



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)

Lớp: ĐHCNTT



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 (3 + 4)

04

Các thao tác trên dữ liệu kiểu mảng (6 + 6)

05

Hàm và phạm vi của biến (4 + 6)



NỘI DUNG MÔN HỌC (2)



06

Kiểu dữ liệu con trỏ (3 + 4)

07

Kiểu dữ liệu chuỗi & mẫu tin (4 + 6)

08

Kiểu tập tin (2 + 4)



Tài liệu tham khảo



Tài liệu bắt buộc chính:

[1] Nguyễn Thị Mỹ Dung, ***Bài giảng Lập trình căn bản***, Khoa SP Toán Tin – ĐH Đồng Tháp, năm 2021.

Tài liệu tham khảo:

[2] HN Aptech, GT Lập trình C căn bản, eBook, 2009.

[3] Hà Đại Dương & cs, BG Kỹ thuật lập trình, Học viện Kỹ thuật Quân sự, 2013.

[4] Easy Programming, C Programming Language The ULtimate Beginner's Guide, eBook, 2016

[5] Noel Kalicharan, Learn to Program with C: Learn to Program using the Popular C Programming Language, eBook, 2015



HÌNH THỨC ĐÁNH GIÁ



- Kiểm tra giữa kỳ: 0.4

- ❖ **Kiểm tra tính chuyên cần:** Đi học đầy đủ và báo cáo nhóm (0.1)
- ❖ **Kiểm tra lý thuyết:** trên lớp (0.15)
- ❖ **Kiểm tra thực hành :** bài tập tổng hợp (0.15)

- Thi kết thúc môn học: 0.6

- ❖ Nội dung từ chương 1 □ 8
- ❖ Thời gian: 90 phút
- ❖ **Được sử dụng tài liệu**



MỘT SỐ KHÁI NIỆM CƠ BẢN



- I. Giới thiệu ngôn ngữ lập trình**
- II. Phân loại ngôn ngữ lập trình**
- III. Chương trình dịch**
- IV. Thuật toán**
- V. Cấu trúc chung của chương trình viết bằng ngôn ngữ lập trình C.**
- VI. Các thao tác soạn thảo, biên dịch & chạy chương trình**
- VII. Phong cách lập trình**
- VIII. Bài tập**



1. Giới thiệu ngôn ngữ lập trình

- **Giới thiệu NNLT**
- Phân loại NNLT
- Chương trình dịch
- Thuật toán
- Cấu trúc NNLT C
- Soạn thảo, Biên dịch

➤ **Khái niệm NNLT:**

- Một chương trình máy tính (hay còn được gọi là phần mềm hoặc ứng dụng) là một tập hợp các câu lệnh, chúng nói cho máy tính biết phải làm những gì.

- Những **tập hợp các câu lệnh** được thiết kế để **ra lệnh cho máy tính** được gọi là Ngôn ngữ lập trình.



1. Giới thiệu ngôn ngữ lập trình

- **Giới thiệu NNLT**
- Phân loại NNLT
- Chương trình dịch
- Thuật toán
- Cấu trúc NNLT C
- Soạn thảo, Biên dịch

➤ **NNLT C:**

- Một chương trình máy tính (hay còn được gọi là phần mềm hoặc ứng dụng) là một tập hợp các câu lệnh, chúng nói cho máy tính biết phải làm những gì.

- Những **tập hợp các câu lệnh** được thiết kế để **ra lệnh cho máy tính** được gọi là Ngôn ngữ lập trình.



1. Giới thiệu ngôn ngữ lập trình

- Giới thiệu NNLT
- Phân loại NNLT
- Chương trình dịch
- Thuật toán
- Cấu trúc NNLT C
- Soạn thảo, Biên dịch

➤ Ưu điểm:

- C là ngôn ngữ mềm dẻo. C được dùng để viết HĐH, các chương trình điều khiển, soạn thảo văn bản, đồ hoạ, bảng tính...
- C là ngôn ngữ dễ thích nghi.
- C là ngôn ngữ có ít **từ khoá**.
- C là ngôn ngữ có cấu trúc Module, đó chính là sử dụng **hàm**.



1. Giới thiệu ngôn ngữ lập trình

- Giới thiệu NNLT
- Phân loại NNLT
- Chương trình dịch
- Thuật toán
- Cấu trúc NNLT C
- Soạn thảo, Biên dịch

➤ **Khuyết điểm**

- Cú pháp thuộc loại lạ và khó học nếu chưa học qua bất kỳ NNLT nào.
- Một số ký hiệu của C có nhiều nghĩa.
*Ví dụ: **dấu “*”** là toán tử **nhân**, là toán tử **không định hướng**, là toán tử **thay thế**...*
- Việc truy nhập tự do vào dữ liệu, việc trộn lẫn các kiểu dữ liệu... làm cho chương trình có phần bất ổn.



2. Phân loại NNLT

- Giới thiệu NNLT
- **Phân loại NNLT**
- Chương trình dịch
- Thuật toán
- Cấu trúc NNLT C
- Soạn thảo, Biên dịch

➤ Chia làm 3 cấp:

- Ngôn ngữ cấp cao: Pascal, Java, C#, Python,...
- Ngôn ngữ cấp trung: C
- Ngôn ngữ cấp thấp - ngôn ngữ máy (chương trình viết dạng nhị phân): Assembly



*Từ ngôn ngữ cấp cao để máy tính có thể thực thi phải thông qua một **chương trình dịch**: **biên dịch** hoặc **thông dịch**.*



3. Chương trình dịch

- Giới thiệu NNLT
- Phân loại NNLT
- **Chương trình dịch**
- Thuật toán
- Cấu trúc NNLT C
- Soạn thảo, Biên dịch

➤ **Chương trình dịch có 2 loại là: biên dịch & thông dịch**

- **Biên dịch:** kiểm tra, phát hiện lỗi, xác định chương trình nguồn có dịch được không. ***Dịch toàn bộ chương trình nguồn thành một chương trình đích*** có thể thực hiện trên máy.

- **Thông dịch:** lần lượt ***dịch từng câu lệnh ra ngôn ngữ máy rồi thực hiện*** ngay câu lệnh vừa dịch được hoặc thông báo lỗi nếu không dịch được



Xác định biên dịch hay thông dịch?





Chương trình dịch

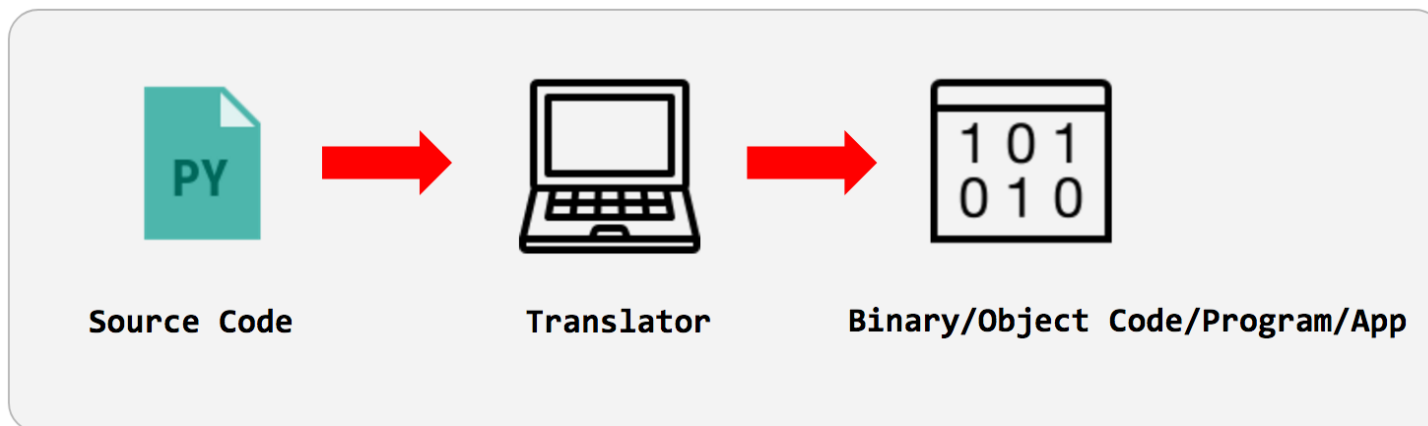
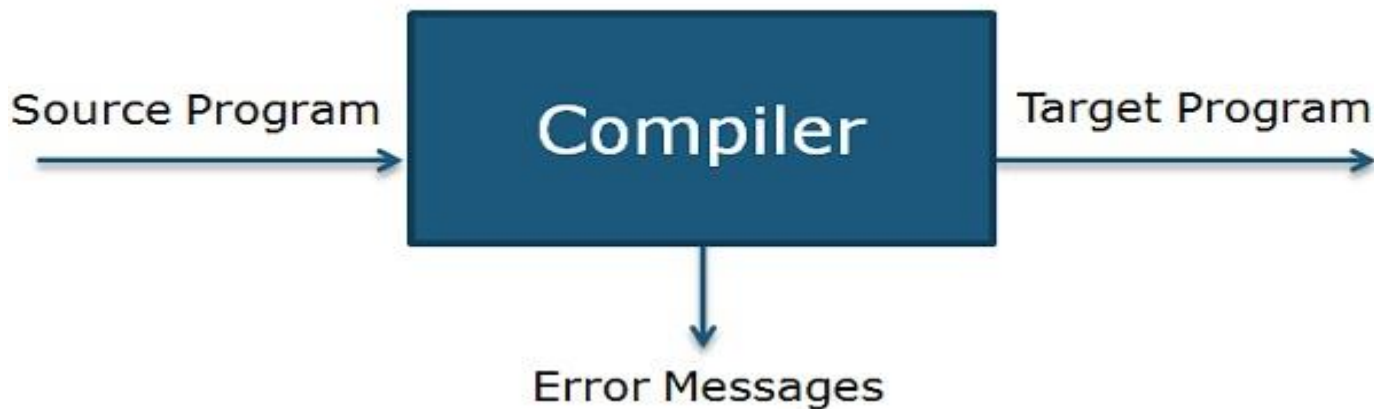
- Giới thiệu NNLT
- Phân loại NNLT
- **Chương trình dịch**
- Thuật toán
- Cấu trúc NNLT C
- Soạn thảo, Biên dịch

Tiêu chí	Trình biên dịch	Trình thông dịch
Đầu vào	Toàn bộ trường trình	Chỉ một dòng code
Đầu ra	Mã đối tượng trung gian	Không tạo ra bất kì mã đối tượng trung gian nào
Cơ chế hoạt động	Việc biên dịch sẽ phải hoàn thành công việc trước khi thực thi	Việc biên dịch và thực thi sẽ là đồng thời
Tốc độ	Nhanh hơn	Chậm hơn
Bộ nhớ	Yêu cầu bộ nhớ nhiều hơn do việc tạo mã đối tượng	Nó đòi hỏi ít bộ nhớ hơn vì nó không tạo mã đối tượng trung gian
Errors	Hiển thị tất cả các lỗi sau khi biên dịch, tất cả cùng một lúc	Hiển thị lỗi của từng dòng một
Phát hiện error	Rất khó khăn	Tương đối dễ
Các ngôn ngữ lập trình	C, C++, C#, Scala, typescript	PHP, Perl, Python, Ruby



Chương trình dịch

- Giới thiệu NNLT
- Phân loại NNLT
- **Chương trình dịch**
- Thuật toán
- Cấu trúc NNLT C
- Soạn thảo, Biên dịch



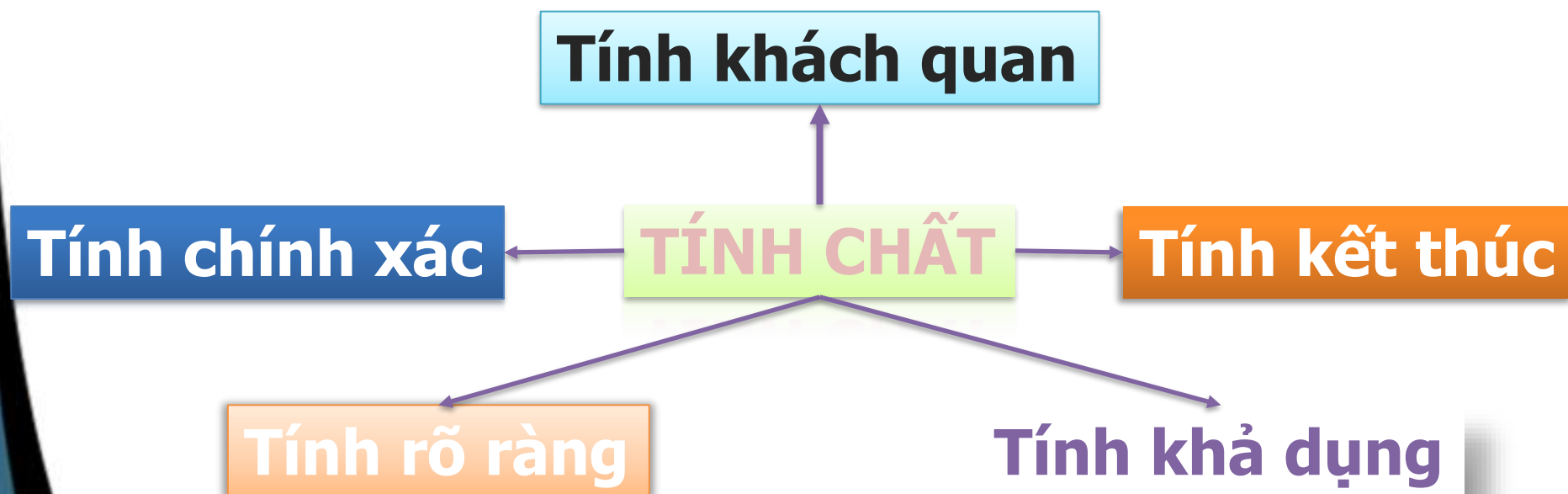


4. Thuật toán

- Giới thiệu NNLT
- Phân loại NNLT
- Chương trình dịch
- **Thuật toán**
- Cấu trúc NNLT C
- Soạn thảo, Biên dịch

➤ Khái niệm:

Thuật toán hay giải thuật (Algorithm) là một tập hợp hữu hạn các hướng dẫn được xác định rõ ràng, có thể thực hiện được bằng máy tính, thường **để giải quyết một lớp vấn đề hoặc để thực hiện một phép tính.**



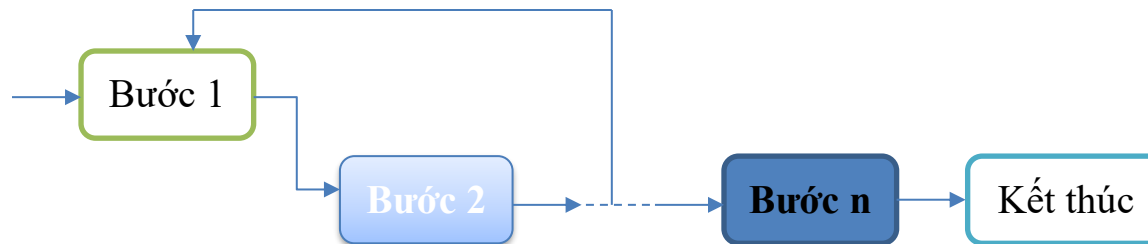


4. Thuật toán

- Giới thiệu NNLT
- Phân loại NNLT
- Chương trình dịch
- **Thuật toán**
- Cấu trúc NNLT C
- Soạn thảo, Biên dịch

➤ **Biểu diễn thuật toán:** Có 3 cách để mô tả

① Sử dụng ngôn ngữ tự nhiên.



Minh họa

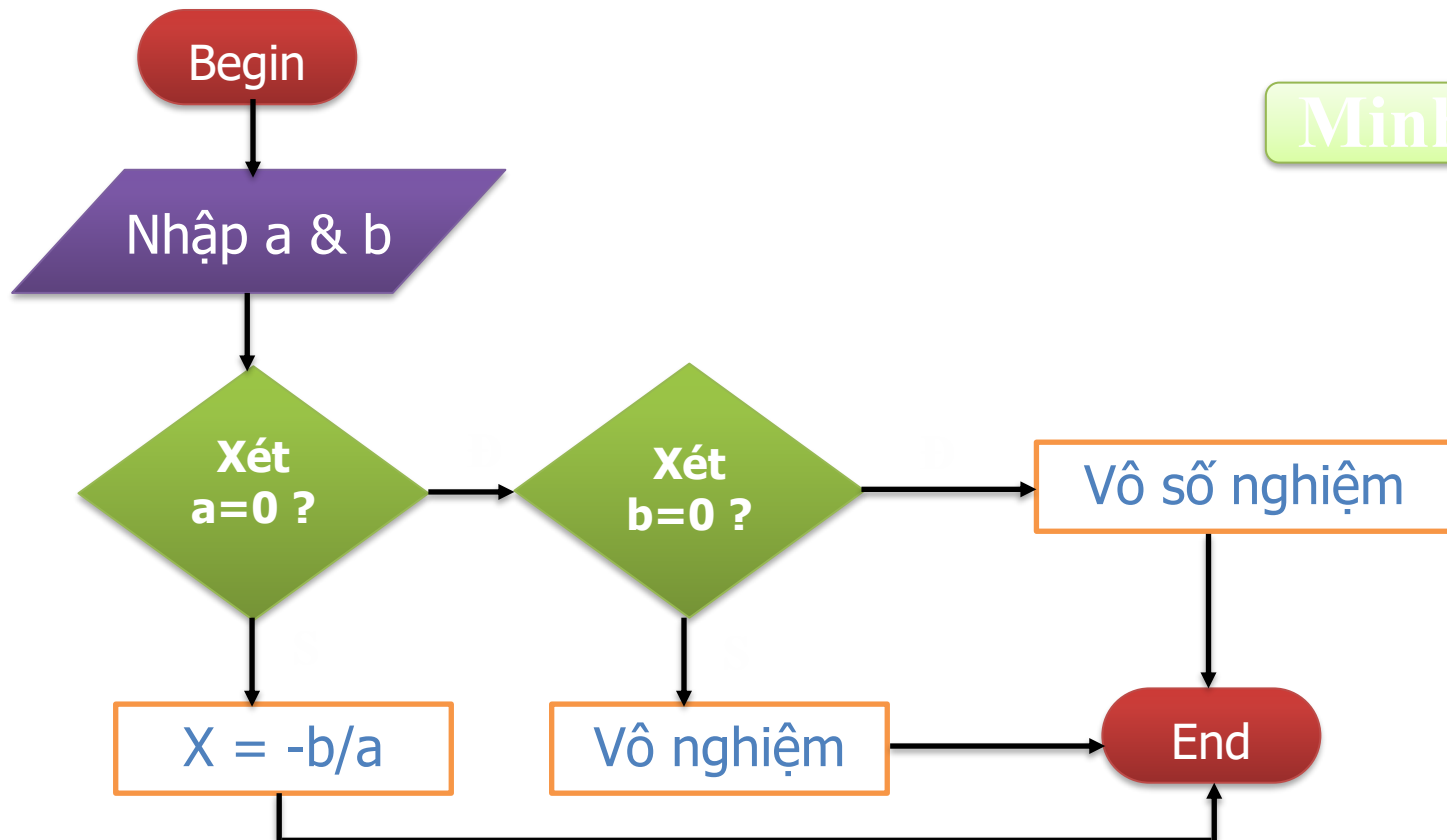


4. Thuật toán

- Giới thiệu NNLT
- Phân loại NNLT
- Chương trình dịch
- **Thuật toán**
- Cấu trúc NNLT C
- Soạn thảo, Biên dịch

➤ **Biểu diễn thuật toán:** Có 3 cách để mô tả

② Sử dụng lưu đồ giải thuật





4. Thuật toán

- Giới thiệu NNLT
- Phân loại NNLT
- Chương trình dịch
- **Thuật toán**
- Cấu trúc NNLT C
- Soạn thảo, Biên dịch

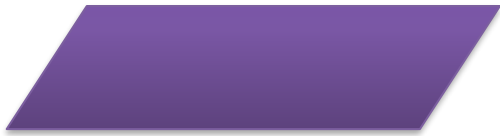
② Sử dụng lưu đồ giải thuật: Một số ký hiệu



Bắt đầu hoặc kết thúc



Thao tác xử lý



Dữ liệu đầu vào/ đầu ra



Thao tác lựa chọn



Hướng đi dữ liệu



4. Thuật toán

- Giới thiệu NNLT
- Phân loại NNLT
- Chương trình dịch
- **Thuật toán**
- Cấu trúc NNLT C
- Soạn thảo, Biên dịch

③ Sử dụng mã giả: có thể vay mượn NNLT nào đó

Đầu vào: a, b thuộc \mathbb{R}

Đầu ra: nghiệm phương trình $ax + b = 0$

```
If  $a = 0$  Then
```

```
Begin
```

```
    If  $b = 0$  Then
```

```
        Writeln “Phương trình vô số nghiệm”
```

```
    Else
```

```
        Writeln “Phương trình vô nghiệm”
```

```
End
```

```
Else
```

```
    Xuất “Phương trình có nghiệm  $x = -b/a$ ”
```



5. Cấu trúc chung của NNLT C

- Giới thiệu NNLT
- Phân loại NNLT
- Chương trình dịch
- Thuật toán
- **Cấu trúc NNLT C**
- Soạn thảo, Biên dịch

➤ Cấu trúc chung

Lệnh tiền xử lý

Kiểu dữ liệu tự định nghĩa

Khai báo nguyên mẫu hàm

Chương trình chính

Định nghĩa hàm



5. Cấu trúc chung của NNLT C

- Giới thiệu NNLT
- Phân loại NNLT
- Chương trình dịch
- Thuật toán
- **Cấu trúc NNLT C**
- Soạn thảo, Biên dịch

➤ Các thành phần

```
#include
```

```
typedef
```

```
int [void | ... ] <ten_ham> ();
```

```
main() {  
    ...};
```

```
int [void | ... ] <ten_ham> () {  
    ...};
```

```
// Ghi chú trên dòng  
/* Ghi chú nhiều  
   dòng */
```

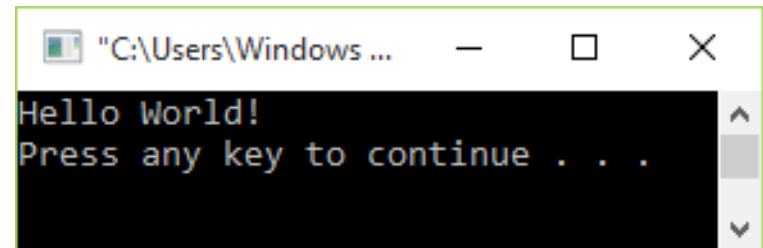


Cấu trúc chung của NNLT C

- Giới thiệu NNLT
- Phân loại NNLT
- Chương trình dịch
- Thuật toán
- **Cấu trúc NNLT C**
- Soạn thảo, Biên dịch

Ví dụ: Chương trình đầu tiên

```
#include <stdio.h>
int main()
{
    printf("Hello World!\n");
    return 0;
}
```



Click xem kết quả



6. Soạn thảo và biên dịch

- Giới thiệu NNLT
- Phân loại NNLT
- Chương trình dịch
- Thuật toán
- Cấu trúc NNLT C
- **Soạn thảo, Biên dịch**

➤ *Khi lập trình với ngôn ngữ C thông qua các bước:*

- Soạn thảo chương trình nguồn (C, Notepad, C free,...), lưu lại với phần mở rộng *.C hoặc *.CPP.
- Thông qua chương trình dịch Turbo C++ 3.0 (hoặc: DevC++; C Free, Visual C++, Code Blocks,...)
- Một số trang web Compiler online:

https://rextester.com/l/c_online_compiler_gcc

<https://www.jdoodle.com/c-online-compiler/>

https://www.onlinegdb.com/online_c_compiler

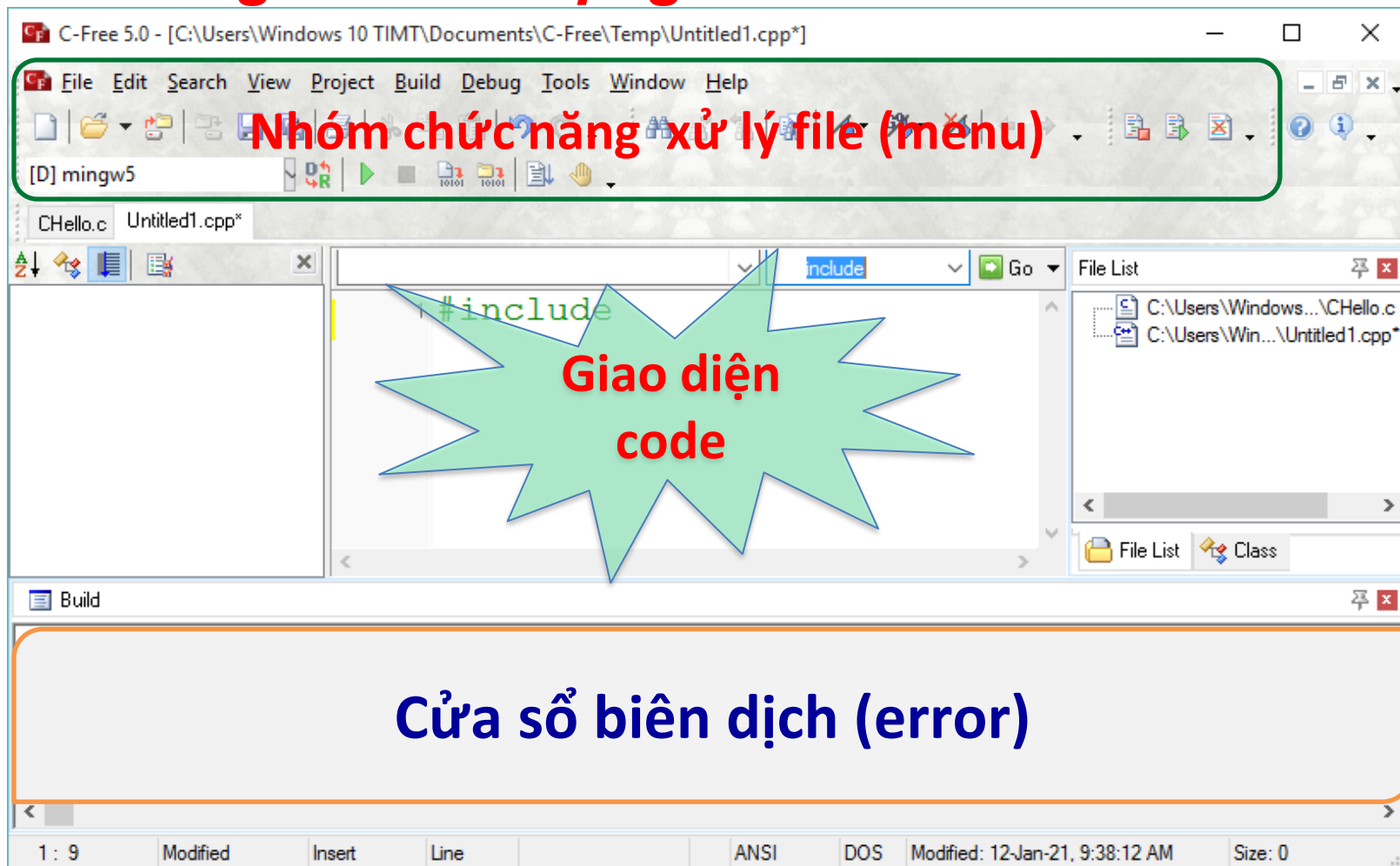
...



6. Soạn thảo và biên dịch

- Giới thiệu NNLT
- Phân loại NNLT
- Chương trình dịch
- Thuật toán
- Cấu trúc NNLT C
- **Soạn thảo, Biên dịch**

➤ Hướng dẫn sử dụng C Free 5.0

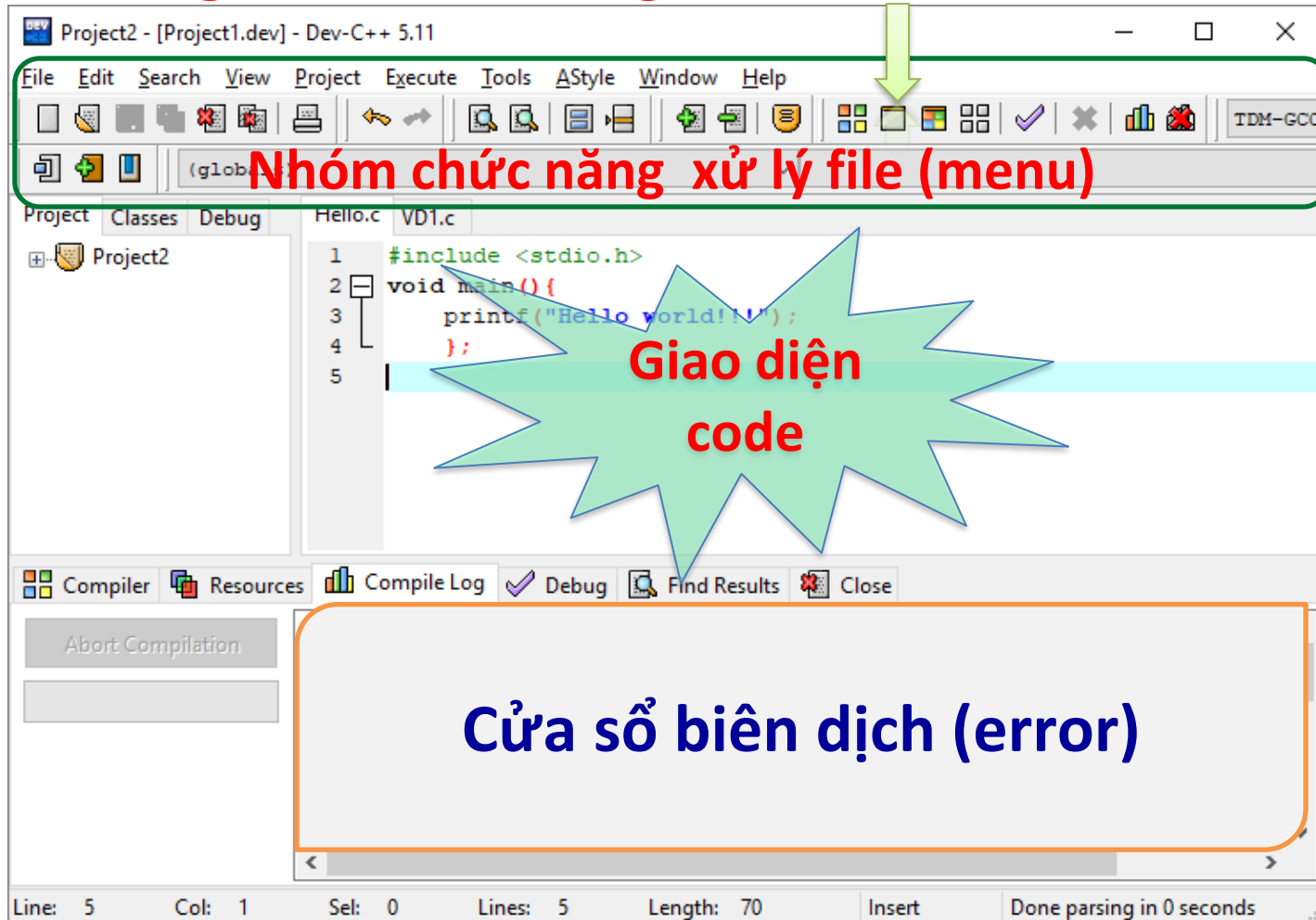




6. Soạn thảo và biên dịch

- Giới thiệu NNLT
- Phân loại NNLT
- Chương trình dịch
- Thuật toán
- Cấu trúc NNLT C
- Soạn thảo, Biên dịch

➤ Hướng dẫn sử dụng DevC++

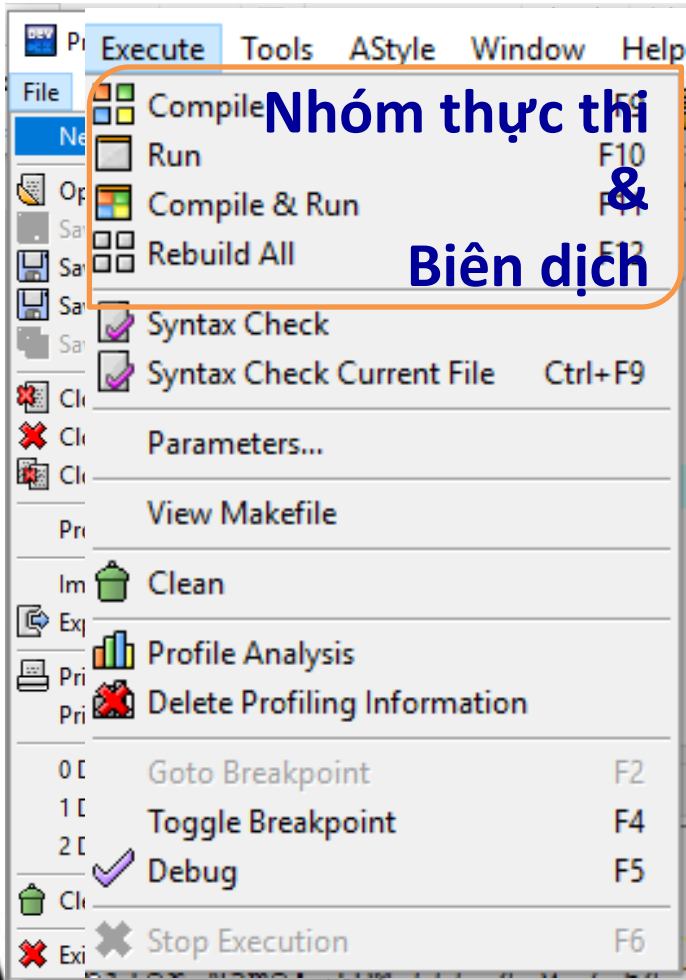




6. Soạn thảo và biên dịch

- Giới thiệu NNLT
- Phân loại NNLT
- Chương trình dịch
- Thuật toán
- Cấu trúc NNLT C
- **Soạn thảo, Biên dịch**

Hướng dẫn sử dụng DevC++ (tt)



Soạn thảo chương trình nguồn

Cách 1:

File / New / Source File (Ctrl N)

Cách 2:

File / New / Project...

Biên dịch chương trình

Run – F10 : Thực thi

Compile – F9: Biên dịch



7. Phong cách lập trình



Lưu ý:

- Mỗi câu lệnh nên đặt trên 1 dòng;
- Nên ghi chú trước những đoạn chương trình;

```
1 #include <stdio.h>
2 int main()
3 {
4     printf("Hello World!\n");
5     return 0;
6 }
```

```
1- /******
2
3         Online C Compiler.
4         Code, Compile, Run and Debug C program online.
5         Write your code in this editor and press "Run" button.
6
7         *****
8
9 #include <stdio.h>
10
11 int main()
12 {
13     printf("Hello World");
14
15     return 0;
16 }
17
```

1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 int main() {
5 // Write C code here
6 printf("Hello world");
7
8 return 0;
9 }

C++14 (Gcc 6.3)

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     // your code goes here
6
7     return 0;
8 }
```



Câu hỏi ôn tập???



Sử dụng phương pháp biểu diễn thuật toán để mô tả các bài toán:

1. Giải và biện luận phương trình bậc 2: $ax^2 + bx + c = 0$.
2. Tìm số nhỏ nhất trong dãy số nguyên.
3. Xác định số nguyên tố của một số nguyên dương.
4. Tìm USCLN của 2 số nguyên dương.
5. Tính tổng của một dãy số.
6. Sắp xếp dãy số theo thứ tự tăng.



TỔNG KẾT CHƯƠNG 1



- Tìm hiểu về NNLT;
- Thuật toán và biểu diễn thuật toán;
- Giới thiệu DevC++, C Free, C compiler Online;
- Cấu trúc chương trình C;
- Thực hành demo mẫu.

