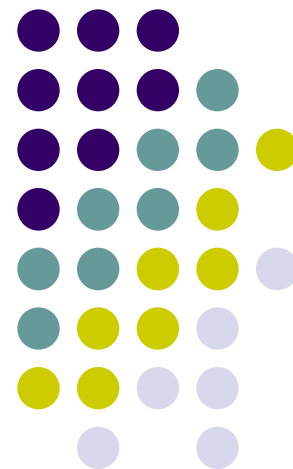


LẬP TRÌNH CĂN BẢN

Phần 1 GIỚI THIỆU VỀ CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

N.C. Danh

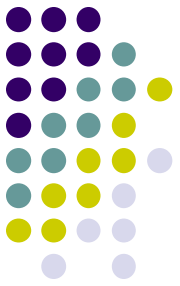




Nội dung chương

- Từ bài toán đến chương trình
- Giải thuật
 - *Khái niệm giải thuật*
 - *Các đặc trưng của giải thuật*
 - *Ngôn ngữ biểu diễn giải thuật*
 - *Một số giải thuật cơ bản*
 - *Các cấu trúc suy luận cơ bản của giải thuật*
 - *Từ giải thuật đến chương trình*
- Kiểu dữ liệu
- Khái niệm về ngôn ngữ lập trình
- Chương trình dịch

Từ Bài Toán Đến Chương Trình



- **Các bước giải bài toán bằng máy tính**
 - Mô tả các bước giải bài toán
 - Vẽ sơ đồ xử lý
 - Viết chương trình xử lý bằng ngôn ngữ giả
 - Chọn ngôn ngữ lập trình và chuyển chương trình từ ngôn ngữ giả sang ngôn ngữ lập trình
 - Thực hiện chương trình: nhập vào các tham số, nhận kết quả



Giải Thuật

- Khái niệm giải thuật
- Các đặc trưng của giải thuật
- Ngôn ngữ biểu diễn giải thuật
- Một số giải thuật cơ bản
- Các cấu trúc suy luận cơ bản của giải thuật
- Từ giải thuật đến chương trình



Khái Niệm Giải Thuật

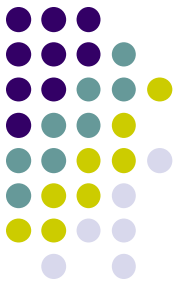
- **Ví dụ:** Hoán đổi chất lỏng trong 2 bình A (nước mắm) và B (rượu):
 - Yêu cầu phải có thêm một bình thứ ba gọi là bình C.
 - **Bước 1:** Đổ rượu từ bình B sang bình C.
 - **Bước 2:** Đổ nước mắm từ bình A sang bình B.
 - **Bước 3:** Đổ rượu từ bình C sang bình A.
- *“Giải thuật là một dãy các thao tác trên những dữ liệu vào sao cho sau một hữu hạn bước ta thu được kết quả của bài toán”.*



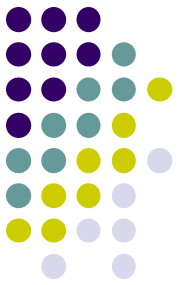
Các Đặc Trưng Của Giải Thuật

- Tính kết thúc
 - *Số bước là hữu hạn*
- Tính xác định
 - *Máy phải thực hiện được*
 - *Cho cùng kết quả trên các máy khác nhau*
- Tính phổ dụng
- Tính hiệu quả
 - *Thời gian*
 - *Tài nguyên máy*

Ngôn Ngữ Biểu Diễn Giải Thuật

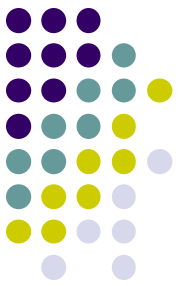


- Ngôn ngữ tự nhiên
- Ngôn ngữ sơ đồ
- Ngôn ngữ giả








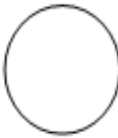


Ngôn Ngữ Tự Nhiên

- Là ngôn ngữ của chúng ta
- Ví dụ: Giải thuật giải phương trình bậc nhất $ax+b=0$.
 - Bước 1:** Nhận giá trị của các tham số a, b .
 - Bước 2:** Xét giá trị của a xem có bằng 0 hay không?
Nếu $a=0$ thì làm bước 3, nếu a khác không thì làm bước 4.
 - Bước 3:** (a bằng 0) Nếu b bằng 0 \Rightarrow pt vô số nghiệm.
Nếu b khác 0 \Rightarrow pt vô nghiệm.
 - Bước 4:** (a khác 0) Ta kết luận phương trình có nghiệm $x=-b/a$.

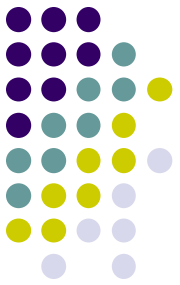


Ngôn Ngữ Sơ Đồ (1)

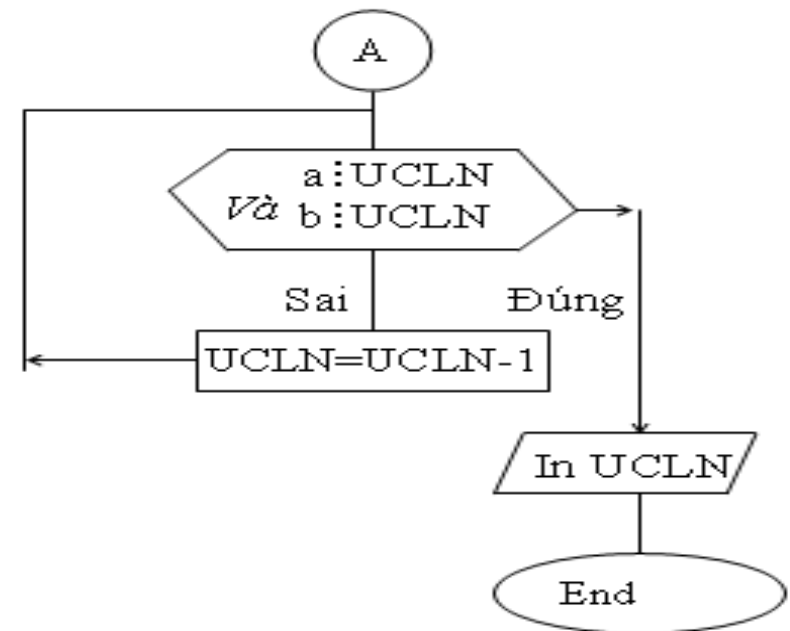
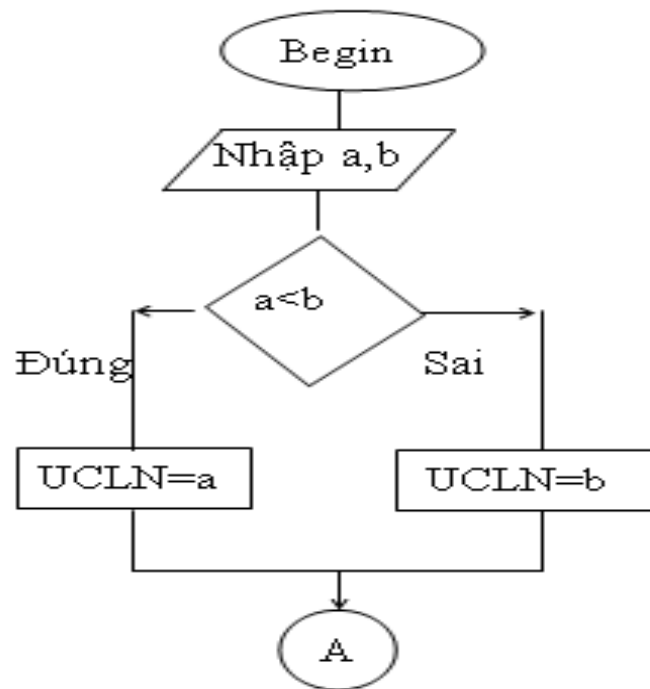
- Mô tả giải thuật bằng bằng các sơ đồ hình khối đã được (quy ước trước)

Khối	Tác dụng (Ý nghĩa của hành động)	Khối	Tác dụng (Ý nghĩa của hành động)
	Bắt đầu/ Kết thúc		Đường đi
	Nhập / Xuất		Chương trình con
	Thi hành		Khối nối
	Lựa chọn		Lời chú thích

Ngôn Ngữ Sơ Đồ (2)



- **Ví dụ:** Dùng lưu đồ để biểu diễn giải thuật tìm UCLN nêu trên như sau:

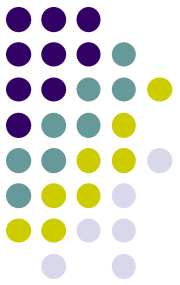




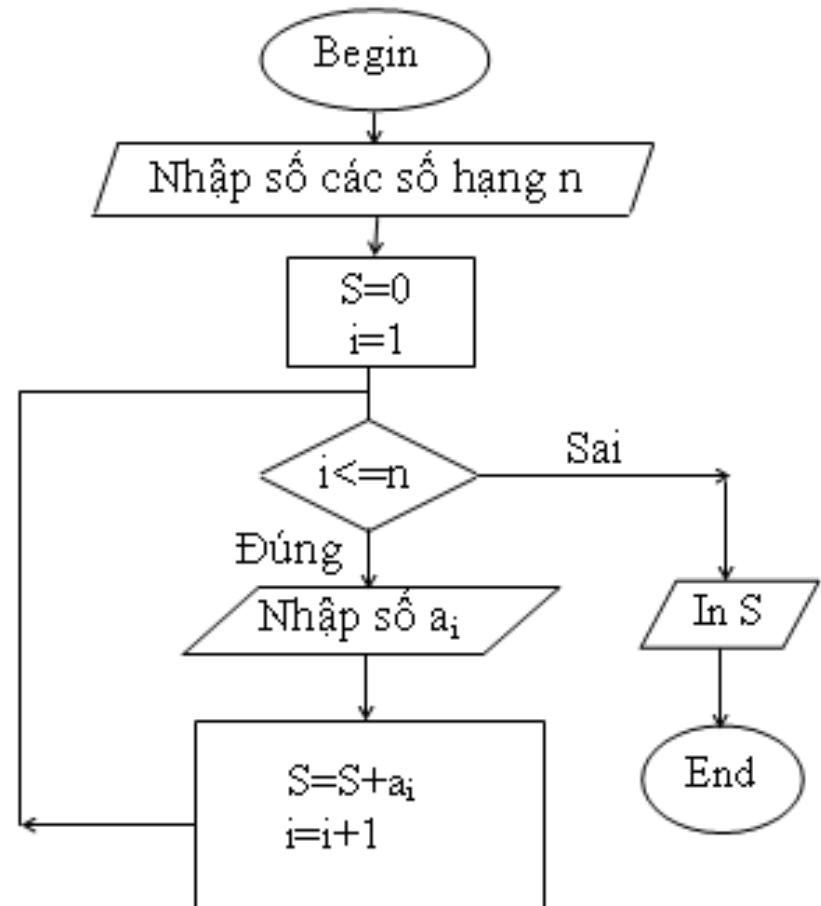
Ngôn Ngữ Giả

- Là một sự kết hợp giữa ngôn ngữ tự nhiên với các cấu trúc câu lệnh của một ngôn ngữ lập trình.
- **Ví dụ:** Giải thuật giải phương trình bậc nhất $ax+b=0$.
 - Nhập vào a, b
 - *If $a==0$ then*
 - If $b==0$ then*
Kết luận phương trình vô số nghiệm
 - else*
Kết luận phương trình vô nghiệm
 - else*
Kết luận phương trình có nghiệm $x=-b/a$

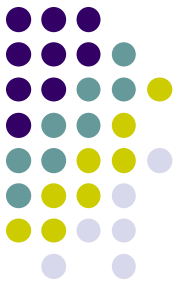
Một Số Giải Thuật Cơ Bản (1)



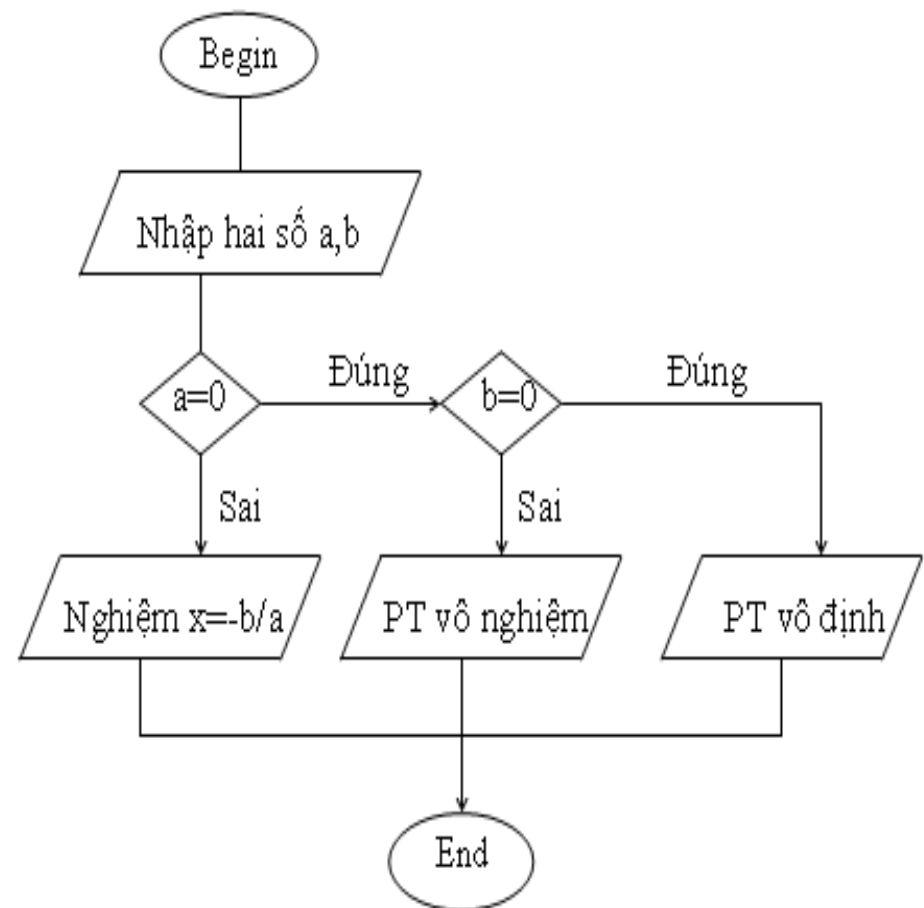
- **Ví dụ 1:** Yêu cầu:
 - Nhập vào 1 dãy n số hạng a_1, a_2, \dots, a_n
 - Tính tổng S:
$$S = a_1 + a_2 + a_3 + \dots + a_n$$
 - In S ra màn hình



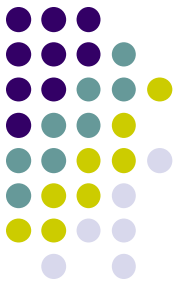
Một Số Giải Thuật Cơ Bản (2)



- **Ví dụ 2:** Yêu cầu:
 - Nhập vào 2 số a và b là 2 hệ số của pt: $ax+b=0$
 - Cho biết nghiệm của phương trình.

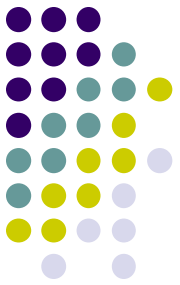


Các Cấu Trúc Suy Luận Cơ Bản Của Giải Thuật (1)



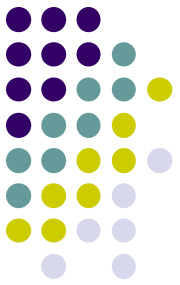
- Giải thuật được thiết kế theo 3 cấu trúc suy luận cơ bản:
 - **Tuần tự (Sequential):**
 - Các công việc được thực hiện tuần tự, công việc này nối tiếp công việc kia.
 - **Cấu trúc lựa chọn (Selection)**
 - Lựa chọn một công việc để thực hiện căn cứ vào một điều kiện nào đó
 - **Cấu trúc 1:** Nếu < điều kiện > (đúng) thì thực hiện < công việc >
 - **Cấu trúc 2:** Nếu < điều kiện > (đúng) thì thực hiện < công việc 1 >, ngược lại (điều kiện sai) thì thực hiện < công việc 2 >
 - **Cấu trúc 3:** Trường hợp < i > thực hiện < công việc i >

Các Cấu Trúc Suy Luận Cơ Bản Của Giải Thuật (2)



- ***Cấu trúc lặp (Repeating)***
 - Lặp lại thực hiện một công việc **không** hoặc **nhiều lần** căn cứ vào một điều kiện nào đó.
 - Có 2 dạng như sau:
 - Lặp với **số lần xác định**
 - Lặp với **số lần không xác định**

Từ Giải Thuật Đến Chương Trình



- Cả 2 đều là **tập các chỉ thị (instruction)** – làm thế nào để giải quyết 1 công việc (task).
- **Giải thuật**
 - *Nói chuyện với con người, dễ hiểu.*
 - *Dùng ngôn ngữ đơn giản (English) – không viết bằng mã.*
- **Chương trình**
 - *Nói chuyện với máy tính.*
 - *Có thể được xem như 1 diễn tả hình thức (formal expression) của 1 giải thuật.*



Kiểu Dữ Liệu

- Ví dụ:
 `int x,y;`
 `float r=3.25;`
- “Kiểu dữ liệu là một *tập hợp các giá trị có cùng một tính chất* và *tập hợp các phép toán thao tác trên các giá trị đó*”.
- Có 2 loại
 - Kiểu dữ liệu *sơ cấp*
 - Kiểu dữ liệu *có cấu trúc*



Kiểu Dữ Liệu Sơ Cấp

- “*Kiểu dữ liệu sơ cấp là kiểu dữ liệu mà giá trị của nó là đơn nhất*”.
- **Ví dụ:** Kiểu `int` trong C
 - là kiểu *sơ cấp*
 - gồm các số nguyên từ `-32768..32767`
 - và các phép toán: `+, -, *, /, %...`



Kiểu Dữ Liệu Có Cấu Trúc

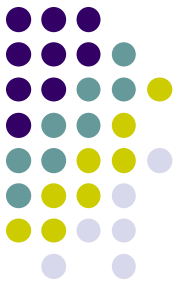
- “*Kiểu dữ liệu có cấu trúc là kiểu dữ liệu mà các giá trị của nó là sự kết hợp của các giá trị khác*”.
- **Ví dụ :** Kiểu chuỗi ký tự trong C.
 - là kiểu có cấu trúc.
 - Ví dụ: `char *chuoil = “Chao cac ban!”;`



Ngôn Ngữ Lập Trình

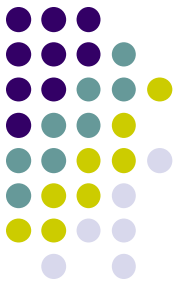
- Khái niệm về ngôn ngữ lập trình
- Chương trình dịch

Khái Niệm Về Ngôn Ngữ Lập Trình

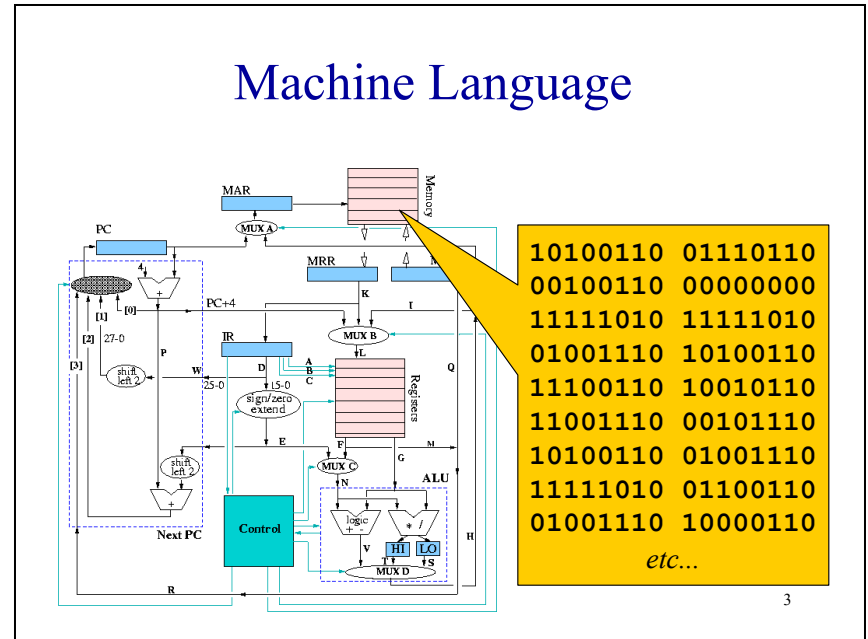


- Ngôn ngữ lập trình là một ngôn ngữ dùng để viết chương trình cho máy tính
- Ta có thể chia ngôn ngữ lập trình thành các loại sau:
 - *Ngôn ngữ máy*
 - *Hợp ngữ*
 - *Ngôn ngữ cấp cao*

Ngôn Ngữ Máy (machine language)



- Là các **chỉ thị** dưới dạng **nhị phân**, can thiệp trực tiếp vào trong các **mạch điện tử**.
- Có thể được **thực hiện ngay** không cần qua bước trung gian nào.
- Tuy nhiên chương trình viết bằng ngôn ngữ máy dễ sai sót, cồng kềnh và khó đọc, khó hiểu vì toàn những con số **0** và **1**.





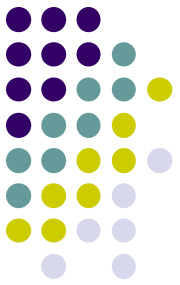
Hợp Ngữ (Assembly language)

- Bao gồm tên **các câu lệnh** và **quy tắc viết** các câu lệnh đó.
- Tên các câu lệnh bao gồm hai phần:
 - **Phần mã lệnh** (English) chỉ phép toán cần thực hiện
 - **Phần địa chỉ** chứa toán hạng của phép toán đó.
- Để máy thực hiện được một chương trình viết bằng **hợp ngữ** thì chương trình đó phải được dịch sang **ngôn ngữ máy**. Công cụ thực hiện việc dịch đó được gọi là **Assembler**.

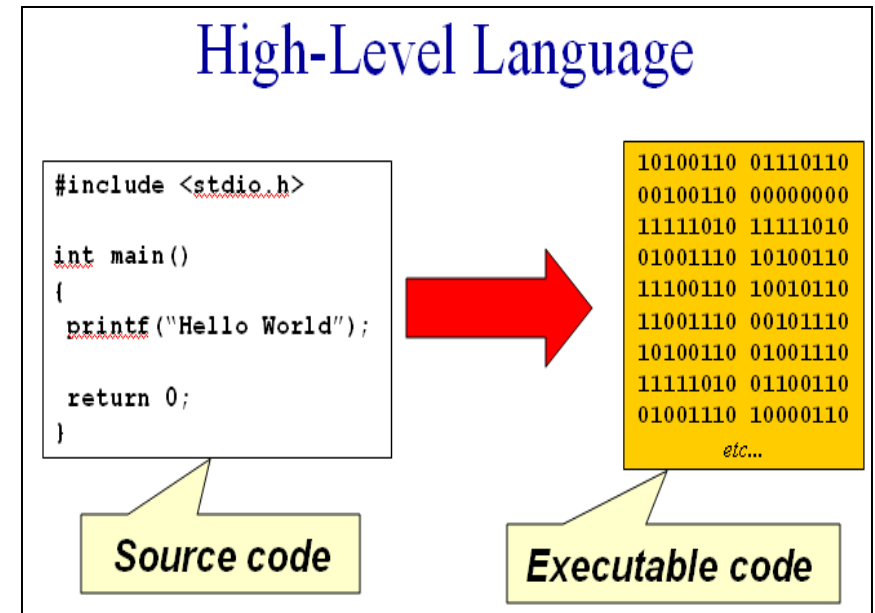
Assembly Language

INPUT a ; Nhập giá trị cho a từ bàn phím
LOAD a ; Đọc giá trị a vào thanh ghi tổng A
PRINT a ; Hiển thị giá trị của a ra màn hình.
INPUT b
ADD b ; Cộng giá trị của thanh ghi tổng A
; với giá trị b

Ngôn Cấp Cao (*High level language*)



- Rất gần với ngôn ngữ con người.
- Một chương trình viết bằng ngôn ngữ cấp cao được gọi là **chương trình nguồn** (source programs).
- Để máy tính "hiểu" và thực hiện được các lệnh trong chương trình nguồn thì phải có một **chương trình dịch** để dịch chương trình nguồn thành dạng chương trình có khả năng thực thi.





Chương Trình Dịch

- Được dùng để chuyển một **chương trình nguồn** sang **chương trình đích**.
- **Có 2 dạng:**
 - **Thông dịch (interpreter):**
 - Dịch từng lệnh một, dịch tới đâu thực hiện tới đó.
 - **Ví dụ:** ngôn ngữ LISP.
 - **Biên dịch (compiler):**
 - Dịch toàn bộ chương trình nguồn thành chương trình đích rồi sau đó mới thực hiện.
 - **Ví dụ:** Pascal, C...



Hết chương