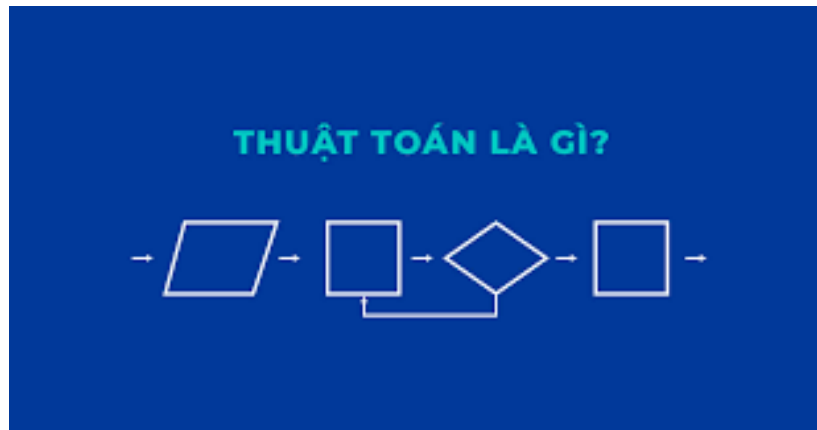
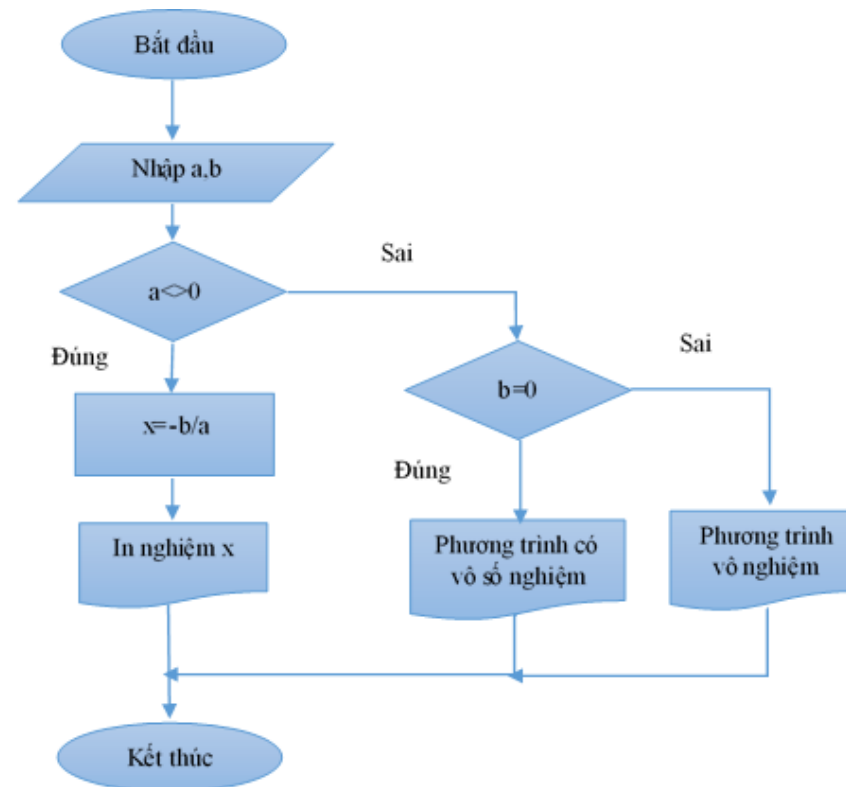


Chương 01: TỔNG QUAN



GVHD: ThS. Hồ Khôi
Email: khoihue@gmail.com

Thuật toán – Giải thuật



Giới thiệu

Thuật toán là một tập các quy tắc (hay quy trình cụ thể) để giải quyết một vấn đề nào đó với số lần thực hiện là hữu hạn, dựa trên các dữ liệu đầu vào được cung cấp cho quá trình xử lý của chương trình

VD: Thuật toán giải phương trình bậc nhất $aX + b = c$

(Với a, b, c là các số thực)

1 – Nếu $a = 0$

** $b = c$ thì phương trình có vô số nghiệm (Nghiệm bất kỳ)*

** $b \neq c$ thì phương trình vô nghiệm*

2 – Nếu $a \neq 0$

Phương trình có 1 nghiệm duy nhất $x = (c - b) / a$

Đặc tính của thuật toán

- ❖ Tính chính xác: Các bước thực thi phải được phát biểu chính xác, chặt chẽ.
- ❖ Tính duy nhất: Kết quả của mỗi bước được định nghĩa một cách duy nhất và chỉ phụ thuộc vào nhập liệu và kết quả của các bước trước đó.
- ❖ Tính hữu hạn: Thuật toán ngừng sau một khi một số hữu hạn các chỉ thị lệnh được thực hiện.
- ❖ Nhập liệu: – thuật toán nhận nhập liệu (có thể trống).

Đặc tính của thuật toán

- ❖ Nhập liệu: thuật toán nhận nhập liệu (có thể trống).
- ❖ Xuất liệu: thuật toán tạo ra xuất liệu.
- ❖ Tính tổng quát: thuật toán có thể được vận dụng cho một tập hợp nhập liệu.

Các phương pháp biểu diễn

❖ Biểu diễn bằng ngôn ngữ tự nhiên

(Native language)

❖ Biểu diễn bằng mã giả

(Pseudo code)

❖ Biểu diễn bằng lưu đồ

(Flow chart)

Minh họa bằng ngôn ngữ tự nhiên

- ☐ Dùng ngôn ngữ thường ngày để liệt kê các bước của thuật toán.
- ☐ Dài dòng, có thể nhập nhằng khó hiểu.
- ☐ Không thể hiện rõ cấu trúc của thuật toán.
- ☐ Không có một quy tắc cố định.
- ☐ Có thể viết thụt lùi vào bên phải để phân cấp cho dễ đọc.

Minh họa bằng ngôn ngữ tự nhiên

1 – Nhập các giá trị a, b, c

2 – Nếu $a = 0$

Nếu $b = c$ thì thông báo phương trình có vô số nghiệm

nếu $b \neq c$ thì thông báo phương trình vô nghiệm

3 – Nếu $a \neq 0$

Thông báo: Phương trình có 1 nghiệm duy nhất

$$x = (c - b) / a$$

Minh hoạ bằng mã giả

Khi thể hiện thuật toán bằng mã giả, người ta thường *vay mượn* các cú pháp của một ngôn ngữ lập trình nào đó để thể hiện thuật toán. Việc dùng mã giả vừa tận dụng được các khái niệm trong ngôn ngữ lập trình, vừa giúp người cài đặt dễ dàng nắm bắt nội dung thuật toán

Nhập a, b, c

If $a = 0$ then

if $b = c$ then

write('có vô số nghiệm');

if $b \neq c$ then

write('phương trình vô nghiệm');

Else

begin

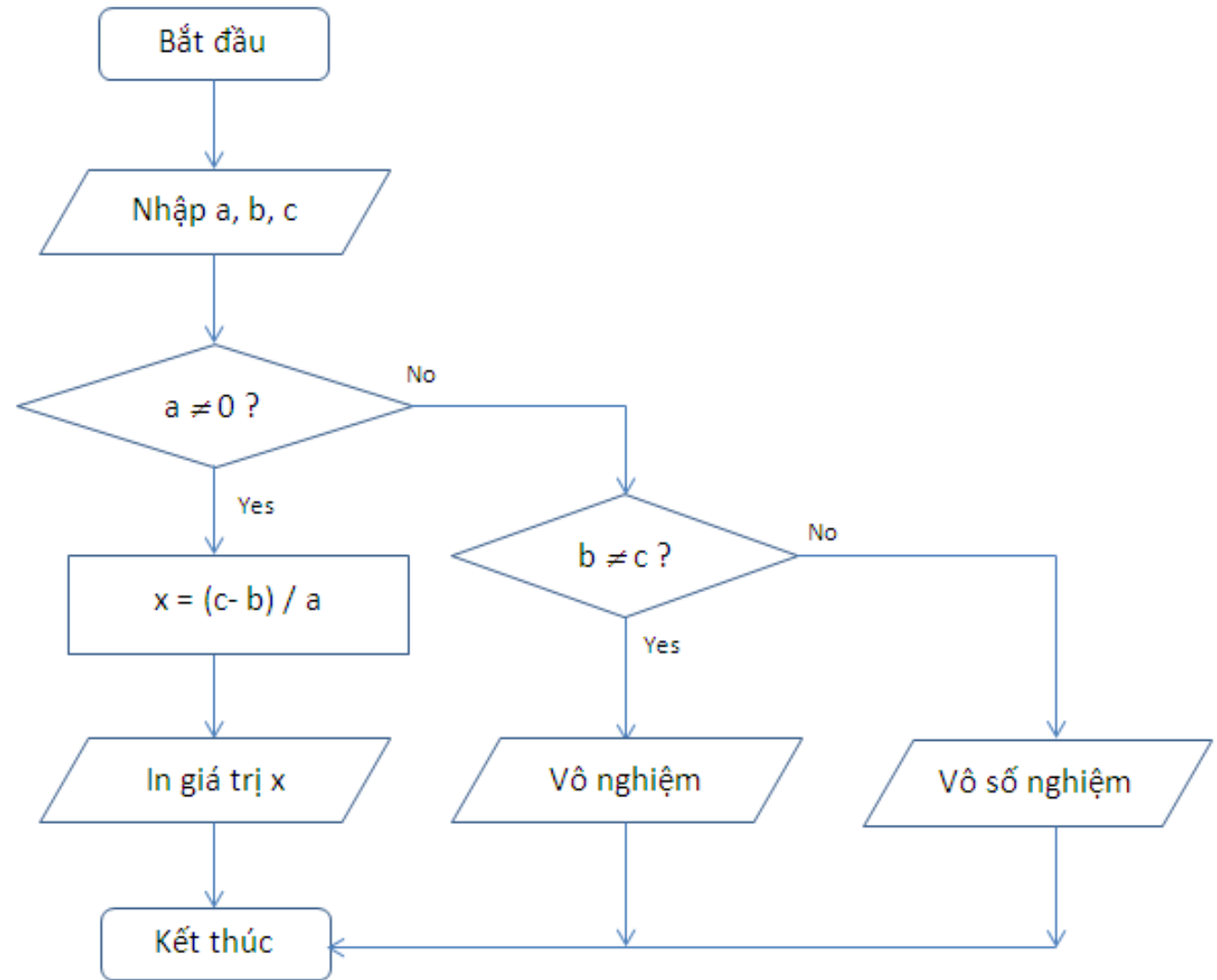
$x := (c - b) / a;$

write('Phương trình có 1 nghiệm duy nhất $x = , x$);

end

Minh hoạ bằng lưu đồ


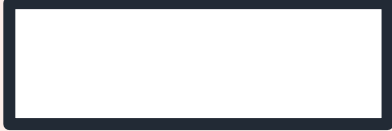
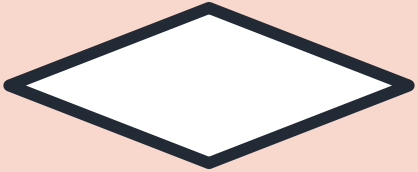
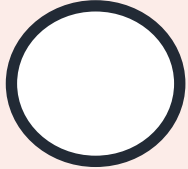
Lưu đồ hay sơ đồ khối là một công cụ trực quan để diễn đạt các thuật toán. Biểu diễn thuật toán bằng lưu đồ sẽ giúp người đọc theo dõi được sự phân cấp các trường hợp và quá trình xử lý của thuật toán



Biểu diễn thuật toán bằng lưu đồ

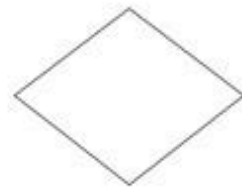
Lưu đồ thuật toán là công cụ dùng để biểu diễn thuật toán của chương trình, việc mô tả quá trình nhập (*input*), xuất dữ liệu (*output*) và các luồng xử lý (*processing*) đều được thể hiện dựa trên các ký hiệu hình học.

Ký hiệu sử dụng trên lưu đồ

Ký hiệu - Biểu tượng	Ý nghĩa sử dụng trong lưu đồ
	Biểu thị bắt đầu hoặc kết thúc chương trình
	Biểu thị hướng xử lý trong một quá trình / chương trình
	Biểu thị thông tin vào hoặc thông tin ra (<i>Nhập hoặc xuất dữ liệu</i>) trong chương trình
	Biểu thị một hoạt động (<i>Hay một quá trình</i>) trong thuật toán
	Biểu thị một quyết định khi đứng trước một vấn đề logic cần phải lựa chọn (<i>hoặc phân nhánh xử lý trong chương trình</i>).
	Điểm nối trên lưu đồ (<i>Sử dụng khi lưu đồ có kích thước lớn, phức tạp</i>)

Biểu diễn bằng lưu đồ (tt)

- Chọn lựa theo một điều kiện nào đó:
 - Biểu diễn bằng một hình thoi, bên trong chứa biểu thức điều kiện.
 - Ví dụ: thao tác "nếu $X > Y$ thì thực hiện thao tác in X, ngược lại thực hiện thao tác in Y" là thao tác chọn lựa

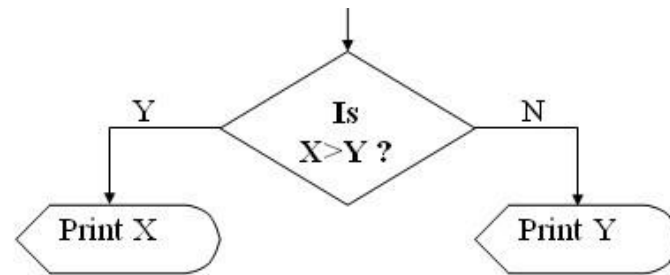


Decision

Decision point between two or more paths in your flowchart

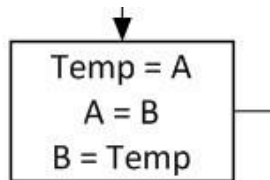
Biểu diễn bằng lưu đồ (tt)

- Thao tác chọn lựa: có thể có hai hướng đi
 - một hướng ứng với điều kiện thỏa
 - một hướng ứng với điều kiện không thỏa.
- 2 cung có nhãn
 - Đ/Đúng,Y/Yes
 - S/Sai,N/No



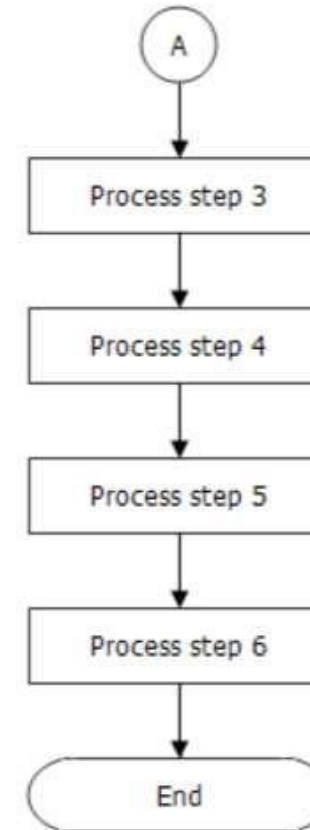
Biểu diễn bằng lưu đồ (tt)

- Xử lý, hành động:
 - Biểu diễn bằng một hình chữ nhật, bên trong chứa nội dung xử lý.



Biểu diễn bằng lưu đồ (tt)

- Quá trình thực hiện các thao tác:
 - Đường đi – route
 - Biểu diễn bằng cung có hướng
 - nối giữa 2 thao tác: thực hiện lần lượt

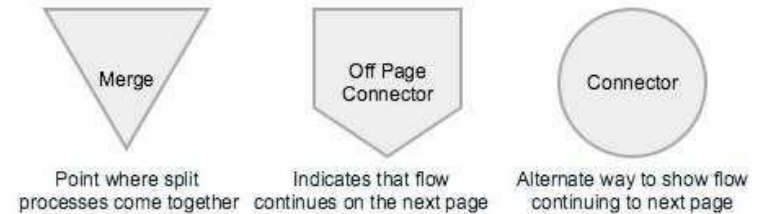


Biểu diễn bằng lưu đồ (tt)

- Điểm cuối (terminator)
 - Biểu diễn bằng hình ovan
 - Điểm khởi đầu
 - chỉ có cung đi ra
 - bên trong ovan ghi chữ: bắt đầu/start/begin
 - Điểm kết thúc
 - Chỉ có cung đi vào
 - bên trong ovan ghi chữ: kết thúc/end
- Mỗi lưu đồ chỉ có 1 điểm bắt đầu và 1 điểm kết thúc.

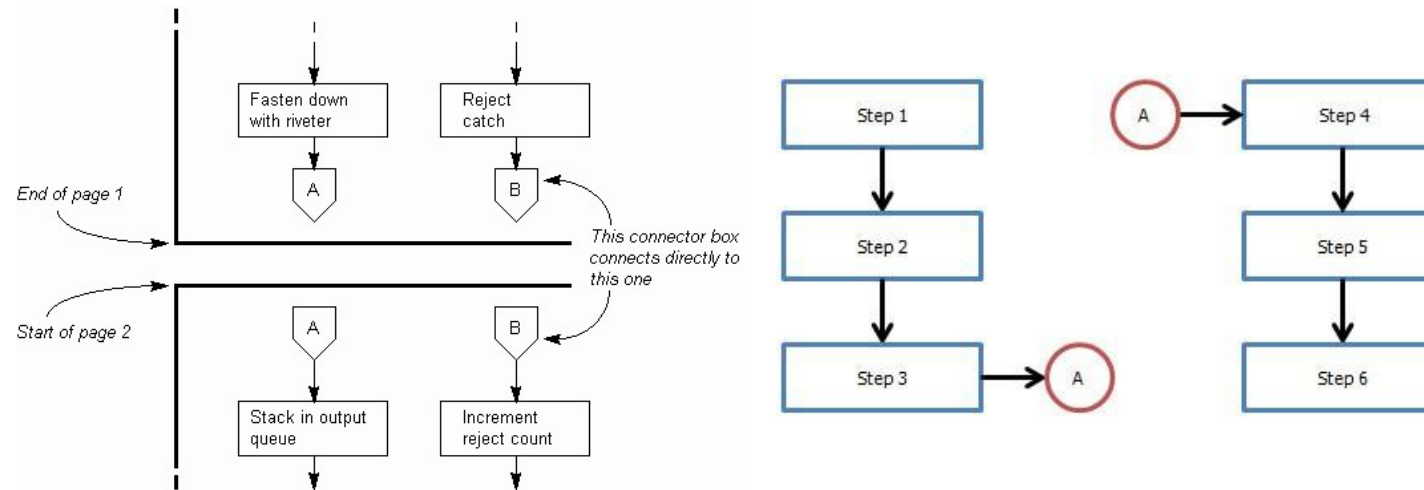
Biểu diễn bằng lưu đồ (tt)

- Điểm nối (connector)
 - Nối các phần khác nhau của một lưu đồ
 - Nối sang trang
 - Sử dụng với lưu đồ phức tạp
 - Giảm độ rắc rối
 - Đặt ký hiệu liên hệ giữa các điểm nối

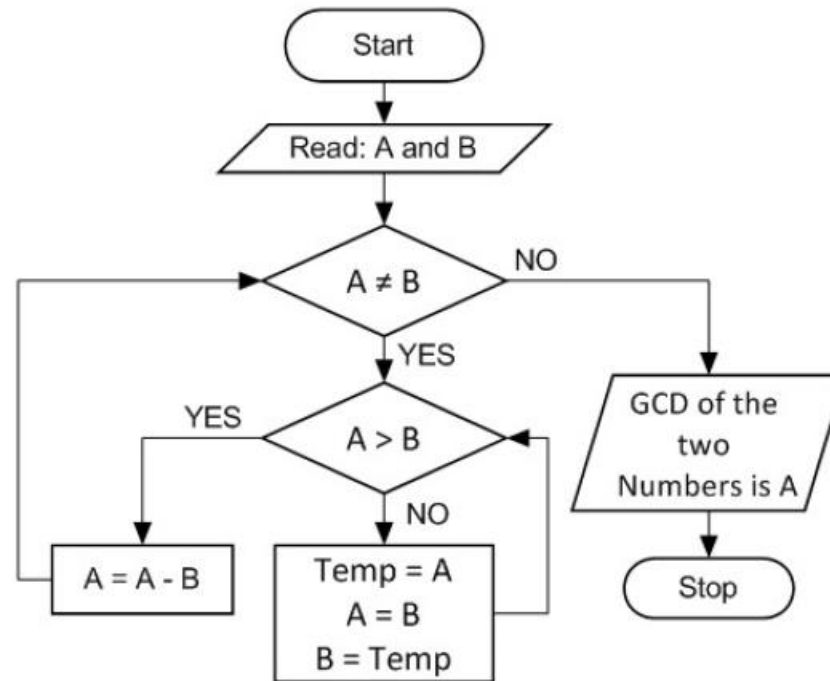


Biểu diễn bằng lưu đồ (tt)

- Điểm nối (connector)



Ví dụ: Thuật toán tính USCLN của 2 số



Độ phức tạp thuật toán

- Độ phức tạp của thuật toán là hàm đánh giá tính hiệu quả của một thuật toán.
- Để đánh giá hiệu quả của một thuật toán, có thể xét số các phép tính phải thực hiện khi thực hiện thuật toán này.
- Thông thường số các phép tính được thực hiện phụ thuộc vào cỡ của bài toán, tức là độ lớn của đầu vào.

Độ phức tạp thuật toán

- Độ phức tạp thuật toán là một hàm phụ thuộc đầu vào. Tuy nhiên trong những ứng dụng thực tiễn, chúng ta không cần biết chính xác hàm này mà chỉ cần biết một ước lượng đủ tốt của chúng.
- Để ước lượng độ phức tạp của một thuật toán ta thường dùng khái niệm bậc O-lớn và bậc Θ (bậc [Theta](#)).

Độ phức tạp thuật toán

- Gọi n là kích thước bài toán, $R(n)$ là tổng tài nguyên cần dung, giả sử tìm được hàm $g(n)$ và có hằng số C sao cho:

$$R(n) \leq C.g(n)$$

Thì ta nói thuật toán có độ phức tạp $O(g(n))$.

Độ phức tạp thuật toán

- $O(1)$: Độ phức tạp hằng số.
- $O(n)$: Độ phức tạp tuyến tính
- $O(P(n))$: Độ phức tạp đa thức.
- $O(\log n)$: Độ phức tạp logarit.
- $O(2^n)$: Độ phức tạp hàm mũ.

Độ phức tạp thuật toán

- Gọi n là kích thước bài toán, $R(n)$ là tổng tài nguyên cần dung, giả sử tìm được hàm $h(n)$ và các hằng số C_1, C_2 sao cho:

$$C_1 \cdot h(n) \leq R(n) \leq C_2 \cdot h(n)$$

Thì ta nói thuật toán có độ phức tạp đúng bằng $\Theta(h(n))$.

Thuật giải

- Có nhiều bài toán cho đến nay vẫn chưa tìm ra một cách giải theo kiểu thuật toán và cũng không biết là có tồn tại thuật toán hay không.
- Có nhiều bài toán đã có thuật toán để giải nhưng không chấp nhận được vì thời gian giải theo thuật toán đó quá lớn hoặc các điều kiện cho thuật toán khó đáp ứng.

Thuật giải

- Có những bài toán được giải theo những cách giải vi phạm thuật toán nhưng vẫn chấp nhận được.
- Tính đúng của thuật toán bây giờ không còn bắt buộc đối với một số cách giải bài toán, nhất là các cách giải gần đúng.

Thuật giải

- Trong thực tiễn có nhiều trường hợp người ta chấp nhận các cách giải thường cho kết quả tốt nhưng ít phức tạp và hiệu quả.
- Khái niệm thuật toán được mở rộng như trên gọi là **thuật giải**.
- Thuật giải thường được dùng nhiều trong trí tuệ nhân tạo.

Tóm tắt

- Một thuật toán là một dãy hữu hạn các thao tác cần thực thi để đi đến lời giải cho một bài toán hay một vấn đề nào đó.
- Thuật toán phải có tính chính xác, tính duy nhất, tính hữu hạn, tính tổng quát.
- Ta có thể biểu diễn thuật toán bằng ngôn ngữ tự nhiên, bằng mã giả hoặc bằng lưu đồ.

Tóm tắt

- Độ phức tạp của thuật toán là một hàm theo kích thước của bài toán, nó cho biết tính hiệu quả khi thực hiện bài toán.
- Độ phức tạp của thuật toán được ký hiệu là $O(h(n))$ hoặc $\Omega(h(n))$.
- Thuật giải là khái niệm thuật toán được mở rộng để có thể đi đến lời giải chấp nhận được nhưng có độ phức tạp tốt hơn (nhiều) so với thuật toán (nếu tồn tại).

Một số ví dụ

- Biểu diễn thuật toán tính ước số chung lớn nhất của số nguyên dương.
- Biểu diễn thuật toán giải phương trình bậc 2.
- Biểu diễn thuật toán tính giá trị lớn nhất của một dãy các số thực.
- Biểu diễn thuật toán tính căn bậc hai của một số thực A với sai số không quá epsilon.

(Xem như bài tập)

Các ví dụ

- Viết thuật toán tìm tổng bình phương của một dãy. Xác định độ phức tạp của thuật toán.
- Viết thuật toán tìm giá trị của phần tử lớn nhất của một dãy các số thực. Xác định độ phức tạp của thuật toán.
- Viết thuật toán Euclide tìm ước số chung lớn nhất của 2 số nguyên.
- Viết thuật toán tính gần đúng giá trị của $\sin(x)$.

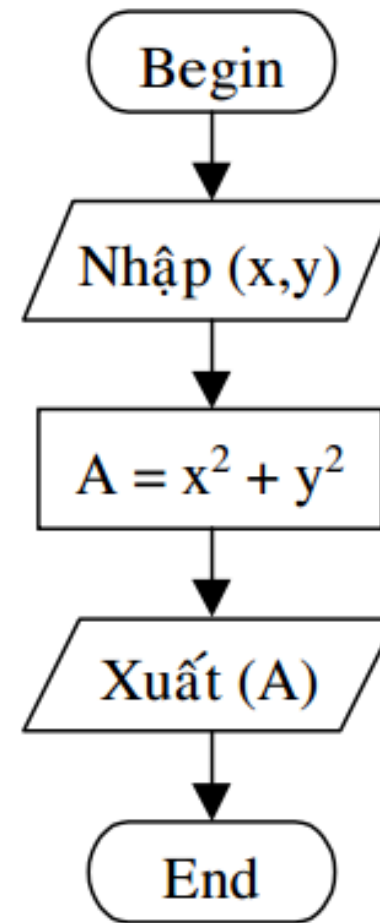
Các hình thức biểu diễn

- ✓ Biểu diễn tuần tự
- ✓ Biểu diễn phân nhánh
(*Có lựa chọn hướng xử lý*)
- ✓ Biểu diễn chu trình
(*Cấu trúc lặp*)

Cấu trúc tuần tự

Yêu cầu:

- Nhập 2 số x , y . Sau đó tính tổng bình phương của 2 số và in kết quả ra màn hình

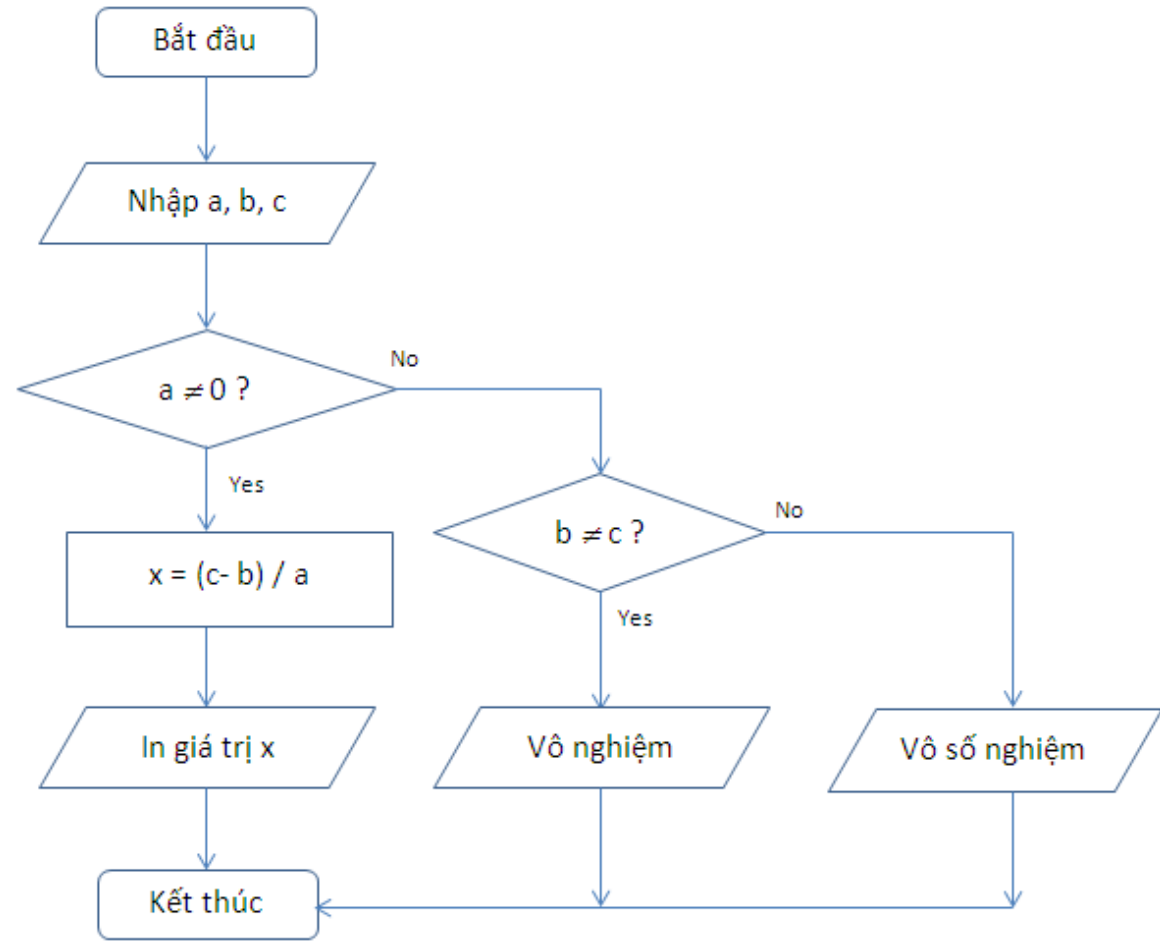


Cấu trúc phân nhánh

Yêu cầu:

- Giải phương trình

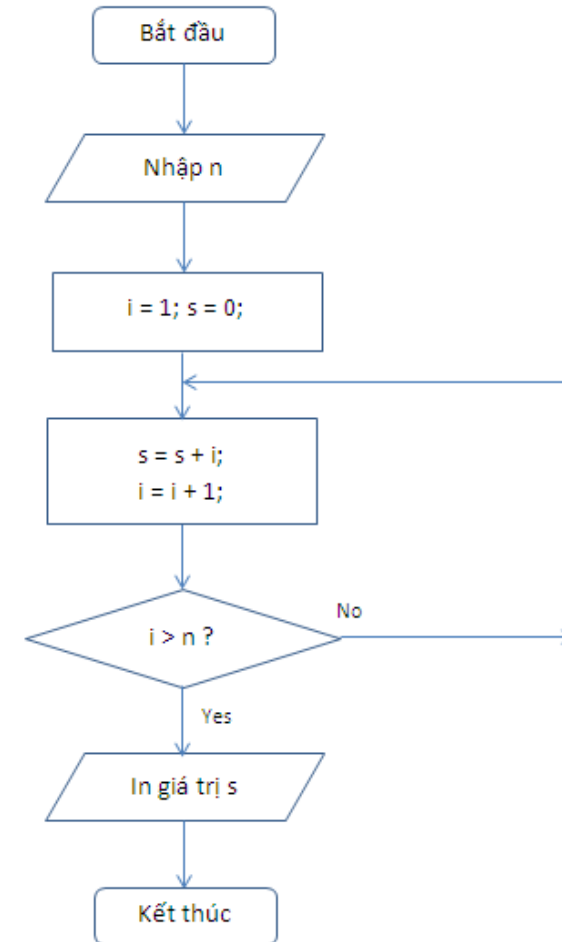
$$P(x): aX + b = c$$



Lặp, xử lý theo chu trình

Yêu cầu:

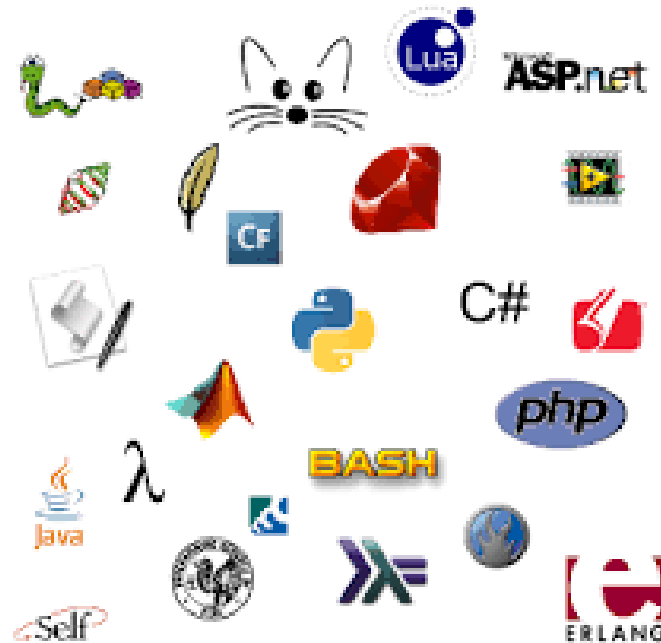
- Tính tổng các số tự nhiên từ 1 đến n
(n được nhập từ bàn phím)



Các ví dụ

- Tìm ước chung lớn nhất của 2 số nguyên A và B :
 - ✓B1: Nhập 2 số nguyên A và B
 - ✓B2: Gán $A = |A|$, $B = |B|$
 - ✓B3: Nếu $A=0$ và $B=0$ thì B9
 - ✓B4: Nếu $A=0$ và $B \neq 0$ thì B10
 - ✓B5: Nếu $B=0$ và $A \neq 0$ thì B11
 - ✓B6: Gán dư của phép chia A cho B vào biến D ($D = A \bmod B$)
 - ✓B7: Nếu $D = 0$ thì chuyển sang B10
 - ✓B8: Gán $A := B$; $B := D$; $D := A \bmod B$ chuyển về B7
 - ✓B9: Thông báo UCLN không tồn tại , chuyển về Bkt
 - ✓B10: Thông báo kết quả : Ước số chung lớn nhất là số B , chuyển về Bkt
 - ✓B11: Thông báo kết quả : Ước số chung lớn nhất là số A
 - ✓Bkt: Kết thúc

Ngôn ngữ lập trình



Ngôn ngữ lập trình

Ngôn ngữ lập trình (*Programming language*) là một dạng ngôn ngữ được thiết kế và chuẩn hóa (So với ngôn ngữ tự nhiên) để truyền các chỉ thị cho máy tính (hoặc các thiết bị khác có bộ xử lý: *Smartphone, Tablet, Smart TV, ...*).

Ngôn ngữ lập trình thường được dùng để tạo ra các chương trình nhằm phục vụ cho việc điều khiển máy tính hoặc mô tả các thuật toán xử lý của chương trình máy tính.

Ngôn ngữ lập trình thường được chia làm 3 dạng

- ❖ **Ngôn ngữ máy** (*Machine language*)
- ❖ **Hợp ngữ** (*Assembly language*)
- ❖ **Ngôn ngữ lập trình cấp cao** (*Higher-Level language*)

Chương trình dịch

Do máy tính (*Và các thiết bị cho phép lập trình*) chỉ có thể hiểu được ngôn ngữ máy, cho nên một chương trình sau khi đã được lập trình (*Viết bằng ngôn ngữ cấp cao, hợp ngữ*) cần phải chuyển đổi thành ngôn ngữ máy thì mới có thể thi hành được.

Những công cụ làm nhiệm vụ chuyển đổi cho mục đích này thường được gọi là ***chương trình dịch***.

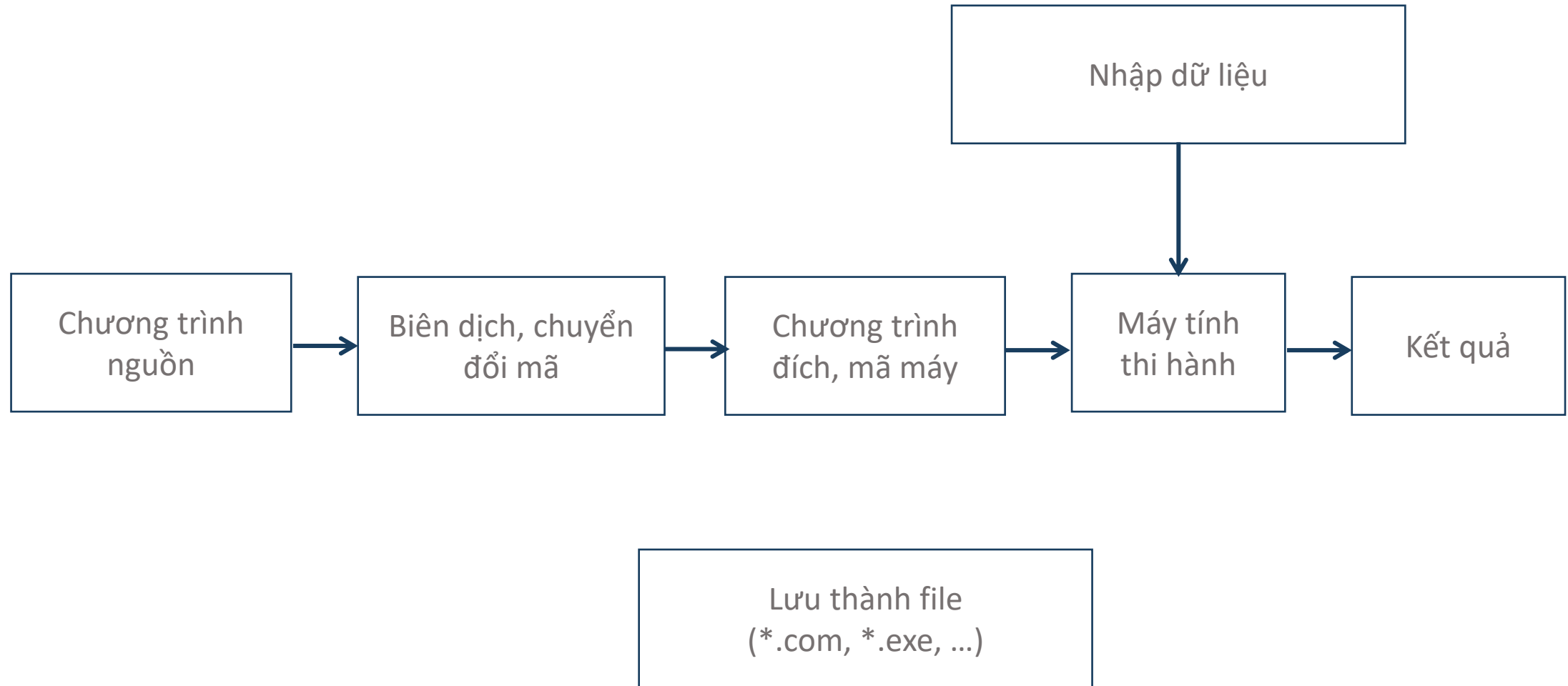
Chương trình dịch thường được phân biệt ở một trong hai dạng:

- ❖ **Trình thông dịch** (*Interpreter*)
- ❖ **Trình biên dịch** (*Compiler*)

Trình biên dịch - Compiler

- Làm nhiệm vụ chuyển đổi một chương trình đã được viết bằng một ngôn ngữ lập trình nào đó (*Còn gọi là chương trình nguồn*) thành ngôn ngữ máy (*Chương trình đích*).
- Quá trình chuyển đổi từ chương trình nguồn thành chương trình đích thường được gọi là **thời gian dịch** (*Compile-time*) và thời gian thực thi chương trình sau khi đã biên dịch thành công được gọi là **thời gian thực thi** (*Run-time*)

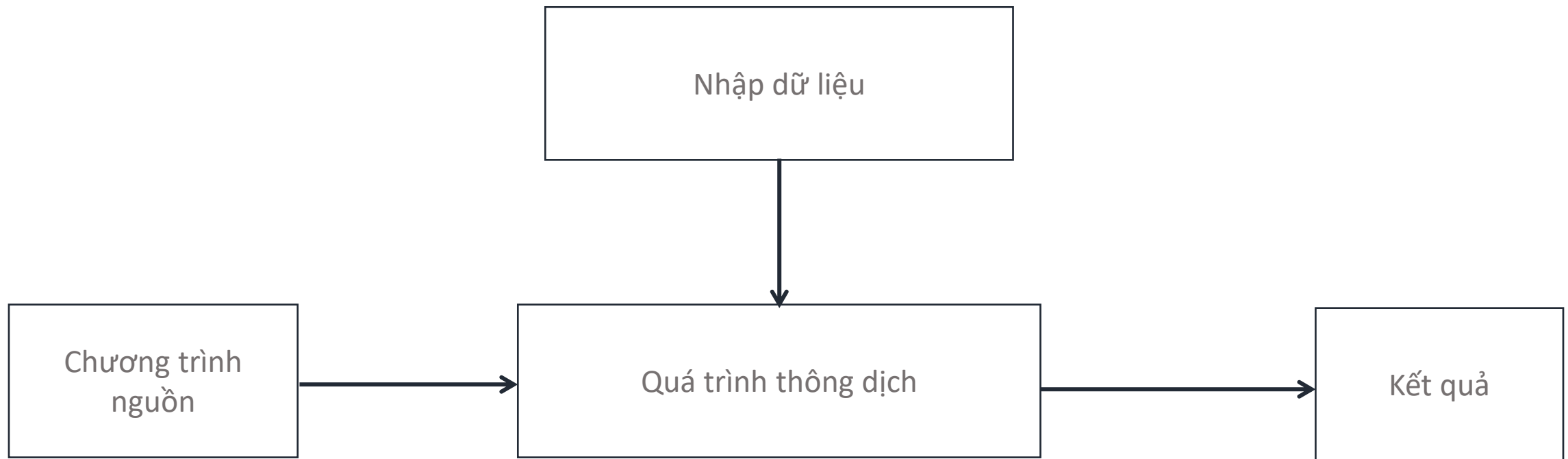
Cơ chế biên dịch



Trình thông dịch - Interpreter

- ❑ Những phần mềm có khả năng đọc và chuyển đổi mã nguồn của một chương trình đã được viết bằng ngôn ngữ lập trình ra mã máy để ra lệnh cho máy tính thi hành được gọi là trình thông dịch.
- ❑ Khác với trình biên dịch, trình thông dịch sẽ dịch từng câu lệnh từ chương trình nguồn theo yêu cầu thực thi.
- ❑ Như vậy, thời gian dịch diễn ra đồng thời với thời gian thực thi chương trình, quá trình này được gọi là ***Thông dịch***

Cơ chế thông dịch



Bài tập

- Cho một số yêu cầu thực tế để sinh viên thực hiện vẽ lưu đồ cho mục tiêu cần giải quyết.
- VD:
 - ❖ Vẽ lưu đồ cho bài toán chia hai số nguyên dương (*input: 2 số*)
 - ❖ Vẽ lưu đồ cho bài toán tính tiền 1 loại sản phẩm trong siêu thị (*input: số lượng, đơn giá*)
 - ❖ Vẽ lưu đồ cho việc xếp loại học lực dựa trên điểm trung bình (*input: điểm trung bình*)
 - ❖ ...

Ôn tập

- Các khái niệm thuật toán, giải thuật. Các đặc tính của thuật toán.
Các phương pháp biểu diễn thuật toán
- Lưu đồ, ký hiệu sử dụng trên lưu đồ. Các hình thức biểu diễn trên lưu đồ.
- Ngôn ngữ lập trình, ngôn ngữ lập trình thường được chia làm bao nhiêu loại ?
- Khái niệm về chương trình dịch, các cơ chế biên dịch, thông dịch

Tài liệu tham khảo

- Jose M. Garrido, “**Object-Oriented Programming: From Problem Solving to Java**”
- Paul Deitel, Harvey Deitel, “**Java : How to program**”, 9th edition, 2012
- Oracle, “**The Java™ Tutorials**”,
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html>, 6:20PM,
18/01/2018
- Java tutorial, <https://www.javatpoint.com/java-tutorial> , 6:20PM, 18/01/2018

THANK YOU VERY MUCH



Q/A: khoihue@gmail.com