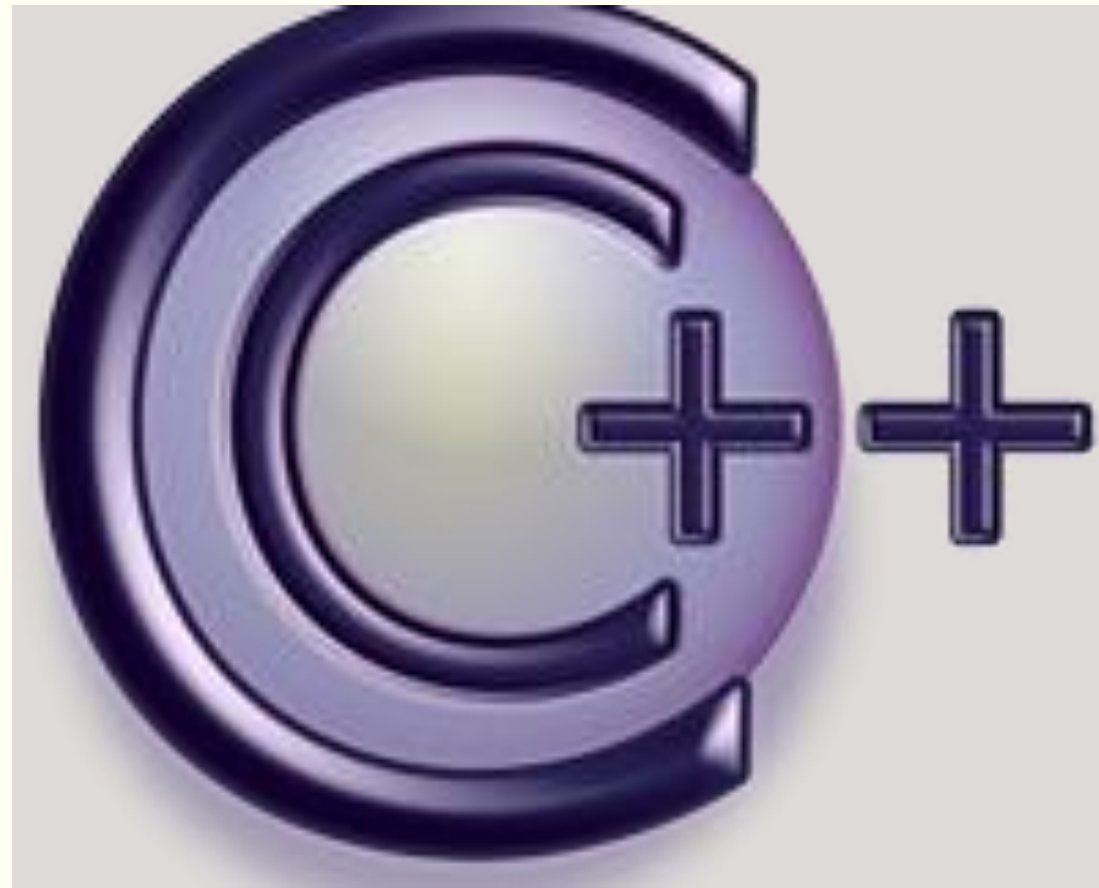


BÀI 2

CÁC KHÁI NIỆM CƠ BẢN TRONG C++



Giảng viên: **Vũ Thương Huyền – Khoa Công nghệ thông tin – ĐH Thủy Lợi**

Email: **huyenvt@wru.edu.vn**

NỘI DUNG

- Thuật toán và phương pháp biểu diễn thuật toán
- Sử dụng **Dev-C++**
- Cấu trúc chương trình C++
- Các khái niệm cơ bản trong C++
 - ✓ Định danh phép gán
 - ✓ Dữ liệu
 - ✓ Biểu thức, khối lệnh
 - ✓ Toán tử
 - ✓ Hàm trong thư viện C++



THUẬT TOÁN

- **Thuật toán:** là một tập hữu hạn các chỉ thị (bước) khi được thực thi sẽ chuyển thông tin đầu vào thành thông tin đầu ra.
- **Tại sao cần thuật toán?** Máy tính chỉ giải quyết được vấn đề khi đã có hướng dẫn giải rõ ràng và đúng
- **Các đặc trưng:**
 - Tính hữu hạn
 - Tính đúng
 - Đầu vào, đầu ra
 - Tính hiệu quả: dựa trên khối lượng tính toán, không gian và thời gian
 - Tính tổng quát: áp dụng cho mọi trường hợp

THUẬT TOÁN

▪ **Ví dụ:** Thuật toán để giải phương trình bậc nhất : $ax + b = c$ (a, b, c là các số thực):

- Đầu vào: các hệ số a, b, c
- Đầu ra: nghiệm của phương trình
- Các bước thuật giải:

1. Cho các giá trị a, b, c
2. Nếu $a = 0$
 - Nếu $b = c$ thì “phương trình có vô số nghiệm”
 - Nếu $b \neq c$ thì “phương trình vô nghiệm”
3. Nếu $a \neq 0$
 - Phương trình có duy nhất 1 nghiệm $x = (c-b)/a$

PHƯƠNG PHÁP BIỂU DIỄN THUẬT TOÁN

1. Bảng ngôn ngữ tự nhiên:

- Liệt kê bằng lời các bước của thuật toán
- Đơn giản, không cần kiến thức nền tảng
- Dài dòng

2. Bảng mã giả:

- Là bản mô tả ngắn gọn, giúp con người có thể hiểu dễ dàng
- Độc lập với môi trường phát triển

**Ví dụ: giải phương trình bậc 2 $a \cdot x^2 + b \cdot x + c = 0$*

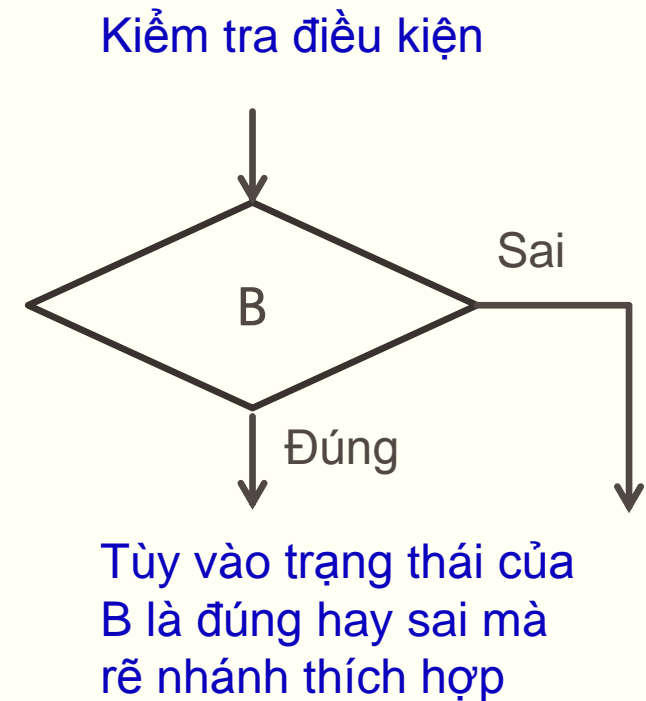
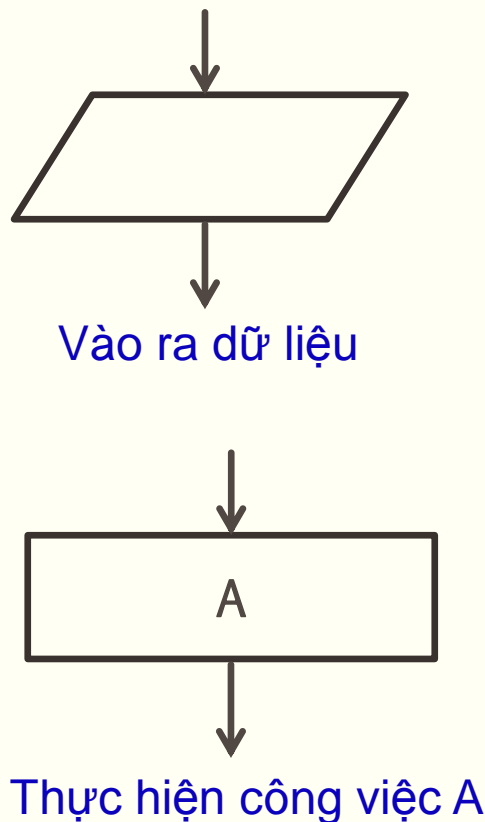
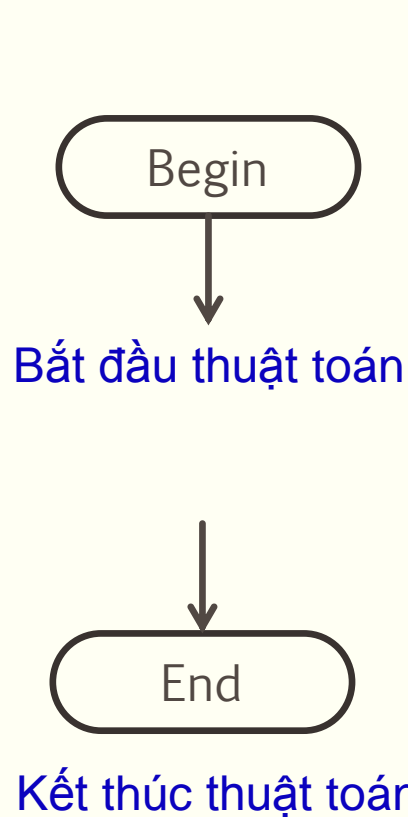
PHƯƠNG PHÁP BIỂU DIỄN THUẬT TOÁN

```
1. Yêu cầu nhập giá trị a, b, c
2. if a = 0 then
3.     if b = 0 then
4.         if c = 0 then
5.             xuất kết quả: phương trình vô số nghiệm
6.         else
7.             xuất kết quả: phương trình vô nghiệm
8.     else
9.         xuất kết quả: phương trình có nghiệm -c/b
10. else
11.     Tính giá trị  $\text{delta} = b^2 - 4*a*c$ 
12.     If  $\text{delta} > 0$  then
13.          $X1 = (-b - \text{sqrt}(\text{delta})) / (2*a)$ 
14.          $X2 = (-b + \text{sqrt}(\text{delta})) / (2*a)$ 
15.         xuất kết quả: phương trình có 2 nghiệm là x1 và b
16.     else
17.         if  $\text{delta} = 0$  then
18.             xuất kết quả: Phương trình có nghiệm kép là  $-b/(2*a)$ 
19.         else
20.             xuất kết quả: phương trình vô nghiệm
```

PHƯƠNG PHÁP BIỂU DIỄN THUẬT TOÁN

3. Bảng lưu đồ:

- sử dụng các khối để biểu diễn thuật toán

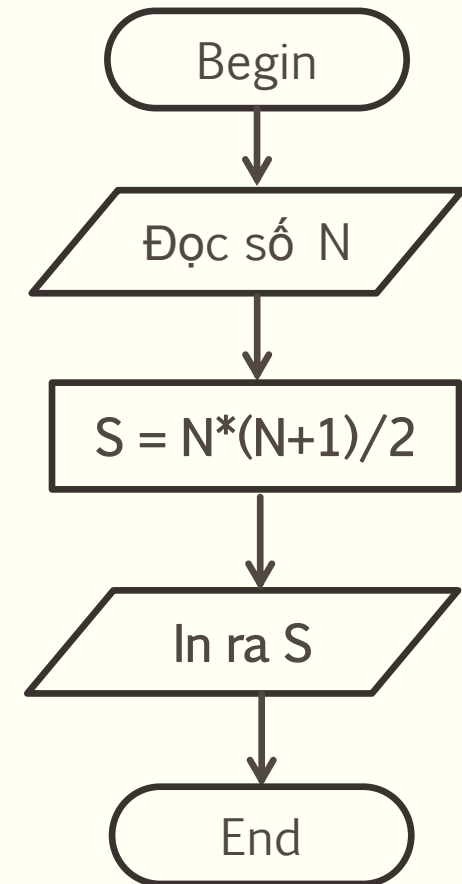
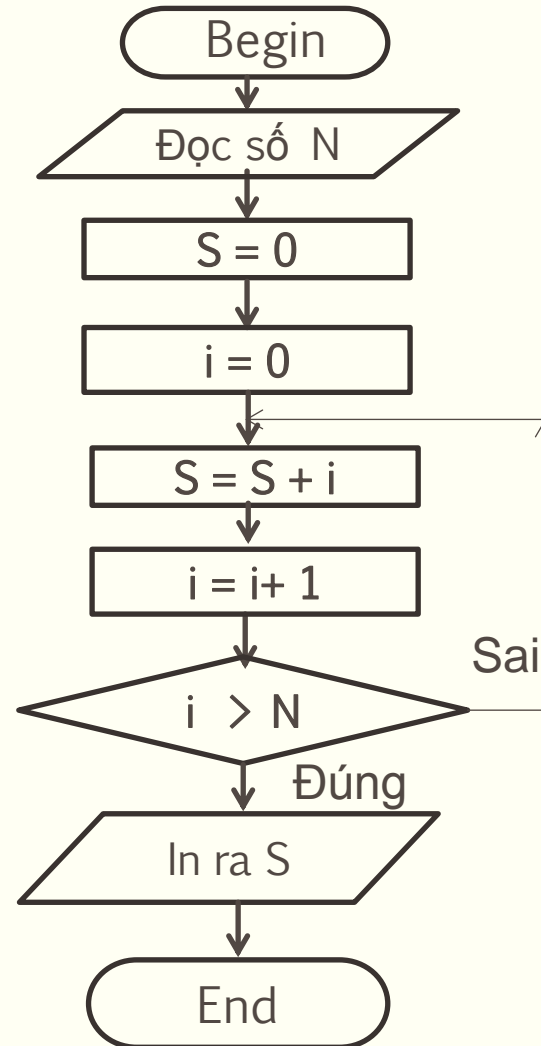


PHƯƠNG PHÁP BIỂU DIỄN THUẬT TOÁN

Ví dụ: Lưu đồ tính tổng của N số nguyên đầu tiên với các thuật toán khác nhau?

➤ **Đầu vào:** số nguyên N

➤ **Đầu ra:** tổng N số



PHƯƠNG PHÁP BIỂU DIỄN THUẬT TOÁN

4. Bảng ngôn ngữ lập trình

- Được thiết kế và chuẩn hóa để truyền các chỉ thị cho máy tính
- Mô tả đầy đủ và rõ ràng thuật toán

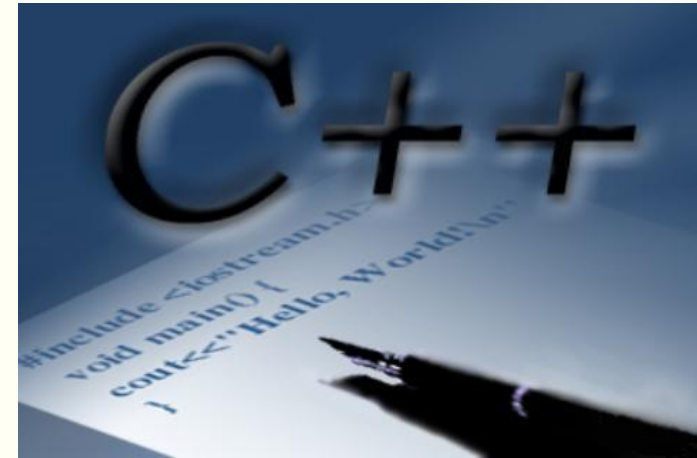


BÀI TẬP

- **Bài 1:** Xây dựng thuật toán tính tổng các số chẵn và tổng các số lẻ trong dãy gồm n số nguyên được nhập.
- **Bài 2:** Xây dựng thuật toán tìm giá trị lớn nhất của một dãy số nguyên có N số.

NGÔN NGỮ LẬP TRÌNH C++

- C++ được phát triển bởi **Bjane Stroustrup** nhằm mở rộng ngôn ngữ C vào năm 1979
- Được ISO duyệt vào năm 1998
- C++ là ngôn ngữ lập trình hướng đối tượng



MÔI TRƯỜNG PHÁT TRIỂN C++

Dev-C++

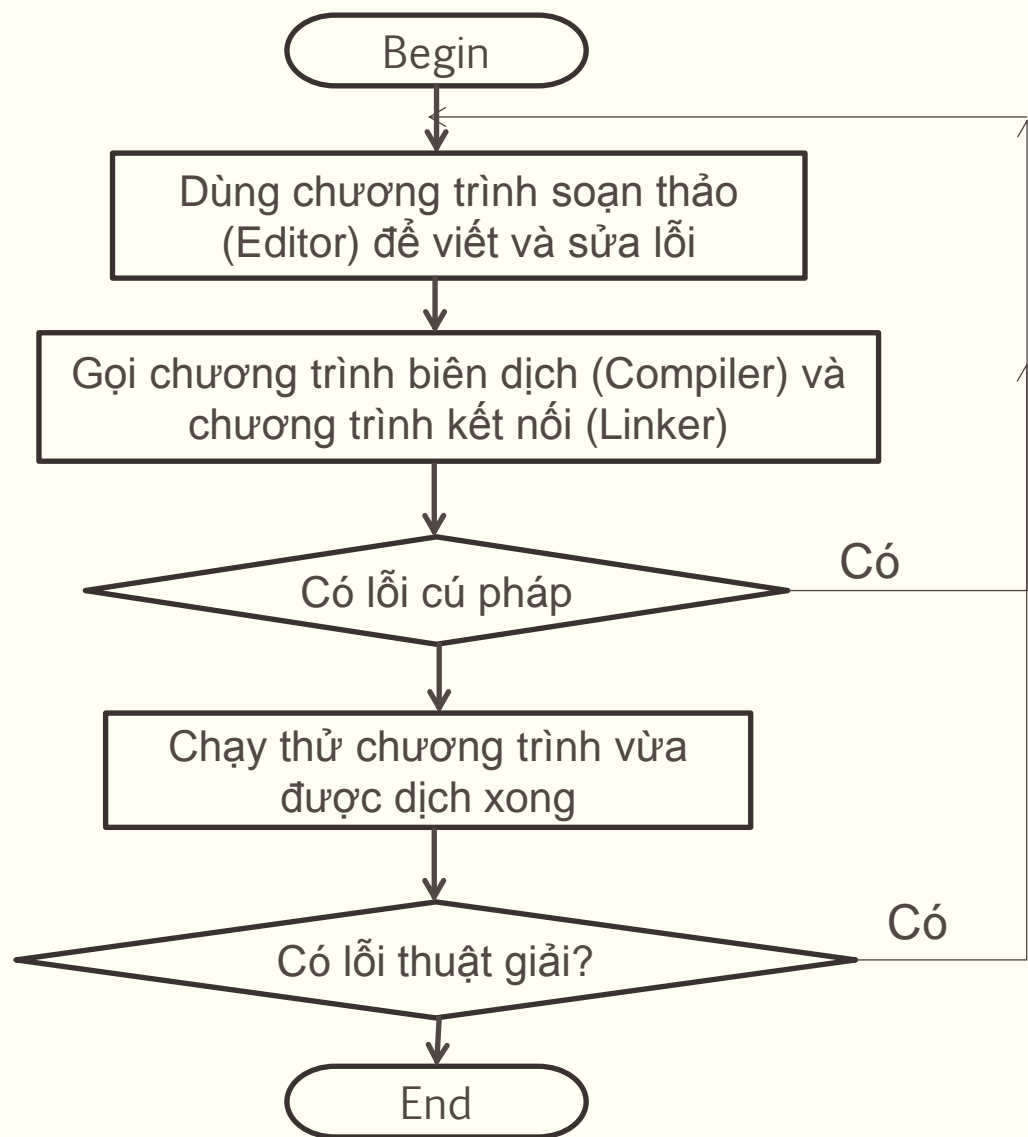
- Hỗ trợ đầy đủ tính năng cho ngôn ngữ lập trình C/C++
- Hỗ trợ soạn mã nguồn
- Sử dụng trình MinGW của GCC làm trình biên dịch

▪ Đường dẫn download:

<http://sourceforge.net/projects/orwelldevcpp/files/latest/download>



CÁC BƯỚC CƠ BẢN ĐỂ VIẾT CHƯƠNG TRÌNH



➤ Editor là chương trình để viết mã nguồn

➤ Tên các tệp mã nguồn có dạng ***.cpp**

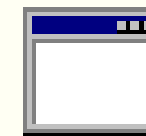


Hello.cpp

➤ Dịch mã nguồn ra tệp OBJ

➤ Liên kết các tệp OBJ qua Linker

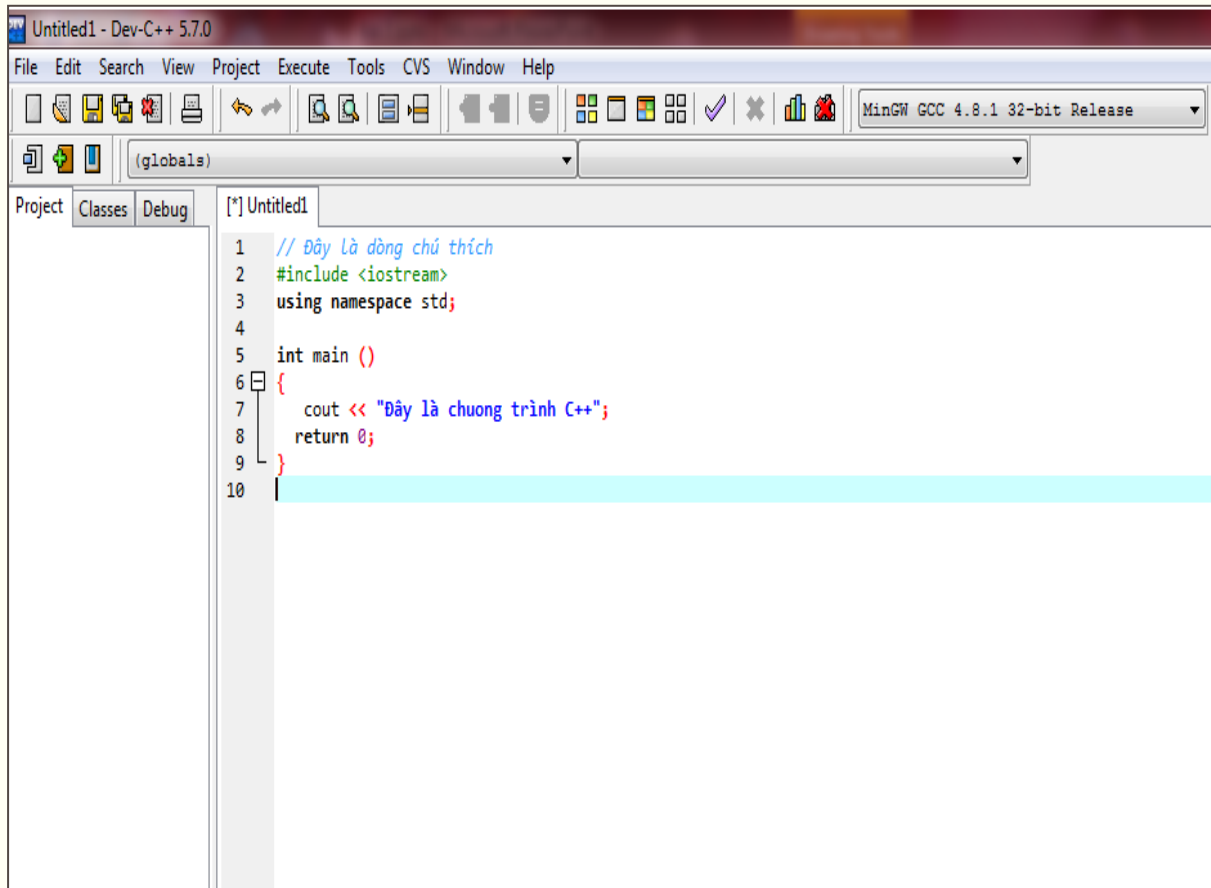
➤ Kết quả trả ra file **.exe**



Hello.exe

CÁC BƯỚC CƠ BẢN ĐỂ VIẾT CHƯƠNG TRÌNH

Dev-C++

A screenshot of the Dev-C++ 5.7.0 IDE. The window title is "Untitled1 - Dev-C++ 5.7.0". The menu bar includes File, Edit, Search, View, Project, Execute, Tools, CVS, Window, and Help. The toolbar contains various icons for file operations and execution. The compiler is set to "MinGW GCC 4.8.1 32-bit Release". The left sidebar shows "Project", "Classes", and "Debug" tabs. The main editor area shows a C++ program with the following code:

```
1 // Đây là dòng chú thích
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     cout << "Đây là chương trình C++";
8     return 0;
9 }
10
```

➤ Tạo mới 1 file mã nguồn:

File -> New -> Source File (Ctrl + N)

➤ Lưu nội dung file mã nguồn:

File -> Save (Ctrl + S)

➤ Biên dịch chương trình:

Execute -> Compile (F9)

➤ Chạy chương trình:

Execute -> Run (F10)

➤ Biên dịch và chạy chương trình:

Execute -> Compile & Run (F11)

CẤU TRÚC CHƯƠNG TRÌNH C++

(Xem mục 2.2 trong giáo trình)

Ví dụ:

```
1. // Đây là dòng chú thích
2. #include <iostream>
3. using namespace std;
4.
5. int main ()
6. {
7.     cout << "Day la chuong trinh C++";
8.     return 0;
9. }
```

CẤU TRÚC CHƯƠNG TRÌNH C++

(Xem mục 2.2 trong giáo trình)

Giải thích:

- **Dòng 1:** Là dòng chú thích, bắt đầu bằng `//`
- **Dòng 2:** Khai báo sử dụng thư viện
`#include <tênthưviện>`
- **Dòng 3:** Khai báo sử dụng không gian tên
`using namespace tên;`

- **Dòng 4:** Dòng trắng
- **Dòng 5:** Khai báo hàm, `main` là hàm đặc biệt, gọi đầu tiên khi chương trình chạy
- **Dòng 6, 9:** Bắt đầu và kết thúc khối lệnh của hàm `main`
- **Dòng 7:** xuất xâu ký tự ra màn hình
- **Dòng 8:** Câu lệnh thoát khỏi hàm `main`



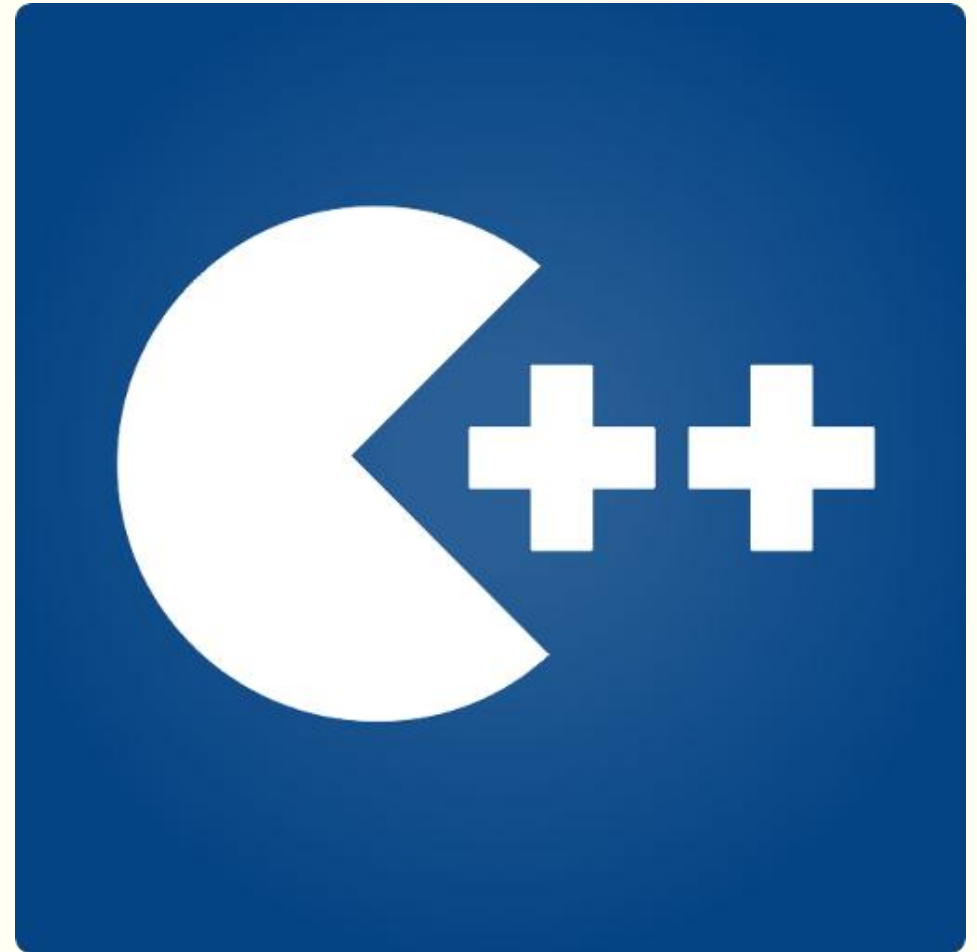
TÌM LỖI SAI

- Tìm những lỗi sai trong chương trình sau:

```
1. / Chương trình 2
2. include <iostream
3. usingnamespace std
4.
5. int main
6. {
7.     cout << "Bài 2";
8.     cout << "Các khái niệm cơ bản trong C++";
9.     return 0;
10.
```

CÁC KHÁI NIỆM CƠ BẢN TRONG C++

- Từ khóa
- Định danh
- Câu lệnh và khối lệnh
- Chú thích
- Các kiểu dữ liệu cơ sở
- Biến
- Hằng
- Toán tử
- Các hàm trong thư viện



TỪ KHÓA

- Các từ cơ bản của ngôn ngữ, không thể định nghĩa lại

<code>alignas (since C++11)</code>	<code>enum</code>	<code>return</code>
<code>alignof (since C++11)</code>	<code>explicit</code>	<code>short</code>
<code>and</code>	<code>export(1)</code>	<code>signed</code>
<code>and_eq</code>	<code>extern</code>	<code>sizeof</code>
<code>asm</code>	<code>false</code>	<code>static</code>
<code>auto(1)</code>	<code>float</code>	<code>static_assert (since C++11)</code>
<code>bitand</code>	<code>for</code>	<code>static_cast</code>
<code>bitor</code>	<code>friend</code>	<code>struct</code>
<code>bool</code>	<code>goto</code>	<code>switch</code>
<code>break</code>	<code>if</code>	<code>template</code>
<code>case</code>	<code>inline</code>	<code>this</code>
<code>catch</code>	<code>int</code>	<code>thread_local (since C++11)</code>
<code>char</code>	<code>long</code>	<code>throw</code>
<code>char16_t (since C++11)</code>	<code>mutable</code>	<code>true</code>
<code>char32_t (since C++11)</code>	<code>namespace</code>	<code>try</code>
<code>class</code>	<code>new</code>	<code>typedef</code>
<code>compl</code>	<code>noexcept (since C++11)</code>	<code>typeid</code>
<code>const</code>	<code>not</code>	<code>typename</code>
<code>constexpr (since C++11)</code>	<code>not_eq</code>	<code>union</code>
<code>const_cast</code>	<code>nullptr (since C++11)</code>	<code>unsigned</code>
<code>continue</code>	<code>operator</code>	<code>using(1)</code>
<code>decltype (since C++11)</code>	<code>or</code>	<code>virtual</code>
<code>default(1)</code>	<code>or_eq</code>	<code>void</code>
<code>delete(1)</code>	<code>private</code>	<code>volatile</code>
<code>do</code>	<code>protected</code>	<code>wchar_t</code>
<code>double</code>	<code>public</code>	<code>while</code>
<code>dynamic_cast</code>	<code>register</code>	<code>xor</code>
<code>else</code>	<code>reinterpret_cast</code>	<code>xor_eq</code>

ĐỊNH DANH

(Xem mục 2.2 trong giáo trình)

- Là đặt tên cho các đối tượng trong chương trình
- Các đối tượng:
 - Biến
 - Hằng số
 - Hàm
 - Kiểu dữ liệu
- Quy tắc đặt tên định danh:
 - ✓ Bắt đầu bằng các chữ cái (**a..z, A..Z**) hoặc “_”
 - ✓ Có thể chứa chữ số (**0..9**) hoặc kí tự “_”
 - ✓ C++ phân biệt chữ hoa, chữ thường
 - ✓ **Không trùng với các từ khóa C++**
 - ✓ Không chứa khoảng trống ở giữa
 - ✓ Không chứa các ký tự đặc biệt





Định danh nào sau đây là đúng/sai? Vì sao?

1. CamelCase
2. camelCase
3. PI
4. 20_tuoi
5. return
6. nghiem1
7. m&m
8. chieu cao



ĐỊNH DANH

- Quy tắc con lạc đà:
 - **Tên biến, hàm:** viết thường từ đầu tiên, viết hoa chữ cái đầu từ tiếp theo
 - **Tên hằng số:** viết hoa toàn bộ, sử dụng dấu gạch dưới để phân cách
 - **Kiểu dữ liệu mới:** viết hoa chữ cái đầu tiên của tất cả các từ

Định danh	Loại đối tượng
HANG_SO	Hằng số
a,b, i,j, chieuDai, chieuRong	Biến
tinhTong, tinhTrungBinh	Hàm
SinhVien, KhachHang	Kiểu dữ liệu mới

LỆNH, KHỐI LỆNH

(Xem mục 2.2.1 trong giáo trình)

▪ Câu lệnh:

- Câu lệnh là thực hiện một chỉ thị
- Kết thúc bằng dấu chấm phẩy “;”

```
// Đây là dòng chú thích
#include <iostream>
using namespace std;

int main ()
{
    cout << "Đây là chương trình C++";
    return 0;
}
```

▪ Khối lệnh:

- Tập hợp nhiều câu lệnh
- Nhóm trong cặp dấu ngoặc {}
- Các câu lệnh có thể viết trên cùng một dòng
- Các câu lệnh thực hiện tuần tự từ trên xuống dưới.

```
// Đây là dòng chú thích
#include <iostream>
using namespace std;

int main ()
{
    cout << "Đây là chương trình C++";
    return 0;
}
```

CHÚ THÍCH

(Xem trang 56, 57 trong giáo trình)

- Là lời giải thích vắn tắt cho câu lệnh hay đoạn chương trình
- Không ảnh hưởng đến kết quả thực thi của chương trình
- **Cách viết chú thích:**
 - Chú thích một dòng: sử dụng **//**
 - Chú thích nhiều dòng: sử dụng **/*** và ***/**
- **Ví dụ:**

```
/* Chương trình tính diện tích tam giác
   Nhập ba giá trị cho ba diện tích
   Tính diện tích tam giác và hiển thị kết quả */

#include <iostream>
using namespace std;

int main ()
{
    //Các câu lệnh bắt đầu từ đây
}
```


CÁC KIỂU DỮ LIỆU CƠ SỞ

Kiểu dữ liệu	Mô tả	Cỡ	Dải dữ liệu
char	Kiểu kí tự	1byte	signed: -128 to 127 unsigned: 0 to 255
short int(short)	Kiểu số nguyên	2bytes	signed: -32768 to 32767 unsigned: 0 to 65535
int	Kiểu số nguyên	4bytes	signed: -2147483648 -> 2147483647 unsigned: 0 to 4294967295
long int	Kiểu số nguyên	4bytes	signed: -2147483648 -> 2147483647 unsigned: 0 to 4294967295
bool	Kiểu logic	1byte	true or false
float	Kiểu số thực	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	Kiểu số thực	8bytes	+/- 1.7e +/- 308 (~15 digits)

CÁC KIỂU DỮ LIỆU CƠ SỞ

▪ Ví dụ:

➤ Ngày, tháng, năm: **int, short int**

- ngày:12, tháng: 1, năm: 2014

➤ Số sinh viên trong lớp: **int, short int**

- 40

➤ Chiều dài, chiều rộng của hình chữ nhật: **float, double**

- chiều dài: 23.123 (cm), chiều rộng: 12.243 (cm)

➤ Các chữ cái: **char**

- 'a', 'b', 'A', ..., 'Z'

➤ Các chữ số: **char**

- '0', '1', ... '9'



Các giá trị sau thuộc kiểu dữ liệu chuẩn nào?

a) 15.0

b) 1992

c) '?'

e) 15.172

d) $3 < 7$

h) $12/5$





Chọn kiểu dữ liệu phù hợp cho các trường hợp sau:

1. Tuổi của sinh viên
2. Người dùng có muốn thêm màu sắc hay không?
3. Pi (3.14159265)
4. Số trang của một văn bản
5. Giá của một cổ phiếu (2 số sau dấu phẩy)
6. Người sử dụng lựa chọn chức năng trên menu bằng các chữ cái



BIẾN

- Biến là nơi để chứa dữ liệu
- Mọi biến đều có tên và kiểu dữ liệu
- **Cú pháp khai báo:**

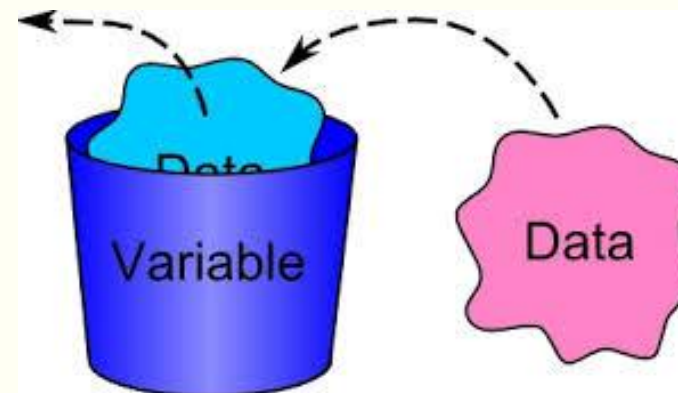
➤ Cách 1: **KieuDulieu** **tenBien**;

➤ Cách 2: **KieuDulieu** **tenBien1**, **tenBien2**;

➤ Cách 3: **KieuDulieu** **tenBien** = **giaTri**;

KieuDulieu **tenBien**(**giaTri**);

KieuDulieu **tenBien1** = **giaTri1**, **tenBien2** = **giaTri2**;



Mọi biến phải được khai báo trước khi sử dụng

Giá trị truyền vào cho biến phải phù hợp với kiểu dữ liệu khai báo

BIẾN

- Ví dụ 1: ➤ Cách 1: `int tuoi;` //Biến “tuoi” là kiểu số nguyên

`double chieuCao;`

➤ Cách 2: `int i, j;`

➤ Cách 3: `double vanToc = 5.2;`

`double giaToc = 9.81, thoiGian (1.5);`

- Ví dụ 2:

```
#include <iostream>
using namespace std;

int main ()
{
    int soLuong = 10.5; //Bien nay chi nhan gia tri 10
    float gia = 300.5;
    cout<<soLuong*gia;
    return 0;
}
```



Những khai báo biến sau đúng hay sai?



- 1. `int cats=5; dogs=5;`
- 2. `int ten bien;`
- 3. `int nValue1, double dValue2;`
- 4. `int nValue3(7), nValue4(8);`



Cần tạo bao nhiêu biến lưu dữ liệu đầu vào và kiểu dữ liệu tương ứng là gì?




- **Bài 1:** Nhập hai cạnh hình chữ nhật và tính diện tích hình chữ nhật đó.
- **Bài 2:** Nhập họ tên và tuổi sinh viên, hiển thị lên màn hình trên cùng một dòng.

BIẾN

- Phạm vi sử dụng của biến:

```
#include <iostream>
using namespace std;
int main ()
{
    int a;
    cin>>a; //Nhap gia tri cho a tu ban phim
    int b = a*2;
    if(a>0)
    {
        int c= b + a;
    }
    cout<< c;
    return 0;
}
```

Chưa khai báo ở phạm vi ngoài



TOÁN TỬ GÁN

(Xem mục 2.3.1 trong giáo trình)

- Là phép gán giá trị, biểu thức, hằng số, hàm vào **biến**

- **Dạng thức:**

```
tenBien = Bieu_thuc;
```

- Ví dụ:

```
x = 100;
```

```
y = PI * r * r;
```

```
z = x / 2;
```

```
s = "ABC";
```

```
a = 'A';
```

```
c = true;
```

- Gán liên tiếp: *nhiều biến gán cùng một giá trị.*

```
x = y = z = 200;
```

TOÁN TỬ GÁN

- Phép gán mở rộng:

$$X += y \quad \longleftrightarrow \quad X = X + y$$

$$X -= y \quad \longleftrightarrow \quad X = X - y$$

$$X *= y \quad \longleftrightarrow \quad X = X * y$$

$$X /= y \quad \longleftrightarrow \quad X = X / y$$

$$X \% = y \quad \longleftrightarrow \quad X = X \% y$$



Giá trị cuối cùng của **a**, **b** là bao nhiêu?

```
// Ví dụ về phép gán
#include <iostream>
using namespace std;

int main ()
{
    int a, b;           // a:?, b:?
    a = 10;             // a:?, b:?
    b = 4;              // a:?, b:?
    a = b;              // a:?, b:?
    b += a;             // a:?, b:?
    a += 7;             // a:?, b:?

    cout << "a:" << a;
    cout << " b:" << b;
}
```

HẰNG SỐ

- Hằng số chứa giá trị không đổi trong chương trình
- Giá trị của hằng số phải xác định ngay khi khai báo

- **Khai báo:**

- Cách 1: Dùng từ khóa tiền xử lý **#define**

- Cú pháp: **#define TEN_HANG_SO giatri**

- Cách 2: Dùng từ khóa **const**

- Cú pháp: **const kieu_dulieu TEN_HANG_SO = giatri;**

Không có dấu “= “ và “;”

HẲNG SỐ

▪ Ví dụ:

```
#include <iostream>
using namespace std;
#define GIA_TOC 9.81

int main ()
{
    const double V0 = 1.5;
    double t;
    cout<<"Nhập thời gian t=";
    cin>>t;
    cout<<" Vận tốc tại thời điểm "<<t<<" là : "<<V0 + GIA_TOC*t;
    return 0;
}
```

Tìm từ GIA_TOC, thay bằng 9.81

BIỂU THỨC VÀ TOÁN TỬ

(Xem mục 2.3 trong giáo trình)

- **Biểu thức (expression):** là tổ hợp các **toán tử** và **toán hạng**

- **Toán tử:** là các phép toán
- **Toán hạng:** là các biến, hằng số, hàm, giá trị cụ thể...

- **Ví dụ:**

$$7 + 2 * 5$$
$$2 * r * PI$$
$$a * (b + c / d)$$
$$5 + (c = 3 + 8)$$

- **Toán tử bao gồm:**

- Toán tử gán
- Toán tử so sánh
- Toán tử số học
- Toán tử điều kiện
- Toán tử logic

TOÁN TỬ SỐ HỌC

▪ Toán tử hai ngôi:

Toán tử	Chức năng	Ví dụ
+	Cộng	$x = a + 2;$
-	Trừ	$a = 7 - 5;$
*	Nhân	$b = 7 * 5;$
/	Chia	$c = 5 / 2;$ $d = 5 / 2.0;$
%	Lấy số dư của phép chia với số nguyên	$e = 4 \% 3; // e = 1$ $e = 4 \% 3.0; // \text{Lỗi}$



Chú ý với phép chia với số nguyên

TOÁN TỬ SỐ HỌC

▪ Toán tử một ngôi:

Toán tử	Chức năng	Ví dụ
-	Lấy số đối	$x = -20;$ $y = -x;$
++	Tăng một giá trị	$x++;$ $// x = x + 1$ $++y;$ $// y = y + 1$
--	Giảm một giá trị	$x--;$ $// x = x - 1$ $--y;$ $// y = y - 1$

TOÁN TỬ SỐ HỌC

- **Toán tử một ngôi – tăng/giảm tiền tố:** tăng hoặc giảm giá trị **trước khi** sử dụng giá trị của toán hạng

- Ví dụ:

```
a = 10;  
b = 5;  
c = a * ++b;    // c= ?
```

- **Toán tử một ngôi – tăng/giảm hậu tố:** tăng hoặc giảm giá trị **sau khi** sử dụng giá trị của toán hạng

- Ví dụ:

```
a = 10;  
b = 5;  
c = a * b++;    // c= ?
```

TOÁN TỬ LOGIC

(Xem mục 3.2 trong giáo trình)

Toán tử	Chức năng	Ví dụ
&&	Trả kết quả là True khi cả 2 toán hạng đều là True	<pre>((5 == 5) && (3 > 6)) // trả về false (true && false)</pre>
 	Trả về kết quả là True khi chỉ một trong 2 toán hạng là True	<pre>((5 == 5) (3 > 6)) // trả về true (true false)</pre>
!	Chuyển đổi giá trị từ True thành False và ngược lại	<pre>!(5 == 5) // trả về false !(6 <= 4) // trả về true !true // trả về false !false // trả về true</pre>

TOÁN TỬ ĐIỀU KIỆN BA NGÔI

- Cú pháp:

biethucdieukien ? Biethuc1 : Biethuc2

- Nếu biểu thức điều kiện là **True** thì trả về biểu thức 1, ngược lại trả về biểu thức 2

- Ví dụ:

(7==5) ? 4 : 3 // trả về ?

(7==5+2) ? 4 : 3 // trả về ?

(5>3) ? a : b // ?

(a>b) ? a : b // ?

TOÁN TỬ QUAN HỆ

(Xem mục 3.2 trong giáo trình)

Toán tử	Mô tả	Ví dụ
==	So sánh bằng	<code>7 == 5</code> // trả về false <code>(b=2) == 5</code> // trả về false
!=	Khác	<code>(3 != 2)</code> // trả về true
<	Nhỏ hơn	<code>(5 < 5)</code> // trả về false
>	Lớn hơn	<code>(3 > 2)</code> // trả về true
<=	Nhỏ hơn hoặc bằng	<code>(6 <= 6)</code> // trả về true
>=	Lớn hơn hoặc bằng	<code>(6 >= 4 +2)</code> // trả về true

TOÁN TỬ BIT

Toán tử	Mô tả	Ví dụ
&	AND	$0 \& 0 = 0$ $0 \& 1 = 0$ $1 \& 0 = 0$ $1 \& 1 = 1$
 	OR	$0 0 = 0$ $0 1 = 1$ $1 0 = 1$ $1 1 = 1$
^	XOR	$0 \wedge 0 = 0$ $0 \wedge 1 = 1$ $1 \wedge 0 = 1$ $1 \wedge 1 = 0$
~	NOT	$\sim 0 = 1$ $\sim 1 = 0$
<<	Dịch bit sang trái	$0001 \ll 1 = 0010$
>>	Dịch bit sang phải	$1000 \gg 1 = 0100$

ĐỘ ƯU TIÊN CỦA TOÁN TỬ

STT	Toán tử	Mô tả
1	++ -- ()	Tăng, giảm hậu tố, dấu ngoặc
2	++ -- ~ !	Tăng, giảm tiền tố, NOT bit, phủ định
3	* /	Nhân, chia
4	+ -	Cộng, trừ
5	<< >>	Dịch bit sang trái, Dịch bit sang phải
6	< > <= >=	So sánh hơn kém
7	== !=	So sánh bằng, khác
8	&	AND bit

STT	Toán tử	Mô tả
9	^	XOR
10	 	OR
11	&&	Toán tử logic AND
12	 	Toán tử logic OR
13	= *= /= %= += -=	Toán tử gán



Tính giá trị các biến sau?

```
int a= 10;  
int b = ++a + 10;           // b=?
```

```
int a= 10;  
int b = 10 + a++;           // b=?
```

```
int a= 10, b=5;  
b= ((a*3 > 27) && (b+5<18)) ? a*b : --a;  
// b=?
```




Tính giá trị các biểu thức sau?

1. $(5 > 3 \ \&\& \ 4 < 8)$
2. $(4 > 6 \ \&\& \ \text{true})$
3. $(3 \geq 3 \ || \ \text{false})$
4. $(\text{true} \ || \ \text{false}) ? 4 : 5$
5. $7 / 4$
6. $14 \% 5$
8. $7 / -2$
9. $-5 \% 2;$



CÁC HÀM TOÁN HỌC

(Xem mục 2.5.1 trong giáo trình)

- Thêm khai báo **#include <cmath>** trên phần tiền xử lý
- Hàm lượng giác:

Hàm	Mô tả
cos (x)	Tính cosin của một góc x đo bằng radian
sin (x)	Tính sin của một góc x đo bằng radian
tan (x)	Tính tag của một góc x đo bằng radian
acos (x)	Tính acos trả ra giá trị radian
asin(x)	Tính asin trả ra giá trị radian
atan(x)	Tính actag trả ra giá trị radian

cmath

CÁC HÀM TOÁN HỌC

■ Ví dụ:

```
/* cos example */
#include <iostream>
#include <cmath>          /* cos */
using namespace std;

#define PI 3.14159265

int main ()
{
    double param, result;
    param = 60.0;
    result = cos ( param * PI / 180.0 );
    cout<<"cosin của góc "<<param<< " độ là "<< result;
    return 0;
}
```

CÁC HÀM TOÁN HỌC

▪ Hàm mũ và logarit:

Hàm	Mô tả
exp(x)	Tính e^x , x có kiểu double , float
log (x)	Tính logarit cơ số e của x, x có kiểu double , float
log10 (x)	Tính logarit cơ số 10 của x, x có kiểu double , float


▪ Hàm lũy thừa, căn, lấy giá trị tuyệt đối:

Hàm	Mô tả
pow(x, y)	Tính x^y , x,y có kiểu double , float
sqrt(x)	Tính căn bậc 2 của x, x có kiểu double , float
cbrt (x)	Tính căn bậc 3 của x, x có kiểu double , float
abs(x)	Lấy giá trị tuyệt đối,

CÁC HÀM TOÁN HỌC


▪ Ví dụ:

e^3  `exp(3)`

$\log 10$  `log(10)`

2^5  `pow(2, 5)`

$2^{\frac{1}{3}}$  `pow(2, 1/3.0)`

$\sqrt{9}$  `pow(9, 1/2.0)`
`sqrt(9)`

CÁC HÀM TOÁN HỌC

▪ Hàm làm tròn:

Hàm	Mô tả
ceil(x)	Trả về số nguyên nhỏ nhất không nhỏ hơn x (làm tròn lên), x có kiểu double , float
floor (x)	Trả về số nguyên lớn nhất không lớn hơn x (làm tròn xuống), x có kiểu double , float

▪ Ví dụ:

```
ceil(2.2) // = ?
```

```
ceil(3.7) // = ?
```

```
ceil(-2.2) // = ?
```

```
ceil(-3.7) // = ?
```





Hãy viết các biểu thức sau theo ngôn ngữ C++

▪ Bài 1:

$$\sqrt{x + \sqrt{x + \sqrt{x}}}$$

▪ Bài 2:

$$x^3 + \log_2(x^4 + 2) - \sqrt[3]{x - 4}$$





THẢO LUẬN!



CHUẨN BỊ CHO BÀI SAU!

- **Nhập xuất:** *Mục 2.9 trong giáo trình*
- **Tệp tin (file):** *Mục 2.9.3 trong giáo trình*