



TRƯỜNG ĐẠI HỌC THỦY LỢI
Khoa Công nghệ thông tin
Bộ môn Tin học và KTTT

NHẬP MÔN LẬP TRÌNH

INTRODUCTION TO COMPUTER PROGRAMMING

Giảng viên: TS.GVC Bùi Thị Thanh Xuân

Email: xuanbtt@tlu.edu.vn

Điện thoại: 0902001581



Chương 2. Tổng quan về ngôn ngữ C



Chương 2. Tổng quan về ngôn ngữ C



- ❖ Làm quen với lập trình C
- ❖ Dữ liệu và biểu thức
- ❖ Nhập xuất trong C



Sự ra đời của C



❖ Nhu cầu viết lại HĐH Unix cho các hệ máy tính khác nhau

➤ Dùng ngôn ngữ Assembly

- Công việc nặng nề, phức tạp
- Khó chuyển đổi chương trình giữa các hệ máy tính khác nhau

➤ Cần ngôn ngữ mới

- Đơn giản việc lập trình
- Tính khả chuyển cao

❖ C ra đời tại Bell Lab thuộc tập đoàn AT&T

➤ Tác giả Brian W. Kernighan & Dennis Ritchie

➤ Dựa trên ngôn ngữ BCPL & B

➤ Phát triển năm 1970, hoàn thành 1972



Ngôn ngữ lập trình C



❖ Đặc điểm

- Ngôn ngữ lập trình hệ thống
- Tính khả chuyển, linh hoạt cao
- Có thể mạnh trong xử lý dữ liệu số, văn bản, cơ sở dữ liệu,...

❖ Phạm vi sử dụng

- Viết các chương trình hệ thống
 - Hệ điều hành Unix có 90% mã C, 10% mã hợp ngữ
- Các trình điều khiển thiết bị (device driver)
- Xử lý ảnh



Ngôn ngữ lập trình C



❖ Các phiên bản

- ANSI C: C chuẩn (1989)
- Các phiên bản khác xây dựng dựa trên ANSI C
 - Đưa thêm thư viện
 - Bổ sung cho thư viện chuẩn của ANSI C

❖ Các trình biên dịch phổ biến

- Microsoft Visual C/C++ (Microsoft)
- gcc (GNU Project)



Khung chương trình C



Khai báo các tệp tiêu đề

#include

Khai báo các đối tượng toàn cục

- Định nghĩa kiểu dữ liệu mới
- Các biến, hằng
- Các nguyên mẫu hàm (prototype)

Định nghĩa hàm main()

{

}

Định nghĩa các hàm đã khai báo nguyên mẫu



Chương trình đầu tiên: Hello world!



```
1. #include <stdio.h>
2. int main() { //Không cần tham số dòng lệnh
3.     printf("Hello world! \n");
4.     return 0; //Trả về giá trị 0
5. }
```

- ❖ Khai báo thư viện **stdio.h**, đây là thư viện vào ra chuẩn (standard input output) chứa khai báo hàm **printf**
- ❖ Điểm bắt đầu thực hiện của chương trình. Máy tính thực hiện các câu lệnh nằm trong cặp ngoặc {} của **main()**
- ❖ Hàm **printf** in ra một xâu. Chú ý dấu xuống dòng (\n)
- ❖ Hàm main trả về giá trị 0 (0 thường dùng để thể hiện chương trình hoạt động bình thường, không có lỗi).



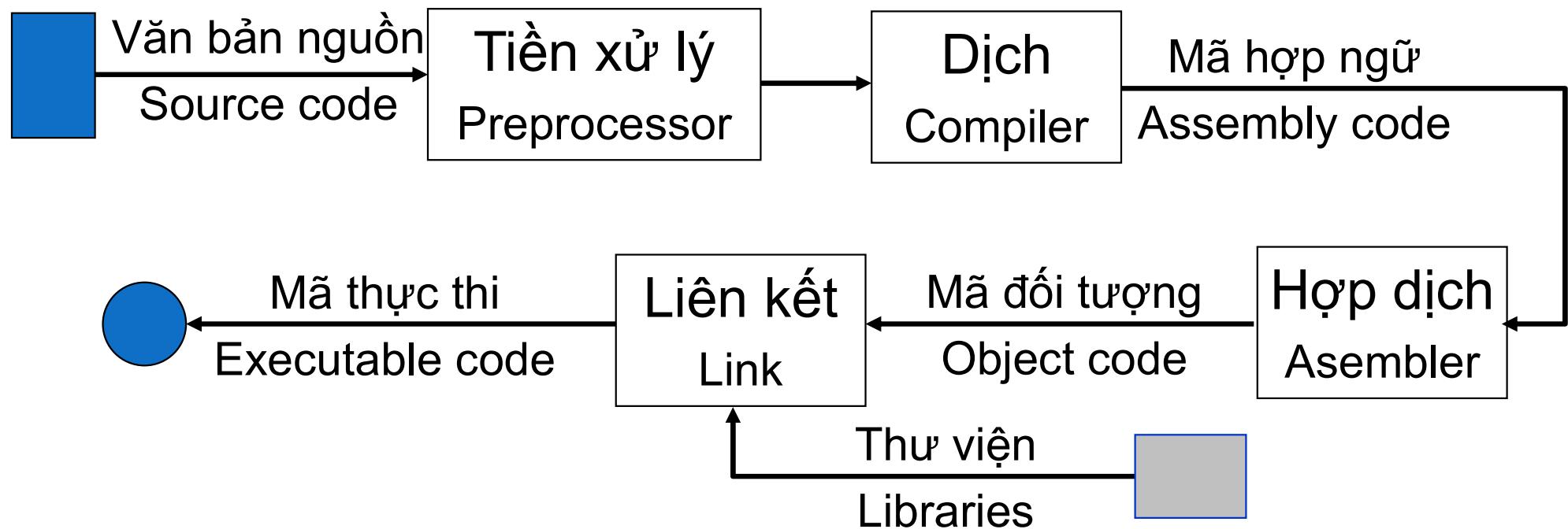
Biên dịch chương trình



❖ Chương trình được viết bằng ngôn ngữ bậc cao phải được dịch ra mã máy để thực thi

➤ Công việc dịch được thực hiện bởi trình biên dịch (compiler)

❖ Các giai đoạn dịch chương trình



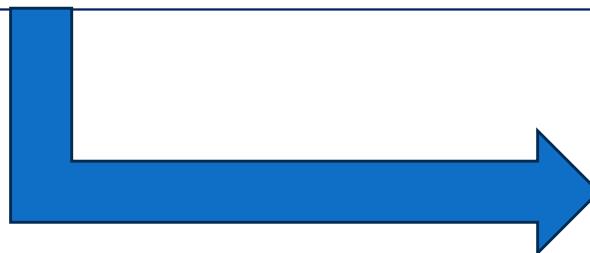


Biên dịch chương trình



vd1.c

```
1 #include <stdio.h>
2 int main(){
3     printf("Dai hoc Thuy loi\nChao Tan sinh vien K65!");
4     return 0;
5 }
```



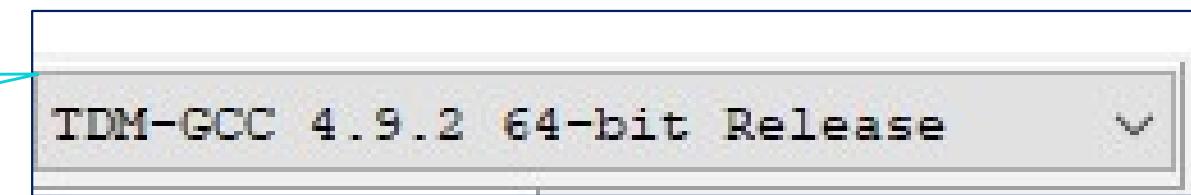
Biên dịch xong
được file vd1.exe

Command Prompt

```
C:\Users\longt\OneDrive\Desktop>vd1.exe
Dai hoc Thuy loi
Chao Tan sinh vien K65!
```

- ❖ **GCC (GNU Compiler Collection)** là một tập hợp các trình biên dịch (compilers) được phát triển bởi dự án GNU. GCC hỗ trợ các ngôn ngữ lập trình như C, C++, Objective-C, Fortran, Ada và Java

Trình biên dịch
TDM-GCC với
Dev-C++





Các công cụ để lập trình C



1. Visual Studio Code
2. Clion
3. Visual Studio
4. Xcode
5. Eclipse
6. Netbean IDE
7. Code::Blocks
8. Dev C++
9. PyCharm
10. CodeLite





Các công cụ để lập trình C



❖ Compiler:

- Công cụ biên dịch mã nguồn thành chương trình
- Visual C, gcc...

❖ IDE: Integrated Development Environment

- Cung cấp giao diện giữa lập trình viên và compiler
- Thực hiện tính năng: biên dịch, thực thi, gỡ lỗi...
- MS Visual Studio, Eclipse, Code::Blocks,...

Dev-Cpp 5.1.1





Cài đặt Dev-Cpp



Dev-C++ download | SourceForge + ↻

sourceforge.net/projects/orwelldevcpp/

SOURCEFORGE Help

Open Source Software Business Software Resources Sync your GitHub to SourceForge

ghibli.com/mashup
Plug AI into SQL... 0.2K Stars 869 Forks 115 Open Issues

Build AI into SQL Learn More

Advertisement

Home / Browse / Development / Integrated Development Environments (IDE) / Dev-C++

 **Dev-C++**
A free, portable, fast and simple C/C++ IDE
Brought to you by: [orwelldevcpp](#)

OPEN SOURCE EXCELLENCE SOURCEFORGE

★★★★★ 138 Reviews Downloads: 156,608 This Week Last Update: 2016-11-29

  **Download**  Get Updates  Share This

Windows | BSD



Cài đặt Dev-Cpp



❖ Tải Dev-C++

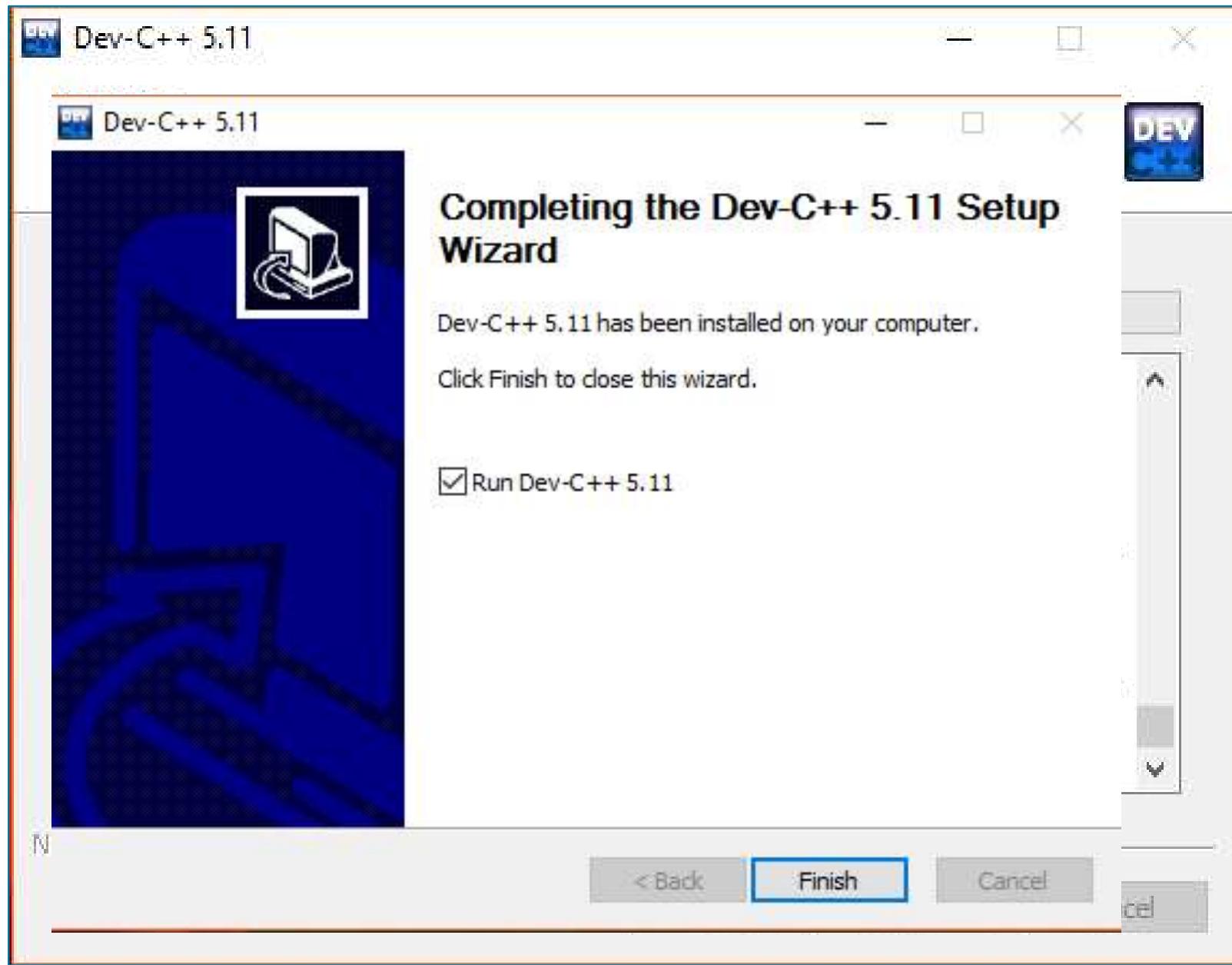
- Vào Google ⇒ Dev-C++
- Tìm đến phiên bản thích hợp, thực hiện tải về
 - Dev-Cpp 5.11 TDM-GCC 4.9.2 Setup.exe



❖ Thực thi file tải về, làm theo hướng dẫn

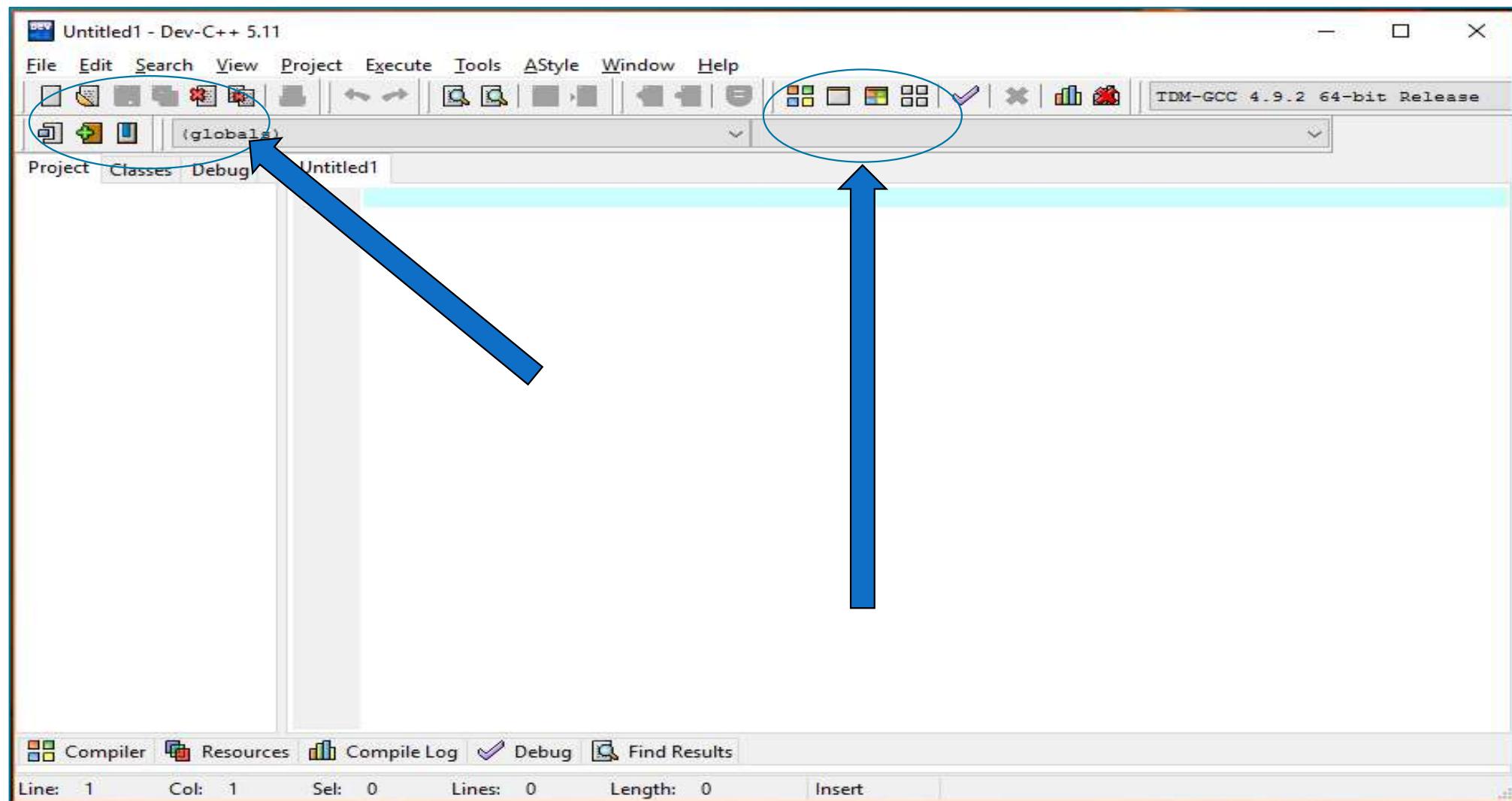


Cài đặt Dev-Cpp





Màn hình giao diện Dev-C++





Màn hình giao diện Dev-C++



The screenshot shows the Dev-C++ integrated development environment. The menu bar includes File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and search. The project manager shows a single file named 'hello.c'. The code editor displays the following C program:

```
1 #include <stdio.h>
2 int main(){
3     printf("Hello world");
4     return 0;
5 }
```

The output window on the right shows the program's execution results:

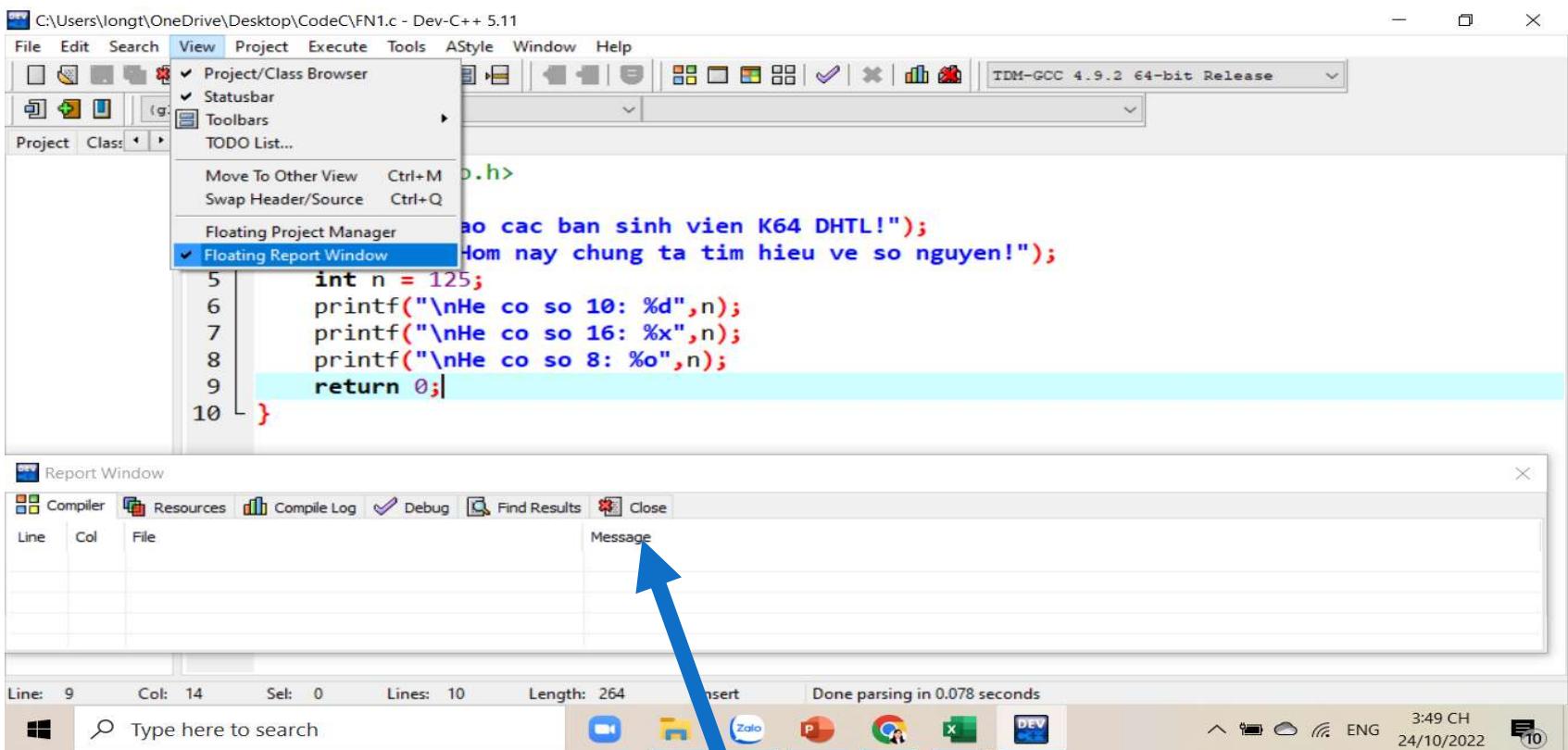
```
Hello world
-----
Process exited after 0.766 seconds with return value 0
Press any key to continue . . .
```



Khắc phục lỗi không thấy hộp thoại thông báo lỗi trong Dev C++



❖ Bước 1: Vào menu View / Floating Report Window



❖ Bước 2: Click vào nút **Close**



Các phần tử cơ bản



- ❖ Tập ký tự
- ❖ Từ khóa
- ❖ Định danh
- ❖ Các kiểu dữ liệu
- ❖ Hằng
- ❖ Biến
- ❖ Hàm
- ❖ Biểu thức
- ❖ Câu lệnh
- ❖ Chú thích



1. Tập ký tự



- ❖ Ký tự là các phần tử cơ bản tạo nên chương trình
- ❖ Chương trình: Tập các câu lệnh nhằm giải quyết nhiệm vụ đặt ra
- ❖ Câu lệnh: là các từ (từ vựng) liên kết với nhau theo cú pháp của ngôn ngữ lập trình
 - Ví dụ:
 while(i < N){ }
 for(i = 0; i<n; i++)
- ❖ Các từ: Tổ hợp các ký tự theo nguyên tắc xây dựng từ vựng
 - Ví dụ: TenFile, BaiTap2...



Tập ký tự trong C



- ❖ 26 chữ cái hoa: A B C ... X Y Z
- ❖ 26 chữ cái thường: a b c ... x y z.
- ❖ 10 chữ số: 0 1 2 3 4 5 6 7 8 9.
- ❖ Các kí hiệu toán học: + - * / = < >
- ❖ Các dấu ngăn cách: . ; , : space tab
- ❖ Các dấu ngoặc: () [] { }
- ❖ Các kí hiệu đặc biệt: _ ? \$ & # ^ \ ! ' " ~
- ...



2. Từ khóa (keyword)



- ❖ Được định nghĩa sẵn trong C
- ❖ Dành riêng cho các mục đích xác định
 - Đặt tên cho kiểu dữ liệu:
 - int, float, double...
 - Mô tả các lệnh, các cấu trúc lập trình
 - if, else, while, case, for...



Từ khóa thông dụng



❖ Lưu ý: Tất cả từ khóa trong C đều viết bằng **chữ cái thường**.

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



3. Định danh (Identifier)

❖ Định danh (Tên) là một dãy các kí tự dùng để gọi tên các đối tượng trong chương trình.

➤ Các đối tượng trong chương trình

- Biến
- Hằng số
- Hàm
- Kiểu dữ liệu

❖ Định danh có thể được đặt bởi

➤ Ngôn ngữ lập trình → các từ khóa
➤ Người lập trình



Quy tắc đặt định danh trong C



- ❖ Định danh được bắt đầu bởi **chữ cái** hoặc **dấu gạch dưới " _** (underscore)
- ❖ Các kí tự tiếp theo chỉ có thể là: **chữ cái**, **chữ số** hoặc **dấu gạch dưới " _**
- ❖ Định danh do người lập trình đặt **không được trùng với các từ khóa** của C
- ❖ Độ dài định danh tùy thuộc phiên bản C
- ❖ Chú ý: C là ngôn ngữ có **phân biệt chữ hoa và chữ thường**



Ví dụ



❖ Định danh hợp lệ:

- i, x, y, a, b, _function,
- _MY_CONSTANT, PI, gia_tri_1

❖ Định danh không hợp lệ

- 1_a, 3d, 55x (bắt đầu bằng chữ số)
- so luong, sin() (có kí tự không hợp lệ, dấu cách, dấu ngoặc..)
- int, char (trùng với từ khóa của C)



Một số quy ước (code convention)



- ❖ Định danh tên có **tính gợi nhớ**
- ❖ Nên sử dụng **dấu gạch dưới** để phân tách các định danh gồm nhiều từ
 - Có thể dùng cách viết hoa chữ cái đầu mỗi từ
 - Ví dụ: sinh_vien, sinhVien, SinhVien
- ❖ Quy ước thường được sử dụng:
 - **Hằng số dùng chữ cái in hoa**
 - Ví dụ: PI, EPSILON,...
 - Các biến, hàm, cấu trúc dùng **chữ cái thường**
 - Biến điều khiển vòng lặp: i, j, k...
 - Hàm: NhapDuLieu, TimKiem,...
 - Cấu trúc: SinhVien, MathHang,...



4. Các kiểu dữ liệu

- ❖ Một kiểu dữ liệu là một tập hợp các giá trị mà một dữ liệu thuộc kiểu dữ liệu đó có thể nhận được.
 - Ví dụ: Một đối tượng kiểu **char** của C sẽ là
 - Một số nguyên (Số nguyên có dấu, 1 byte)
 - Giá trị thuộc khoảng: [-128...127]
- ❖ Trên một kiểu dữ liệu, xác định một số phép toán đối với các dữ liệu thuộc kiểu dữ liệu tương ứng.
- ❖ Với số nguyên, thêm từ khóa **unsigned** để chỉ ra số không dấu



4. Các kiểu dữ liệu



Kiểu dữ liệu	Kích thước (bytes)	Miền	Ký tự định dạng
short int	2	-32,768 to 32,767 hay [-2^15, 2^15-1]	%hd
unsigned short int	2	0 to 65,535	%hu
unsigned int	4	0 to 4,294,967,295	%u
int	4 (2)	-2,147,483,648 to 2,147,483,647	%d
long int	4	-2,147,483,648 to 2,147,483,647	%ld
unsigned long int	4	0 to 4,294,967,295	%lu
long long int	8	-(2^63) to (2^63)-1	%lld
unsigned long long int	8	0 đến 18,446,744,073,709,551,615 hay [0, 2^64-1]	%llu
char	1	-128 to 127	%c
unsigned char	1	0 to 255	%c
float	4	1.2E-38 to 3.4E+38	%f
double	8	1.7E-308 to 1.7E+308	%lf
long double	10-16	3.4E-4932 to 1.1E+4932	%Lf



Các kiểu kết hợp



Với số nguyên, thêm từ khóa **unsigned** để chỉ ra số **không dấu**

Kiểu dữ liệu	Ý nghĩa	Kích thước	Miền dữ liệu
unsigned char	Số nguyên không dấu	1 byte	0 ÷ 255
unsigned short	Số nguyên không dấu	2 byte	0÷65.535
unsigned int	Số nguyên không dấu	2 hoặc 4 byte	
unsigned long	Số nguyên không dấu	4 byte	0 ÷ 4,294,967,295
unsigned long int			
long double	Số thực dấu phẩy động,	10 byte/16 byte	$\pm 3.4E-4932 \div \pm 1.1E+4932$
void	Là kiểu đặc biệt, không có kích thước		



Minh họa kích thước kiểu dữ liệu trên Turbo C



File Edit Search Run Compile Debug Project Options Window Help
[] Help 1=[]

Data Types

Type	Length	Range
unsigned char	8 bits	0 to 255
char	8 bits	-128 to 127
enum	16 bits	-32,768 to 32,767
unsigned int	16 bits	0 to 65,535
short int	16 bits	-32,768 to 32,767
int	16 bits	-32,768 to 32,767
unsigned long	32 bits	0 to 4,294,967,295
long	32 bits	-2,147,483,648 to 2,147,483,647
float	32 bits	3.4 * (10**-38) to 3.4 * (10**+38)
double	64 bits	1.7 * (10**-308) to 1.7 * (10**+308)
long double	80 bits	3.4 * (10**-4932) to 1.1 * (10**+4932)

F1 Help on help Alt-F1 Previous topic Shift-F1 Help index Esc Close help



Minh họa chạy trên Turbo C



The screenshot shows the Turbo C IDE interface. The menu bar includes File, Edit, Search, Run, Compile, Debug, Project, Options, Window, and Help. The title bar displays "C:\TURBOC3\BIN>TC". The main window has two panes: the left pane shows the source code, and the right pane shows the output of the program. The output pane displays the following text:

```
C:\TURBOC3\BIN>TC
char: 1 bytes
short int: 2 bytes
int: 2 bytes
long int: 4 bytes
float: 4 bytes
double: 8 bytes
long double: 10 bytes
```

The source code in the left pane is:

```
printf("char: %d bytes\n", sizeof(a1));
printf("short int: %d bytes\n", sizeof(a2));
printf("int: %d bytes\n", sizeof(a3));
printf("long int: %d bytes\n", sizeof(a4));
printf("float: %d bytes\n", sizeof(a5));
printf("double: %d bytes\n", sizeof(a6));
printf("long double: %d bytes\n", sizeof(a7));
return 0;
}
```

The status bar at the bottom indicates "1:1" and provides keyboard shortcuts: F1 Help, ↑↔ Scroll.



Minh họa chạy trên Turbo C



```
1 #include <stdio.h>
2
3 int main(){
4     char a1;
5     short int a2;
6     int a3;
7     long int a4;
8     float a5;
9     double a6;
10    long double a7;
11    printf("char: %d bytes\n", sizeof(a1));
12    printf("short int: %d bytes\n", sizeof(a2));
13    printf("int: %d bytes\n", sizeof(a3));
14    printf("long int: %d bytes\n", sizeof(a4));
15    printf("float: %d bytes\n", sizeof(a5));
16    printf("double: %d bytes\n", sizeof(a6));
17    printf("long double: %d bytes\n", sizeof(a7));
18    return 0;
19 }
```

Running Turbo C Project

```
Turbo C++ Version 3.00 Copyright (c) 1992 Borland International
main.c:
Turbo Link Version 5.0 Copyright (c) 1992 Borland International

        Available memory 4142224
char: 1 bytes
short int: 2 bytes
int: 2 bytes
long int: 4 bytes
float: 4 bytes
double: 8 bytes
long double: 10 bytes

Press any key to continue.
```



Minh họa chạy trên Dev-C++



```
1 #include <stdio.h>
2
3 int main(){
4     char a1;
5     short int a2;
6     int a3;
7     long int a4;
8     float a5;
9     double a6;
10    long double a7;
11    printf("char: %d bytes\n", sizeof(a1));
12    printf("short int: %d bytes\n", sizeof(a2));
13    printf("int: %d bytes\n", sizeof(a3));
14    printf("long int: %d bytes\n", sizeof(a4));
15    printf("float: %d bytes\n", sizeof(a5));
16    printf("double: %d bytes\n", sizeof(a6));
17    printf("long double: %d bytes\n", sizeof(a7));
18    return 0;
19 }
```

```
char: 1 bytes
short int: 2 bytes
int: 4 bytes
long int: 4 bytes
float: 4 bytes
double: 8 bytes
long double: 16 bytes
```

Hàm `sizeof`
để lấy kích
thước của
kiểu dữ liệu



Minh họa chạy trên website



Result Size: 532 x 349

```
#include <stdio.h>

int main(){
    char a1;
    short int a2;
    int a3;
    long int a4;
    float a5;
    double a6;
    long double a7;
    printf("char: %d bytes\n", sizeof(a1));
    printf("short int: %d bytes\n", sizeof(a2));
    printf("int: %d bytes\n", sizeof(a3));
    printf("long int: %d bytes\n", sizeof(a4));
    printf("float: %d bytes\n", sizeof(a5));
    printf("double: %d bytes\n", sizeof(a6));
    printf("long double: %d bytes\n", sizeof(a7));
    return 0;
}
```

char: 1 bytes
short int: 2 bytes
int: 4 bytes
long int: 8 bytes
float: 4 bytes
double: 8 bytes
long double: 16 bytes



Chú ý độ lớn của kiểu dữ liệu



```
1 #include <stdio.h>
2
3 int main(){
4     unsigned char n;//so nguyen khong dau 1 byte
5     printf("Nhập số nguyên không dấu 1 byte, N = ");
6     scanf("%u",&n);
7     printf("In lại số N = %u", n);
8     return 0;
9 }
```

```
Nhập số nguyên không dấu 1 byte, N = 230
In lại số N = 230
-----
Process exited after 4.42 seconds with return value 0
Press any key to continue . . .
```

```
Nhập số nguyên không dấu 1 byte, N = 270
In lại số N = 14
-----
Process exited after 8.141 seconds with error code 14
Press any key to continue . . .
```

Do số **270** nhập vào
vượt quá kiểu
unsigned char nên bị
sai!



Chú ý độ lớn của kiểu dữ liệu



```
1 #include <stdio.h>
2
3 int main(){
4     unsigned short int n;//so nguyen khong dau 2 byte
5     printf("Nhập số nguyên không dấu 2 byte, N = ");
6     scanf("%hu",&n);
7     printf("In lại số N = %hu", n);
8     return 0;
9 }
```

```
Nhập số nguyên không dấu 2 byte, N = 70000
In lại số N = 4464
-----
Process exited after 9.387 seconds with return value 0
Press any key to continue . . .
```

Do số **70000** nhập
vào vượt quá kiểu
unsigned short int
nên bị sai!

```
Nhập số nguyên không dấu 2 byte, N = 53400
In lại số N = 53400
-----
Process exited after 5.319 seconds with return value 0
Press any key to continue . . .
```



Ví dụ kiểu int



Một số phép toán được định nghĩa trên kiểu dữ liệu số nguyên của C

Tên phép toán	Ký hiệu	Ví dụ
Đảo dấu	-	
Cộng; Trừ; Nhân	+ - *	
Chia lấy nguyên	/	17/3 → 5
Chia lấy phần dư	%	17%3 → 2
So sánh	> < >= <= == !=	
<u>Phép toán trên bit:</u> AND; OR; XOR; NOT, Shift,...	& ^ ~ << >>	$3^17 \rightarrow 18$ $\sim 3 \rightarrow -4$



5. Hằng



- ❖ Hằng là đại lượng có giá trị không đổi trong chương trình.
- ❖ Giá trị hằng do người lập trình xác định.
- ❖ Các khái niệm
 - Literals
 - Constant
 - Definition (macro)
- ❖ Các loại hằng
 - Hằng số nguyên
 - Hằng số thực
 - Hằng ký tự
 - Hằng chuỗi/xâu ký tự



Hằng số nguyên (int literal)



Trong C, hằng số nguyên có thể biểu diễn dưới các dạng: cơ số 10, 8, 16

Hệ 10	Hệ 16	Hệ 8
2011	0x7DB	03733
396	0x18C	0614



Ví dụ



```
1 #include <stdio.h>
2 int main(){
3     printf("Chao cac ban sinh vien K64 DHTL!");
4     printf("\nHom nay chung ta tim hieu ve so nguyen!");
5     int n = 125;
6     printf("\nHe co so 10: %d",n);
7     printf("\nHe co so 16: %x",n);
8     printf("\nHe co so 8: %o",n);
9     return 0;
10 }
```

```
Chao cac ban sinh vien K64 DHTL!
Hom nay chung ta tim hieu ve so nguyen!
He co so 10: 125
He co so 16: 7d
He co so 8: 175
```

```
-----  
Process exited after 0.01571 seconds with return value 0  
Press any key to continue . . .
```



Ví dụ



```
1 #include <stdio.h>
2 int main() {
3     int n = 125;
4     printf("%d",n);
5     printf("\n%x",n);
6     printf("\n%o",n);
7     return 0;
8 }
```

125
7d
175

Proce
Press

```
1 #include <stdio.h>
2 int main() {
3     int n = 0x7d;
4     printf("%d",n);
5     printf("\n%x",n);
6     printf("\n%o",n);
7     return 0;
8 }
```

125
7d
175

Proce
Press

```
1 #include <stdio.h>
2 int main() {
3     int n = 0175;
4     printf("%d",n);
5     printf("\n%x",n);
6     printf("\n%o",n);
7     return 0;
8 }
```

125
7d
175

Process e
Press any



Hằng số thực (floating point literal)



- Trong C, hằng số thực có thể biểu diễn dưới các dạng
 - Dạng số thực dấu phẩy tĩnh
 - Dạng số thực dấu phẩy động

Số thực dấu phẩy tĩnh	Số thực dấu phẩy động
3.14159	31.4159E-1
123.456	12.3456E+1 hoặc 1.23456E+2



Hằng ký tự (character literal)



- Hằng ký tự có thể biểu diễn theo hai cách
 - Đặt ký hiệu của ký tự giữa hai dấu nháy đơn
 - Dùng mã ASCII của ký tự:
 - Số thứ tự của ký tự đó trong bảng mã ASCII
 - Là số nguyên → tuân thủ quy tắc biểu diễn số nguyên

Ký tự	Dùng nháy đơn	Dùng mã ASCII
Chữ cái A	'A'	65, 0x41, 0101
Dấu nháy đơn	'\'	39, 0x27, 047
Ký tự tab	'\t'	9, 0x09, 011



Hằng xâu ký tự (string literal)



❖ Hằng xâu kí tự được biểu diễn bằng dãy các kí tự trong cặp dấu nháy kép.

❖ Ví dụ:

"ngon ngu lap trinh C"

"Nhập môn lập trình"

"chúc mừng năm mới"



6. Biến (variable)

- ❖ Biến là đối tượng lưu giữ dữ liệu và có thể thay đổi giá trị khi chương trình được thực thi.
- ❖ Biến được đặt (allocate) trong các ô nhớ thuộc bộ nhớ chính của máy tính.
- ❖ Số ô nhớ dành cho 1 biến phụ thuộc kiểu dữ liệu của biến.
 - VD mỗi biến kiểu float chiếm 4byte trong bộ nhớ.
- ❖ Tên biến phải được đặt theo quy tắc đặt tên.



7. Hàm (function)

❖ Hàm là chương trình con có chức năng

- Nhận dữ liệu đầu vào (các tham số vào)
- Thực hiện một công việc nào đó
- Có thể trả về một kết quả gọi là giá trị của hàm
 - Ví dụ: hàm sin(x)
 - $\sin(3.14/2) \rightarrow 1.000$
 - $\sin(3.14/6) \rightarrow 0.499770$

❖ Có 2 loại hàm:

- ❖ Hàm có sẵn của các thư viện C cung cấp
- ❖ Hàm do người lập trình tự định nghĩa

❖ Chi tiết về hàm sẽ được học ở phần sau



Một số hàm toán học (trong math.h)



Hàm	Ý nghĩa	Ví dụ
<code>sqrt(x)</code>	Căn bậc 2 của x	<code>sqrt(16.0) → 4.0</code>
<code>pow(x,y)</code>	x mũ y (x^y)	<code>pow(2,3) → 8</code>
<code>fabs(x)</code> <code>abs(n)</code>	Trị tuyệt đối của số thực x ($ x $) Trị tuyệt đối của số nguyên n ($ n $)	<code>fabs(-5.5) → 5.5</code> <code>abs(-5) → 5</code>
<code>exp(x)</code>	e mũ x (e^x)	<code>exp(1.0) → 2.71828</code>
<code>log(x)</code>	Logarithm tự nhiên của x ($\ln x$)	<code>log(2.718) → 0.999</code>
<code>log10(x)</code>	Logarithm cơ số 10 của x ($\log x$)	<code>log10(100) → 2.00</code>
<code>sin(x)</code> <code>cos(x), tan(x)</code>	Các hàm lượng giác	
<code>ceil(x)</code>	Số nguyên nhỏ nhất không nhỏ hơn x ($\lceil x \rceil$)	<code>ceil(2.5)=3</code> <code>ceil(-2.5)=-2</code>
<code>floor(x)</code>	Số nguyên lớn nhất không lớn hơn x ($\lfloor x \rfloor$)	<code>floor(2.5)=2</code> <code>floor(-2.5)=-3</code>

#include <math.h>



Một số hàm toán học (trong math.h)



#include <math.h>

```
1 #include <stdio.h>
2 #include <math.h> //Thieu khai bao se co CANH BAO
3
4 int main(){
5     int n; //So nguyen co dau 4 byte
6     float x; //So thuc 4 byte
7     double y; //So thuc 8 byte
8     printf("Nhập số nguyên N = "); scanf("%d",&n);
9     printf("Tri tuyet doi |%d| = %d", n, abs(n));
10    printf("\nNhập số thực x = "); scanf("%f",&x);
11    printf("Tri tuyet doi |%g| = %g", x, fabs(x));
12    printf("\nNhập số thực y = "); scanf("%lf",&y);
13    printf("Tri tuyet doi |%g| = %g", y, fabs(y));
14    return 0;
15 }
```

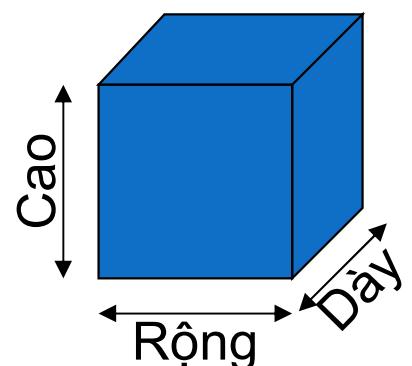
```
Nhap so nguyen N = -35
Tri tuyet doi |-35| = 35
Nhap so thuc x = -35.25
Tri tuyet doi |-35.25| = 35.25
Nhap so thuc y = -35.25
Tri tuyet doi |-35.25| = 35.25
-----
Process exited after 22.04 seconds with return value 0
Press any key to continue . . .
```

SAI: Khai
báo kiểu int,
nhưng nhập
số thực

```
Nhap so nguyen N = -12.25
Tri tuyet doi |-12| = 12
Nhap so thuc x = Tri tuyet doi |0.25| = 0.25
Nhap so thuc y = -12.25
Tri tuyet doi |-12.25| = 12.25
-----
Process exited after 25.36 seconds with return value 0
Press any key to continue . . .
```

8. Biểu thức

- ❖ Biểu thức là sự kết hợp các các toán hạng (operand) với các toán tử (operator) theo một quy tắc xác định.
 - Toán hạng có thể là biến, hằng, hàm, biểu thức...
 - Các toán tử rất đa dạng: cộng, trừ, nhân, chia...
- ❖ Khi biểu thức được tính toán sẽ trả về một kết quả.
- ❖ Ví dụ
 - Thể tích hình hộp: $V = \text{Rộng} * \text{Cao} * \text{Dày}$
 - Phép nhân (*) là toán tử
 - 3 toán hạng Rộng, Cao, Dày





9. Câu lệnh (statement)

- ❖ Câu lệnh diễn tả một hoặc một nhóm các thao tác trong giải thuật.
 - Chương trình được tạo thành từ dãy các câu lệnh.
- ❖ Các câu lệnh trong C, được kết thúc bởi dấu chấm phẩy (;)
 - Chú ý: lệnh khởi (sẽ học sau) không cần kết thúc bằng dấu ;



Phân loại



❖ Câu lệnh đơn:

- Những câu lệnh không chứa câu lệnh khác.
 - Ví dụ: Phép gán, gọi hàm, vào/ra dữ liệu

❖ Các câu lệnh phức:

- Những câu lệnh chứa câu lệnh khác.
 - Ví dụ: Lệnh khối (Tập các lệnh đơn nhóm lại với nhau và đặt trong cặp ngoặc nhọn ""{ }"")
- Các lệnh điều khiển cấu trúc chương trình
 - Ví dụ: Lệnh rẽ nhánh, lệnh lặp..



10. Chú thích (comment)



- ❖ Lời mô tả, giải thích văn tắt cho một câu lệnh, một đoạn chương trình hoặc cả chương trình
 - Giúp việc đọc hiểu chương trình dễ dàng hơn
 - Chú thích không phải là câu lệnh ⇒ không ảnh hưởng tới chương trình
 - Khi gấp chú thích, trình biên dịch sẽ bỏ qua
- ❖ Cách viết chú thích
 - Chú thích một dòng: sử dụng //
 - Chú thích nhiều dòng: sử dụng /* và */



Khung chương trình C



Khai báo các tệp tiêu đề

#include

Khai báo các đối tượng toàn cục

- Định nghĩa kiểu dữ liệu mới
- Các biến, hằng
- Các nguyên mẫu hàm (prototype)

Định nghĩa hàm main()

{

}

Định nghĩa các hàm đã khai báo nguyên mẫu



1. Khai báo các tệp tiêu đề



❖ Liệt kê danh sách thư viện sẽ được sử dụng trong chương trình

- Các hàm của C đều thuộc một thư viện nào đó
- Nếu không khai báo thư viện, trình biên dịch sẽ không hiểu được hàm (có thể báo lỗi)

❖ Cách thức (cú pháp) khai báo

- `#include <ThuVien.h>`
 - Thư viện phải nằm trong thư mục chứa các header file
 - Ví dụ: `#include <stdio.h>`
- `#include "ThuVien.h"`
 - Thư viện nằm trong thư mục hiện tại



2. Khai báo các đối tượng toàn cục



- ❖ Các đối tượng toàn cục có phạm vi sử dụng trong toàn bộ chương trình
 - Các kiểu dữ liệu mới
 - Các hằng, biến
 - Các nguyên hàm
- ❖ Tuân theo nguyên tắc khai báo đối tượng



2. Khai báo các đối tượng toàn cục



❖ Định nghĩa kiểu dữ liệu

➤ Cú pháp:

typedef <ĐịnhNghĩaKiểu> <Tên kiểu>

➤ Ví dụ:

```
typedef unsigned char byte;
```

```
typedef struct{
    float re, im;
}complex;
```



2. Khai báo các đối tượng toàn cục



❖ Khai báo hằng

```
const float PI = 3.1415;
```

```
#define Max 50
```

❖ Khai báo biến

```
int N;
```

```
float Delta, x1, x2;
```



2. Khai báo các đối tượng toàn cục (tiếp)



- ❖ Khai báo các nguyên mẫu hàm
- ❖ Khai báo thông tin về các hàm của người dùng sẽ được sử dụng trong chương trình
 - Tên hàm
 - Danh sách các kiểu tham số sẽ truyền vào
 - Kiểu dữ liệu trả về
- ❖ Ví dụ

```
float DienTichTamGiac(float a, float b, float c);  
int getMax(int Arr []);  
void swap(int * a, int * b); Có thể bỏ tên tham số  
void swap(int *, int *);
```



3. Định nghĩa hàm main()



- ❖ **Bắt buộc phải có**
- ❖ Là hàm đặc biệt trong C, đánh dấu điểm bắt đầu của mọi chương trình C
- ❖ Khi thực hiện một chương trình C, hệ thống sẽ gọi tới hàm main đầu tiên, sau đó sẽ thực hiện lần lượt các câu lệnh (bao gồm cả lời gọi tới các hàm khác) nằm trong hàm main()



3. Định nghĩa hàm main()



❖ Cú pháp

```
void main(){
```

```
.....
```

```
}
```

```
void main(int argc, char * argv[ ]){    ....}
```



3. Định nghĩa hàm main()



❖ Cú pháp

```
int main(){  
    ....;  
    return 0;  
}
```

```
int main(int argc, char * argv[ ]){  
    ....;  
    return 0;  
}
```



4. Định nghĩa các hàm đã khai báo

❖ Định nghĩa các hàm đã khai báo ở phần 3 (Phần khai báo nguyên mẫu - prototype)

➤ Phần khai báo nguyên mẫu mới chỉ khai báo các thông tin cơ bản về hàm, chưa xác định rõ hàm hoạt động như thế nào

❖ Ví dụ

```
float DienTichTamGiac(float a, float b, float c){  
    float p = (a+b+c)/2;  
    return sqrt(p*(p-a)*(p-b)*(p-c));  
}
```

$$S_{\Delta} = \sqrt{p(p - a)(p - b)(p - c)}$$



Chú ý



- ❖ Các phần không bắt buộc phải theo đúng thứ tự
- ❖ Nguyên tắc:
 - Mọi đối tượng cần phải được khai báo trước khi sử dụng
- ❖ Khi định nghĩa hàm được đặt trước hàm main() thì không cần khai báo nguyên mẫu hàm



Một minh họa chương trình C



```
#include <stdio.h>

#define PI 3.14159
const float G = 9.8;
float r, S;

void main(){
    printf("Cho ban kinh r = "); scanf("%f",&r);
    printf("Dien tich hinh tron S = %f\n", PI*r*r);
    printf("Gia toc trung truong G = %2.1f",G);
}
```

Khai báo tệp tiêu đề

Định nghĩa hằng,
Khai báo biến toàn cục

Hàm main()



Ví dụ về kiểu số nguyên, số thực



```
#include <stdio.h>
#include <math.h> //de su dung hang M_PI

void main(){
    int r = 3.5;
    printf("Dien tich hinh tron voi ban kinh r = %d la S = %f.\n", r, M_PI*r*r);
}
```

Khai báo kiểu
nguyên nhưng
khởi tạo là số
thực

Dien tich hinh tron voi ban kinh r = 3 la S = 28.274334.

Process exited after 0.05746 seconds with return value 57
Press any key to continue . . .



Ví dụ 1 chương trình C



```
1 //giai phuong trinh bac 1
2 #include <stdio.h>
3 float a, b, x;
4 void main(){
5     printf("Cho he so cua phuong trinh ax + b = 0.\n");
6     printf("a = "); scanf("%f",&a);
7     printf("b = "); scanf("%f",&b);
8     if(a == 0)
9         if( b == 0)
10             printf("PT vo so nghiem!");
11         else
12             printf("PT khong co nghiem!");
13     else
14         printf("PT co nghiem x = %5.2f", -b/a);
15 }
```



Biểu diễn hằng số (literal)



Kiểu dữ liệu	Ví dụ	Ý nghĩa
Số nguyên	123, -12 012, 03777 0x7F, 0x3fe15 39u 0267u, 0xFFu	Số hệ 10 Số hệ 8 Số hệ 16 Số không dấu
Số nguyên lớn (kiểu long)	12L, 07723L 0xFFL, -10L 0xFFUL, 0xFFFFU	
Số thực	3.1415 -12.3, .327 10e-12, -15.3E12 3.1415F, -12.F	



Khai báo biến



- ❖ Một biến phải được khai báo trước khi sử dụng
- ❖ Cú pháp khai báo:
`<KieuDuLieu> TenBien;`
`<KieuDuLieu> TenBien_1, ..., TenBien_N;`
- ❖ Ví dụ:
`//Khai báo biến x là một số nguyên 4byte có dấu
int x;
//Khai báo các biến y, z là các số thực 4 byte
float y, z;
//Sau khi khai báo, có thể sử dụng
x = 3; y = x + 1;`



Khai báo biến



- ❖ Sau khi khai báo, biến chưa có giá trị xác định.
 - Biến cần được gán giá trị trước khi sử dụng

- ❖ Cho phép kết hợp khai báo và khởi tạo biến

KieuDuLieu TenBien = GiaTriBanDau;

KieuDuLieu Bien1 = GiaTri1, BienN = Gia_TriN;

- ❖ Ví dụ:

//Khai báo biến nguyên a và khởi tạo giá trị bằng 3

int a = 3;

//Khai báo biến thực x,y và khởi tạo giá trị bằng

5.0 và 7.6

float x = 5.0, y = 7.6;

int n; m = 2 * n; ⇒ m=?



Ví dụ về kiểu số nguyên, số thực



```
#include <stdio.h>
#include <math.h> //de su dung hang M_PI

void main(){
    int r = 3.5;
    printf("Dien tich hinh tron voi ban kinh r = %d la S = %f.\n", r, M_PI*r*r);
}
```

Khai báo kiểu
nguyên nhưng
khởi tạo là số
thực

Dien tich hinh tron voi ban kinh r = 3 la S = 28.274334.

Process exited after 0.05746 seconds with return value 57
Press any key to continue . . .



Khai báo hằng (dùng macro)



- ❖ Dùng chỉ thị `#define`

- ❖ Cú pháp:

```
#define Tên_hằng Giá_trị
```

Không có dấu ;

- ❖ Ví dụ:

```
#define MAX_SINH_VIEN 50
```

```
#define CNTT "Cong nghe thong tin"
```

```
#define DIEM_CHUAN 23.5
```



Khai báo hằng



- ❖ Dùng từ khóa **const**

- ❖ Cú pháp:

const Kiểu_Dữ_Liệu Tên_hằng = Giá_trị;

- ❖ Ví dụ:

```
const int MAX_SINH_VIEN = 50;
```

```
const char CNTT[20] = "Cong nghe thong tin";
```

```
const float DIEM_CHUAN = 23.5;
```



Khai báo hằng



- ❖ Chú ý:
- ❖ Giá trị của các hằng phải được xác định ngay khi khai báo.
- ❖ **#define là chỉ thị tiền xử lý**
 - Khi chương trình được biên dịch, <Tên_hằng> sẽ được thay thế bằng <Giá_trị>
 - Dễ đọc, dễ thay đổi
 - Dễ chuyển đổi giữa các nền tảng phần cứng hơn



Ví dụ



```
#include <stdio.h>
#include <math.h> //de su dung hang M_PI

#define PI 3.14159
float r1, r2, s1, s2;

void main(){
    printf("Cho ban kinh r1 = "); scanf("%f",&r1);
    s1 = PI*r1*r1;
    printf("Dien tich hinh tron 1 = %f.\n", s1);
    printf("Cho ban kinh r2 = "); scanf("%f",&r2);
    s2 = M_PI*r2*r2;
    printf("Dien tich hinh tron 2 = %f.",s2);
}
```

```
Cho ban kinh r1 = 1
Dien tich hinh tron 1 = 3.141590.
Cho ban kinh r2 = 1
Dien tich hinh tron 2 = 3.141593.
-----
Process exited after 2.668 seconds with return value 33
Press any key to continue . . .
```



Biểu thức trong C

- ❖ Làm vẽ phải của lệnh gán.
- ❖ Làm toán hạng trong các biểu thức khác.
- ❖ Làm tham số thực sự trong lời gọi hàm.
- ❖ Làm biểu thức kiểm tra trong các cấu trúc điều khiển
 - Cấu trúc lặp: for, while, do while.
 - Cấu trúc rẽ nhánh: if, switch.



Tính toán giá trị biểu thức



- ❖ Các toán hạng được thay thế bởi giá trị tương ứng
- ❖ Các phép tính được thực hiện theo thứ tự

Ví dụ (alpha = 10, beta = 81)

Biểu thức: alpha + sqrt(beta)

: alpha + sqrt(81)

: alpha + 9.0

: 10 + 9.0

: 19.0



Các loại biểu thức



- ❖ Biểu thức số học
- ❖ Biểu thức quan hệ
- ❖ Biểu thức logic



Biểu thức số học



- ❖ Là biểu thức mà giá trị của nó là các đại lượng số học (số nguyên, số thực).
 - Sử dụng các toán tử là các phép toán số học (cộng, trừ, nhân, chia...),
 - Các toán hạng là các đại lượng số học (hằng số, biến, biểu thức khác).
- ❖ Ví dụ: a, b, c là các biến thuộc kiểu số thực.
 - $3 * 3.7$
 - $8 + 6/3$
 - $a + b - c$
- ❖ Chú ý: phép chia số nguyên/số nguyên → số nguyên



Ví dụ



```
1 //thao tac tren so nguyen
2 #include<stdio.h>
3 #include<conio.h>
4
5 int main(){
6     int a = 40, b = 9;
7     printf("a + b = %d", a+b);
8     printf("\na - b = %d", a-b);
9     printf("\na * b = %d", a*b);
10    printf("\na / b = %d", a/b);
11    return 0;
12 }
```

```
a + b = 49
a - b = 31
a * b = 360
a / b = 4
-----
```

```
1 //thao tac tren so thuc
2 #include<stdio.h>
3 #include<conio.h>
4
5 int main(){
6     float a = 40, b = 9;
7     printf("a + b = %f", a+b);
8     printf("\na - b = %f", a-b);
9     printf("\na * b = %f", a*b);
10    printf("\na / b = %f", a/b);
11    return 0;
12 }
```

```
a + b = 49.000000
a - b = 31.000000
a * b = 360.000000
a / b = 4.444444
-----
```



Biểu thức quan hệ



- ❖ Là những biểu thức có sử dụng các toán tử quan hệ như lớn hơn, nhỏ hơn, khác nhau...
- ❖ Chỉ có thể trả về một trong 2 giá trị logic Đúng (TRUE) hoặc Sai (FALSE)
- ❖ Ví dụ

- | | |
|---|---|
| <ul style="list-style-type: none">• $5 > 7$• $9 != 10$• $2 >= 2$• $a > b$ | <ul style="list-style-type: none">• // có giá trị logic là sai, FALSE• // có giá trị logic là đúng, TRUE• // có giá trị logic là đúng, TRUE• // giả sử a, b là 2 biến kiểu int |
|---|---|

- ❖ Chú ý: Ngôn ngữ C coi các giá trị nguyên khác 0 là giá trị logic đúng (TRUE), giá trị 0 là giá trị logic sai (FALSE)



Biểu thức logic



❖ Là biểu thức trả về các giá trị logic Đúng/Sai

➤ Các phép toán logic gồm có

- AND VÀ logic, sử dụng toán tử &&
- OR HOẶC logic, sử dụng toán tử ||
- NOT PHỦ ĐỊNH, sử dụng toán tử !

➤ Biểu thức quan hệ là trường hợp riêng của biểu thức logic.

➤ Biểu thức logic cũng trả về một giá trị số học 0/1



Ví dụ



```
1 #include<stdio.h>
2 #include<conio.h>
3
4 int main(){
5     int a = 12, b = 23;
6     printf("%d", 5>7);
7     printf("\n%d", 9!=10);
8     printf("\n%d", 2>=2);
9     printf("\n%d", a>b);
10    return 0;
11 }
```

```
0
1
1
0
-----
Process exited after 0.1547 seconds
Press any key to continue . . .
```



Các toán tử chính



- ❖ Toán tử số học
- ❖ Toán tử quan hệ
- ❖ Toán tử logic
- ❖ Toán tử logic bit
- ❖ Toán tử gán



Các toán tử số học



Toán tử	Ý nghĩa	Kiểu dữ liệu của toán hạng	Ví dụ (int a = 12; float x=3.0)
-	Đảo dấu	float, double, int, long,.. <i>(Số nguyên hoặc thực)</i>	-12, -12.34, - a, - x
+	Cộng	float, double, int, long,..	12 + x
-	Trừ	float, double, int, long,..	12.0 - x
*	Nhân	float, double, int, long,.. <i>(Số nguyên hoặc thực)</i>	12 * 3.0 12 * 3
/	Chia	Nếu có ít nhất 1 toán hạng là số thực	17.0/3.0 → 5.666667 17/3.0 → 5.666667 17.0/3 → 5.666667
/	Chia lấy thương	Số nguyên int, long,..	17/3 → 5
%	Chia lấy số dư	Số nguyên: int, long,..	17%3 → 2



Các toán tử quan hệ



$<$, $>$, \leq , \geq , $=$, \neq

- ❖ Dùng cho phép so sánh giá trị 2 toán hạng
- ❖ Kết quả phép so sánh là một số nguyên
 - 1 nếu quan hệ có kết quả là đúng,
 - 0 nếu quan hệ có kết quả sai
- ❖ Ví dụ:
 - $6 > 4 \rightarrow$ Trả về giá trị 1
 - $6 < 4 \rightarrow$ Trả về giá trị 0
 - $\text{int } b = (x \neq y);$
Nếu x và y khác nhau, biểu thức đúng và b mang giá trị 1.
Ngược lại biểu thức sai và b mang giá trị 0



Các toán tử logic



- ❖ Sử dụng để xây dựng các biểu thức logic
- ❖ Biểu thức logic có kết quả logic đúng
 - Trả về giá trị 1
- ❖ Biểu thức logic có kết quả logic sai
 - Trả về giá trị 0



Các toán tử logic (tiếp)



❖ Và logic (&&) :

- Cho kết quả đúng (trả về giá trị 1) khi cả 2 toán hạng đều đúng (khác 0)
- Ví dụ: $3 < 5 \text{ && } 4 < 6 \rightarrow 1$; $3 < 5 \text{ && } 5 > 6 \rightarrow 0$

❖ Hoặc logic (||):

- Cho kết quả sai (trả về giá trị 0) chỉ khi cả 2 toán hạng đều sai (bằng 0)
- Ví dụ: $4 || 5 < 3 \rightarrow 1$; $5 < 5 || 2 > 6 \rightarrow 0$

❖ Phủ định logic (!):

- Cho kết quả đúng (1) hoặc sai (0) khi toán hạng là sai (0) hoặc đúng (khác 0)
- Ví dụ: $!3 \rightarrow 0$; $!(2 > 3) \rightarrow 1$;



Toán tử trên bit



- ❖ Toán tử trên bit (bitwise operator) được sử dụng với kiểu số nguyên

Và nhị phân: Op1 & Op2

Hoặc nhị phân : Op1 | Op2

Hoặc có loại trừ nhị phân: Op1 ^ Op2

Đảo bít nhị phân : ~ Op

Dịch trái n bit: $\text{Op} \ll n$

Dịch phải n bit: Op >> n



Toán tử trên bit (tiếp)



```
char Op1 = 83, Op2 = -38, Op = 3;
```

```
char r = Op1 & Op2;
```

01010011

11011010

r = 01010010 → (82)

```
char r = Op1 | Op2;
```

01010011

11011010

r = 11011011 → (-37)

```
char r = Op1 ^ Op2;
```

01010011

11011010

r = 10001001 → (-119)

```
char r = ~ Op2;
```

11011010

r = 00100101 → (37)

```
unsigned char r = Op1 | Op2; r = 11011011 → 219
```



Ví dụ trên C



```
1 //Thao tac tren bit
2 #include<stdio.h>
3 #include<conio.h>
4
5 int main(){
6     int a, b;
7     printf("Cho a = "); scanf("%d",&a);
8     printf("Cho b = "); scanf("%d",&b);
9     printf("Tinh toan theo bit:\n");
10    printf("a&b = %d", a&b);
11    printf("\ta\b = %d", a\b);
12    printf("\ta\b = %d", a^b);
13    printf("\t~b = %d", ~b);
14    return 0;
15 }
```

```
Cho a = 83
Cho b = -38
Tinh toan theo bit:
a&b = 82      a\b = -37      a^b = -119      ~b = 37
-----
Process exited after 3.748 seconds with return value 0
Press any key to continue . . .
```



Toán tử trên bit (tiếp)



```
char Op1 = 83, Op2 = -38, Op = 3;
```

```
char r = Op1 >> Op;
```

0 1 0 1 0 0 1 1
r = 0 0 0 0 1 0 1 0 → (10)

```
char r = Op2 >> Op;
```

1 1 0 1 1 0 1 0
r = 1 1 1 1 1 0 1 1 → (-5)

```
unsigned char Op = 218;  
unsigned char r = Op >> 3;
```

1 1 0 1 1 0 1 0
r = 0 0 0 1 1 0 1 1 → (27)

```
char r = Op2 << 2;
```

r = 0 1 1 0 1 0 0 0 → (104)

(unsigned) int r = Op2<<2 → ?



Ví dụ về dịch bit



```
1 //Thao tac tren bit
2 #include<stdio.h>
3 #include<conio.h>
4
5 int main(){
6     char a = 83, b = -38, c = 3;
7     printf("Tinh toan theo bit:\n");
8     printf("a&b = %d", a&b);
9     printf("\ta|b = %d", a|b);
10    printf("\ta^b = %d", a^b);
11    printf("\t~b = %d", ~b);
12    printf("\na >> c = %d", a>>c);
13    printf("\nb << c = %d", b<<c);
14
15 }
```

```
Tinh toan theo bit:
a&b = 82          a|b = -37          a^b = -119        ~b = 37
a >> c = 10
b << c = -304
-----
Process exited after 0.05981 seconds with return value 0
Press any key to continue . . .
```



Toán tử gán



Biến = Biểu_thức;

- ❖ Là toán tử được sử dụng thường xuyên
 - Biểu thức bên phải dấu bằng được tính toán
 - Giá trị của biểu_thức được gán cho biến ở vế trái
- ❖ Ví dụ:
 - int a, b, c;
 - a = 3;
 - b = a + 5;
 - c = a * b;



Toán tử gán



- ❖ Biểu thức gán là biểu thức nên cũng có giá trị.
 - Giá trị của biểu thức gán bằng giá trị của biểu_thức bên phải toán tử
 - Có thể gán giá trị của biểu thức gán cho một biến khác
 - Có thể sử dụng như một biểu thức bình thường
- ❖ Ví dụ:
 - int a, b, c;
 - a = b = 2007;
 - c = (a = 20) * (b = 30); // c→600



Toán tử gán dạng kết hợp



❖ Toán tử số học:

+= -= *= /= %=

Vídu: $a *= b$ // $a = a * b$

Toán tử trên bit:

&= **|=** **^=** **<<=** **>>=**

Vídu:

`x &= 0x3F` // `x = x & 0x3F`

`s <<= 4` // `s = s << 4`



Kiểu dữ liệu và biểu thức trong C

2.1. Các kiểu dữ liệu cơ bản trong C

2.2. Biểu thức trong C

2.3. Các toán tử trong C

2.4. Một số toán tử khác



Các toán tử



- ❖ Tăng/giảm tự động một đơn vị
- ❖ Lấy địa chỉ
- ❖ Biểu thức điều kiện
- ❖ Toán tử phẩy



Tăng giảm tự động một đơn vị



++	Tăng tự động	++Var, Var++
--	Giảm tự động	--Var, Var--

❖ Tiên tố (hậu tố): biến được tăng(++)/giảm(--) trước (sau) khi sử dụng để tính toán biểu thức

❖ Ví dụ:

```
int a = 5, b, c, d, e;  
b = a++; // b = 5 sau đó a = 6  
c = ++a; // a = 7 rồi tới c = 7  
d = a--; // d = 7 rồi tới a = 6  
e = --a; // a = 5 sau đó e= 5
```



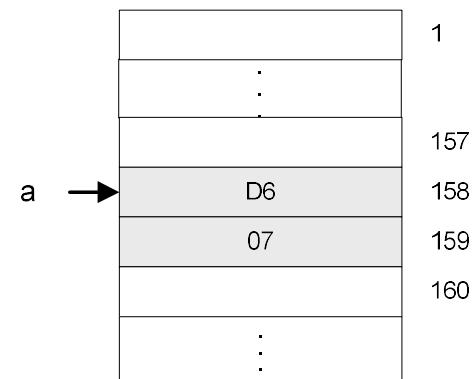
Toán tử lấy địa chỉ

&Tên_bien

- Biến thực chất là một vùng nhớ của máy tính được đặt tên → tên của biến
- Mọi ô nhớ trên bộ nhớ máy tính đều được đánh địa chỉ.
→ Mọi biến đều có địa chỉ
- Ví dụ:

short int a = 2006;

`&a` là địa chỉ của ô nhớ chứa giá trị biến a





Biểu thức điều kiện



Biểu thức1 ? Biểu thức2 : Biểu thức3

- ❖ Nếu Biểu thức 1 $\neq 0$ (giá trị đúng), biểu thức điều kiện trả về giá trị của Biểu thức 2
- ❖ Nếu Biểu thức 1 $= 0$ (giá trị sai) biểu thức điều kiện trả về giá trị của Biểu thức 3.
- ❖ Ví dụ:

float x= 5.2, y = 3.8, z;

$z = (x < y) ? x : y;$

$\rightarrow z = 3.8 // z \min\{x, y\}$

$\Leftrightarrow \text{if } (x < y) z = x; \text{ else } z = y;$



Toán tử phẩy (, comma)



biểu_thức_1, biểu_thức_2, ..

- ❖ Toán tử phẩy (,) cho phép sử dụng nhiều biểu thức tại nơi chỉ cho phép viết một biểu thức
- ❖ Các biểu thức được tính toán từ trái qua phải
- ❖ Giá trị và kiểu của biểu thức là giá trị và kiểu của biểu thức cuối cùng, bên phải
- ❖ Ví dụ:

if (i = 0, a !=b)...

for(i = 0, j = 0; i < 100; **i++, j++**)....



Chuyển kiểu



(Kiểu) biểu thức

❖ Chuyển kiểu tự động

➤ Chương trình dịch tự động chuyển đổi từ kiểu có **phạm vi biểu diễn thấp** tới kiểu có **phạm vi biểu diễn cao**

char → int → long int → float → double → long double

❖ Ép kiểu

➤ Bằng câu lệnh tường minh trong chương trình

➤ Được sử dụng khi muốn tự chuyển kiểu dữ liệu, hoặc thao tác chuyển kiểu tự động không được hỗ trợ



Ví dụ



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

void main(){
    int a, b;
    printf("Cho so nguyen a = "); scanf("%d",&a);
    printf("Cho so nguyen b = "); scanf("%d",&b);
    printf("a%b = %d\n", a%b);
    printf("a/b = %d\n", a/b);
    printf("a/b = %4.2f", (float) a/b); |
}
```

```
Cho so nguyen a = 14
Cho so nguyen b = 5
a%b = 4
a/b = 2
a/b = 2.80
-----
Process exited after 7.081 seconds with return value 10
Press any key to continue . . .
```



Ép kiểu



Ví dụ

```
#include <stdio.h>
#include <conio.h>
int main(){
    long L = 0xABCD;
    float f = 123.456;
    int i;
    i = (int) L;
    printf("\n L = %ld; i = %d(%X)", L, i, i);
    i = (int) f; L = (long) f;
    printf("\n f = %f; L = %ld; i = %d", f, L, i);
    return 0;
}
```

```
L = 11259375; i = 11259375(ABCD)
f = 123.456001; L = 123; i = 123
-----
Process exited after 0.07731 seconds with
Press any key to continue . . .
```



Thứ tự ưu tiên các toán tử



Mức	Toán tử	Chức năng	Chiều
1	<code>-> . [] () ++</code> hậu tố <code>--</code> hậu tố	Lựa chọn, chỉ số...	→
2	<code>++ -- ~ ! + - * & () sizeof</code>	Toán tử 1 ngôi, ép kiểu,...	↓
3	<code>* / %</code>	Toán tử số học lớp nhân	→
4	<code>+ -</code>	Toán tử số học lớp cộng	→
5	<code>>> <<</code>	Dịch bit	→
6	<code>< <= > >=</code>	Toán tử quan hệ	→
7	<code>== !=</code>	Bằng, khác	→
8	<code>&</code>	AND nhị phân	→
9	<code>^</code>	XOR nhị phân	→
10	<code> </code>	OR nhị phân	→
11	<code>&&</code>	AND logic	→
12	<code> </code>	OR logic	→
13	<code>? :</code>	Toán tử phỏng điều kiện	↓
14	<code>= *= += <<= &= ...</code>	Toán tử gán	↓

Chiều kết hợp với các toán hạng



Thứ tự ưu tiên các toán tử



- ❖ Nguyên tắc
- ❖ Biểu thức con trong ngoặc được tính toán trước
- ❖ Phép toán một ngôi đứng bên trái toán hạng được kết hợp với toán hạng đi liền nó.
- ❖ Toán hạng đứng cạnh hai toán tử
 - Nếu hai toán tử có độ ưu tiên khác nhau thì toán tử nào có độ ưu tiên cao hơn sẽ kết hợp với toán hạng
 - Nếu hai toán tử cùng độ ưu tiên thì dựa vào trật tự kết hợp của các toán tử để xác định toán tử được kết hợp với toán hạng.
- ❖ Ví dụ
 - $a < 10 \&& 2 * b < c \equiv (a < 10) \&& ((2 * b) < c)$
 - Chú ý: int x = 5, a = 5 * x++; → a = 25, x = 6



Ví dụ



```
const int N=10;  
float S = 0.0;  
int b;  
S = N/3 +1;  
b=(S>4);
```

S= ? b = ?

```
int a= 3, b=4, c;  
c = a++ * ++b;  
  
a= ?      b= ?      c= ?
```

```
int k, num=30;  
k = num>5 ? (num <=10 ? 100 : 200): 500;  
k=?
```



Ví dụ



```
const int N=10;  
float S= 0.0;  
int b;  
S = N/3 +1;  
b=(S>4);  
  
S= 4      b = 0
```

```
int a= 3, b=4, c;  
c = a++ * ++b;  
  
a=4      b=5      c=15
```

```
int k, num=30;  
k = num>5 ? (num <=10 ? 100 : 200): 500;  
k=200
```



Vào ra dữ liệu



- ❖ Xuất dữ liệu với printf()
- ❖ Nhập dữ liệu với scanf()



Các hàm vào ra cơ bản



❖ In dữ liệu:

➤ printf()

❖ Nhập dữ liệu:

➤ scanf()

❖ Cần khai báo tệp tiêu đề:

#include <stdio.h>



Hàm in dữ liệu (ra màn hình)



❖ printf()

- ❑ Hiển thị ra màn hình các loại dữ liệu cơ bản
 - Số nguyên, số thực, ký tự, xâu ký tự
- ❑ Tạo một số hiệu ứng hiển thị đặc biệt
 - Xuống dòng, sang trang,...
- ❑ Cú pháp:

```
printf(xau_dinh_dang [, DS_tham_so]);
```



Cú pháp



```
printf(Xau_dinh_dang [, DS_tham_so]);
```

- ❖ **Xau_dinh_dang:** Là một xâu qui định cách thức hiển thị dữ liệu ra màn hình máy tính.
 - Bao gồm các nhóm kí tự định dạng
 - Nhóm kí tự định dạng thứ k xác định quy cách hiển thị tham số thứ k trong DS_tham_số
 - Số lượng tham số trong DS_tham_số bằng số lượng nhóm các kí tự định dạng trong xâu định dạng.
- ❖ **DS_tham_so:** Danh sách các biến/biểu thức sẽ được hiển thị giá trị lên màn hình theo cách thức được qui định trong Xau_dinh_dang.



Ví dụ



```
#include <stdio.h>
void main() {
    int a = 5;
    float x = 1.234;
    printf("Hien thi mot bien thuc nguyen %d
va mot so thuc %f",2*a,x);
}
```

Kết quả: Hien thi mot bien thuc nguyen 10 va mot so thuc
1.234000



Xâu định dạng



❖ Các kí tự thông thường:

- Được hiển thị ra màn hình.

❖ Các kí tự điều khiển:

- Dùng để tạo các hiệu ứng hiển thị đặc biệt như xuống dòng ('\n').

❖ Các nhóm kí tự định dạng:

- Xác định quy cách hiển thị các tham số trong phần danh_sach_tham_so.



Nhóm ký tự định dạng



❖ Mỗi nhóm kí tự định dạng chỉ dùng cho một kiểu dữ liệu"

Ví dụ: %d dùng cho kiểu nguyên, %f dùng cho kiểu thực

❖ DS_tham_so phải phù hợp với các nhóm kí tự định dạng trong xau_dinh_dang vẽ:

- Số lượng;
- Thứ tự;
- Kiểu dữ liệu;

Nếu không phù hợp sẽ hiển thị ra kết quả không như ý

```
printf( "%d", 3.14); → -31457
```



Các ký tự định dạng



Ký tự	Kiểu dữ liệu	Kết quả
%i, %d	int, char	Số hệ 10
%o	int, char	Số hệ 8 (in ra không có 0 đằng trước)
%x %X	int, char	Số hệ hexa (chữ thường/chữ hoa)
%u	unsigned int/char	Số hệ 10



Các ký tự định dạng



```
1 #include <stdio.h>
2 int main() {
3     int n = 175;
4     printf("%d",n);
5     printf("\n%i",n);
6     printf("\n%u",n);
7     return 0;
8 }
```

```
175
175
175
-----
Process
Press a
```



Các ký tự định dạng



Ký tự	Kiểu dữ liệu	Kết quả
%ld, %li	long	Số thập phân
%lo	long	Số hệ 8 (không có 0 đằng trước)
%lx, %LX	long	Số hệ hexa (chữ thường/chữ hoa)
%lu	unsigned long	Số thập phân

Nhận xét: Với kiểu long, thêm ký tự l ngay sau dấu %



Các ký tự định dạng



```
1 #include <stdio.h>           175
2 int main() {                  175
3     long int n = 175L;        257
4     printf("%d",n);          AF
5     printf("\n%Ld",n);        af
6     printf("\n%lo",n);        -
7     printf("\n%LX",n);        Process exited after 0.02249
8     printf("\n%lx",n);        Press any key to continue . .
9     return 0;
10 }
```



Các ký tự định dạng



Ký tự	Kiểu dữ liệu	Kết quả
%f, %lf, %Lf	float, double long double	Số thực dấu phẩy tĩnh
%e, %E %g	float/double	Số thực dấu phẩy động Hiển thị dạng thu gọn
%c	int, char	Kí tự đơn lẻ
%s	char []	Hiển thị xâu kí tự kết thúc bởi '\0'
%%		Hiển thị kí tự %



Các ký tự định dạng



```
1 #include <stdio.h>
2 int main() {
3     float x = 17.5;
4     printf("%f",x);
5     printf("\n%g",x);
6     printf("\n%.3f",x);
7     return 0;
8 }
```

```
17.500000
17.5
17.500
-----
Process exited
Press any key t
```



Độ rộng hiển thị



❖ Có dạng "%**m**",

➤ m là một giá trị nguyên, không âm.

➤ m cho biết số chỗ trống dành cho hiển thị biểu thức tương ứng

❖ Ví dụ:

```
int a = 1234;
```

```
printf("%5d",a) → ?1234
```

```
printf("%5d",34)→ ???34
```

? ký hiệu cho dấu trắng (space)



Ví dụ



```
printf("\n%3d%15s%3c", 1, "Nguyen Van A", 'g');
printf("\n%3d%15s%3c", 2, "Tran Van B", 'k');
```

1 Nguyen Van A g

2 Tran Van B k



Độ rộng hiển thị với số thực



❖ Có dạng "%m.n",

- m, n là 2 giá trị nguyên, không âm.
- m cho biết **kích thước để hiển thị số thực**
- n cho biết kích thước dành cho phần thập phân, nếu không đủ C sẽ làm tròn khi hiển thị

❖ Ví dụ:

- `printf("\n%f", 17.346);` → 17.346000
- `printf("\n%.2f", 17.346);` → 17.35
- `printf("\n%.2f", 17.345);` → 17.34 !?
- `printf("\n%8.2f", 17.346);` → □□□17.35
- `printf("\n%8.2f", 17.344);` → ???



Độ rộng hiển thị với số thực



❖ Ví dụ:

```
1 #include <stdio.h>
2 void main() {
3     printf("%f", 17.346);
4     printf("\n%.2f", 17.346);
5     printf("\n%.2f", 17.345);
6     printf("\n%8.2f", 17.346);
7     printf("\n%8.2f", 17.344);
```

```
17.346000
17.35
17.34
17.35
17.34
-----
Process exited after 0.09906 s
Press any key to continue . . .
```



Chú ý



- ❖ Nếu số chỗ cần để hiển thị dữ liệu lớn hơn được cung cấp trong định dạng ⇒ Tự động cung cấp thêm chỗ mới để hiển thị đầy đủ, không cắt bớt nội dung của dữ liệu.
- ❖ Ví dụ:

printf("%2d", 1234); → 1234

printf("%6.3f", 123.456); → 123.456

printf("%12.6e", 123.456); → 1.234560e+02

printf("%12.3e", 123.456); → 1.235e+02



Căn lề trái - căn lề phải



❖ Khi hiển thị dữ liệu có sử dụng tham số độ rộng, để căn lề trái cần thêm dấu trừ - vào ngay sau dấu %:

➤ Ngầm định, căn lề phải

%-

❖ Ví dụ:

```
printf("%-3d%-10s%-5.2f%-3c", 5, "Hello", 7.5, 'g')
```

→ 5□□Hello□□□□□7.50□g□□



Vào ra dữ liệu



- ❖ Xuất dữ liệu với printf()
- ❖ Nhập dữ liệu với scanf()



Mục đích



- ❖ Dùng để nhập dữ liệu từ bàn phím
- ❖ Ký tự đơn lẻ
- ❖ Chuỗi ký tự
- ❖ Số nguyên: hệ 10, 2, 8, 16
- ❖ Số thực
 - Dấu phẩy tĩnh,
 - Dấu phẩy động
- ❖ Cú pháp

```
scanf(xau_dinh_dang [,DS_dia_chi]);
```



Cú pháp



```
scanf(Xau_dinh_dang [,DS_dia_chi]);
```

- ❖ **Xau_dinh_dang**: Gồm các ký tự được qui định cho từng loại dữ liệu được nhập vào.
 - Ví dụ: dữ liệu định nhập kiểu nguyên thì xâu định dạng là : %d
- ❖ **DS_dia_chi**: bao gồm địa chỉ của các biến (toán tử &), phân tách nhau bởi dấu phẩy (,)
- ❖ Phải phù hợp với các ký tự định dạng trong **xau_dinh_dang** về số lượng, kiểu, thứ tự.



Hoạt động



- ❖ Đọc các ký tự được gõ vào từ bàn phím
- ❖ Căn cứ vào xâu định dạng, chuyển thông tin đã nhập sang kiểu dữ liệu phù hợp
- ❖ Gán những giá trị vừa nhập vào các biến tương ứng trong **DS_dia_chi**
- ❖ Ví dụ:

int a;

scanf("%d",&a); →1234_ → a = 1234



Ghi chú



- ❖ Thông tin được gõ vào từ bàn phím, được lưu ở vùng đệm trước khi được xử lý bởi hàm **scanf()**
→ Hàm **scanf()** đọc từ vùng đệm

```
#include <stdio.h>
int main(){
    int a, b;
    scanf("%d",&a);
    scanf("%d",&b);
    printf("%d %d", a, b);
    return 0;
}
```

123 456

123 456



Ghi chú



- ❖ Thông tin được gõ vào từ bàn phím, được lưu ở vùng đệm trước khi được xử lý bởi hàm **scanf()**.
→Hàm **scanf()** đọc dữ liệu từ vùng đệm.

```
#include <stdio.h>
int main(){
    int n;
    char c;
    printf("Nhập số n = "); scanf("%d",&n);
    printf("Nhập ký tự c = "); scanf("%c",&c);
    printf("In ra n: %d, c: %c", n, c);
    return 0;
}
```

Phải thực hiện
xóa bộ nhớ đệm
trước khi nhập ký
tự c

```
Nhập số n = 123
Nhập ký tự c = In ra n: 123, c:
-----
Process exited after 3.941 seconds
Press any key to continue . . .
```

```
Nhập số n = 123a
Nhập ký tự c = In ra n: 123, c: a
-----
Process exited after 2.518 seconds
Press any key to continue . . .
```



Lệnh fflush(stdin) xóa bộ nhớ đệm



Xóa bộ nhớ đệm
trước khi nhập
ký tự c

```
1 #include <stdio.h>
2 int main(){
3     int n;
4     char c;
5     printf("Nhập số n = "); scanf("%d",&n);
6     fflush(stdin); //Xóa bộ nhớ đệm
7     printf("Nhập ký tự c = "); scanf("%c",&c);
8     printf("In ra n: %d, c: %c", n, c);
9     return 0;
10 }
```

```
Nhập số n = 1234
Nhập ký tự c = 2
In ra n: 1234, c: 2
-----
Process exited after 6.091 seconds
Press any key to continue . . .
```

```
Nhập số n = 1234a
Nhập ký tự c = d
In ra n: 1234, c: d
-----
Process exited after 4.712 seconds
Press any key to continue . . .
```



Các ký tự tự định dạng



Kí tự	Khuôn dạng dữ liệu nhập
%c	Đọc kí tự đơn lẻ
%d	Đọc số thập phân
%o	Đọc số hệ 8
%x	Đọc số hệ hexa
%u	Đọc số thập phân không dấu



Các ký tự định dạng



Kí tự	Chú thích
%s	Đọc xâu kí tự tới khi gặp dấu phân cách
%f	Đọc số thực dấu phẩy tĩnh (float)
%ld	Đọc số nguyên kiểu long
%lf	Đọc số thực dấu phẩy tĩnh (double)
%Lf	Đọc số thực dấu phẩy tĩnh (long double)
%e	Đọc số thực dấu phẩy động
%%	Đọc ký tự %



Ký tự định dạng kiểu dữ liệu trong C



Kiểu dữ liệu	Kích thước (bytes)	Ký tự định dạng
short int	2	%hd
unsigned short int	2	%hu
unsigned int	4	%u
int	4	%d, %i
long int	4	%ld, %li
unsigned long int	4	%lu
long long int	8	%lld, %lli
unsigned long long int	8	%llu
char	1	%c
unsigned char	1	%c
float	4	%f
double	8	%lf
long double	10-16	%Lf



Minh họa hiển thị số nguyên



```
Nhap so nguyen int n = 123
Ket qua cua so nguyen n theo %d: 123
Ket qua cua so nguyen n theo %5d: 123
Ket qua cua so nguyen n theo %i: 123
Ket qua cua so nguyen n theo %o: 173
Ket qua cua so nguyen n theo %x: 7b
Ket qua cua so nguyen n theo %X: 7B
Ket qua cua so nguyen n theo %u: 123
(Loi) Ket qua cua so nguyen n theo %f: 0.000000
```

```
Process exited after 1.946 seconds with return value 0
Press any key to continue . . .
```

```
3  int main(){
4      int n;
5      printf("Nhap so nguyen int n = "); scanf("%d", &n);
6      printf("Ket qua cua so nguyen n theo %%d: %d",n);
7      printf("\nKet qua cua so nguyen n theo %%5d: %5d",n);
8      printf("\nKet qua cua so nguyen n theo %%i: %i",n);
9      printf("\nKet qua cua so nguyen n theo %%o: %o",n);
10     printf("\nKet qua cua so nguyen n theo %%x: %x",n);
11     printf("\nKet qua cua so nguyen n theo %%X: %X",n);
12     printf("\nKet qua cua so nguyen n theo %%u: %u",n);
13     printf("\n(Loi) Ket qua cua so nguyen n theo %%f: %f",n);
14     return 0;
15 }
```



Minh họa hiển thị số nguyên



```
1 #include<stdio.h>
2 int main(){
3     char a;
4     printf("He 10: So nguyen a = "); scanf("%d",&a);
5     printf("He 8: a = %o",a);
6     printf("\nHe 16: a = %x",a);
7     printf("\nHe 2: a = ");
8     int i, n;
9     n = sizeof(a)*8;//So bit can bieu dien kieu cua a
10    for (i = n-1; i >= 0; i--) {
11        int k = a >> i;
12        if (k & 1)
13            printf("1");
14        else
15            printf("0");
16    }
17    return 0;
18 }
```

```
He 10: So nguyen a = 127
He 8: a = 177
He 16: a = 7f
He 2: a = 01111111
-----
Process exited after 7.813 seconds
Press any key to continue . . .
```

Câu lệnh **for** sẽ
được giới thiệu sau!



Minh họa hiển thị số nguyên kiểu long



```
Nhap so nguyen Tong (%ld) m = 1234
Ket qua cua so nguyen m theo %d: 1234
Ket qua cua so nguyen m theo %i: 1234
Ket qua cua so nguyen m theo %o: 2322
Ket qua cua so nguyen m theo %x: 4d2
Ket qua cua so nguyen m theo %X: 4D2
Ket qua cua so nguyen m theo %u: 1234
Ket qua cua so nguyen m theo %l: 1234
(Loi) Ket qua cua so nguyen m theo %f: 0.000000
```

```
Process exited after 7.468 seconds with return value 0
Press any key to continue . . .
```

```
3 int main()
4 {
5     long m;
6     printf("Nhap so nguyen long (%ld) m = "); scanf("%ld", &m);
7     printf("Ket qua cua so nguyen m theo %d: %d", m);
8     printf("\nKet qua cua so nguyen m theo %i: %i", m);
9     printf("\nKet qua cua so nguyen m theo %o: %o", m);
10    printf("\nKet qua cua so nguyen m theo %x: %x", m);
11    printf("\nKet qua cua so nguyen m theo %X: %X", m);
12    printf("\nKet qua cua so nguyen m theo %u: %u", m);
13    printf("\nKet qua cua so nguyen m theo %l: %ld", m);
14    printf("\n(Loi) Ket qua cua so nguyen m theo %f: %f", m);
15 }
```



Minh họa hiển thị số thực (1)



```
Nhap so thuc x = 123.25
Ket qua cua so thuc x theo %10f: 123.250000
Ket qua cua so thuc x theo %10.4f: 123.2500
Ket qua cua so thuc x theo %.7f: 123.2500000
Ket qua cua so thuc x theo %e: 1.232500e+002
Ket qua cua so thuc x theo %g: 123.25
(Loi!) Ket qua cua so thuc x theo %d: 0
-----
Process exited after 4.363 seconds with return value 0
Press any key to continue . . .
```

```
3 int main(){
4     float x;
5     printf("Nhap so thuc x = "); scanf("%f", &x);
6     printf("\nKet qua cua so thuc x theo %%10f: %10f",x);
7     printf("\nKet qua cua so thuc x theo %%10.4f: %10.4f",x);
8     printf("\nKet qua cua so thuc x theo %%.7f: %.7f",x);
9     printf("\nKet qua cua so thuc x theo %%e: %e",x);
10    printf("\nKet qua cua so thuc x theo %%g: %g",x);
11    printf("\n(Loi!) Ket qua cua so thuc x theo %%d: %d",x);
12
13 }
```



Minh họa hiển thị số thực (2)



```
3 int main(){
4     double x;
5     printf("Nhập số thực (long double) x = "); scanf("%lf",&x);
6     printf("Kết quả của số thực x theo %%lf: %lf",x);
7     printf("\nKết quả của số thực x theo %%e: %e",x);
8     printf("\nKết quả của số thực x theo %%g: %g",x);
9     printf("\nKết quả của số thực x theo %%10f: %10f",x);
10    printf("\nKết quả của số thực x theo %%10.4f: %10.4f",x);
11    printf("\nKết quả của số thực x theo %%.7f: %.7f",x);
12    printf("\n(Loi!) Kết quả của số thực x theo %%d: %d",x);
13    return 0;
14 }
```

```
Nhap so thuc (long double) x = 123.25
Ket qua cua so thuc x theo %%lf: 123.250000
Ket qua cua so thuc x theo %%e: 1.232500e+002
Ket qua cua so thuc x theo %%g: 123.25
Ket qua cua so thuc x theo %%10f: 123.250000
Ket qua cua so thuc x theo %%10.4f: 123.2500
Ket qua cua so thuc x theo %%.7f: 123.250000
(Loi!) Ket qua cua so thuc x theo %%d: 0
```

```
Process exited after 6.749 seconds with return value 0
Press any key to continue . . .
```



Minh họa hiển thị kí tự, xâu kí tự



```
3 int main(){
4     char c; char s[30];
5     printf("Nhập kí tự c = ");
6     fflush(stdin); scanf("%c", &c);
7     printf("Kết quả của kí tự x theo %%c: %c", c);
8     printf("\nKết quả của kí tự x theo %%5c: %5c", c);
9     printf("\nKết quả của kí tự x theo %%-5c: %-5c", c);
10    printf("\n\nNhập xâu kí tự (scanf()) s = ");
11    fflush(stdin); scanf("%s", &s);
12    printf("Kết quả của xâu kí tự s theo %%s: %s", s);
13    printf("\nKết quả của xâu kí tự s theo %%10s: %10s", s);
14    printf("\nKết quả của xâu kí tự s theo %%-10s: %-10s", s);
15    printf("\n\nNhập xâu kí tự (gets()) s = ");
16    fflush(stdin); gets(s);
17    printf("Kết quả của xâu kí tự s theo %%s: %s", s);
18    printf("\nKết quả của xâu kí tự s theo %%10s: %10s", s);
19    printf("\nKết quả của xâu kí tự s theo %%-10s: %-10s", s);
20    return 0;
21 }
```

```
Nhập kí tự c = A
Kết quả của kí tự x theo %%c: A
Kết quả của kí tự x theo %%5c:      A
Kết quả của kí tự x theo %%-5c: A

Nhập xâu kí tự (scanf()) s = Bach Khoa
Kết quả của xâu kí tự s theo %%s: Bach
Kết quả của xâu kí tự s theo %%10s:      Bach
Kết quả của xâu kí tự s theo %%-10s: Bach

Nhập xâu kí tự (gets()) s = Bach Khoa
Kết quả của xâu kí tự s theo %%s: Bach Khoa
Kết quả của xâu kí tự s theo %%10s: Bach Khoa
Kết quả của xâu kí tự s theo %%-10s: Bach Khoa
```



Ví dụ



```
2 #include <stdio.h>
3 void main(){
4     // khai bao bien
5     int a; float x;
6     char ch;
7     char str[30];
8     // Nhap du lieu
9     printf("Nhập vào một số nguyên: "); scanf("%d",&a);
10    printf("\nNhập vào một số thực: "); scanf("%f",&x);
11    fflush(stdin); //xoa bo nho dem
12    printf("\nNhập vào một ký tự: "); scanf("%c",&ch);
13    printf("\nNhập vào một xâu ký tự: "); scanf("%s",str);
14    // Hien thi du lieu vua nhap vao
15    printf("\nNhững dữ liệu vừa nhập vào");
16    printf("\nSo nguyen: %d", a);
17    printf("\nSo thuc: %5.2f", x);
18    printf("\nKy tu: %c", ch);
19    printf("\nXau ky tu: %s", str);
20 }
```



Ví dụ → Kết quả thực hiện

```
Nhap vao mot so nguyen: 12
Nhap vao mot so thuc: 23.4456
Nhap vao mot ki tu: a
Nhap vao mot xau ki tu: Dai hoc Bach Khoa
Nhung du lieu vua nhap vao
So nguyen: 12
So thuc: 23.45
Ky tu: a
Xau ky tu: Dai
-----
Process exited after 0.00 seconds
Press any key to continue . . .
```

Không hiển thị hết
xâu kí tự vừa nhập
"Dai hoc Bach Khoa"



Các quy tắc cần lưu ý



- ❖ Khi đọc số
- ❖ Hàm scanf() quan niệm rằng mọi kí tự số, dấu chấm (.) đều là kí tự hợp lệ.
Số thực dấu phẩy động, chấp nhận ký tự e/E
- ❖ Khi gấp các dấu phân cách như tab, xuống dòng hay dấu cách (space bar), scanf() sẽ hiểu là kết thúc nhập dữ liệu cho một số



Các quy tắc cần lưu ý



❖ Khi đọc kí tự

Hàm `scanf()` cho rằng mọi kí tự có trong bộ đệm của thiết bị vào chuẩn đều là hợp lệ, kể cả các kí tự tab, xuống dòng hay dấu cách

❖ Khi đọc xâu kí tự:

Hàm `scanf()` nếu gặp các kí tự dấu trắng, dấu tab hay dấu xuống dòng thì nó sẽ hiểu là kết thúc nhập dữ liệu cho một xâu kí tự.



Ví dụ



```
2 #include <stdio.h>
3 void main(){
4     // khai bao bien
5     int a; float x;
6     char ch;
7     char str[30];
8     printf("Nhập vào một số nguyên: "); scanf("%d",&a);
9     printf("\nNhập vào một số thực: "); scanf("%f",&x);
10    fflush(stdin); //xóa bỏ nhớ đệm
11    printf("\nNhập vào một ký tự: "); scanf("%c",&ch);
12    fflush(stdin);
13    printf("\nNhập vào một xâu ký tự: ");
14    gets(str); //Nhập xâu ký tự có chứa dấu cách
15    printf("\nNhững dữ liệu vừa nhập vào");
16    printf("\nSố nguyên: %d", a);
17    printf("\nSố thực: %5.2f", x);
18    printf("\nKý tự: %c", ch);
19    printf("\nXâu ký tự: %s", str);
20 }
```



Ví dụ



Đọc 2 số nguyên, đưa ra tổng, hiệu, tích...

```
#include <stdio.h>

int main(){
    int A, B;
    printf("Nhập vào 2 số nguyên : "); scanf("%d %d",&A,&B);
    printf("\n");
    printf("Tổng %d + %d = %d \n", A, B, A + B);
    printf("Hiệu %d - %d = %d\n", A, B, A - B);
    printf("Tích %d x %d = %d\n", A, B, A * B);
    printf("Th嚮 %d / %d = %.3f\n", A, B, (float)A / B);
    printf("Chia nguyên %d / %d = %d\n", A, B, A / B);
    printf("Chia dư %d %% %d = %d\n", A, B, A % B);
    printf("\n");
    return 0;
}
```



Ví dụ



Đọc 2 số nguyên, đưa ra tổng, hiệu, tích...

```
Nhap vao 2 so nguyen : 17 5
```

```
Tong 17 + 5 = 22
```

```
Hieu 17 - 5 = 12
```

```
Tich 17 x 5 = 85
```

```
Thuong 17 / 5 = 3.400
```

```
Chia nguyen 17 / 5 = 3
```

```
Chia du 17 % 5 = 2
```



Ví dụ



Đọc tọa độ 3 điểm A,B,C và in ra S_{ABC}

```
Nhap vao tọa do diem A : 0 0
Nhap vao tọa do diem B : 6 0
Nhap vao tọa do diem C : 0 8
Dien tich tam giac ABC la: 24.000000
```

```
Nhap vao tọa do diem A : -1 -2
Nhap vao tọa do diem B : 5 -2
Nhap vao tọa do diem C : 6 6
Dien tich tam giac ABC la: 24.000000
```

```
Nhap vao tọa do diem A : 1 1
Nhap vao tọa do diem B : 2 2
Nhap vao tọa do diem C : 4 4
Dien tich tam giac ABC la: 0.000000
```



Ví dụ



Đọc tọa độ 3 điểm A,B,C và in ra diện tích ΔABC

```
#include <stdio.h>
int main(){
    float Ax,Ay, Bx, By, Cx, Cy, AB, BC, CA,p;
    printf("Nhập vào tọa độ điểm A : "); scanf("%f %f",&Ax,&Ay);
    printf("Nhập vào tọa độ điểm B : "); scanf("%f %f",&Bx,&By);
    printf("Nhập vào tọa độ điểm C : "); scanf("%f %f",&Cx,&Cy);

    //Tính độ dài các cạnh của tam giác
    AB = sqrt((Ax-Bx)*(Ax-Bx)+(Ay-By)*(Ay-By));
    BC = sqrt((Bx-Cx)*(Bx-Cx)+(By-Cy)*(By-Cy));
    CA = sqrt((Cx-Ax)*(Cx-Ax)+(Cy-Ay)*(Cy-Ay));
    p = (AB + BC + CA)/2;
    printf("Diện tích tam giác ABC là: %f",sqrt(p*(p-AB)*(p-
BC)*(p-CA)));
    return 0;
}
```



Bài tập



- ❖ Viết chương trình nhập vào từ bàn phím bán kính một đường tròn và đưa ra màn hình diện tích và chu vi đường tròn
- ❖ Ghi chú:
 - cách 1: Giả thiết $\pi = 3.1416$. Cần khai báo hằng PI trong chương trình.
 - cách 2: π là hằng số được khai báo trong tệp tiêu đề **math.h** và có tên là **M_PI**



Minh họa



```
1 #include <stdio.h>
2 #include <math.h>
3 #define PI 3.1416
4 int main(){
5     float r;
6     printf("Cho ban kinh r = "); scanf("%f",&r);
7     printf("Chu vi hinh tron P = %f\n", 2*PI*r);
8     printf("Dien tich hinh tron S = %f\n", PI*r*r);
9     return 0;
10 }
```

```
Cho ban kinh r = 1
Chu vi hinh tron P = 6.283200
Dien tich hinh tron S = 3.141600
```

```
-----
Process exited after 1.585 seconds with return value 33
Press any key to continue . . .
```



Minh họa hằng M_PI



```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(){
5     float r;
6     printf("Cho ban kinh r = "); scanf("%f",&r);
7     printf("Chu vi hinh tron P = %f\n", 2*M_PI*r);
8     printf("Dien tich hinh tron S = %f\n",M_PI*r*r);
9     return 0;
10 }
```

```
Cho ban kinh r = 1
Chu vi hinh tron P = 6.283185
Dien tich hinh tron S = 3.141593
```

```
-----
Process exited after 1.785 seconds with return value 0
Press any key to continue . . .
```



Bài tập



1. Viết chương trình nhập vào từ bàn phím chiều dài 3 cạnh của một tam giác, rồi đưa ra diện tích và các đường cao của tam giác
2. Nhập vào từ bàn phím tọa độ 3 điểm A, B, C rồi đưa ra độ dài các cạnh của tam giác ABC và của đường trung tuyến AM
3. Cho hàm số: $f(x) = x^7 + 5\sqrt[3]{x^5 + 3x^3 + 2} + 12$
Viết chương trình nhập vào 3 số thực a,b,c và đưa ra trung bình cộng của $f(a)$, $f(b)$, $f(c)$
4. Nhập x vào từ bàn phím và tính giá trị của biểu thức

$$A = \frac{\cos 3a + \sqrt[5]{2x^3 + x + 1}}{\log_7(3^{x^2} + 2.14b)} \text{ trong đó } a = \sqrt{2^x + \pi} \text{ và } b = \ln(e^{x+1.23} + 1)$$