

BỘ CÔNG THƯƠNG
ĐẠI HỌC CÔNG NGHIỆP TP. HỒ CHÍ MINH



Bài giảng

NGÔN NGỮ LẬP TRÌNH C
THE C PROGRAMMING LANGUAGE

Lecturer : Le Ngoc Tran, PhD

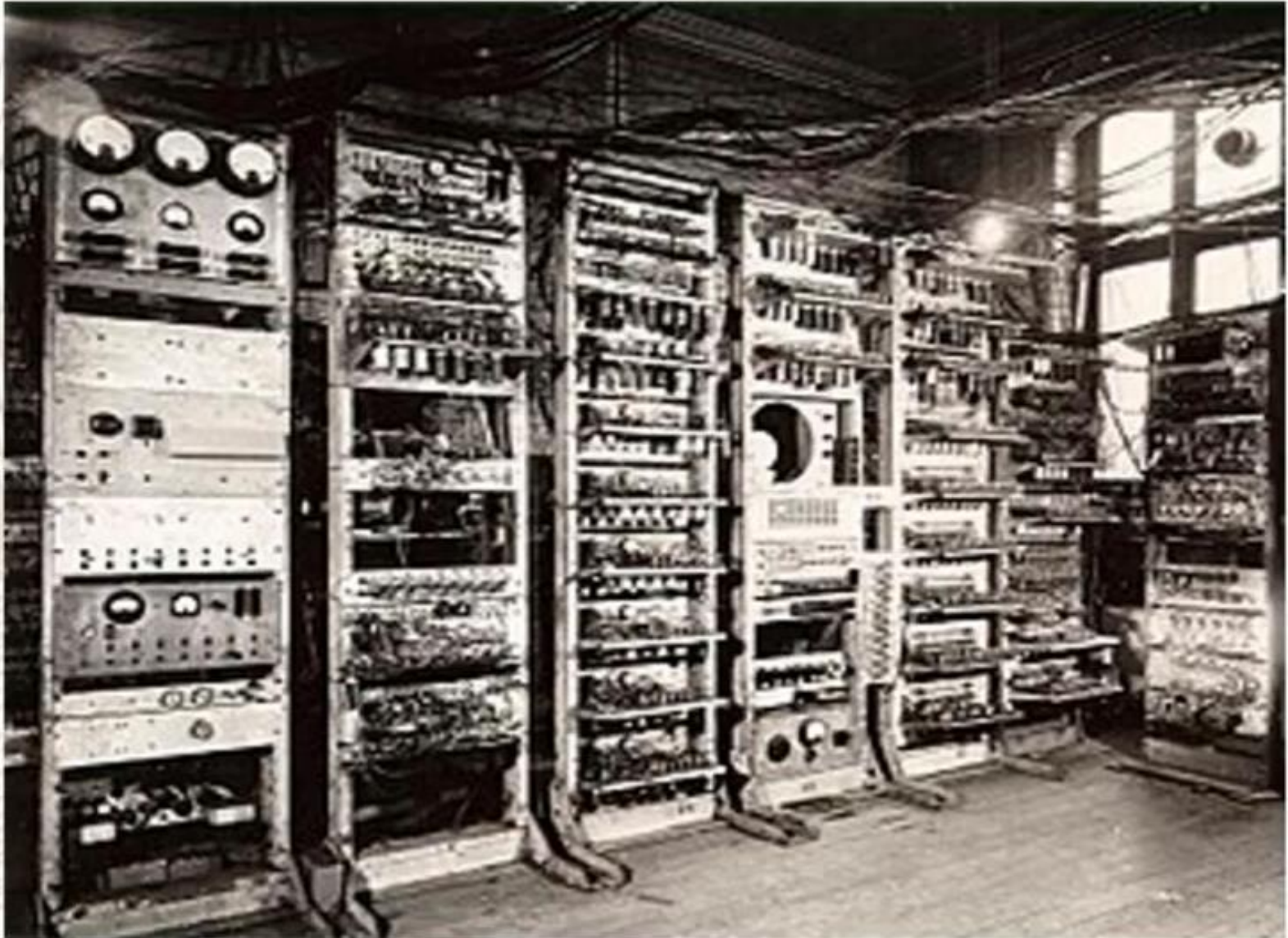
Email : lengoctran@iuh.edu.vn

- ❖ **Chương 1:** Tổng quan về lập trình máy tính và ngôn ngữ lập trình
- ❖ **Chương 2:** Các thành phần cơ bản và các kiểu dữ liệu trong C
- ❖ **Chương 3:** Các lệnh điều kiện và rẽ nhánh
- ❖ **Chương 4:** Các lệnh vòng lặp
- ❖ **Chương 5:** Hàm
- ❖ **Chương 6:** Mảng
- ❖ **Chương 7:** Chuỗi
- ❖ **Chương 8:** Cấu trúc
- ❖ **Chương 9:** Pointer

- ❖ Đặng Thành Tín, **Tin học II**, NXB ĐH Quốc Gia Tp. HCM, 2007
- ❖ 2. Brian W. Kernighan, Dennis M. Ritchie, **The C Programming Language** - 2nd ed, Prentice Hall, 1989.
- ❖ 3. Paul Deitel, Harvey Deitel, C How To Program – 6th ed., Prentice Hall, 2010.
- ❖ 4. Nguyễn Viễn Quốc, Bài giảng ngôn ngữ lập trình C.

TỔNG QUAN VỀ LẬP TRÌNH MÁY TÍNH VÀ NGÔN NGỮ LẬP TRÌNH

1.1. Tổng quan - Máy tính đầu tiên



Manchester Mark 1

1.1. Tổng quan - lập trình có từ khi nào

- ❑ Khi máy tính ra đời vào năm 1837, thì việc hướng dẫn cho máy tính làm việc thời bấy giờ được gọi là **lập trình máy tính**. Tuy nhiên những việc làm đó chưa được tổ chức thành những tập lệnh rõ ràng.
- ❑ Năm 1883, bà Ada Lovelace đã tạo ra ngôn ngữ lập trình đầu tiên ✓
- ❑ Sau đó các mã lệnh được thay thế bằng các tên gọi nhớ và lập trình được ở dạng văn bản (text) rồi dịch sang mã máy, gọi là ngôn ngữ lập trình thế hệ 2
- ❑ Ngôn ngữ bậc cao (High-level programming languages) hay **ngôn ngữ lập trình thế hệ 3 ra đời vào những năm 1950.**

- ❑ 1957: John Backus tạo ra FORTRAN là ngôn ngữ đầu tiên được các lập trình viên thực sự sử dụng.
- ❑ 1957: Grace Hopper phát minh ra ngôn ngữ lập trình hướng thương mại COBOL.
- ❑ 1964: John Kemery và Thomas Kurtz thấy rằng các ngôn ngữ lập trình hiện tại quá khó và họ đã tạo ra một ngôn ngữ đơn giản, dễ sử dụng và gọi nó là BASIC.
- ❑ 1970: Niklaus Wirth tạo ra Pascal
- ❑ 1972: Dennis Ritchie khi đang làm việc tại Bell Labs, ông đã tạo ra ngôn ngữ lập trình **C**.

1. Tổng quan - lịch sử của các ngôn ngữ lập trình

- ❑ 1983: Bjarne Stroustrup bổ sung thêm nhiều tính năng nâng cao mà ông có thể nghĩ ra cho C và đặt cho ngôn ngữ lập trình này là **C++**
- ❑ 1986: Brac Box và Tol Move quyết định tạo thêm một phiên bản C dựa trên Smalltalk, cái được gọi là **Objective-C**.
- ❑ 1991: Guido Van Rossum phát minh ra **Python**
- ❑ 1991: **Java** được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun Microsystems.
- ❑ 1994: Rasmus tung ra một vài liên kết cơ sở dữ liệu mở rộng cho nó và gọi là **PHP**.
- ❑ 1995: Brendan Eich thiết kế một ngôn ngữ sẽ được sử dụng rộng rãi trên mọi trình duyệt và gọi nó là **JavaScript**.

Tổng quan - Các NN lập trình phổ biến

<u>Sep 2020</u>	<u>Sep 2019</u>	Change	Programming Language	Ratings	Change
1	2	▲	<u>C</u>	<u>15.95%</u>	+0.74%
2	1	▼	<u>Java</u>	13.48%	-3.18%
3	3		<u>Python</u>	10.47%	+0.59%
4	4		<u>C++</u>	<u>7.11%</u>	+1.48%
5	5		<u>C#</u>	<u>4.58%</u>	+1.18%
6	6		<u>Visual Basic</u>	4.12%	+0.83%
7	7		<u>JavaScript</u>	2.54%	+0.41%
8	9	▲	<u>PHP</u>	2.49%	+0.62%
9	19	▲▲	<u>R</u>	2.37%	+1.33%
10	8	▼	<u>SQL</u>	1.76%	-0.19%
11	14	▲	<u>Go</u>	1.46%	+0.24%

1.1. Tổng quan - Ngôn ngữ lập trình mới

☐ Rust

☐ Go

☐ Kotlin

☐ TypeScript

☐ F#

☐ Dart



TypeScript



Tổng quan - Lập trình đóng vai trò trong cuộc sống

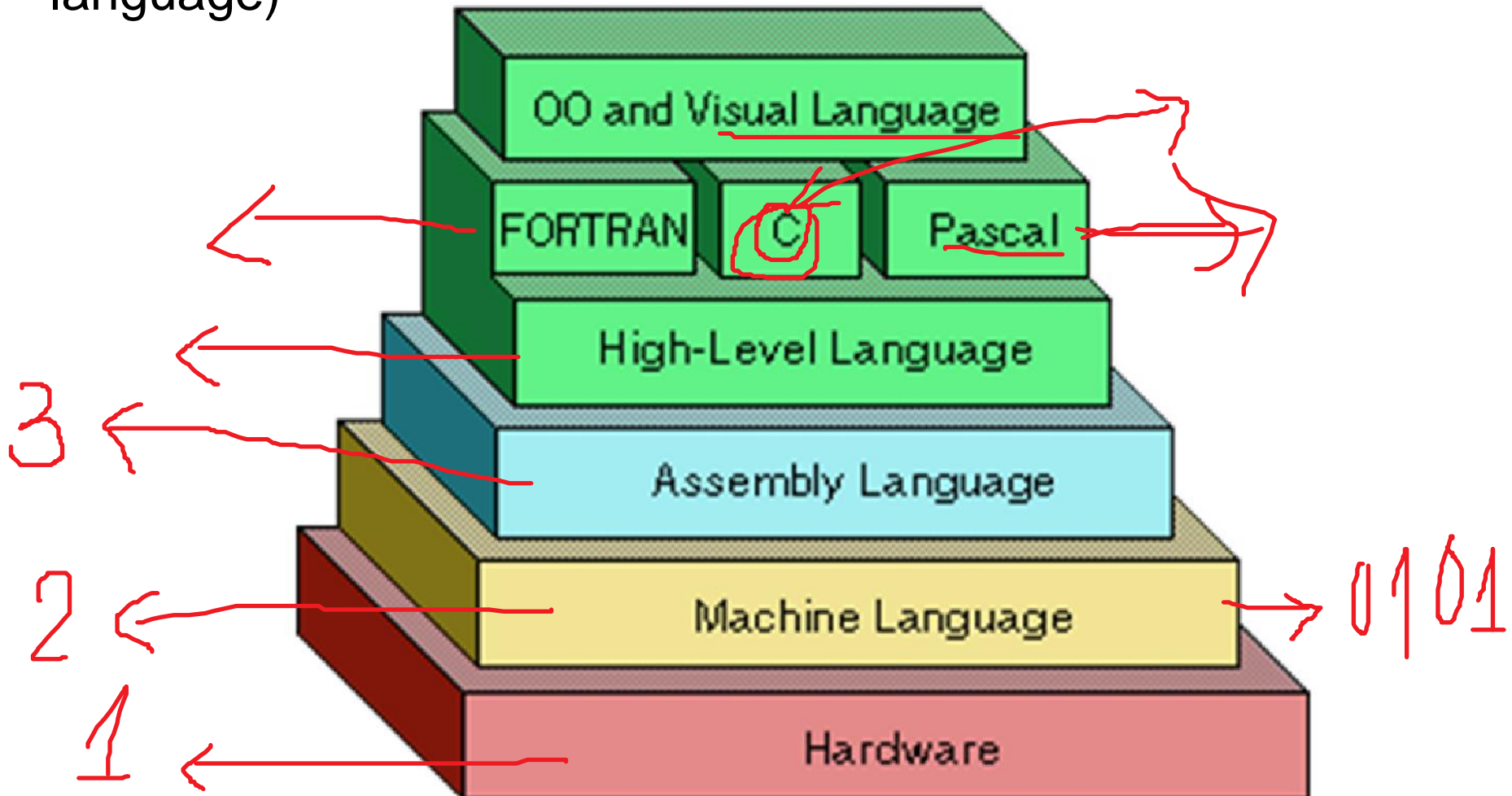
- ❑ Lập trình để tạo ra các phần mềm phục vụ cho đời sống con người. ✓
- ❑ Tất cả các lĩnh vực từ nông nghiệp, khoa học, y tế, khám phá không gian,...đều sử dụng Công nghệ thông tin nói chung và phần mềm nói riêng. ✓
- ❑ Phần mềm có thể làm thay đổi thế giới:
 - ❖ Trí tuệ nhân tạo ✓
 - ❖ Máy học ✓
 - ❖

Tổng quan - Làm thế nào để học tốt lập trình?

- ☐ Có niềm đam mê ✓
- ☐ Thực hành thật nhiều ✓
- ☐ Thảo luận và tranh luận ✓
- ☐ Viết ra những gì chợt đến trong đầu ✓
- ☐ Thu thập kiến thức hàng ngày về lĩnh vực liên quan ✓
- ☐ Rèn luyện sức khỏe ✓
- ☐ Có kỹ năng tìm kiếm thông tin ✓
- ☐ Học tập thêm tiếng anh ✓
- ☐ ...

1.2. Giới thiệu về ngôn ngữ lập trình C

- ❖ C là ngôn ngữ lập trình bậc cao (high-level programming language)



1.2. Giới thiệu về ngôn ngữ lập trình C

- ❖ Các ngôn ngữ phát triển từ ngôn ngữ C
 - C++
 - C#, JAVA, Python, Swift đều có sự kế thừa nhất định từ C
- ❖ Microsoft Windows, Linux, MacOS được phát triển chủ yếu dựa trên ngôn ngữ lập trình C
- ❖ iOS, Android và Windows Phone đều được viết bằng ngôn ngữ C
- ❖ Các phim 3D mà bạn đang xem ngày nay được viết từ ngôn ngữ C và C++
- ❖ Đồng hồ báo thức, lò vi sóng, máy pha café đều được lập trình embedded từ C.

1.2. Giới thiệu về ngôn ngữ lập trình C

❖ Sự phân biệt giữa ngôn ngữ C và C++

<u>Ngôn ngữ C</u>	Ngôn ngữ C++
<input type="checkbox"/> C là ngôn ngữ ra đời trước	<input type="checkbox"/> C++ ra đời sau mở rộng từ C
<input type="checkbox"/> Là ngôn ngữ hướng thủ tục	<input type="checkbox"/> Là ngôn ngữ lập trình hướng đối tượng
<input type="checkbox"/> C là nền tảng của C++	<input type="checkbox"/> C++ không ra đời để thay thế C

1.2. Giới thiệu về ngôn ngữ lập trình C

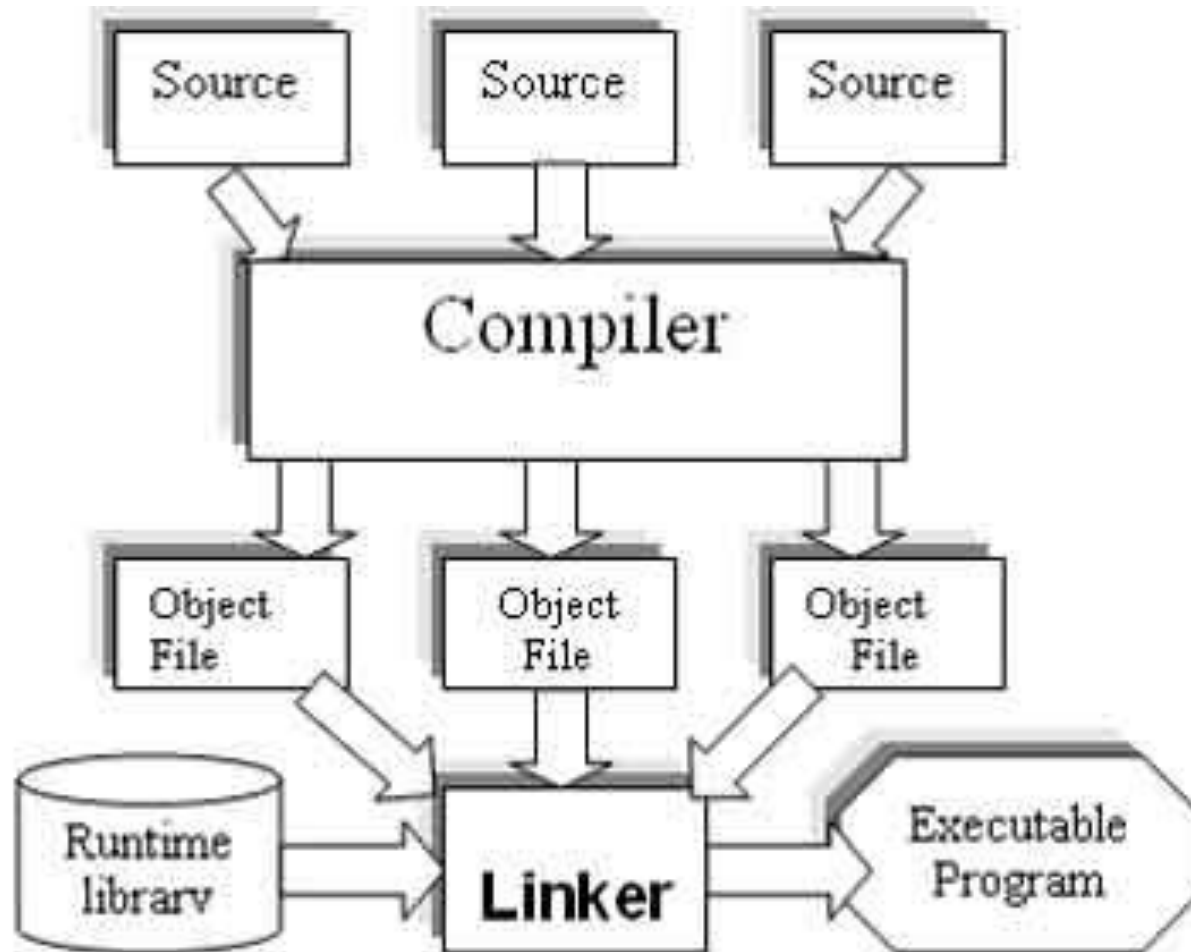
❖ Ngôn ngữ lập trình và Ngôn ngữ máy

Ngôn ngữ lập trình	Ngôn ngữ máy
<input type="checkbox"/> C là ngôn ngữ được tạo ra giúp con người có thể sử dụng bằng ngôn ngữ <u>thông thường</u> (số, chữ,...)	<input type="checkbox"/> Ngôn ngữ máy là một loại ngôn ngữ lập trình trong đó mọi chỉ thị đều được biểu diễn bằng các con số nhị phân (<u>0 và 1</u>)
<input type="checkbox"/> Cần <u>trình biên dịch</u> để chuyển sang ngôn ngữ máy	<input type="checkbox"/> Ngôn ngữ máy lại là ngôn ngữ duy nhất mà bộ vi xử lý (CPU) có thể nhận biết, hiểu được và thực hiện
<input type="checkbox"/> Có rất nhiều ngôn ngữ lập <u>trình khác nhau</u>	<input type="checkbox"/> Chỉ có <u>1 ngôn ngữ máy</u>

1.2. Giới thiệu về ngôn ngữ lập trình C

- ❖ C kết hợp các tính năng của ngôn ngữ bậc cao với ngôn ngữ bậc thấp, nên một số tài liệu xếp **ngôn ngữ C thuộc loại ngôn ngữ bậc trung** (middle level language) → cho phép lập trình linh hoạt, hiệu quả → được xem là ngôn ngữ lập trình chuyên nghiệp.
- ❖ **Các thao tác thực hiện khi lập trình:**
 - ■ Soạn thảo (edit) → text file (có đuôi là ‘.c’)
 - ■ Biên dịch (compile) → object file (có đuôi là ‘.o’ hay ‘.obj’)
 - ■ Liên kết (link) → file chạy được (có đuôi là ‘.exe’)

1.2. Giới thiệu về ngôn ngữ lập trình C



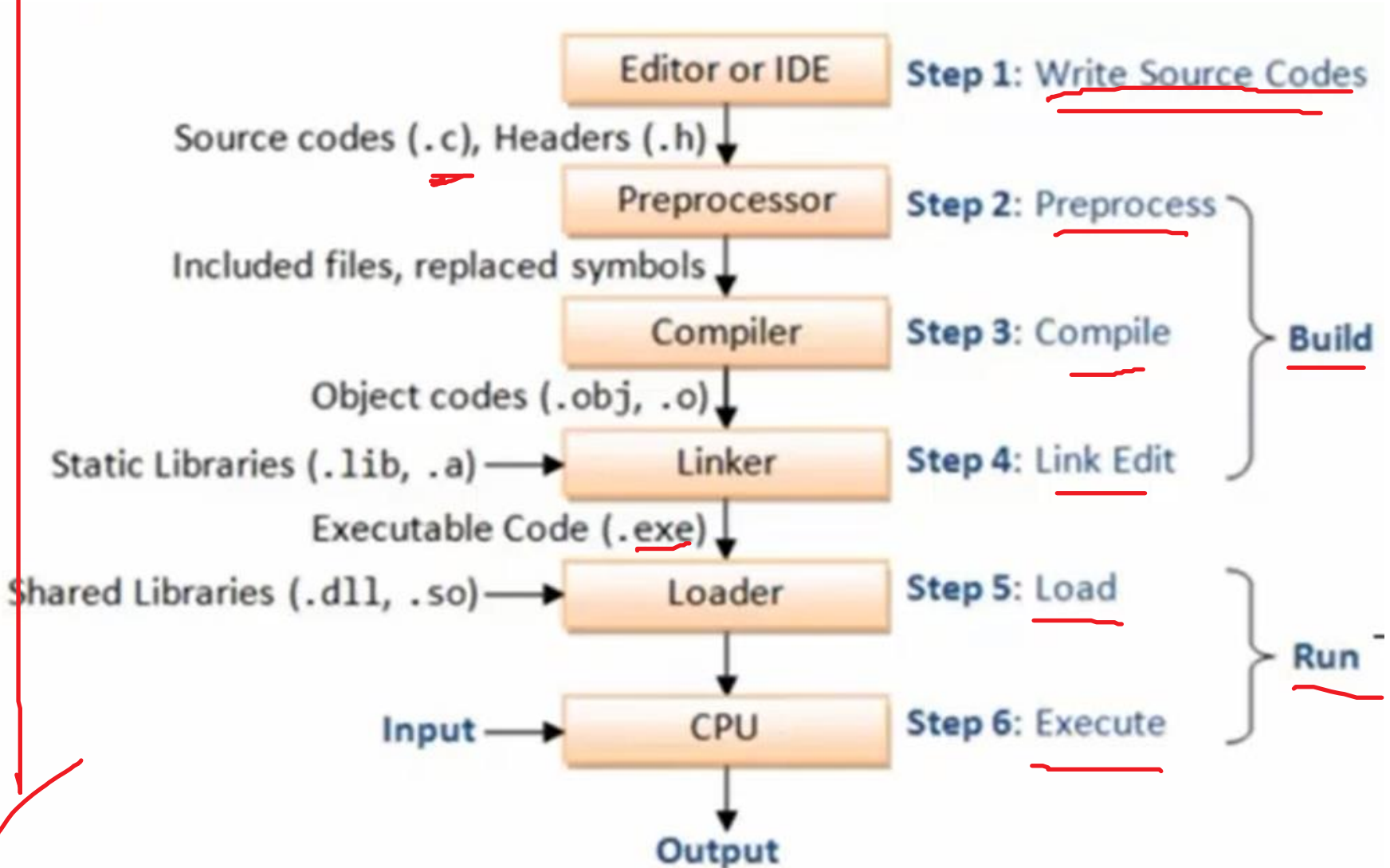
1.2. Giới thiệu về ngôn ngữ lập trình C

❖ Trình biên dịch (compiler)

- ❑ Khi biên dịch từ mã nguồn, trình biên dịch sẽ:
 - ❑ Kiểm tra các cú pháp mã nguồn, ✓
 - ❑ Thực hiện các phép kiểm tra logic ✓
 - ❑ Đảm bảo các dữ liệu được sắp xếp đúng trình tự ✓
- ❑ Trong quá trình biên dịch các lỗi được phát hiện sẽ được liệt kê thành danh sách để lập trình viên sửa đổi các lỗi một cách dễ dàng

1.2. Giới thiệu về ngôn ngữ lập trình C

Trình tự các bước trong lập trình C



1.2. Giới thiệu về ngôn ngữ lập trình C

- ❖ **IDE** (integrated development environment - môi trường phát triển tích hợp) là phần mềm cho phép người lập trình có thể soạn thảo, biên dịch, liên kết, thực thi và gỡ rối (debug) trong cùng một môi trường.
- ❖ Là một phần mềm bao gồm những gói phần mềm khác nhau giúp phát triển ứng dụng phần mềm
- Dev-C++
 - C-Free
 - **Visual Studio Code**
 - Xcode
 - Netbean
 - Eclipse
 - CodeLite

1.2. Giới thiệu về ngôn ngữ lập trình C

❖ Visual Studio Code

- ❑ VSC là công cụ phát triển tích hợp mã nguồn mở của Microsoft
- ❑ Hỗ trợ rất nhiều các Extension giúp phát triển nhiều loại ngôn ngữ khác nhau trong đó có ngôn ngữ C
- ❑ Extensuon C/C++ hỗ trợ phát triển chương trình C rất mạnh cũng do Microsoft phát triển (*hỗ trợ cho chuẩn mới nhất là C18*)

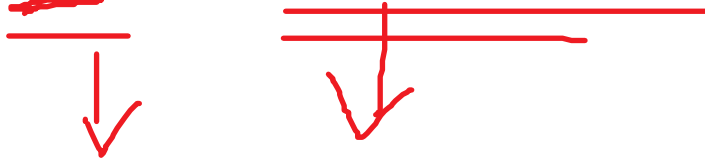
PHỤ LỤC 1B →

1.2. Giới thiệu về ngôn ngữ lập trình C

- ❖ Kỹ sư ngành cơ điện tử học ngôn ngữ lập trình C để làm gì?
 - Để có kỹ năng lập trình, kỹ năng phân tích trình bày giải thuật giải quyết vấn đề.
 - Để có thể lập trình cho các bộ vi xử lý, vi điều khiển, DSP, ...
 - 8051 (Keil C)
 - PIC (C18, C30, Hi-Tech C, CCS)
 - AVR (Code Vision)
 - TMS320 (Code Composer Studio)
 - Để có thể lập trình cho các hệ thống cơ điện tử, những hệ thống đo lường và điều khiển có dùng máy tính, ...
 - Để có thể dễ dàng tiếp cận với các ngôn ngữ lập trình hướng đối tượng như C++, Java, Visual C++, Visual C#, ...

1.3 Trình bày giải thuật

- ❖ 2 công cụ thường dùng để trình bày giải thuật là lưu đồ và **pseudocode** (ngôn ngữ giả)




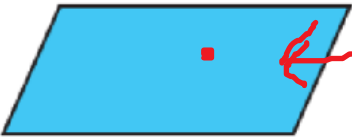
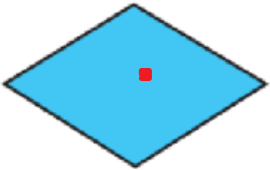


1.3.1. Lưu đồ (Flowchart)

- ❖ Lưu đồ cho phép trình bày giải thuật một cách trực quan. Nhờ lưu đồ giải thuật giải quyết vấn đề, lập trình viên có thể dễ dàng chuyển thành chương trình.

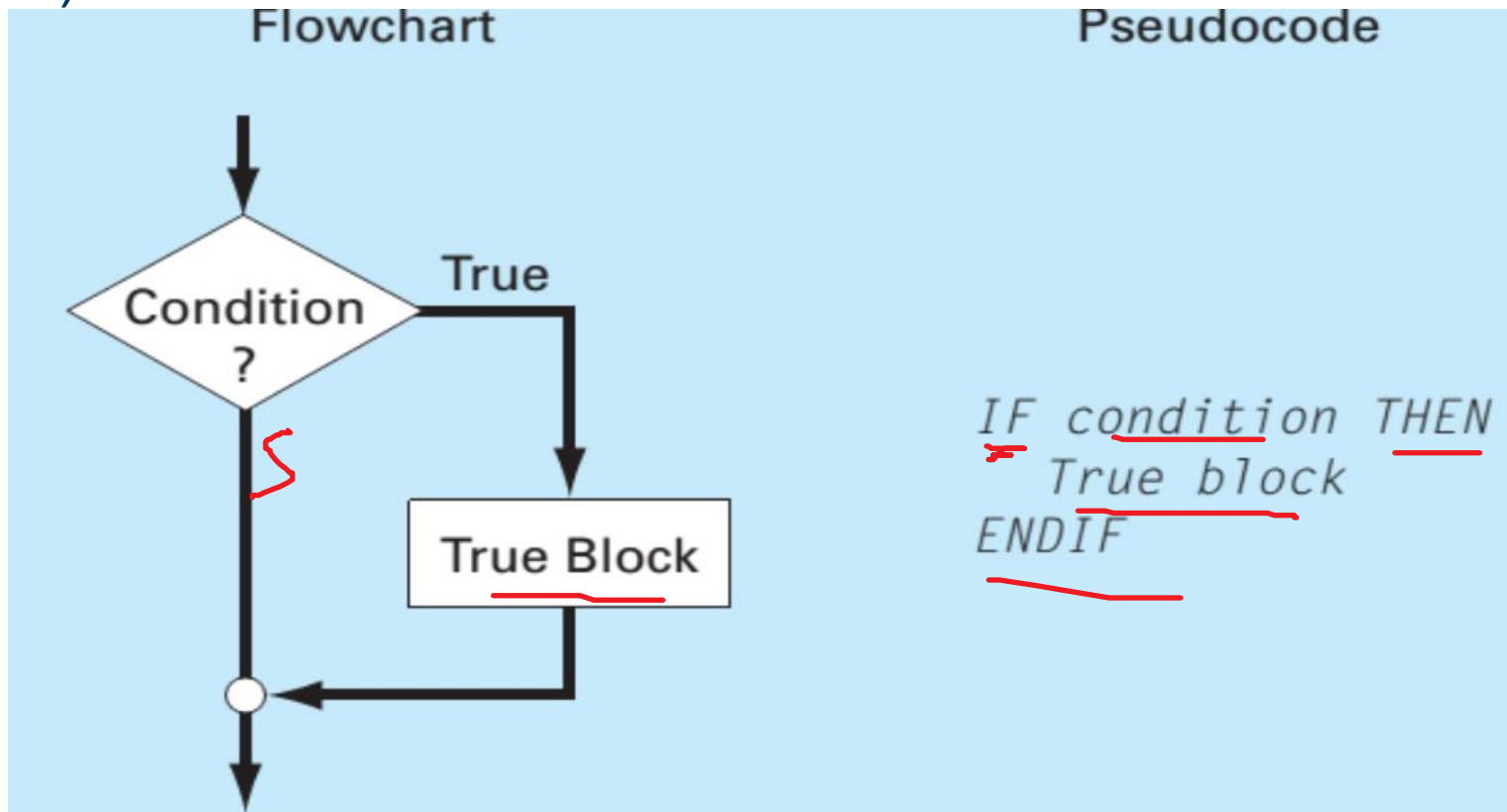
1.3.1. Lưu đồ (Flowchart)

❖ Một số quy ước:

Ký hiệu	Ý nghĩa
	Khối bắt đầu/kết thúc
	Chỉ chiều thực hiện
	Khối xử lý
	Khối vào/ra
	Khối quyết định

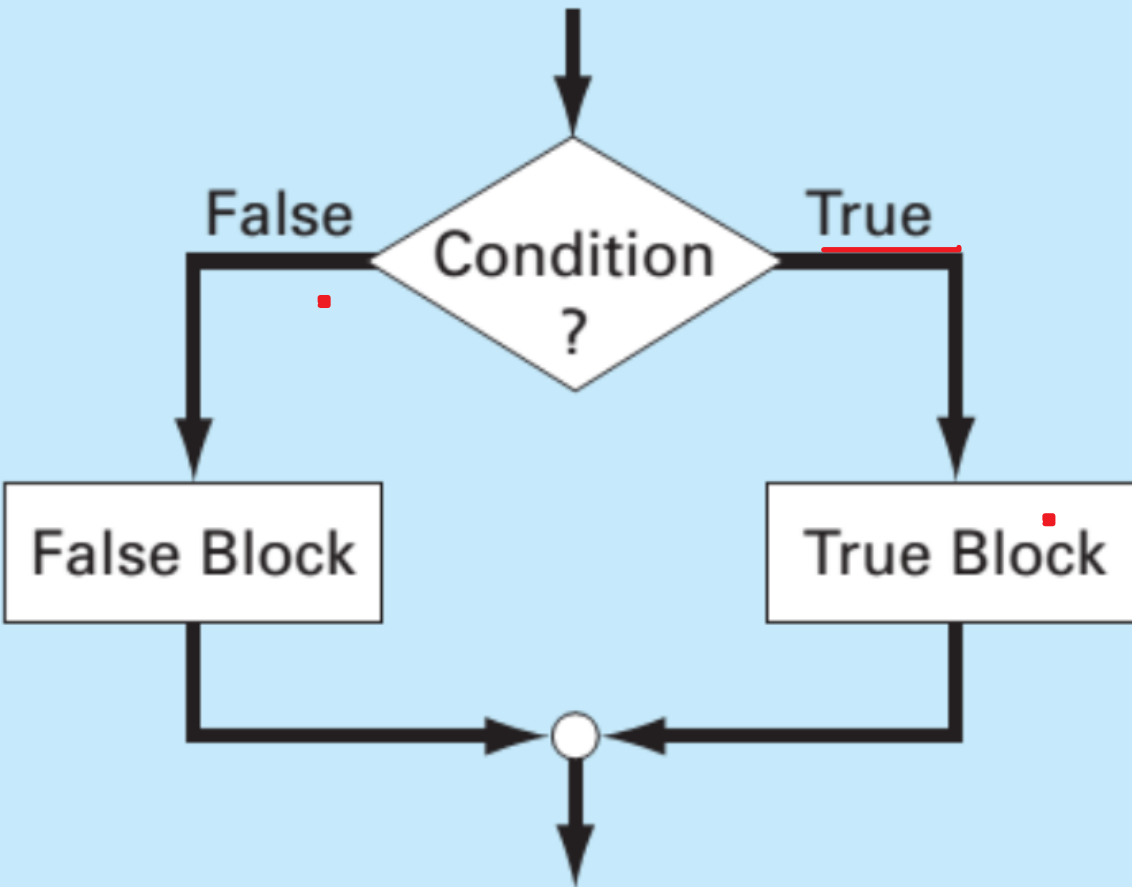
1.3.2. Ngôn ngữ giả (Pseudocode)

- ❖ Pseudocode là những mã tương tự như mã chương trình, dùng để mô tả giải thuật.
- ❖ Một số ví dụ **lưu đồ** và **pseudocode**:
 - a) **Cấu trúc IF/THEN**



1.3.2. Ngôn ngữ giả (Pseudocode)

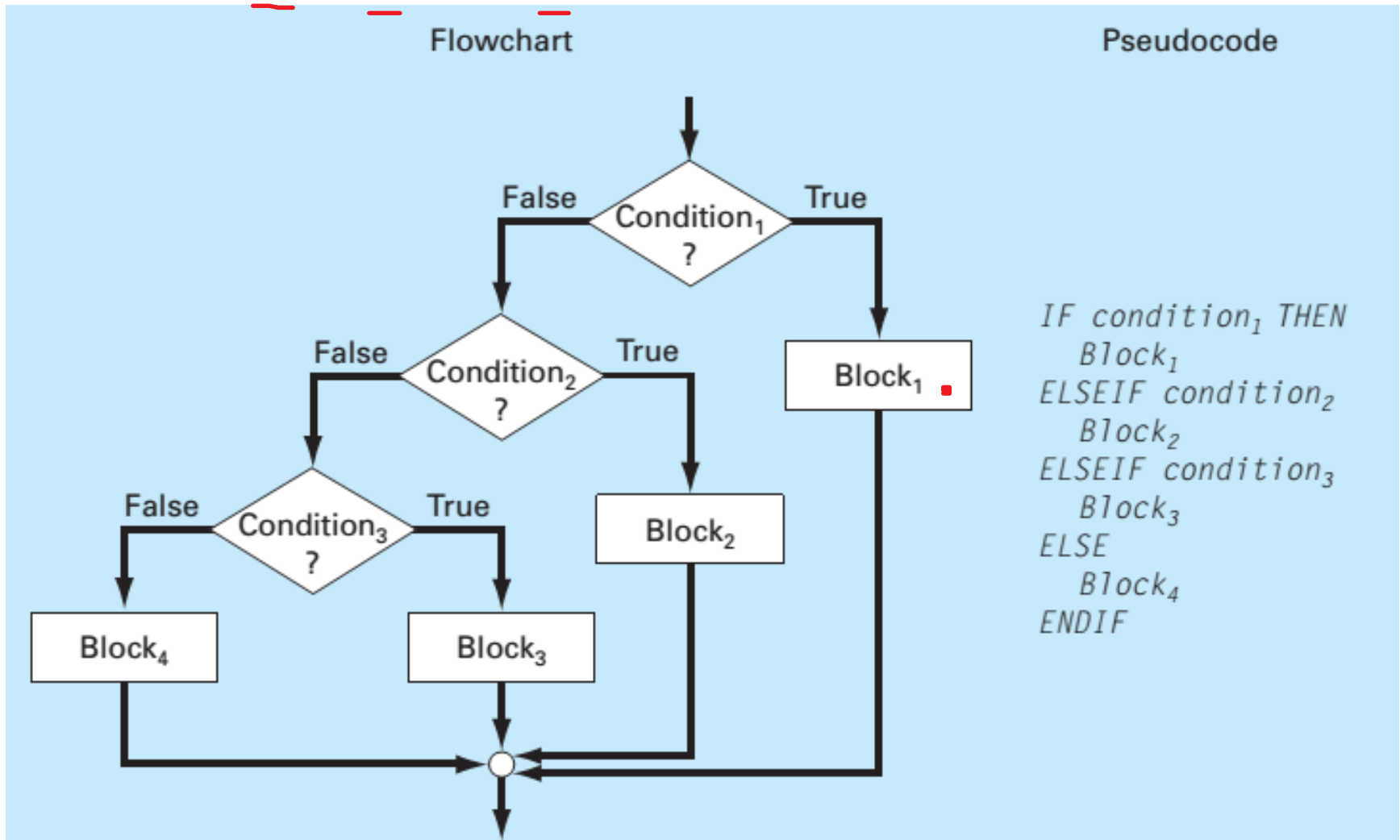
b) Cấu trúc IF/THEN/ELSE



```
IF condition THEN  
    True block  
ELSE  
    False block  
ENDIF
```

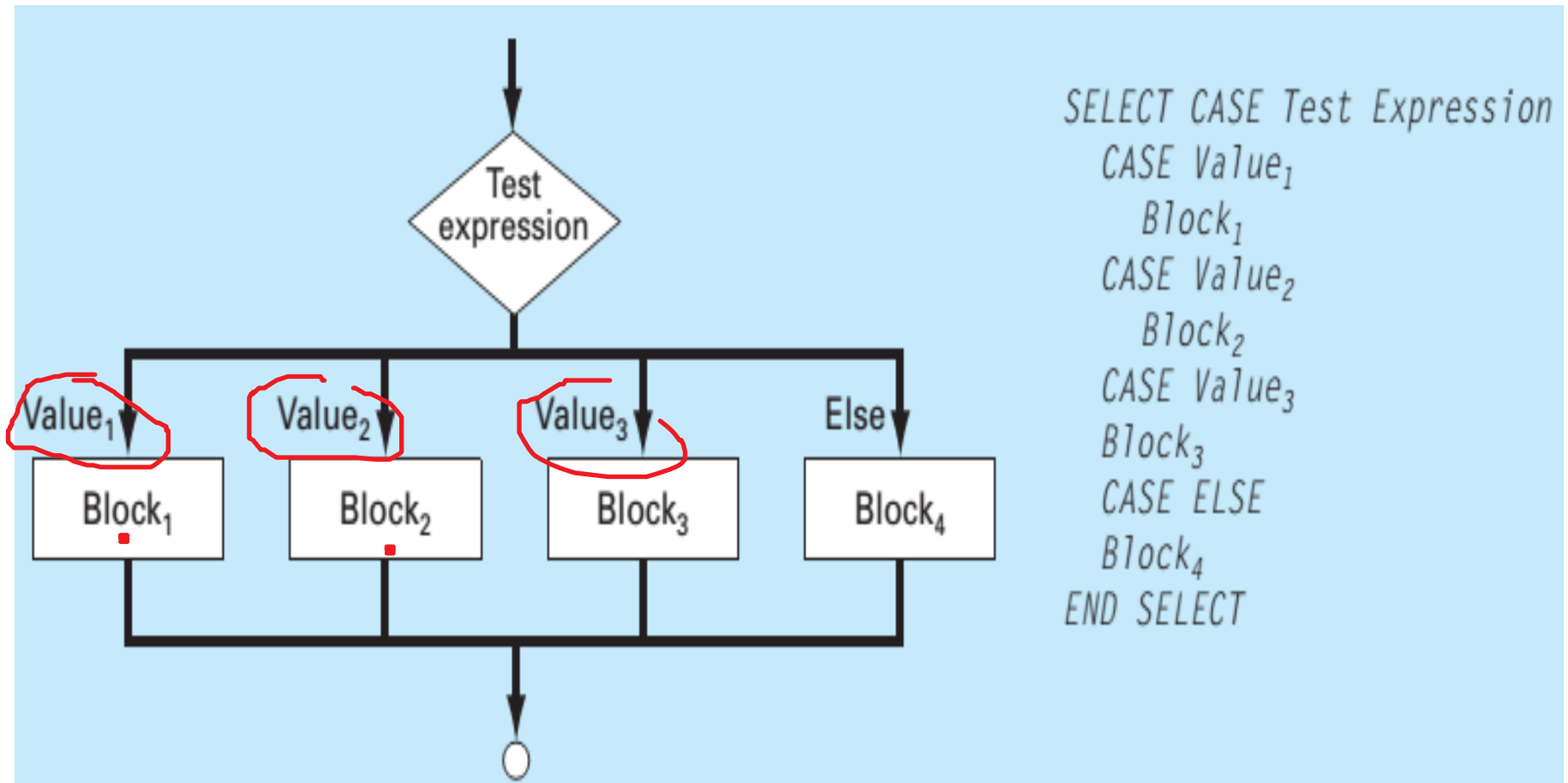
1.3.2. Ngôn ngữ giả (Pseudocode)

c) Cấu trúc IF/THEN/ELSE lồng nhau



1.3.2. Ngôn ngữ giả (Pseudocode)

d) Cấu trúc CASE



1.4. Một số ví dụ

VD1: Chương trình in ra màn hình câu "Hello, world!".

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

- Hàng 2-6: hàm `main()`, trả về số kiểu `int` (integer – số nguyên). Trị trả về là 0 nếu chương trình kết thúc thành công. Chương trình C bao giờ cũng có một hàm **`main()`**. Hàm này được thực thi đầu tiên khi chạy chương trình. Phần bên trong dấu `{...}` sau tên hàm là thân hàm.
- Hàng 4: Hàm **`printf()`** in ra màn hình chuỗi ký tự nằm giữa 2 dấu ngoặc kép. **`printf()`** là hàm có sẵn trong C, thuộc thư viện vào ra chuẩn (**`stdio`**). Để liên kết với thư viện này cần phải khai báo **`#include <stdio.h>`** ở đầu chương trình. **`'\n'`** là ký tự xuống hàng.
- Hàng 5: Trả về giá trị 0 và kết thúc hàm. (Trị trả về của hàm `main()` luôn là 0.)
- Dấu chấm phẩy dùng để kết thúc một lệnh.

1.4. Một số ví dụ

- ❖ VD: Chương trình in ra màn hình bảng lũy thừa 2 của các số nguyên từ 1 đến 10.

1! 2! 3

```
1  #include <stdio.h>
2  int main()
3  {
4      int n, n2;
5      n = 1;
6      while (n<=10)
7      {
8          n2 = n * n;
9          printf("%2d %5d\n", n, n2);
10         n = n + 1;
11     }
12     return 0;
13 }
```

- Hàng 4: khai báo biến n và $n2$ thuộc kiểu `int` (kiểu nguyên). Mọi biến cần phải khai báo kiểu trước khi sử dụng.
- Hàng 9: `%2d` và `%5d` là định dạng chuỗi xuất tương ứng cho biến n và biến $n2$ là những số nguyên thập phân độ dài tương ứng là 2 và 5.
- Hàng 6-11: vòng lặp `while`, được hiểu là "trong khi $n \leq 10$, thực hiện khối lệnh bên trong dấu `{...}`".

1.4. Một số ví dụ

- ❖ Bảng dưới đây liệt kê một số định dạng thường dùng trong printf() và scanf()

Định dạng	Ý nghĩa
<u>%c</u>	Ký tự kiểu char ✓
<u>%d</u>	Số nguyên hệ thập phân ✓
<u>%u</u> ✓	Số nguyên không dấu ✓
<u>%x</u> ✓	Số nguyên hệ thập lục phân (hexa) ✓
<u>%o</u> ✓	Số nguyên hệ bát phân (octal) ✓
<u>%f</u> ✓	Số dấu chấm động (float) ✓
<u>%lf</u> ✓	Số dấu chấm động (double) (xem ghi chú bên dưới)
<u>%e</u> ✓	Số dấu chấm động dạng mũ ✓
<u>%s</u> ✓	Chuỗi ký tự ▪

- ❖ Ghi chú:

- Đối với printf(), %f dùng cho cả kiểu float và double.
- Đối với scanf(), %f dùng cho kiểu float; %lf cho kiểu double. ✓

1.4. Một số ví dụ

VD: Chương trình nhập 2 số kiểu float, in ra màn hình tổng của 2 số đó.

```
#include <stdio.h>
int main(void)
{
    float a, b;
    printf("a = ");
    scanf("%f", &a);
    printf("b = ");
    scanf("%f", &b);
    printf("a + b = %6.2f", a+b);
    return 0;
}
```

- Trong chương trình trên, scanf("%f", &a) là hàm nhập dữ liệu từ bàn phím vào cho biến a. Với định dạng %f, những ký tự nhập vào từ bàn phím sẽ được chuyển đổi thành số dấu chấm động (kiểu float).
- Trong lệnh scanf(), luôn có ký tự '&' trước biến được gán giá trị. Điều này sẽ được lí giải trong chương 7.

- Trong hàm printf() cuối cùng, với định dạng "%6.2f" kết quả của a+b sẽ được in ra màn hình với độ dài là 6 (kể cả dấu chấm), trong đó có 2 số sau dấu chấm thập phân.
- Nếu khai báo biến a hay b là kiểu double thì chuỗi định dạng trong hàm scanf() cho biến tương ứng phải đổi thành "lf"; nếu không, kết quả sẽ sai. (Một số trình dịch không báo lỗi!)

