

# IT001 - Nhập môn Lập Trình

## BÀI 02 – THUẬT TOÁN



- Sau khi học xong buổi học, sinh viên có khả năng:
  - Hiểu được khái niệm cơ bản như bài toán, thuật toán, các tiêu chuẩn của thuật toán, các phương pháp biểu diễn thuật toán.
  - Áp dụng lưu đồ (sơ đồ khối) hay mã giả để mô tả một số thuật toán đơn giản;
  - Diễn tả quá trình thực hiện thuật toán trên bộ dữ liệu cụ thể



1. Khái niệm về vấn đề/bài toán.
2. Các bước giải quyết vấn đề/bài toán bằng máy tính
3. Khái niệm về thuật toán
4. Sự cần thiết của thuật toán
5. Các tiêu chuẩn của thuật toán
6. Các phương pháp biểu diễn thuật toán.
7. Một số ví dụ về thuật toán
8. Lập bảng trên giấy để theo dõi hoạt động của một thuật toán
9. Độ phức tạp thuật toán



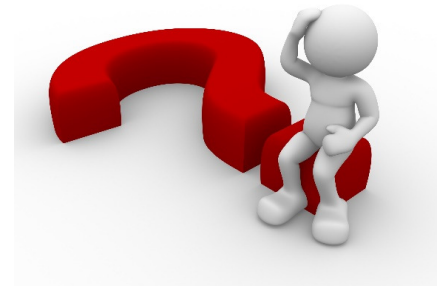
# 1. Khái niệm về vấn đề/bài toán

- “Bài toán” hay “Vấn đề”

- Vấn đề có nghĩa rộng hơn bài toán
- Bài toán là một loại vấn đề mà để giải quyết phải liên quan ít nhiều đến tính toán: bài toán trong vật lý, hóa học, xây dựng, kinh tế...

- Hai loại vấn đề

- Theorema: là vấn đề cần được khẳng định tính đúng sai.
- Problema: là vấn đề cần tìm được giải pháp để đạt được một mục tiêu xác định từ những điều kiện ban đầu nào đó.



# 1. Khái niệm về vấn đề/bài toán



- Biểu diễn vấn đề-bài toán

- $A \rightarrow B$
- A: Giả thiết, điều kiện ban đầu
- B: Kết luận, mục tiêu cần đạt



- Giải quyết vấn đề-bài toán

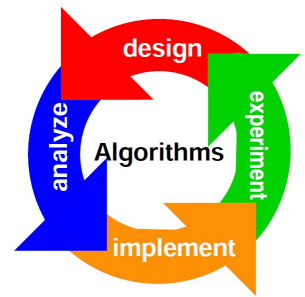
- Từ A dùng một số hữu hạn các bước suy luận có lý hoặc hành động thích hợp để đạt được B
- Trong Tin học, A là đầu vào, B là đầu ra



## 2. Các bước giải quyết vấn đề/bài toán bằng máy tính



- Máy tính không thể dùng để giải quyết các vấn đề liên quan đến hành động vật lý hoặc biểu thị cảm xúc
- Máy tính chỉ làm được những gì mà nó được bảo phải làm. Máy tính không thông minh, nó không thể tự phân tích vấn đề và đưa ra giải pháp.
- Lập trình viên là người phân tích vấn đề, tạo ra các chỉ dẫn để giải quyết vấn đề (chương trình), và máy tính sẽ thực hiện các chỉ dẫn đó
- Phương án giải quyết bài toán được gọi là thuật toán/giải thuật trong tính toán



### 3. Khái niệm về thuật toán

- Thuật toán - Algorithm

- Là **tập hợp** (dãy) **hữu hạn** các **chỉ thị** (hành động) được **định nghĩa rõ ràng** nhằm **giải quyết một bài toán cụ thể** nào đó.
- Thuật toán để giải một bài toán là một dãy hữu hạn các thao tác được sắp xếp theo một trình tự xác định sao cho sau khi thực hiện dãy thao tác đó, từ Input của bài toán, ta nhận được Output cần tìm.





## 4. Sự cần thiết của thuật toán

- Tại sao sử dụng máy tính để xử lý dữ liệu?
  - Nhanh hơn.
  - Nhiều hơn.
  - Giải quyết những bài toán mà con người không thể hoàn thành được.
- Làm sao đạt được những mục tiêu đó?
  - Nhờ vào sự tiến bộ của kỹ thuật: tăng cấu hình máy  $\Rightarrow$  chi phí cao 😞
  - Nhờ vào các thuật toán hiệu quả: thông minh và chi phí thấp 😊

***"Một máy tính siêu hạng vẫn không thể cứu vãn một thuật toán tồi!"***





## 5. Các tiêu chuẩn của thuật toán

- **Tính chính xác/đúng:**

- Quá trình tính toán hay các thao tác máy tính thực hiện là chính xác.
- Khi kết thúc, giải thuật phải cung cấp kết quả đúng đắn.

- **Tính phổ dụng/tổng quát:**

- Có thể áp dụng cho một lớp các bài toán có đầu vào tương tự nhau.

- **Tính kết thúc/hữu hạn:**

- Thuật toán phải dừng sau một số bước hữu hạn.



## 5. Các tiêu chuẩn của thuật toán

- **Tính rõ ràng/hiệu quả:**

- Các câu lệnh minh bạch được sắp xếp theo thứ tự nhất định.

- **Tính khách quan/xác định:**

- Được viết bởi nhiều người trên máy tính nhưng kết quả phải như nhau.
- Trong cùng một điều kiện hai bộ xử lý cùng thực hiện, thuật toán phải cho những kết quả giống nhau.

## 6. Các phương pháp biểu diễn thuật toán.



- a. Dùng ngôn ngữ tự nhiên.
- b. Dùng lưu đồ - sơ đồ khối (flowchart)
- c. Dùng mã giả (pseudocode)
- d. So sánh ưu nhược điểm của các phương pháp



## 6.a) Dùng ngôn ngữ tự nhiên



- Sử dụng ngôn ngữ thường ngày để liệt kê các bước của thuật toán.
- Phương pháp biểu diễn này không yêu cầu người viết thuật toán cũng như người đọc thuật toán phải nắm các quy tắc.
- Tuy vậy, cách biểu diễn này:
  - Thường dài dòng,
  - Không thể hiện rõ cấu trúc của thuật toán,
  - Đôi lúc gây hiểu lầm hoặc khó hiểu cho người đọc.
- Gần như không có một quy tắc cố định nào trong việc thể hiện thuật toán bằng ngôn ngữ tự nhiên.

## 6.a) Dùng ngôn ngữ tự nhiên



•VD: Giải phương trình  $ax+b=0$

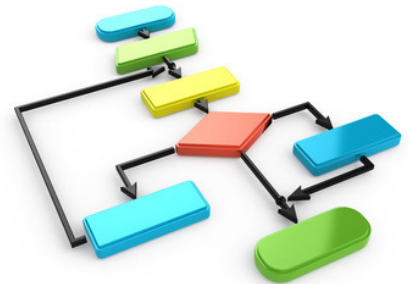
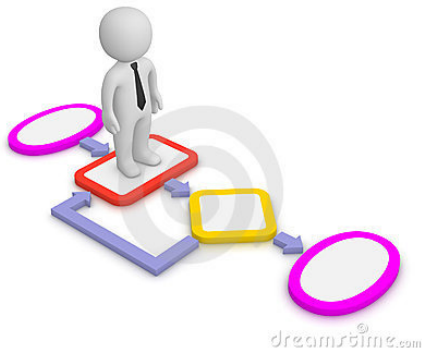
Đầu vào:  $a, b$  thuộc  $\mathbb{R}$

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

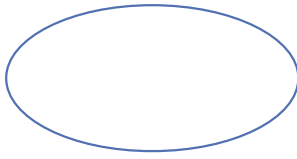
1. Nhập 2 số thực  $a$  và  $b$ .
2. Nếu  $a = 0$  thì
  - 2.1. Nếu  $b = 0$  thì
    - 2.1.1. Phương trình vô số nghiệm
    - 2.1.2. Kết thúc thuật toán.
  - 2.2. Ngược lại
    - 2.2.1. Phương trình vô nghiệm.
    - 2.2.2. Kết thúc thuật toán.
3. Ngược lại
  - 3.1. Phương trình có nghiệm.
  - 3.2. Giá trị của nghiệm đó là  $x = -b/a$
  - 3.3. Kết thúc thuật toán.

## 6.b) Dùng lưu đồ - 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.
- Phương pháp lưu đồ thường được dùng trong những thuật toán có tính rắc rối, khó theo dõi được quá trình xử lý.



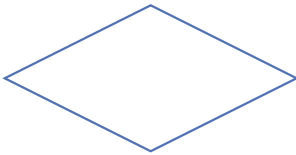
## 6.b) Dùng lưu đồ - sơ đồ khối



Khối giới hạn  
Chỉ thị bắt đầu và kết thúc.



Khối vào ra (input/output)  
Nhập/Xuất dữ liệu.



Khối lựa chọn (decision)  
Tùy điều kiện sẽ rẽ nhánh.



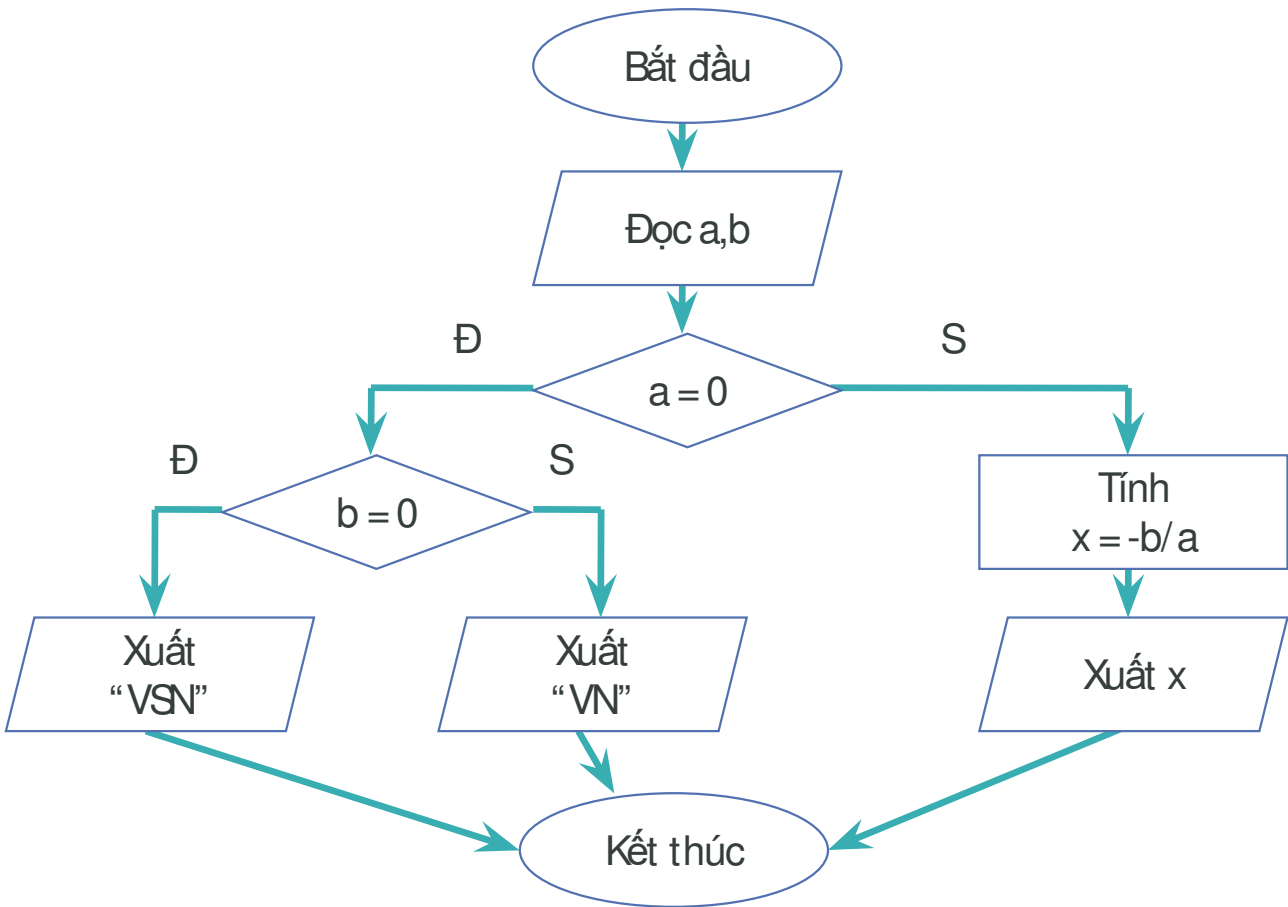
Khối thao tác (process)  
Ghi thao tác cần thực hiện.



Đường đi (route)  
Chỉ hướng thao tác tiếp theo.

## 6.b) Dừng lưu đồ - sơ đồ khối

- VD: Giải phương trình  $ax+b=0$







## 6.c) Dùng mã giả

- Ngôn ngữ tựa ngôn ngữ lập trình:
  - Dùng cấu trúc chuẩn hóa, chẳng hạn tựa Pascal, C.
  - Dùng các ký hiệu toán học, biến, hàm.
- Ưu điểm:
  - Dễ công kênh hơn lưu đồ khối.
- Nhược điểm:
  - Không trực quan bằng lưu đồ khối.

## 6.c) Dùng mã giả



- VD: Giải phương trình  $ax+b=0$

Đầu vào:  $a, b$  thuộc  $\mathbb{R}$

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

If  $a = 0$  Then

Begin

    If  $b = 0$  Then

        Xuất "Phương trình vô số nghiệm"

    Else

        Xuất "Phương trình vô nghiệm"

End

Else

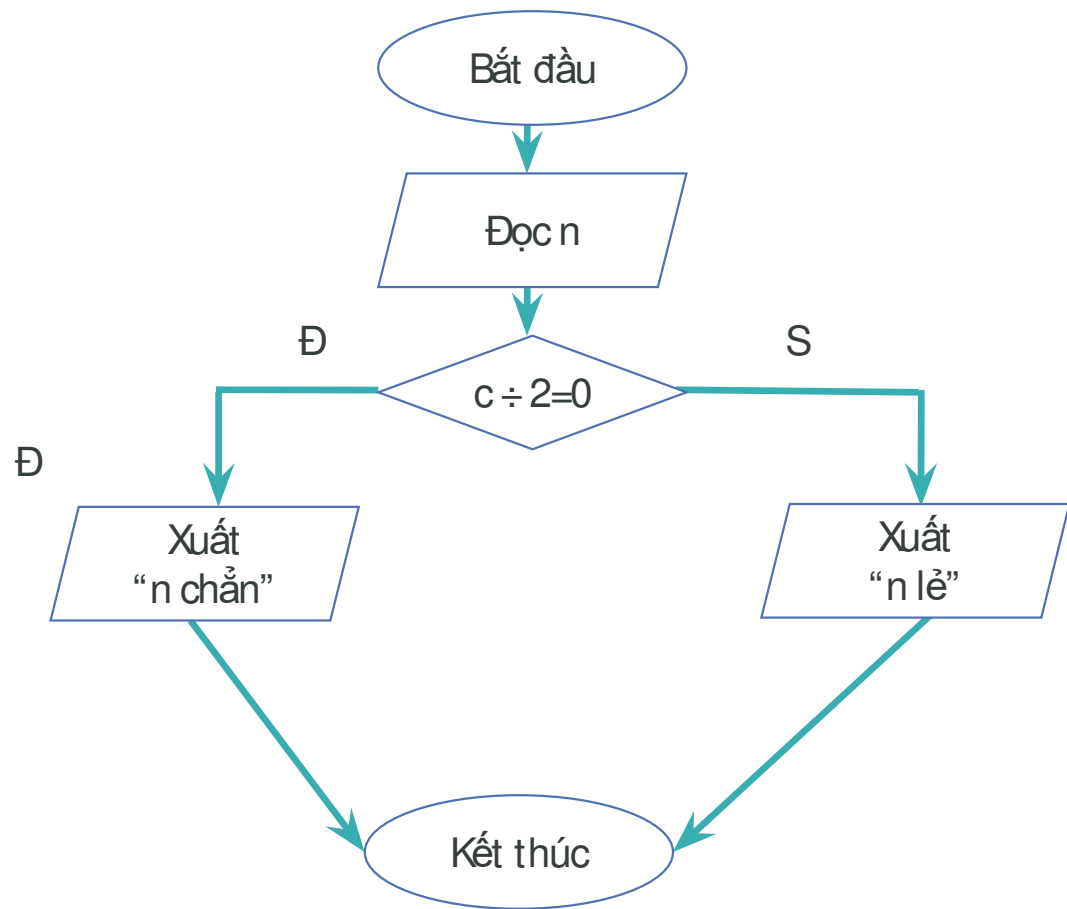
    Xuất "Phương trình có nghiệm  $x = -b/a$ "



## 7. Một số ví dụ về thuật toán

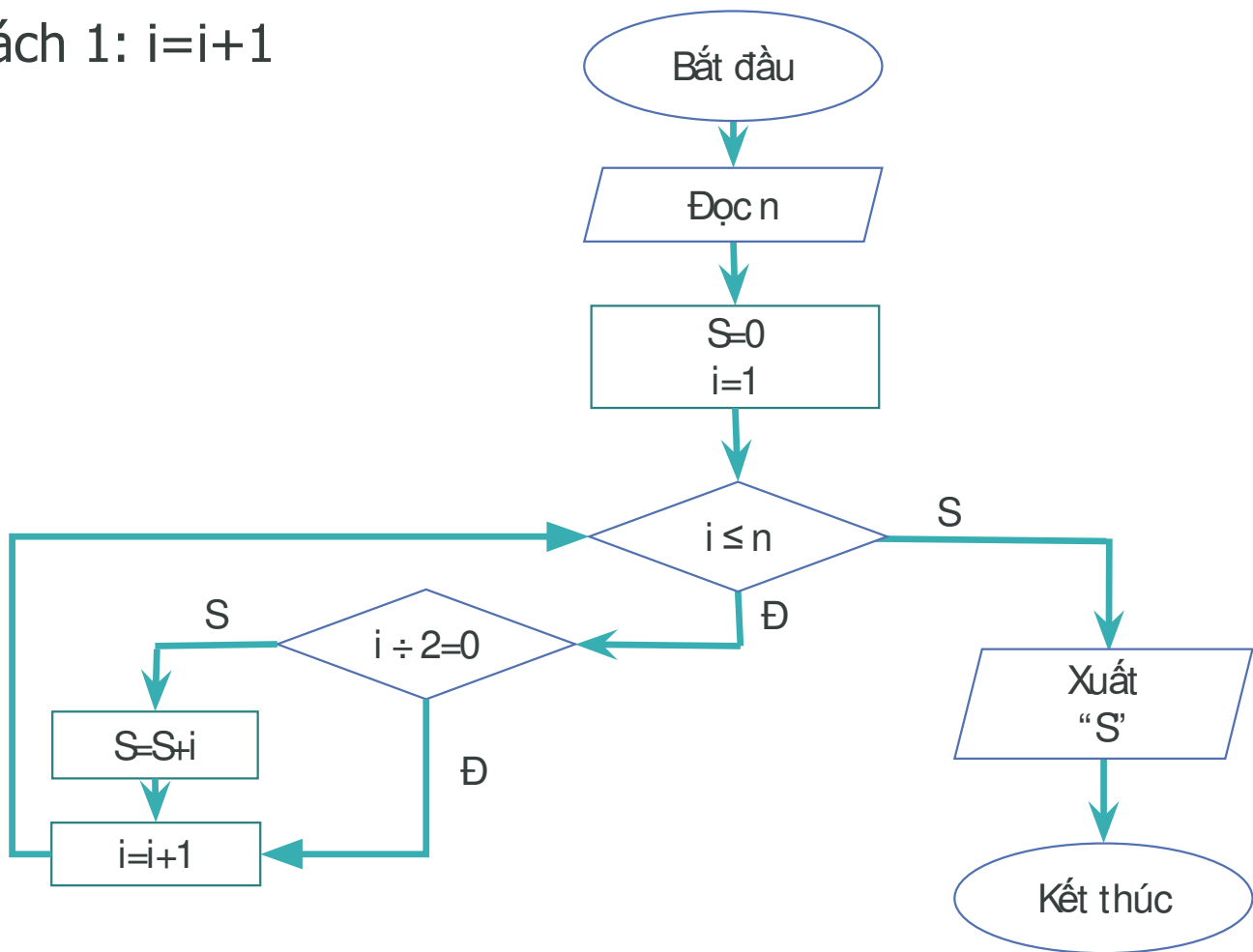
- Ví dụ 1: Vẽ lưu đồ thuật toán Kiểm tra tính chẵn lẻ của một số nguyên
- Ví dụ 2: Vẽ lưu đồ thuật toán Tính tổng các số nguyên dương lẻ từ 1 đến  $n$
- Ví dụ 3: Vẽ lưu đồ thuật toán Tìm nghiệm của phương trình bậc hai một ẩn
- Ví dụ 4: Vẽ lưu đồ thuật toán Liệt kê tất cả ước số của số nguyên dương  $n$

## 7.a) Ví dụ 2: Kiểm tra tính chẵn lẻ



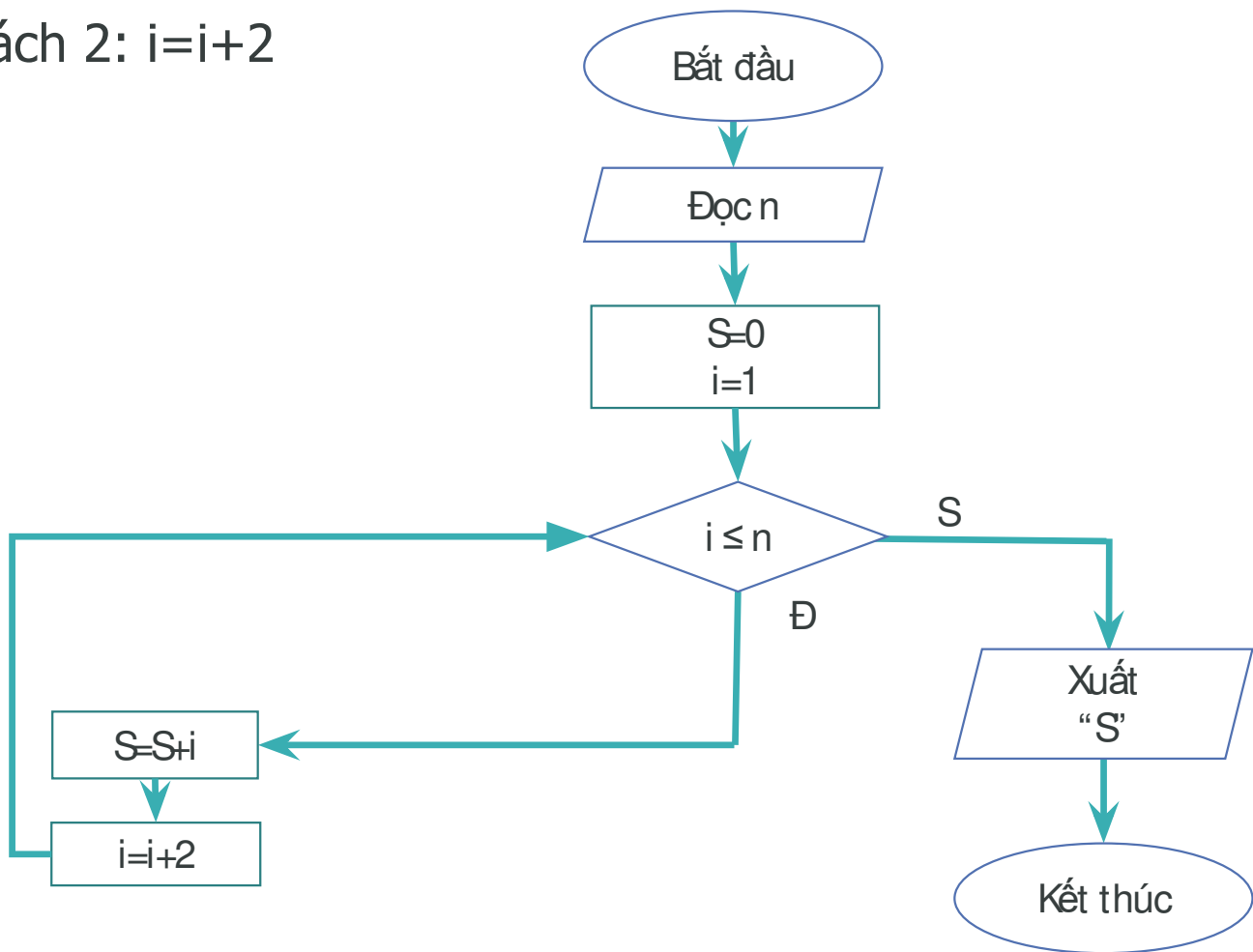
## 7.b) Ví dụ 2: Tổng các số nguyên dương lẻ $1 \rightarrow n$

- Cách 1:  $i=i+1$



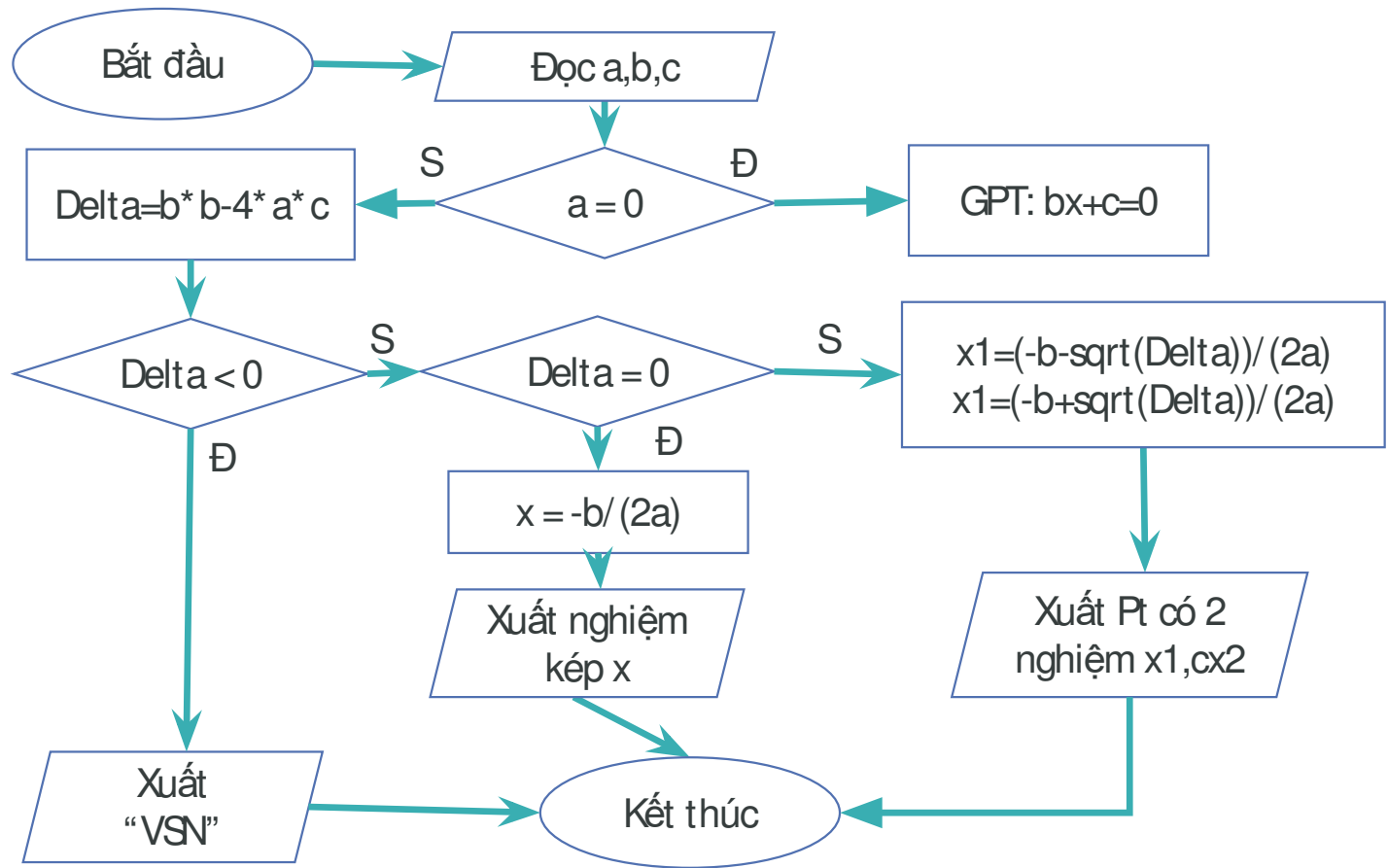
## 7.b) Ví dụ 2: Tổng các số nguyên dương lẻ $1 \rightarrow n$

• Cách 2:  $i=i+2$



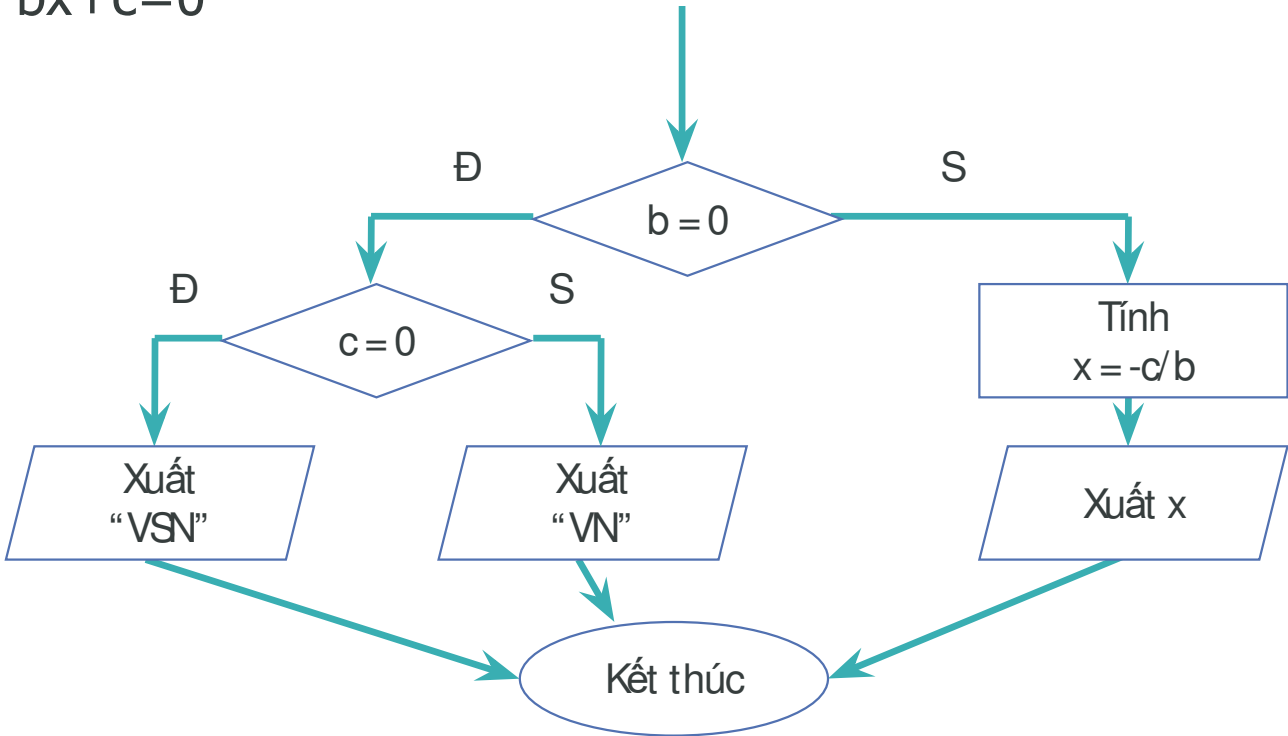
## 7.c) Ví dụ 3: Giải phương trình bậc 2

- Giải pt:  $ax^2+bx+c=0$



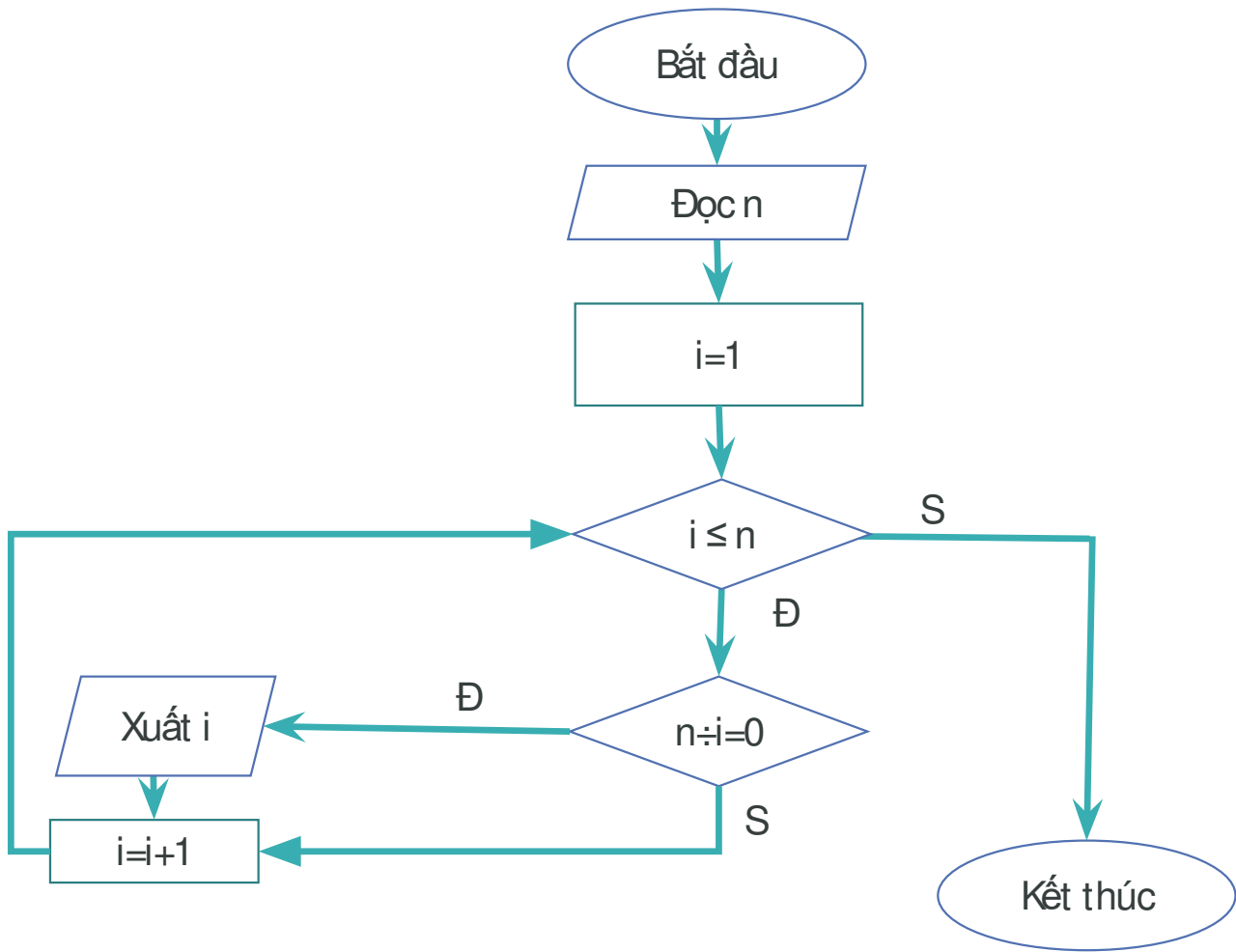
## 7.c) Ví dụ 3: Giải phương trình bậc 2

• GPT:  $bx+c=0$



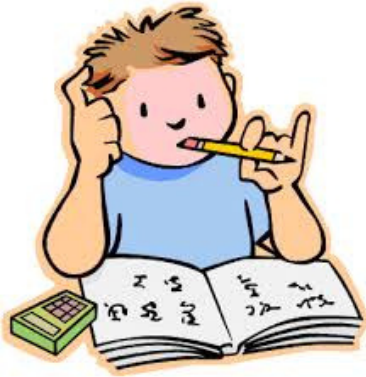


## 7.d) Ví dụ 4: Liệt kê các ước số của n



## 8. Lập bảng trên giấy để theo dõi hoạt động của một thuật toán

- Chuẩn bị các bộ dữ liệu kiểm thử: dữ liệu nhập và kết quả mong đợi
- Chạy thử, ghi nhận kết quả, đánh giá đúng sai





1. Giải phương trình bậc nhất  $ax+b=0$
2. Kiểm tra một số nguyên  $n$  là số nguyên tố không?
3. Tính giá trị biểu thức:  $S = 1+2+\dots+n$
4. Nhập vào số nguyên dương  $n$ . Tính tổng các chữ số của số đó.
5. Tìm số lớn nhất trong 3 số





## 9. Độ phức tạp thuật toán

### 1. Tính hiệu quả của giải thuật

Để giải một bài toán có thể có nhiều giải thuật khác nhau. Cần lựa chọn một giải thuật tốt theo hai tiêu chuẩn:

- Đơn giản, dễ hiểu, dễ lập trình.
- Thời gian thực hiện nhanh, dùng ít tài nguyên máy tính.

Tiêu chuẩn 2 là tính hiệu quả của giải thuật. Đánh giá độ phức tạp của giải thuật là đánh giá thời gian thực hiện giải thuật đó.



## 9. Độ phức tạp thuật toán

### 2. *Đánh giá thời gian thực hiện giải thuật*

Thời gian thực hiện giải thuật phụ thuộc: Ngôn ngữ lập trình, chương trình dịch, hệ điều hành, phần cứng của máy,...

Mặt khác phải lập trình mới đo được thời gian thực hiện giải thuật.

Cần có cách đánh giá khác sao cho:

- Không phụ thuộc máy, ngôn ngữ lập trình, chương trình dịch.
- Không cần triển khai chương trình thực hiện giải thuật.
- Chỉ dựa vào phân tích bản thân giải thuật.

Tổng số phép toán sơ cấp cần thiết để thực hiện giải thuật là cách làm đáp ứng được các yêu cầu trên.

## 9. Độ phức tạp thuật toán



- Đánh giá giá thuật toán theo hướng tiệm xấp xỉ tiệm cận qua các khái niệm  $O()$ .
- Ưu điểm: Ít phụ thuộc môi trường cũng như phần cứng hơn.
- Nhược điểm: Phức tạp.
- Các trường hợp độ phức tạp quan tâm:
  - Trường hợp tốt nhất (phân tích chính xác)
  - Trường hợp xấu nhất (phân tích chính xác)
  - Trường hợp trung bình (mang tính dự đoán)



## 9. Độ phức tạp thuật toán

### • Sự Phân Lớp Theo Độ Phức Tạp Của Thuật Toán

#### • Sử dụng ký hiệu BigO

- Hằng số :  $O(c)$
- $\log N$  :  $O(\log N)$
- $N$  :  $O(N)$
- $N \log N$  :  $O(N \log N)$
- $N^2$  :  $O(N^2)$
- $N^3$  :  $O(N^3)$
- $2^N$  :  $O(2^N)$
- $N!$  :  $O(N!)$



Độ phức tạp tăng dần