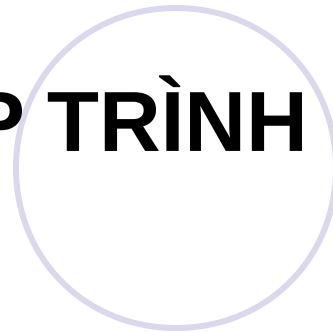
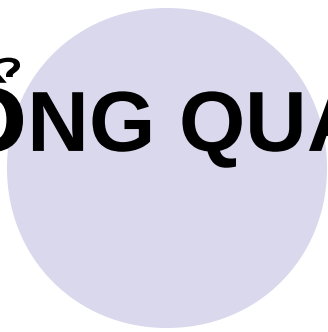
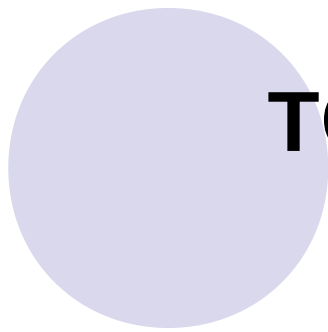
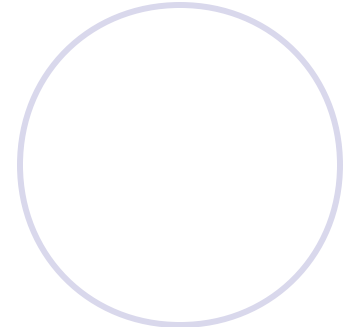
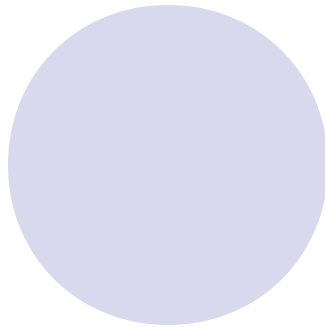
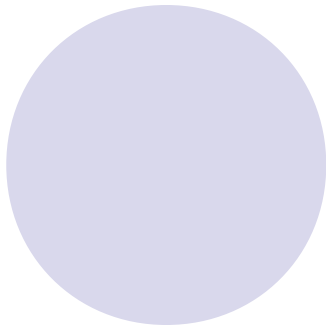


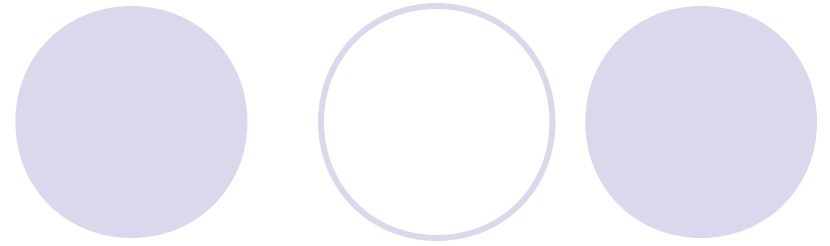
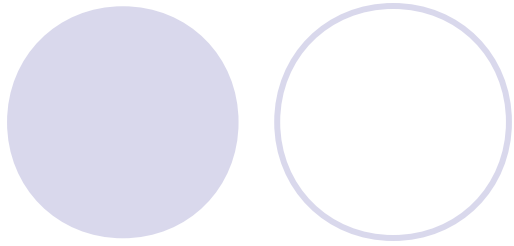
Mở đầu



TỔNG QUAN VỀ LẬP TRÌNH

Tại sao phải học $+ -x /$ khi có
máy tính bỏ túi rồi???





Word → soạn thảo

Excel → tính toán

...

Có quá nhiều ứng dụng rồi vậy học trình để
làm gì

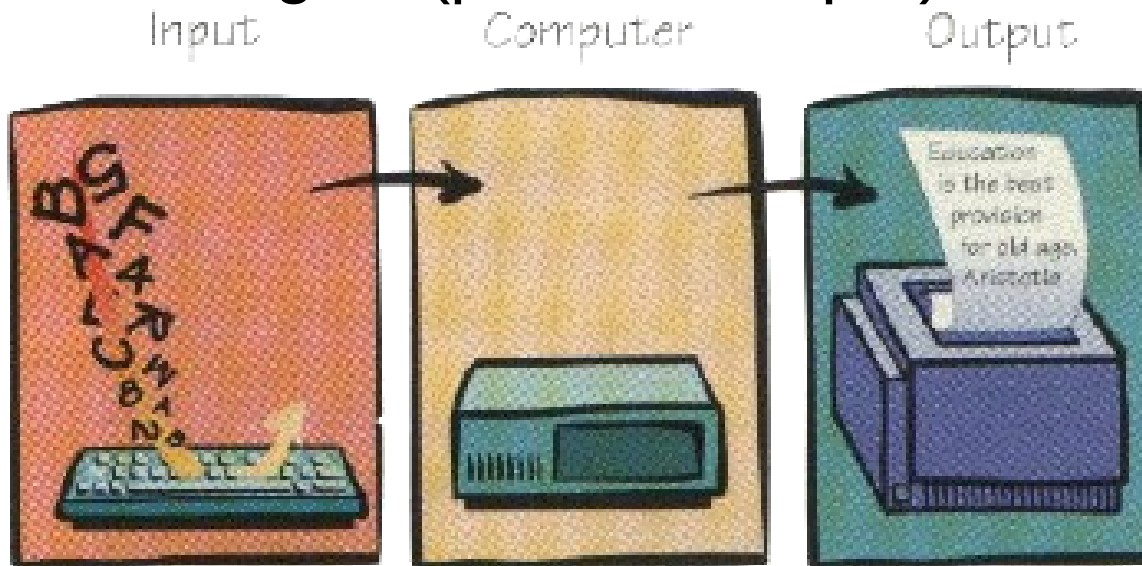
→ Học để tạo ra những sản phẩm như thế

Vậy mình đi sau thì sao mà bằng họ

- Đúng
- Nhưng từ những cái mình đã biết mình sẽ làm được nhiều cái hay hơn
- Ví dụ: máy tính có thể nhìn được
- Quan sát được
- Nhiều vấn đề đang bỏ ngỏ

Quá trình xử lý thông tin trên máy tính

- Máy tính là công cụ xử lý thông tin.
 - Nhận thông tin (Receive input)
 - **Xử lý thông tin (process information)**
 - Xuất thông tin (produce output)



Lập trình là gì?

Dạy cho máy tính thực hiện yêu cầu

Chỉ cách lưu trữ dữ liệu → cấu trúc dữ liệu

Chỉ cách xử lý dữ liệu → thuật toán xử lý

Chỉ cách trình bày dữ liệu → giao diện

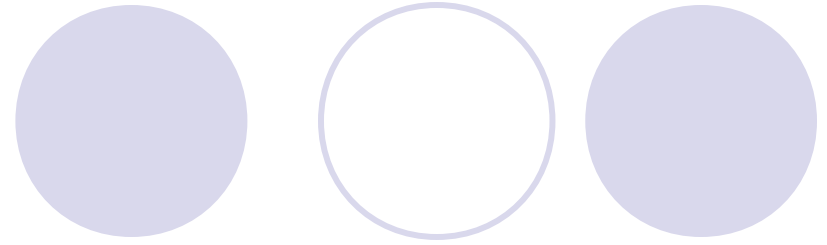
Dùng một ngôn ngữ lập trình và kỹ thuật
vận dụng ngôn ngữ lập trình để thực hiện
các yêu cầu trên

Chương trình

- Chương trình = thuật toán + cấu trúc dữ liệu
 - Program = Algorithm + Data Structure

- Chương trình = giao diện + thuật toán + cấu trúc dữ liệu
 - Program = interface + Algorithm + Data Structure

Cấu trúc dữ liệu



- Mô tả dữ liệu
- Mô tả đặc tính
- Mô tả cách lưu trữ
- Mô tả thao tác trên dữ liệu
- Truy xuất dữ liệu
-

Xử lý thông tin – Thuật toán

- **Thuật toán** là một tập các hướng dẫn nhằm thực hiện một công việc nào đó.
- Đối với việc giải quyết một vấn đề - bài toán thì **thuật toán** có thể hiểu là một tập hữu hạn các hướng dẫn rõ ràng để người giải toán có thể theo đó mà giải quyết được vấn đề.
- Như vậy, **thuật toán** là một phương pháp thể hiện lời giải của vấn đề - bài toán.

Nguyên tắc lập trình

- Điều kiện để quyết định
- "Thực thi được" xét trong điều kiện hiện tại của bài toán
- Tính "dừng" tránh lặp vô tận hoặc bị quẩn

Các đặc trưng khác của thuật toán tính

- **1. Đầu vào và đầu ra (input/output)** : mọi thuật toán, dù có đơn giản đến mấy cũng phải nhận dữ liệu đầu vào, xử lý nó và cho ra kết quả cuối cùng.
- **2. Tính hiệu quả (effectiveness)** : tính hiệu quả của thuật toán được đánh giá dựa trên một số tiêu chuẩn như khối lượng tính toán, không gian và thời gian khi thuật toán được thi hành. Tính hiệu quả của thuật toán là một yếu tố quyết định để đánh giá, chọn lựa cách giải quyết vấn đề-bài toán trên thực tế. Có rất nhiều phương pháp để đánh giá tính hiệu quả của thuật toán. Trong mục 3 của chương , ta sẽ tìm hiểu một tiêu chuẩn được dùng rộng rãi là độ phức tạp của thuật toán.
- **3. Tính tổng quát (generalliness)** : thuật toán có tính tổng quát là thuật toán phải áp dụng được cho mọi trường hợp của bài toán chứ không phải chỉ áp dụng được cho một số trường hợp riêng lẻ nào đó

Biểu diễn thuật toán



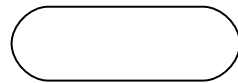
- Ngôn ngữ tự nhiên
- Lưu đồ - sơ đồ khối (flowchart)
- Dùng mã giả (pseudocode)
 - Vay mượn cú pháp của một ngôn ngữ lập trình nào đó để thể hiện thuật toán và một phần ngôn ngữ tự nhiên

Lưu đồ



- Công cụ trực quan diễn đạt thuật toán giúp người đọc theo dõi được sự phân cấp các trường hợp và quá trình xử lý của thuật toán
- Thường dùng diễn đạt cho các thuật toán phức tạp
- Ưu điểm
 - Mới nhìn qua ta có thể dễ dàng hiểu được tốt hơn là đọc một bản mô tả dài dòng
 - Chương trình có thể được xem xét và gỡ rối một cách dễ dàng
 - Cung cấp các sườn liệu cho chương trình một cách hiệu quả
 - Giúp dễ dàng giải thích chương trình hoặc thảo luận về một giải pháp

Các ký hiệu dùng trong lưu đồ



Điểm bắt đầu hoặc kết thúc chương trình

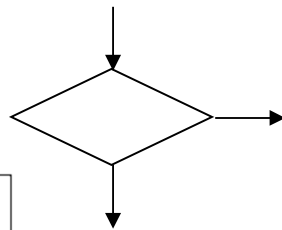
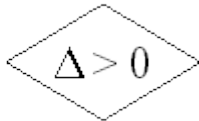
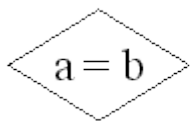
Tăng lên 1



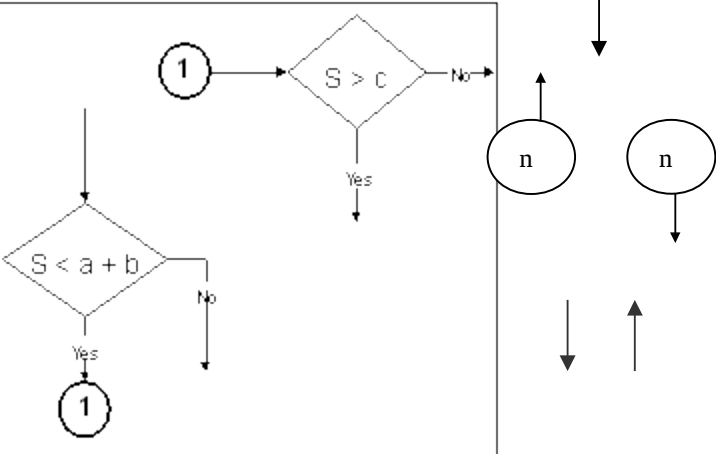
Thao tác xử lý



Thao tác nhập xuất



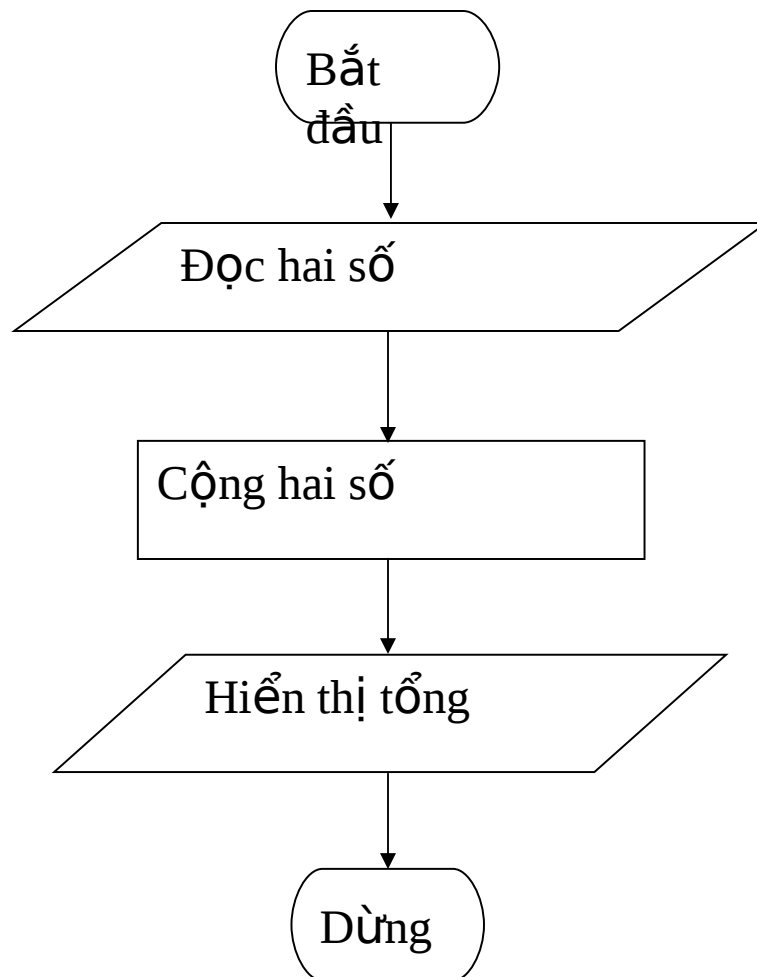
Thao tác lựa chọn



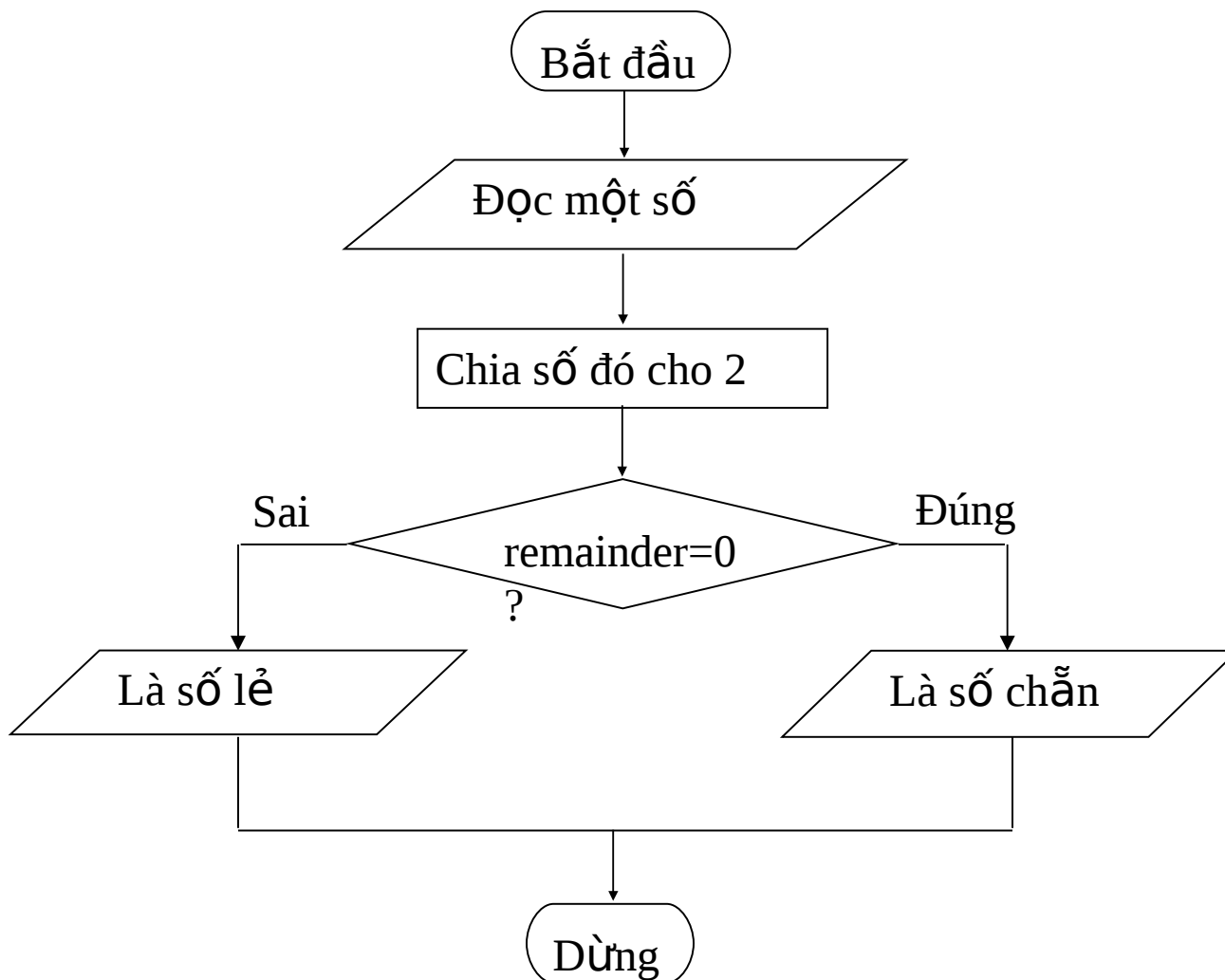
Điểm nối

Đường đi

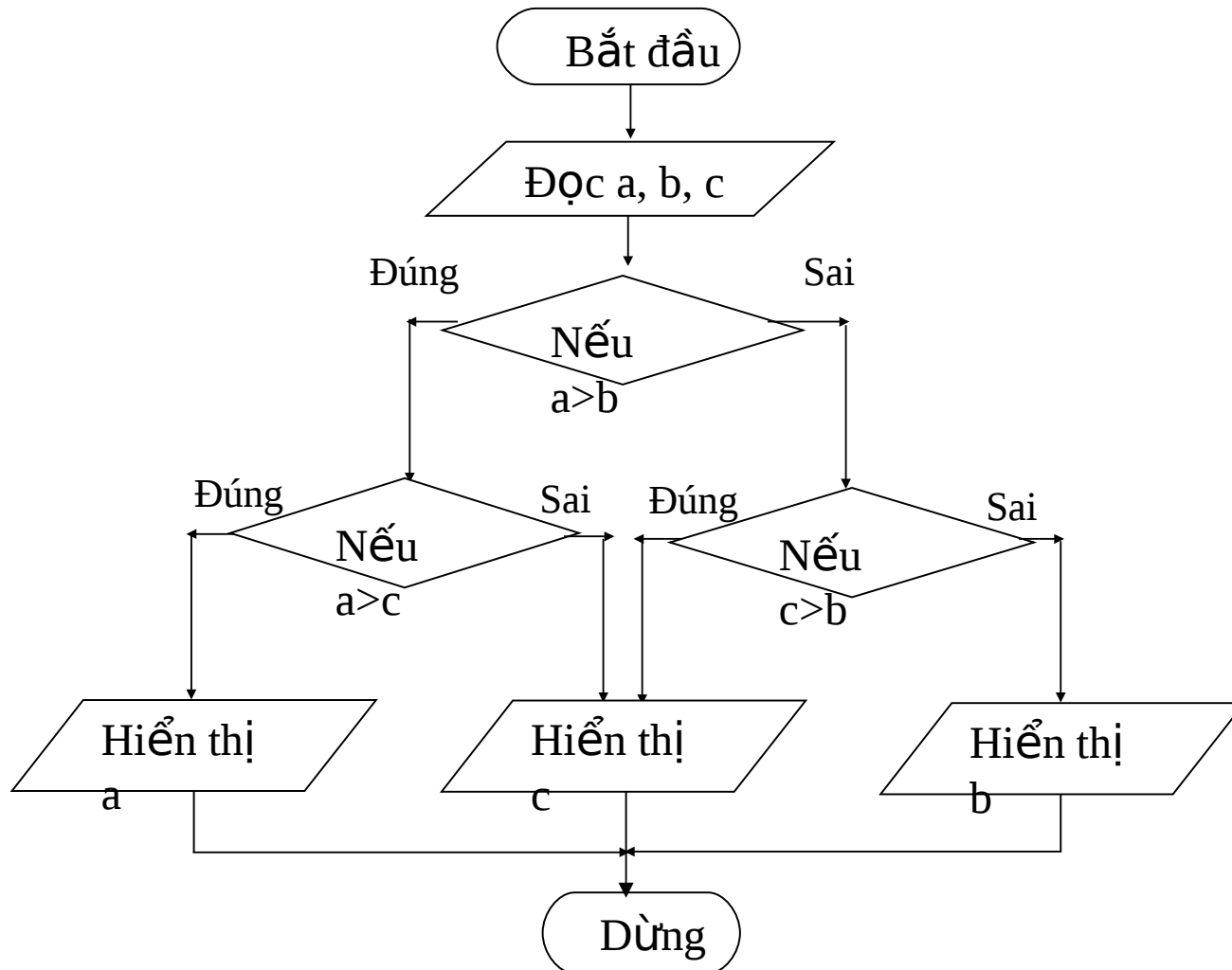
Tính tổng của hai số



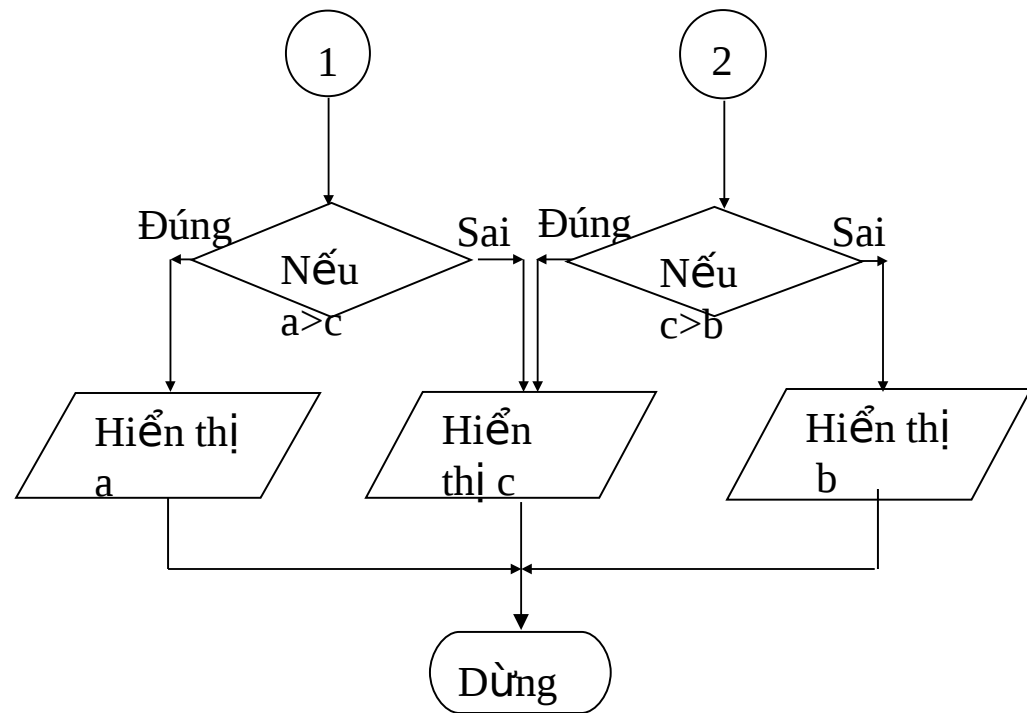
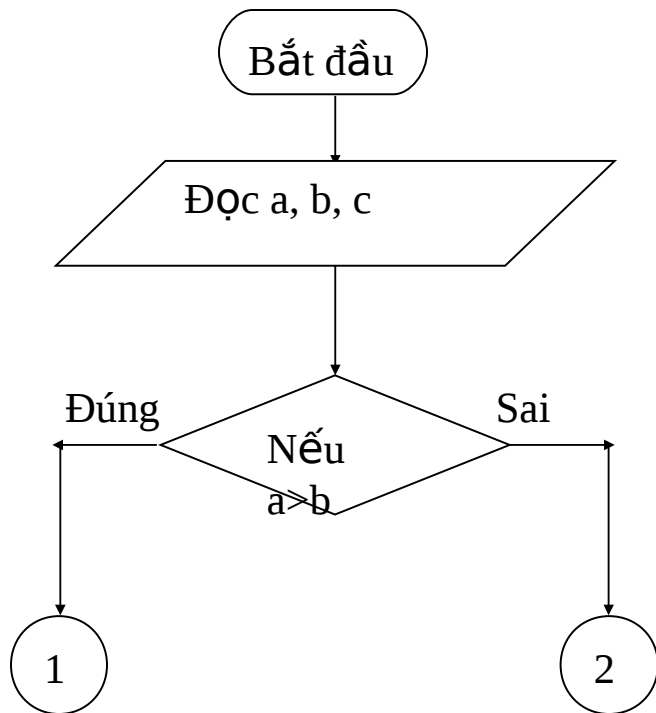
Kiểm tra một số là số chẵn hay số lẻ



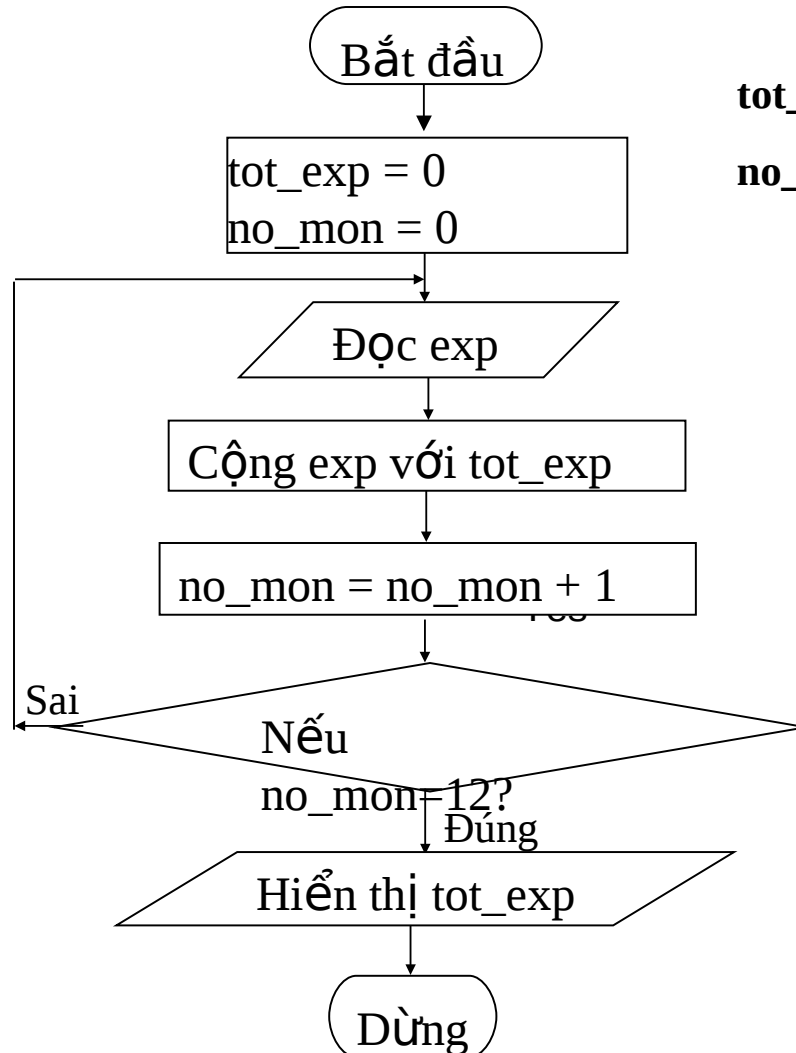
Tìm số lớn nhất trong 3 số



Tìm số lớn nhất trong 3 số



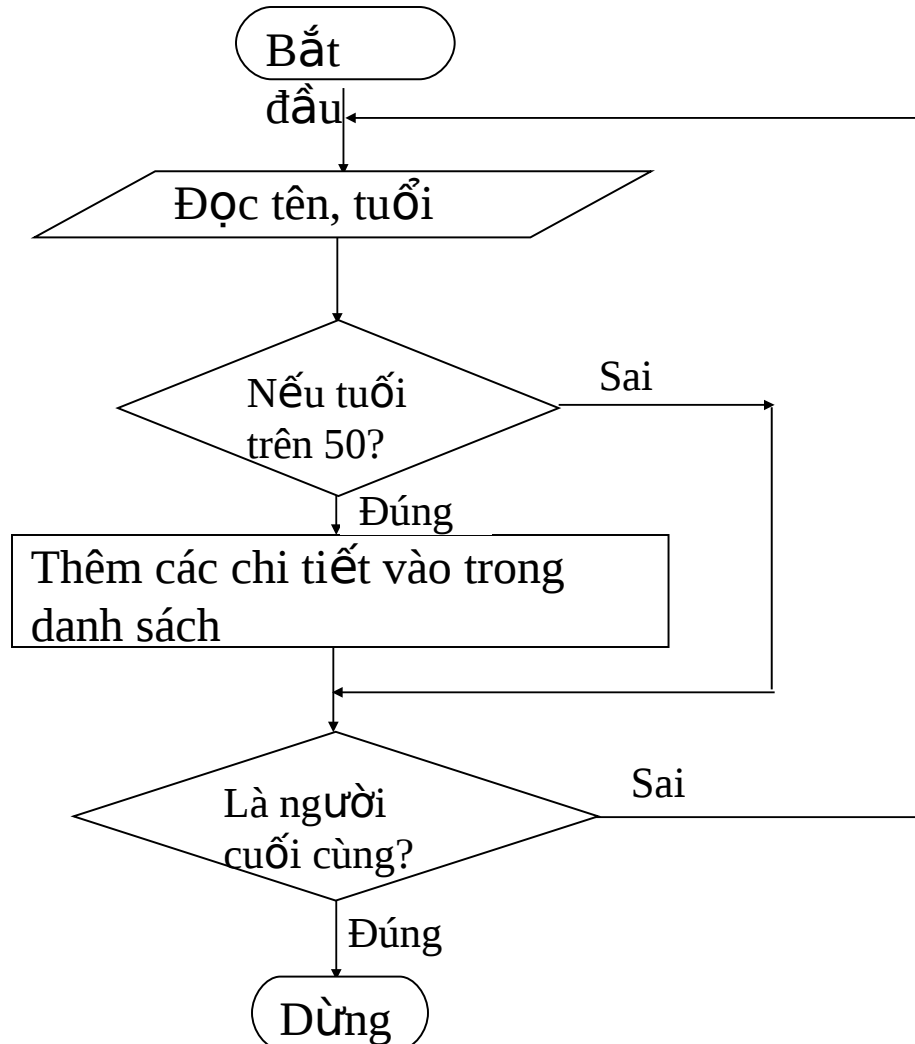
Tổng chi phí hàng tháng trong năm



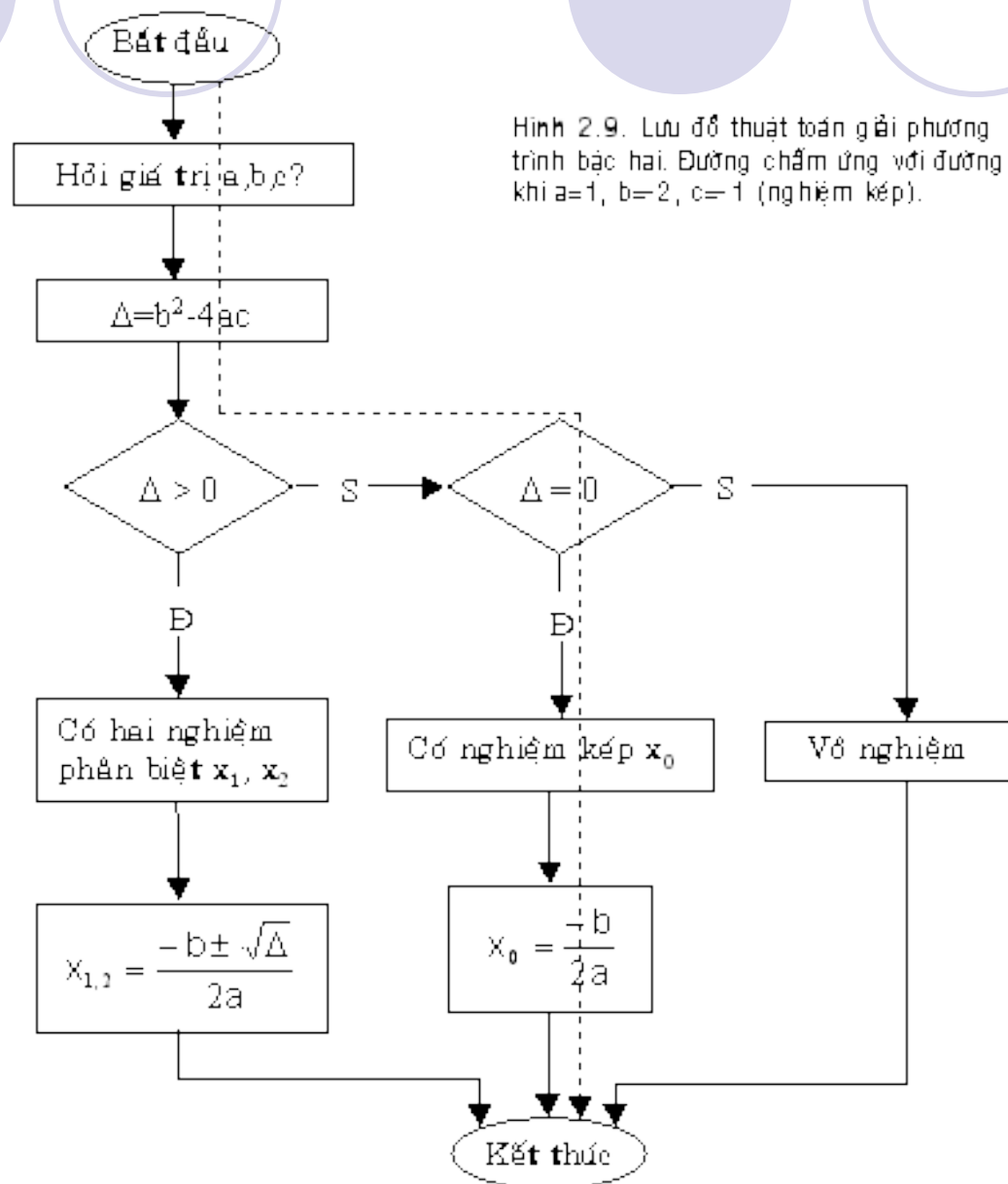
tot_exp – Total expenditure

no_mon – Number of months

Đọc các thông tin như tên và tuổi và lưu lại những người có tuổi trên 50



Đường đi giải phương trình bậc hai



Hình 2.9. Lưu đồ thuật toán giải phương trình bậc hai. Đường chấm ứng với đường đi khi $a=1$, $b=2$, $c=-1$ (ngheem kép).

Lời khuyên khi vẽ lưu đồ

- Trước tiên hãy tập trung vào mặt luận lý của vấn đề và vẽ một số đường đi chính của lưu đồ
- Thêm vào tất cả các nhánh và vòng lặp
- Một lưu đồ chỉ có một điểm Start (bắt đầu) và một điểm Stop (dừng)
- Mỗi bước trong chương trình không cần thể hiện trong lưu đồ
- Nên nhớ rằng những người sử dụng hoặc lập trình viên khác phải có thể hiểu lưu đồ một cách dễ dàng

Thuật toán giải phương trình bậc hai

$ax^2+bx+c=0$

1. Yêu cầu cho biết giá trị của 3 hệ số a, b, c

2. Nếu $a=0$ thì

2.1. giải phương trình bậc 1

2.2. Kết thúc thuật toán.

3. Trường hợp $a \neq 0$ thì

3.1. Tính giá trị $D = b^2-4ac$

3.2. Nếu $D > 0$ thì

3.2.1. Phương trình có hai nghiệm phân biệt x_1 và x_2

$X_1=(-b -\sqrt{D})/2a$, $X_2=(-b +\sqrt{D})/2a$

3.2.2. Kết thúc thuật toán.

3.3. Nếu $D = 0$ thì

3.3.1. Phương trình có nghiệm kép x_0

3.3.2. Giá trị của nghiệm kép là
 $X_1=-b /2a$

3.3.3. Kết thúc thuật toán

3.4. Nếu $D < 0$ thì

3.4.1. Phương trình vô nghiệm.

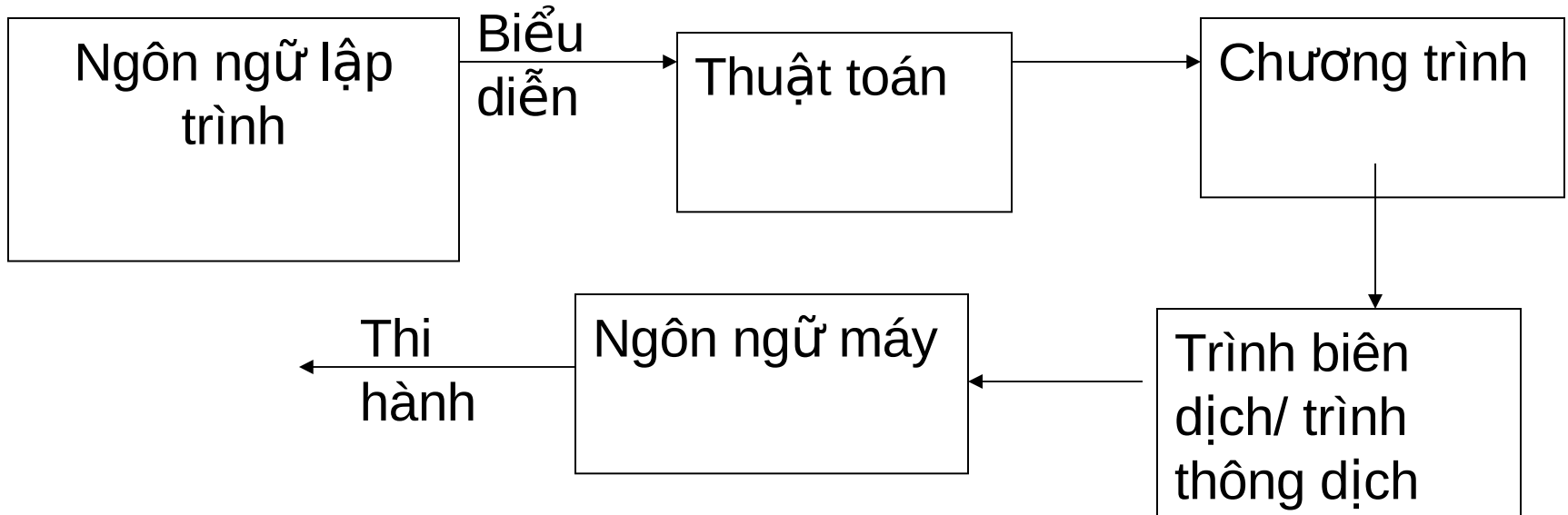
3.4.2. Kết thúc thuật toán.

Ví dụ về mã giả

- if $\Delta > 0$ then begin
- $x_1 = \frac{-b - \sqrt{\Delta}}{2a}$
- $x_2 = \frac{-b + \sqrt{\Delta}}{2a}$
- xuất kết quả : phương trình có hai nghiệm là x_1 và x_2
- end

- else
- if $\Delta = 0$ then
- xuất kết quả : phương trình có nghiệm kép là $-b/(2a)$
- else {trường hợp $\Delta < 0$ }
- xuất kết quả : phương trình vô nghiệm

Lập trình



Ngôn ngữ lập trình

- ngôn ngữ lập trình

- một tập từ ngữ và ký hiệu cho phép lập trình viên hoặc người dùng có thể nói chuyện với máy tính. ngôn ngữ lập trình cũng có các luật được gọi là cú pháp (syntax) để đảm bảo ngôn ngữ đó được vận dụng một cách chính xác.

- chương trình máy tính

- Đó là một tập các chỉ thị (instruction) được sắp xếp theo một trật tự định trước nhằm hướng dẫn máy tính thực hiện các thao tác, hành động cần thiết để đáp ứng một mục tiêu đã định trước của con người như truy xuất dữ liệu, tìm kiếm, giải bài toán, ... Các chỉ thị này có thể được viết bằng nhiều ngôn ngữ lập trình khác nhau.

Các loại ngôn ngữ lập trình thông dụng

- Ngôn ngữ máy
- Hợp ngữ
- Ngôn ngữ cấp cao
- **BASIC, COBOL, C, FORTRAN, PASCAL,**
- **Visual Basic, visual C, C++, C#, Delphi**
- **JAVA, PHP, ASP, JSP**

Trình biên dịch- Trình thông dịch

- Trình biên dịch : sẽ chuyển đổi toàn bộ chương trình sang mã máy, rồi chứa kết quả vào đĩa để có thể thi hành về sau. Chương trình ngôn ngữ cấp cao được chuyển đổi được gọi là chương trình nguồn (source program) và chương trình ngôn ngữ máy được tạo ra được gọi là chương trình đối tượng (object program) hoặc mã đối tượng (object code) : C, PASCAL...
- Trình thông dịch : thay vì chuyển đổi toàn bộ chương trình nguồn như trình biên dịch, trình thông dịch chỉ chuyển đổi một mệnh đề của chương trình và thực hiện đoạn mã kết quả ngay, sau đó nó tiếp tục chuyển đổi mệnh đề thứ 2 rồi thi hành đoạn mã kết quả thứ 2 và cứ thế. Khi sử dụng trình thông dịch, mỗi lần chạy chương trình là mỗi lần chương trình nguồn được thông dịch sang ngôn ngữ máy. Không có chương trình đối tượng nào được tạo ra. JAVA cript,PHP,ASP...

Chương trình

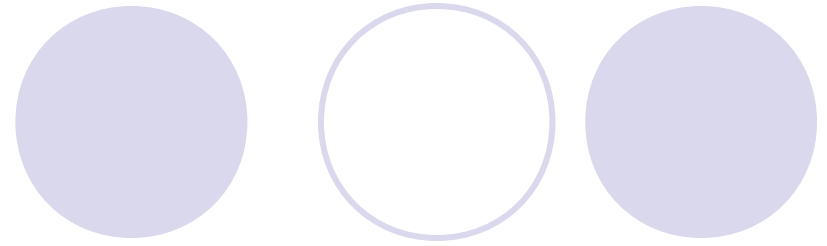
- Các xử lý → thuật toán
- Giao diện: tương tác với người sử dụng
 - Dòng lệnh
 - Đồ hoạ



Ví dụ

- Bài toán : tính a^3 , với a là một con số nhập vào, in kết quả ra màn hình
- Input : số a
- Xử lý kết quả = $a*a*a$
- Output: kết quả a^3

Bảng dòng lệnh



```
đây là chương trình tính lập phương  
hãy nhập vào 1 số a : 5  
lập phương của 5 là 125
```

Xử lý

```
Project1 - Form1 (Code)
Command2 Click
Private Sub Command1_Click(Index As Integer)
Dim a As Double, kq As Double
a = Val(Text1.Text)
kq = a * a * a
Text2.Text = Str(kq)
End Sub
Private Sub Command2_Click(Index As Integer)
End
End Sub
```

Form1

Đây là chương trình
Tính lập phương của 1 số

Nhập vào 1 số a:

Kết quả là

Khi chạy chương trình

Form1

Đây là chương trình
Tính lập phương của 1 số

Nhập vào 1 số a:

Kết quả là

Form1

Đây là chương trình
Tính lập phương của 1 số

Nhập vào 1 số a:

Kết quả là

Nội dung



- Khái niệm thuật toán
- Độ phức tạp của thuật toán
- Lưu đồ thuật toán
- Ngôn ngữ lập trình

Giới thiệu



- Sử dụng máy tính để
 - Giải quyết các vấn đề
 - Thực hiện tính toán
- Chương trình
 - Là tập hợp các lệnh được cung cấp cho máy tính để giải quyết vấn đề

Thuật toán (Algorithm)

- Một tập *hữu hạn* các hướng dẫn *rõ ràng* để người giải toán có thể theo đó mà giải quyết được vấn đề
- Phương pháp thể hiện lời giải của vấn đề - bài toán
- Trong khoa học máy tính, thuật toán được định nghĩa là một dãy *hữu hạn* các bước *không mập mờ* và *có thể thực thi được*, quá trình hành động theo các bước này phải *dừng* và cho được *kết quả như mong muốn*
- Tính hữu hạn, tính xác định và tính đúng của thuật toán

Thuật toán (Algorithm)

- Đánh giá thuật toán dùng để chọn lớp trưởng cho một lớp học
 1. Lập danh sách tất cả học sinh trong lớp
 2. Sắp thứ tự danh sách học viên
 3. Chọn học sinh đứng đầu danh sách để làm lớp trưởng

Thuật toán (Algorithm)

- Thuật toán chọn lớp trưởng cho một lớp học đã sửa đổi
 1. Lập danh sách tất cả học sinh trong lớp theo hai thông tin: Họ và Tên; Điểm trung bình cuối năm
 2. Sắp hạng học sinh theo điểm trung bình theo thứ tự giảm dần (từ điểm cao đến điểm thấp), hai học sinh có cùng điểm trung bình sẽ có cùng hạng
 3. Nếu chỉ có một học sinh hạng nhất thì chọn học sinh đó làm lớp trưởng, trường hợp có nhiều học sinh đồng hạng nhất thì tiến hành

Thuật toán (Algorithm)

- Đánh giá thuật toán dùng để tính tổng các số nguyên dương lẻ trong khoảng từ 1 đến n :
 1. Hỏi giá trị của n
 2. $S = 0$
 3. $i = 1$
 4. Nếu $i = n+1$ thì sang bước 8, ngược lại sang bước 5
 5. Cộng thêm i vào S
 6. Cộng thêm 2 vào i
 7. Quay lại bước B4
 8. Tổng cần tìm chính là S

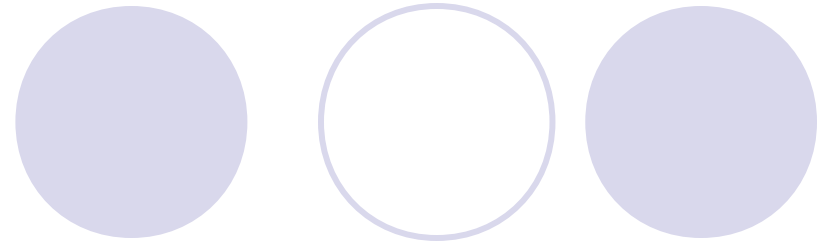
Thuật toán (Algorithm)

- Ngoài 3 đặc trưng chính là tính hữu hạn, tính xác định và tính đúng, thuật toán còn có thêm 3 đặc trưng phụ khác:
 - Đầu vào và đầu ra (Input/Output)
 - Tính hiệu quả (Effectiveness)
 - Tính tổng quát (Generalliness)

Thuật toán (Algorithm)

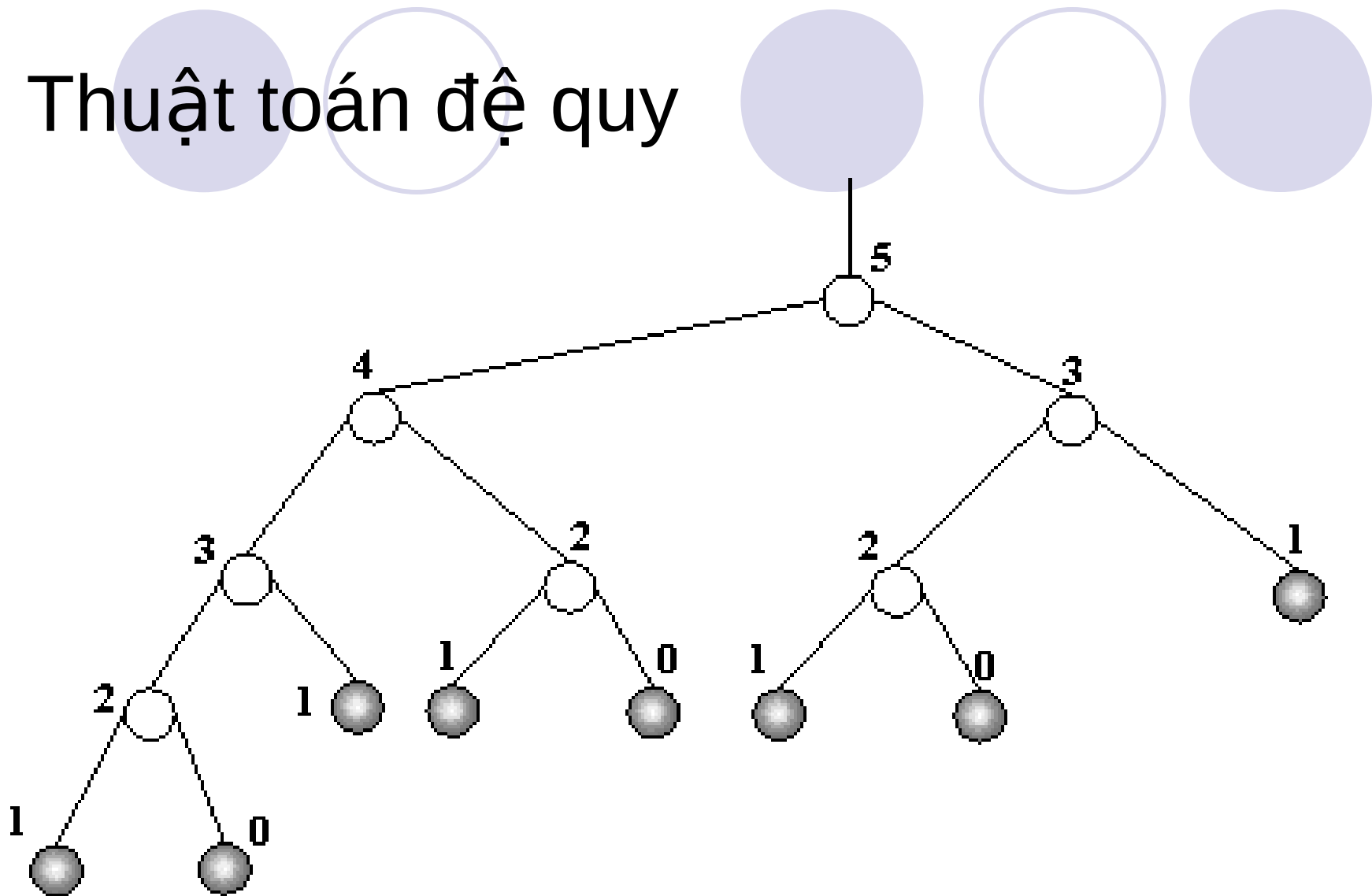
- Hãy viết các thuật toán
 - Giải phương trình bậc hai $ax^2+bx+c=0$
 - Tìm hộp có trọng lượng nặng nhất trong n hộp với một cái cân. Bài toán tổng quát: cho một tập A hữu hạn, hãy tìm phần tử lớn nhất của A
 - Thuật toán Eulid tìm ước số chung lớn nhất của hai số nguyên dương a và b

Thuật toán đệ quy



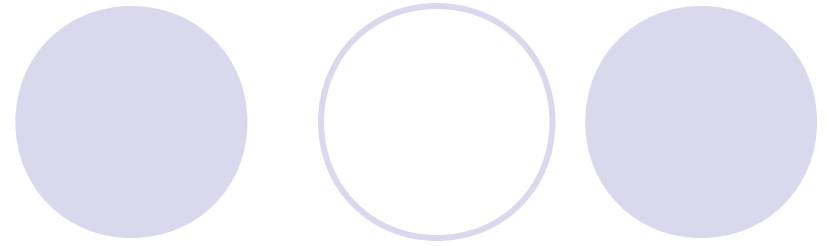
- Đưa bài toán hiện tại về một bài toán cùng loại, cùng tính chất (đồng dạng) nhưng ở cấp độ thấp hơn và quá trình này tiếp tục cho đến lúc bài toán được đưa về một cấp độ mà tại đó có thể giải được. Từ kết quả ở cấp độ này, lần ngược để giải được bài toán ở cấp độ cao đến lúc ban đầu
 - Định nghĩa giai thừa: $0! = 1$, $n! = (n-1)!n$ với mọi $n > 0$
 - Định nghĩa dãy số Fibonacci: $f_0 = 1$, $f_1 = 1$, $f_n = f_{n-1} + f_{n-2}$

Thuật toán đệ quy



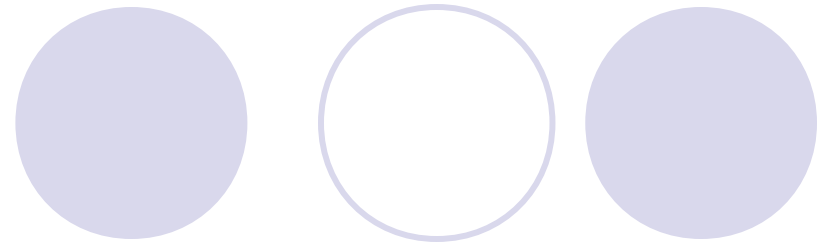
Hình 2.12. Quá trình tính toán số hạng thứ năm của dãy Fibonacci

Thuật toán đệ quy

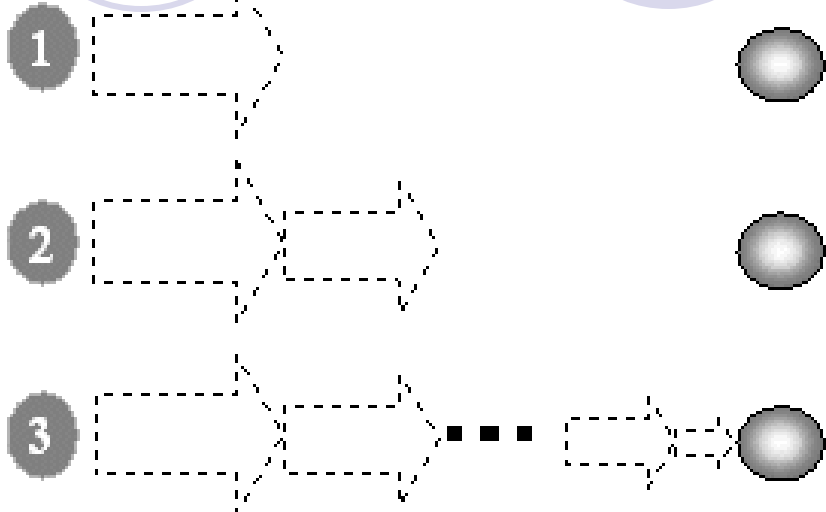


- Mọi thuật toán đệ quy gồm hai phần:
 - Phần cơ sở: các trường hợp không cần thực hiện lại thuật toán (không gọi đệ quy), chính là các điểm dừng
 - Phần đệ quy: có yêu cầu gọi đệ quy, tức yêu cầu thực hiện lại thuật toán với cấp độ dữ liệu thấp hơn

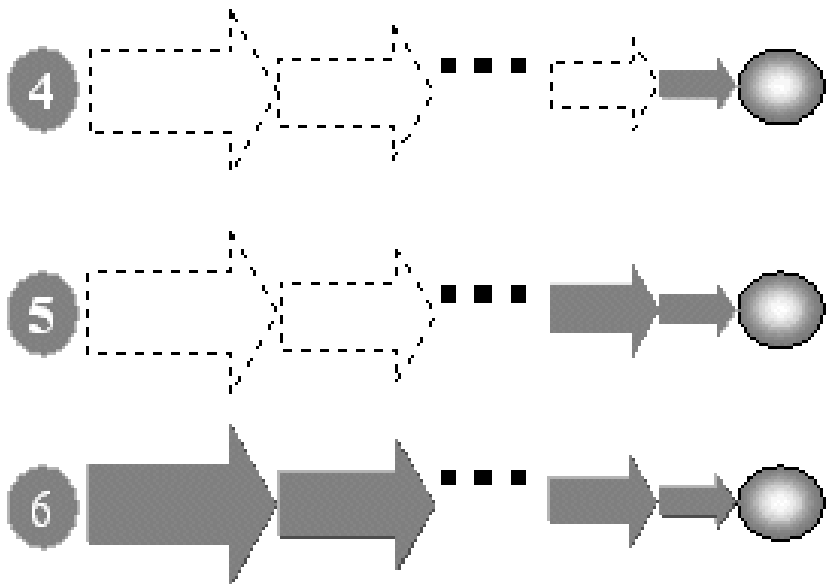
Thuật toán đệ quy



Quá trình đệ quy



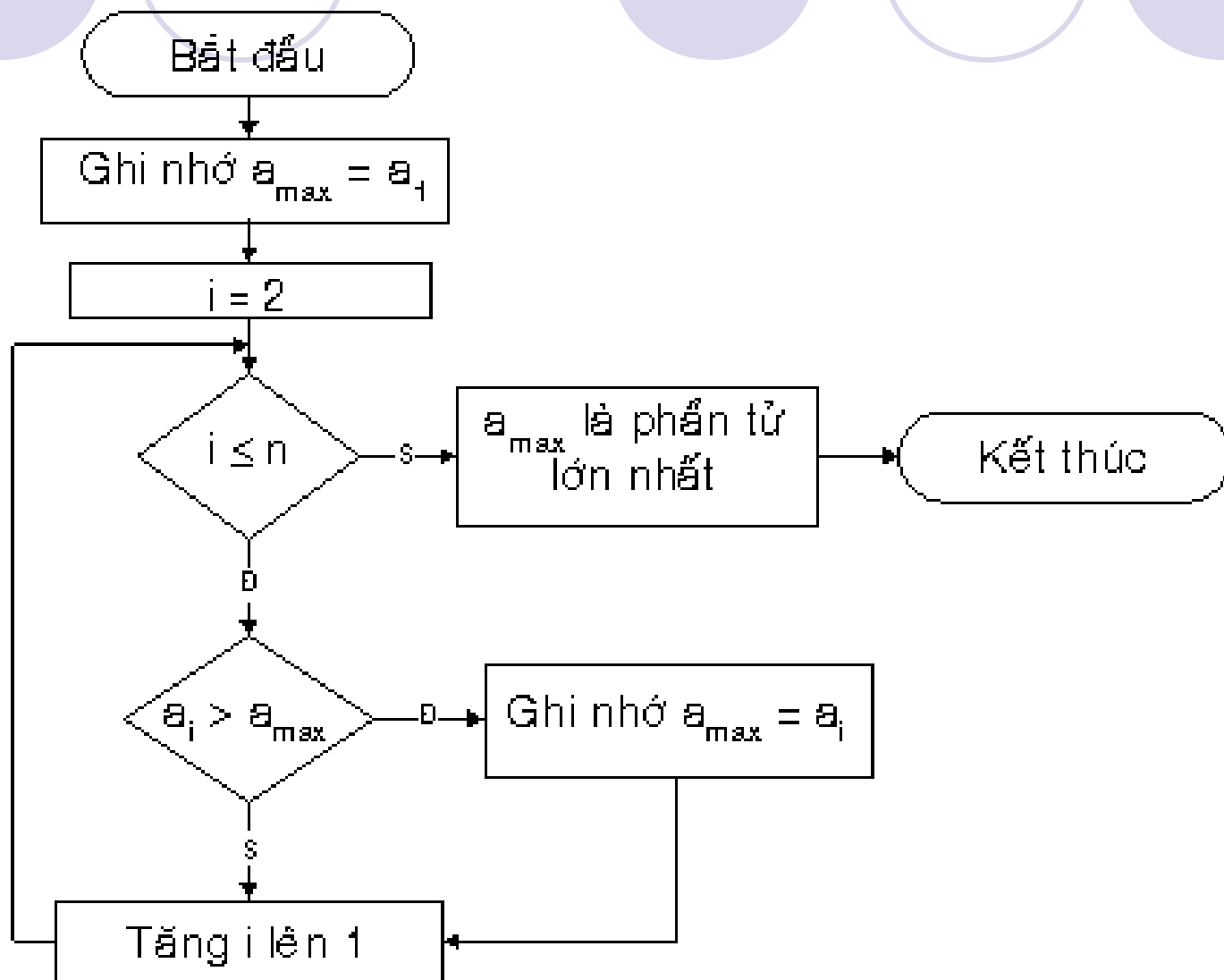
Quá trình giải



Độ phức tạp của thuật toán

- Phân tích thuật toán: tìm một cách đánh giá về thời gian và “không gian” cần thiết để thực hiện thuật toán
- Đánh giá về thời gian của thuật toán không phải là xác định thời gian tuyệt đối mà là mối liên quan giữa dữ liệu đầu vào và chi phí để thực hiện thuật toán $T=f(\text{input})$
- Thông thường chỉ quan tâm đến mối liên quan giữa *độ lớn* của dữ liệu đầu vào và chi phí $T=f(n)$ trong trường hợp *tốt nhất* và *xấu nhất*

Độ phức tạp của thuật toán



Lưu đồ thuật toán tìm phần tử lớn nhất

Độ phức tạp của thuật toán

- Chi phí chung = chi phí so sánh + chi phí gán
- Trong mọi trường hợp, phép so sánh $a_i > a_{\max}$ luôn được thực hiện và số lần thực hiện là $n-1$: chi phí cố định (bất biến) của bài toán
- Trường hợp tốt nhất: số lớn nhất ở đầu dãy. Do đó phép gán $a_{\max} = a_i$ không cần thực thi, như vậy chi phí chung cho trường hợp này chính là chi phí cố định $T=f(n)=n-1$
- Trường hợp xấu nhất: số lớn nhất ở cuối dãy. Do đó phép gán $a_{\max} = a_i$ luôn cần thực thi, số lần thực thi là $n-1$, như vậy chi phí chung cho trường hợp này là $T=f(n)=2(n-1)=2n-2$
- Độ phức tạp của thuật toán là $O(n)$, tuyến tính

Phương pháp lập trình

- Lập trình cấu trúc có các đặc trưng
 - Tính đơn thể
 - Cấu trúc điều khiển: phối hợp cấu trúc tuần tự, cấu trúc chọn và cấu trúc lặp
 - Tính vào/ra đơn
- Lập trình hướng đối tượng: một tiếp cận mới trong lập trình sử dụng khái niệm đối tượng, kết hợp cả dữ liệu và các câu lệnh của chương trình

Các bước xây dựng chương trình

- Bước 1. Xác định vấn đề - bài toán.
- Bước 2. Lựa chọn phương pháp giải.
- Bước 3. Xây dựng thuật toán hoặc thuật giải.
- Bước 4. Cài đặt chương trình.
- Bước 5. Hiệu chỉnh chương trình.
 - Ở bước 4, nói chung chúng ta không tránh khỏi sai sót. Ở bước 5 này chúng ta cho chương trình chạy thử để phát hiện và điều chỉnh các sai sót nếu tìm thấy.
 - Có hai loại lỗi:
 - **Lỗi cú pháp** là lỗi do không tuân thủ đúng các quy tắc viết chương trình trên một ngôn ngữ lập trình cụ thể.
 - **Lỗi ngữ nghĩa** là lỗi làm sai lạc ý nghĩa hoặc dẫn đến bế tắc của chương trình. Lỗi cú pháp thường dễ phát hiện và hiệu chỉnh hơn lỗi ngữ nghĩa. Cần phải nói rằng việc hiệu chỉnh chương trình khá phức tạp, mất nhiều thời gian và công sức. Việc xây dựng tốt, phù hợp, đầy đủ các bộ dữ liệu để kiểm chứng chương trình là hết sức quan trọng, giúp phát hiện ra các lỗi ngữ nghĩa của chương trình cũng như có thể có vấn đề gì đó bị bỏ sót.
- Bước 6. Thực hiện chương trình.

Bài tập



- Xác định đầu vào và đầu ra của chương trình
- Bài toán về tam giác
- Bài toán về hcn
- Bài toán về giải phương trình bậc 2
- Bài toán về nhập một biểu thức toán cho ra kết quả
- Bài toán tính tổng N số nguyên đầu tiên
- Bài toán tính $N!$
- Đổi tiền việt sang USD và ngược lại
-

Bài tập

- 8). nhập vào 1 ký tự và cho biết đó là nguyên âm hay phụ âm
- 9) nhập vào 1 chuỗi và cho biết chuỗi đó có bao nhiêu nguyên âm
- 10) nhập vào 1 chuỗi và cho biết chuỗi đó có bao nhiêu phụ âm
- 11) nhập vào 3 số và cho biết số lớn nhất và nhỏ nhất
- 12) nhập vào 1 số n và cho biết số chẵn hay lẻ
- 13) Nhập vào một số và cho biết số này chia hết cho 2 hay không (3,4,5...)
- 14. Nhập vào một số N và tính tổng những số chẵn
- 15. Nhập vào một số N và tính tổng những số lẻ
- 16. Nhập vào một số N và tính tổng những chia hết cho 2
- 17. Nhập vào một số N và tính tổng những số không chia hết cho 2