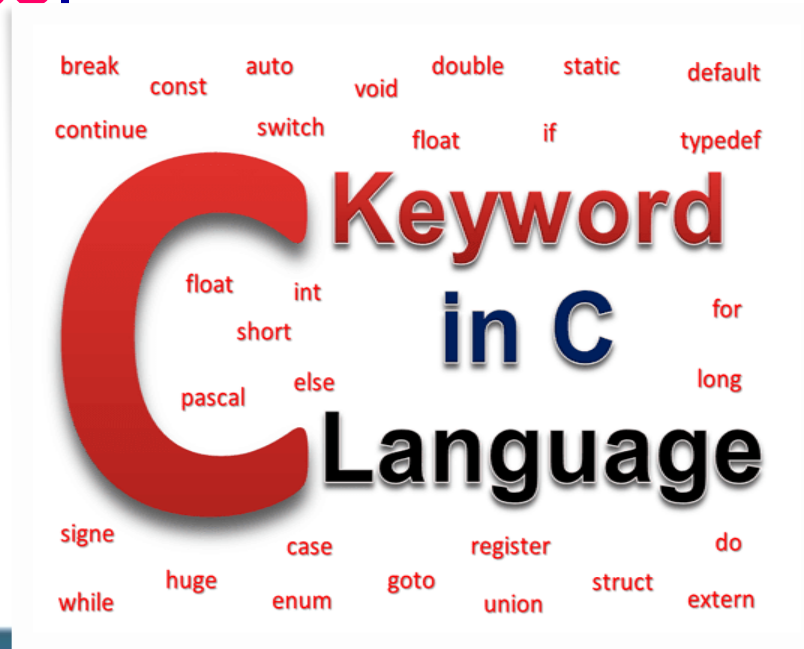
1. Từ khóa

- **Từ khóa**
- Các phép toán
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- Nhập xuất

➤ Khái niệm

Là các từ dành riêng được **định nghĩa sẵn** trong ngôn ngữ C, mang một ý nghĩa nào đó.

Không thể dùng từ khóa để **làm định danh, tên biến hay hằng số**.





1. Từ khóa

- **Từ khóa**
- Các phép toán
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
 - Nhập xuất

➤ Một số từ khóa

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



1. Từ khóa

- **Từ khóa**
- Các phép toán
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- Nhập xuất



Chú ý:

- *Không được dùng từ khoá đặt tên cho các hằng, biến, mảng, hàm.*
- *Từ khoá không được viết hoa mà phải viết thường.*
- *C phân biệt chữ hoa và chữ thường.*





2. Các phép toán

- Từ khóa
- **Các phép toán**
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
 - Nhập xuất

➤ Phép toán

- **Các phép toán số học**: cộng (+), trừ (-), nhân (*), chia (/) trên các kiểu dữ liệu: char, int, float, double
- **Các phép toán tăng** (++), **giảm** (--). Các phép toán có 2 dạng:

Cú pháp:

++<biến> hay **<biến>++**

--<biến> hay **<biến>--**



Diễn giải:

- **Phép toán tăng**: $i = i + 1 \leftrightarrow i++$ hoặc $++i$






- **Phép toán giảm**: $i = i - 1 \leftrightarrow i--$ hoặc $--i$



2. Các phép toán

- Từ khóa
- **Các phép toán**
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- Nhập xuất

Ví dụ: Sự khác nhau giữa dạng `++i` và `i++` trong phép gán. Cho `i = 3` và `j = 5`

PHÉP TOÁN	TƯƠNG ĐƯƠNG	KẾT QUẢ
<code>i = ++j;</code>	<code>j = j+1; i = j</code> 	
<code>i = j++;</code>	<code>i = j; j = j+1;</code> 	
<code>i++;</code>	<code>i = i+1;</code> 	
<code>j = ++i + 5;</code>	<code>i = i+1; j = i+5;</code> 	
<code>j = i++ + 5;</code>	<code>j = i+5; i = i+1;</code> 	



Viết chương trình hiển thị kết quả phép toán tăng, giảm?



2. Các phép toán

- Từ khóa
- **Các phép toán**
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- Nhập xuất

➤ Toán tử so sánh (quan hệ)

Trả về giá trị là true (đúng) hoặc false (sai)

Ví dụ: **int A=5, B=6;**

MÔ TẢ	MINH HỌA	KẾT QUẢ
< (nhỏ hơn)	$A < B$	true
<= (nhỏ hơn hoặc bằng)	$A \leq B$	true
> (lớn hơn)	$A > B$	false
>= (lớn hơn hoặc bằng)	$A \geq B$	false
== (bằng)	$A == B$	false
!= (khác)	$A != B$	true



2. Các phép toán

- Từ khóa
- **Các phép toán**
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- Nhập xuất

➤ Toán tử so sánh (tt)

Ví dụ:

```
#include <stdio.h>
int main(){
    int a = 5, b = 6;
    printf("%d == %d is %d \n", a, b, a == b);
    printf("%d > %d is %d \n", a, b, a > b);
    printf("%d < %d is %d \n", a, b, a < b);
    printf("%d != %d is %d \n", a, b, a != b);
    printf("%d >= %d is %d \n", a, b, a >= b);
    printf("%d <= %d is %d \n", a, b, a <= b);
}
```



2. Các phép toán

- Từ khóa
- **Các phép toán**
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- Nhập xuất

➤ Các toán tử logic

&& (and) trả về **true** khi cả 2 toán hạng đều đúng. Ngược lại trả về **false**.

- $(5 < 6) \ \&\& \ (7 < 8) \rightarrow true$
- $(5 > 6) \ \&\& \ (7 < 8) \rightarrow false$
- $(5 > 6) \ \&\& \ (7 > 8) \rightarrow false$

|| (or) trả về **true** khi ít nhất một trong 2 toán hạng đúng. Ngược lại trả về **false**.

- $(5 < 6) \ || \ (7 < 8) \rightarrow true$
- $(5 > 6) \ || \ (7 < 8) \rightarrow true$
- $(5 > 6) \ || \ (7 > 8) \rightarrow false$



2. Các phép toán

- Từ khóa
- **Các phép toán**
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
 - Nhập xuất

➤ Các toán tử logic (tt)

! (not) trả về **true** khi toán hạng (đằng sau dấu ! sai).
Ngược lại trả về **false**.

- **!(7>8)** -> *true*
- **!(7<8)** -> *false*

Ví dụ:

```
#include <stdio.h>
int main(){
    int a = 5;
    printf("\n%d", (a > 0 && a % 2 == 0));
    printf("\n%d", (a % 2 == 0 || a % 5 == 0));
    printf("\n%d", !(a == 5));
}
```



2. Các phép toán

- Từ khóa
- **Các phép toán**
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
 - Nhập xuất

➤ Toán tử điều kiện

Toán tử điều kiện: có ký hiệu ?

Cú pháp:

<điều-kiện>?<biểu-thức-1>:<biểu-thức-2>

Diễn giải:

<điều kiện> **đúng**: giá trị là <biểu-thức-1>

<điều kiện> **sai**: giá trị là <biểu-thức-2>

Ví dụ:

```
int a = 1, b = (a)?1:2;  
b+=1;
```





2. Các phép toán

- Từ khóa
- **Các phép toán**
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
 - Nhập xuất

➤ **Toán tử phẩy:** ký hiệu ",", "

Diễn giải:

Biểu thức, bao gồm nhiều cặp biểu thức và được tính từ trái sang phải. Kết quả của toàn bộ biểu thức là kết quả của biểu thức bên phải.

Ví dụ:

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





2. Các phép toán

- Từ khóa
- **Các phép toán**
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- Nhập xuất

➤ Thứ tự ưu tiên của các phép toán

Loại	Toán tử	Thứ tự ưu tiên
Toán tử	() [] -> . ++ - -	Trái sang phải
Toán tử một ngôi	+ - ! ~ ++ - - (type) * & sizeof	Phải sang trái
Tính nhân	* / %	Trái sang phải
Tính cộng	+ -	Trái sang phải
Dịch chuyển bit	<< >>	Trái sang phải
So sánh không bằng	< <= > >=	Trái sang phải
So sánh ngang bằng	== !=	Trái sang phải



2. Các phép toán

- Từ khóa
- **Các phép toán**
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- Nhập xuất

➤ Thứ tự ưu tiên của các phép toán

Loại	Toán tử	Thứ tự ưu tiên
Phép AND bit	&	Trái sang phải
Phép XOR bit	^	Trái sang phải
Phép OR bit		Trái sang phải
Phép AND logic	&&	Trái sang phải
Phép OR logic		Trái sang phải
Điều kiện	? :	Phải sang trái
Gán	= += -= *= = %= >>= <<= &= ^= =	Phải sang trái
Dấu phẩy	,	Trái sang phải



3. Kiểu dữ liệu

- Từ khóa
- Các phép toán
- **Kiểu dữ liệu**
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- Nhập xuất

➤ **Khái niệm**

Một kiểu dữ liệu được định nghĩa: **Là một tập hợp các giá trị mà một biến thuộc kiểu đó có thể nhận được.** Trên đó xác định một số phép toán. Có nhiều kiểu dữ liệu:

- **Kiểu vô hướng**: kiểu số nguyên, kiểu số thực, kiểu ký tự,...
- **Kiểu dữ liệu có cấu trúc**: kiểu mảng, kiểu cấu trúc, kiểu tập tin.
- **Kiểu con trỏ.**



3. Kiểu dữ liệu

- Từ khóa
- Các phép toán
- **Kiểu dữ liệu**
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- Nhập xuất

Sơ đồ tổng quát về kiểu dữ liệu





3. Kiểu dữ liệu

- Từ khóa
- Các phép toán
- **Kiểu dữ liệu**
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- Nhập xuất

➤ Kiểu số nguyên:

- Dữ liệu số không có thành phần thập phân. Khoảng lớn của nó phụ thuộc vào độ dài bộ nhớ lưu trữ.
- Số nguyên 32 bits : từ 2147483647 ($2^{31} - 1$) đến -147483648. Số nguyên 16 bits : từ 32767 ($2^{15} - 1$) đến -32768.

Các kiểu số nguyên cơ bản sau đây

Character (**char**), Integer (**int**), Short integer (**short**), Long integer (**long**), Unsigned char (**unsigned char**).

Các phép tính số học với số nguyên: cộng (+), trừ (-), nhân (*), chia (/), chia lấy dư (%).



3. Kiểu dữ liệu

- Từ khóa
- Các phép toán
- **Kiểu dữ liệu**
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
 - Nhập xuất

➤ Khai báo: biến int

Cú pháp:

<kiểu số nguyên> <danh_sách_tên_biến>;

Ví dụ:

int a, b;

long tong = 0;

Định dạng cho số nguyên:

%d (int)

%ld (long)



Chú ý: Số nguyên chia cho số nguyên sẽ được một số nguyên.

Ví dụ: **9/2 = 4**



3. Kiểu dữ liệu

- Từ khóa
- Các phép toán
- **Kiểu dữ liệu**
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
 - Nhập xuất

➤ Kiểu số thực:

- Số thực có ít nhất 6 số lẻ thập phân. Các phép tính giữa số thực và số thực hoặc với số nguyên cho kết quả kiểu số thực (float).

- **Các kiểu số thực:** float, double, long double.

- **Định dạng:** %f (%e,%E), %lf (%f, %e, %E), %lf (%f, %e, %E).

- **Các phép tính trên kiểu số thực:** cộng (+), trừ (-), nhân (*), chia (/).



3. Kiểu dữ liệu

- Từ khóa
- Các phép toán
- **Kiểu dữ liệu**
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
 - Nhập xuất

➤ Khai báo: biến float

Cú pháp:

<kiểu số thực> <danh_sách_tên_biến>;

Ví dụ:

float bankinh;

double tich, giaithua;

Định dạng cho số thực:

%f (float, double)

%lf (long double)

Chú ý:



- Đối với số thực không có phép toán chia lấy dư (%)
- Số nguyên (a) chia cho số nguyên (b) để được số thực ta thực hiện chuyển đổi kiểu như sau: **(float)a/b;**



3. Kiểu dữ liệu

- Từ khóa
- Các phép toán
- **Kiểu dữ liệu**
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- Nhập xuất

➤ Kiểu ký tự và khai báo: **biến char**

Cú pháp:

```
<kiểu_ký_tự> <danh_sách_biến>;
```

Ví dụ:

```
char c;
```

```
char ten[7];
```

Định dạng cho ký tự:

```
%c (char)
```



Chú ý:

- Một số hàm xử lý ký tự: **toascii**, **toupper**, **tolower**. Khi sử dụng các hàm này phải khai báo thư viện: **ctype.h**



3. Kiểu dữ liệu

- Từ khóa
- Các phép toán
- **Kiểu dữ liệu**
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
 - Nhập xuất

➤ Một số ký tự đặc biệt:

Ký tự	Diễn giải
\n	Xuống dòng
\t	Khoảng tab
\b	backspace
\0	Kết thúc chuỗi
\r	Kết thúc dòng
\\	In dấu \
\'	In dấu `
\"	In dấu "



3. Kiểu dữ liệu

- Từ khóa
- Các phép toán
- **Kiểu dữ liệu**
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- Nhập xuất

➤ Kiểu logic:

- Trong C không có kiểu logic. Nhưng chúng ta vẫn ngầm định kiểu số logic như kiểu số nguyên.

Ví dụ:

0: tương ứng với giá trị false,

1: tương ứng với giá trị true

* Các phép toán trên kiểu logic

Toán tử	Phép toán logic
&&	AND
	OR
!	NOT

* Các biểu thức quan hệ:

Phép toán	Ý nghĩa
!=	Khác nhau
==	Bằng nhau
>=	Lớn hơn bằng
<=	Bé hơn bằng
>	Lớn hơn
<	Bé hơn



4. Tên và cách đặt tên

- Từ khóa
- Các phép toán
- Kiểu dữ liệu
- **Tên & đặt tên**
- Khai báo
- Lệnh, khối lệnh
- Nhập xuất

➤ **Tên:** Chương trình dùng rất nhiều tên: tên biến, tên chương trình, tên hằng, tên hàm...

Tên được đặt với bất kỳ tên hợp lệ nào.

Tên hợp lệ:

không được bắt đầu bằng chữ số,
không được chứa các ký tự đặc biệt,
không chứa khoảng trắng,
không được là từ khoá.

Ví dụ:

Turbo C

turbo c

TURBO C



- Tên chữ hoa và tên chữ thường là khác nhau;
- Tên hàm phải có nghĩa và đúng cú pháp.



5. Khai báo

- Từ khóa
- Các phép toán
- Kiểu dữ liệu
- Tên & đặt tên
- **Khai báo**
- Lệnh, khối lệnh
- Nhập xuất

➤ **Biến:** Là đại lượng mà giá trị của nó có thể thay đổi trong quá trình thi hành chương trình.

Ví dụ:

int a, b, c;

float d, c, e;

char c;

Khởi tạo biến:

Ngay trên dòng khai báo ta có thể gán giá trị cho biến.

Ví dụ: **int** a, b=0;

Truy xuất đến địa chỉ biến :

Một số hàm của C thường dùng đến địa chỉ của biến, điển hình như hàm **scanf**. Để nhận địa chỉ biến dùng toán tử: **&**

Cú pháp: **&tên_biến**

Ví dụ: **&a**



5. Khai báo

- Từ khóa
- Các phép toán
- Kiểu dữ liệu
- Tên & đặt tên
- **Khai báo**
- Lệnh, khối lệnh

➤ **Hằng**: là đại lượng không thay đổi giá trị trong quá trình thi hành chương trình.

Khai báo:

Dùng toán tử **#define** nhằm định nghĩa hằng tượng trưng.

Cú pháp:

#define <tên_hằng> <giá_trị_hằng>

Ví dụ:

#define max 100

#define str "DH DongThap"

- **Biến hằng** được định nghĩa nhờ từ khoá const với cú pháp như sau:

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

Ví dụ: **const** int max = 100;



5. Khai báo

- Từ khóa
- Các phép toán
- Kiểu dữ liệu
- Tên & đặt tên
- **Khai báo**
- Lệnh, khối lệnh
 - Nhập xuất

➤ Một số loại hằng:

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 ngoặc 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: "Turbo C"

Lưu ý:



Hằng ký tự có thể tham gia các phép toán như mọi số nguyên khác.
Một hằng chuỗi được trử tận cùng bằng ký tự nul ("\\0").



5. Khai báo

- Từ khóa
- Các phép toán
- Kiểu dữ liệu
- Tên & đặt tên
- **Khai báo**
- Lệnh, khối lệnh
 - Nhập xuất

➤ Biểu thức:

Biểu thức là tổ hợp các **hằng**, **biến** và **các toán tử** xác định các phép toán.

Ví dụ: $(2 + 3 * i) / 10$

➤ Phép gán:

- Phép gán đơn: $\text{<biến> = <biểu_thức>;}$

Ví dụ: $a = 5;$

- Phép gán kép:

Ví dụ: $a = b = c = 10$ //a và b và c đều = 10

Lưu ý:



Trong C cho phép thực hiện một số phép gán đặc biệt như: tự cộng, tự trừ, tự nhân, tự chia nguyên, & tự chia dư.



6. Lệnh và khối lệnh

- Từ khóa
- Các phép toán
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- **Lệnh, khối lệnh**
 - Nhập xuất

➤ **Lệnh:**

là một tổ hợp có nghĩa (ở 2 vế) các biểu thức và chấm dứt bởi dấu chấm phẩy (;).

Ví dụ:

`i = (2 + 3) * 10; i = 2 + 3 * 10;`

`j = 6 % 4; k = i + j;`

`return 0;`

`exit(0);`

`printf ("Ket thuc chuong trinh");`



6. Lệnh và khối lệnh

- Từ khóa
- Các phép toán
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- **Lệnh, khối lệnh**
 - Nhập xuất

➤ **Khối lệnh:**

Hay **Cụm câu lệnh** là nhóm câu lệnh bắt đầu bằng ngoặc **{** mở và kết thúc bằng ngoặc **}** đóng.

Khối lệnh được thi hành như một câu lệnh bởi bộ biên dịch C.

Ví dụ:

```
for ( . . . ) {  
    s3 = s1 + s2;  
    mul = s3 * c;  
    remainder = sum % c;  
}
```



7. Lệnh nhập, xuất

- Từ khóa
- Các phép toán
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- **Nhập xuất**

➤ Lệnh xuất (printf):

Hàm **printf()** trong C được sử dụng để in ra các ký tự (**%c**); chuỗi (**%s**); các giá trị số nguyên (**%d, %ld**), số thực (**%f, %lf**) hay số thập phân, bát phân,... và hiển thị lên màn hình biên dịch.

➤ Lệnh nhập (scanf):

Hàm **scanf()** trong C được sử dụng để đọc vào ký tự, chuỗi, các giá trị số mà người dùng nhập từ bàn phím.

Hàm **scanf()** nhận vào **tham số là địa chỉ** (**<tên_biến>**) của một biến đã được khai báo.



7. Lệnh nhập, xuất

- Từ khóa
- Các phép toán
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- **Nhập xuất**

Ví dụ 1: Thực hiện các lệnh nhập xuất

```
#include <stdio.h>
int main(){
    // Xuất printf()
    printf("Hello World!");
    // Nhập 1 số nguyên
    int a;
    printf("\nNhap a = ");
    scanf("%d", &a);
    // xuất có định dạng (format)
    printf("Gia tri a = %d", a);
```

```
int i;
for (i = 1; i < 5; i++) {
    printf("\nLan");
    printf("\n%d", i);
}
}
```

Click xem kết quả



7. Lệnh nhập, xuất

- Từ khóa
- Các phép toán
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- **Nhập xuất**

Ví dụ 2: Thực hiện các lệnh nhập xuất trên số thực

*/*Viết chương trình nhập vào chiều dài, chiều rộng của hình chữ nhật. Tính: chu vi, diện tích*/*

```
#include <stdio.h>
#include <conio.h>
int main(){
int cd, cr;
long cv, dt;
printf ("\nNhap chieu dai,
chieu rong: ");
scanf ("%d, %d",&cd, &cr);
```

//Tính chu vi

```
cv = (cd + cr) * 2;
```

//Tính diện tích

```
dt = (cd * cr);
```

//In chu vi, diện tích

```
printf ("\nChu vi (%d, %d) =
%d", cd, cr, cv);
```

```
printf ("\nDiện tích (%d, %d) =
%d", cd, cr, dt);
```

```
getch();
```

```
}
```



7. Lệnh nhập, xuất

- Từ khóa
- Các phép toán
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- **Nhập xuất**

Ví dụ 3: Thực hiện các lệnh nhập xuất trên số nguyên

*/*Viết chương trình nhập vào 3 số nguyên dương. In ra trung bình có 2 số lẻ thập phân*/*

Click xem kết quả

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
int main(){
    int a, b, c;
    float tb;
    printf ("\nNhap a, b, c = ");
    scanf ("%d, %d, %d",&a, &b, &c);
    tb = (float) (a+b+c)/3;
    printf ("\nTrung bình = %3.2f", tb);
    getch();
}
```




7. Lệnh nhập, xuất

- Từ khóa
- Các phép toán
- Kiểu dữ liệu
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- **Nhập xuất**

Ví dụ 4: Thực hiện các lệnh nhập xuất trên ký tự

*/*Viết ch/tr nhập vào một ký tự. In ra ký tự nhập, in ra mã ASCII của ký tự nhập*/*

Click xem kết quả

```
#include <stdio.h>
#include <conio.h>
#include <ctype.h>
int main(){
    char kt;
    printf ("\nNhap ky tu: ");
    scanf ("%c", &kt);
    printf ("\nKy tu nhap: %c", kt);
    printf ("\nMa Ascii cua ky tu: %d",
            (kt));

    getch();
}
```



8. Chuyển đổi kiểu

-
- Tên & đặt tên
- Khai báo
- Lệnh, khối lệnh
- Nhập xuất
- **Chuyển đổi kiểu**

Hay gọi là **ép kiểu** là cách để chuyển đổi một biến từ kiểu dữ liệu này **sang** kiểu dữ liệu khác.

- Chuyển đổi kiểu theo chiều:

char □ int □ **long** □ float □ double □ long double

Ví dụ:

- *Số nguyên (a) chia cho số nguyên (b) để được số thực ta thực hiện chuyển đổi kiểu như sau:*

(float)a/b;



9. Bài tập



2.1. Viết chương trình nhập vào 2 số nguyên a, b. In lên màn hình theo yêu cầu.

- Dòng 1: Tổng 2 số
- Dòng 2: Hiệu 2 số
- Dòng 3: Tích 2 số
- Dòng 4: Chia lấy dư 2 số (mod)

2.2. Viết chương trình nhập vào chiều dài 3 cạnh tam giác. Tính chu vi và diện tích tam giác.

$$CV = a + b + c$$

$$DT = \text{sqrt} (CV/2 * (CV/2-a) * (CV/2-b) * (CV/2-c))$$

2.3. Viết chương trình nhập vào một số nguyên dương có đúng 3 chữ số, in ra số đảo ngược.

VD: nhập vào: 123 □ in ra: 321



9. Bài tập (tt)



2.4. Viết chương trình nhập vào bán kính của một hình tròn in ra chu vi, diện tích của hình tròn.

2.5. Viết ch/tr nhập vào một ký tự thường. Yêu cầu:

- Dòng 1: in ký tự nhập
- Dòng 2: in ký tự kế tiếp
- Dòng 3: in ký tự hoa của ký tự nhập.
- Dòng 4: in mã ascii của ký tự nhập.

2.6. Viết chương trình nhập 3 số nguyên hoặc số thực. Tính tổng 3 số vừa nhập.

2.7. Sử dụng hàm toán học sqrt để in căn bậc 2 của một số nguyên nhập vào.



TỔNG KẾT CHƯƠNG 2



- Tìm hiểu về từ khóa, phép toán
- Tìm hiểu về các kiểu dữ liệu: số nguyên, số thực, ký tự, kiểu logic;
- Nhập xuất trên kiểu dữ liệu;
- Thực hành kiểu dữ liệu.

