

NHẬP MÔN LẬP TRÌNH

CHƯƠNG 2: GIỚI THIỆU VỀ THUẬT TOÁN

ThS. Nguyễn Thị Ngọc Diễm
diemntn@uit.edu.vn



- 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



- Vấn đề, bài toán (Problem)
- Giải quyết vấn đề -> giải pháp (solution)
- Mục tiêu lập trình: Xây dựng chương trình máy tính để giải quyết vấn đề thay cho con người
- Diễn đạt?
 - Con người diễn đạt bằng NNTN
 - Máy tính thì thế nào?



1. Khái niệm về vấn đề/ bài toán

“Bài toán” hay “Vấn đề”

- Vấn đề là gì?

- Có 1 số người quan niệm rằng : “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

Mô hình vấn đề: $A \rightarrow B$

- A: giả thiết, điều kiện ban đầu (đầu vào - input)
 - B: kết luận, mục tiêu cần đạt (đầu ra - output)
- => Suy luận hay giải pháp cần xác định = một số hữu hạn bước

Bài tập: Phát biểu vấn đề (define a problem)???



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.

Hỏi: Quy trình giải quyết vấn đề bằng máy tính?

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.
- Mở rộng (máy tính): Là **tập hợp** (dãy, bước) **hữu hạn** các **chỉ thị** (hành động) được **định nghĩa rõ ràng** và **có thể thực thi** nhằm giải quyết một bài toán cụ thể nào đó. Quá trình hành động theo các bước này **phải dừng** và cho được **kết quả như mong muốn**.



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 -> 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 rõ ràng/xác định** = không mập mờ + có thể thực thi được

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

- **Tính kết thúc/hữu hạn:**

- Các câu lệnh minh bạch được sắp xếp theo thứ tự nhất định.
- Thuật toán phải dừng sau một số bước hữu hạn

5. Các tiêu chuẩn của thuật toán



- **Tính phổ dụng/tổng quát:**

- Có thể áp dụng cho một lớp các bài toán có đầu vào tương tự nhau.
- Phải áp dụng được cho mọi trường hợp của bài 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



- 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.



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.



- Ngôn ngữ tựa ngôn ngữ lập trình:
 - Vay mượn các cú pháp của một NNLT nào đó (C, Pascal, ...), vẫn dung một phần ngôn ngữ tự nhiên.
 - Dùng các ký hiệu toán học, biến, hàm, cú pháp trong NNLT, giúp người cài đặt dễ dàng nắm bắt nội dung thuật toán
- Ưu điểm:
 - Đỡ công kênh hơn lưu đồ khối.
- Nhược điểm:
 - Không trực quan bằng lưu đồ khối.



- Vay mượn ngôn ngữ nào đó (ví dụ Pascal) để biểu diễn thuật toán.

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  $x = -b/a$ "
```



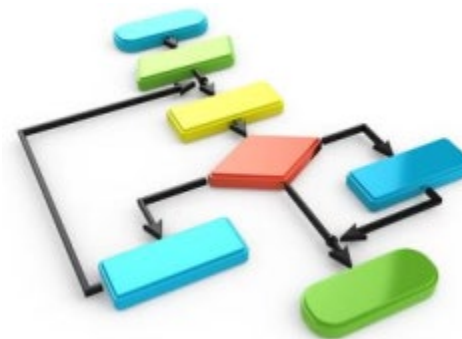

Đầu vào: a, b thuộc \mathbb{R}

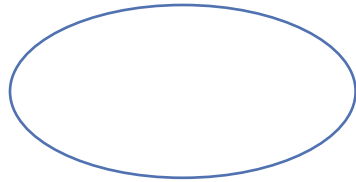
Đầu ra: nghiệm phương trình $ax + b = 0$

```
START
  READ a, b
  IF a=0 THEN
    IF b=0 THEN
      WRITE "VSN"
    ELSE
      WRITE "VN"
    ENDIF
  ELSE
     $x = -b/a$ 
    WRITE x
  ENDIF
END
```



- 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.

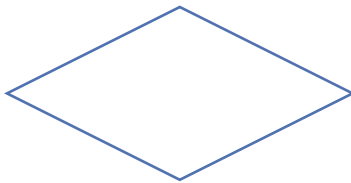




Khối giới hạn
Chỉ thị bắt đầu và kết thúc.



Khối vào ra
Nhập/Xuất dữ liệu.



Khối lựa chọn
Tùy điều kiện sẽ rẽ nhánh.



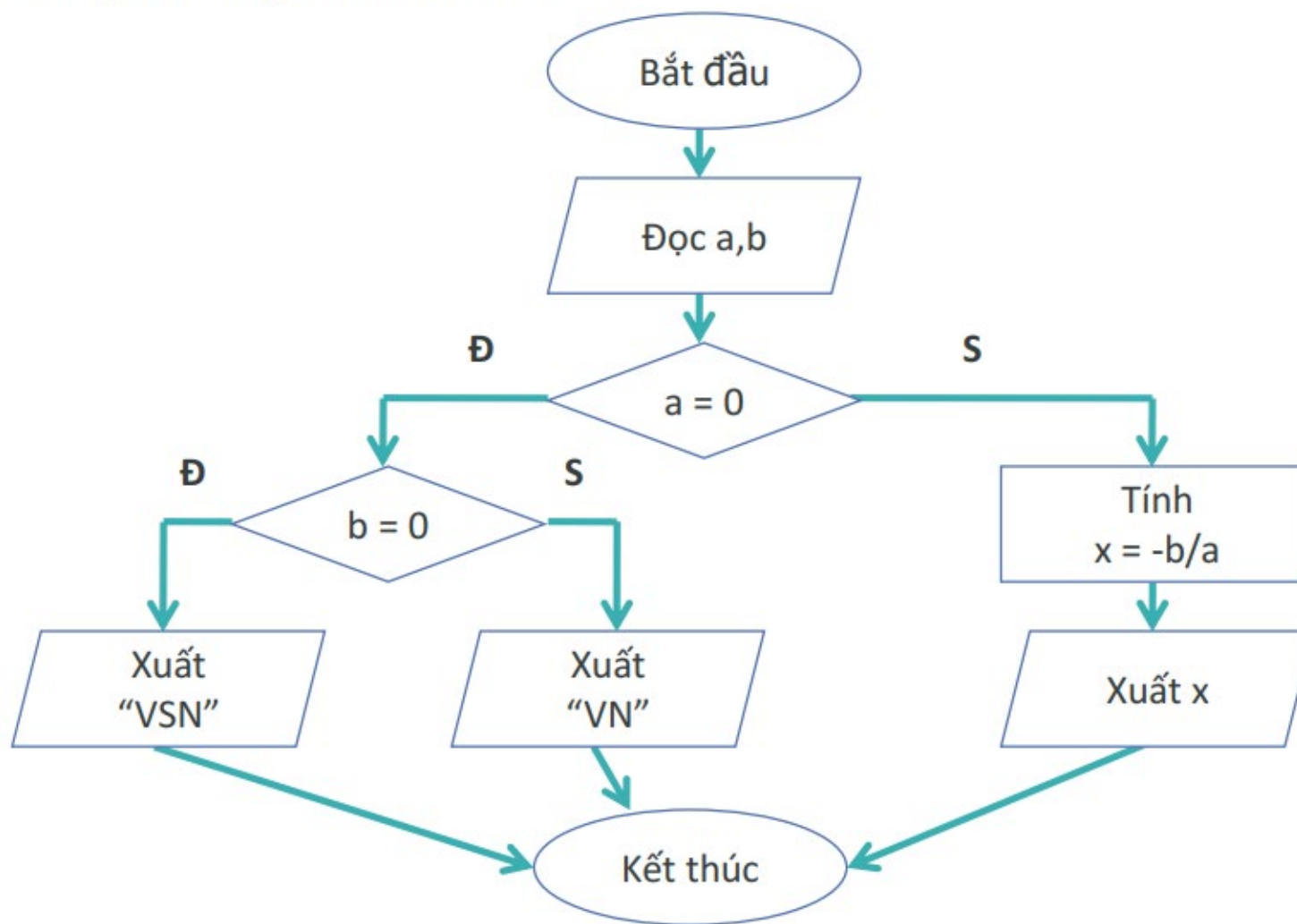
Khối thao tác
Ghi thao tác cần thực hiện.



Đường đi
Chỉ hướng thao tác tiếp theo.



- VD: Giải phương trình $ax+b=0$





```
#include <stdio.h>
#include <conio.h>

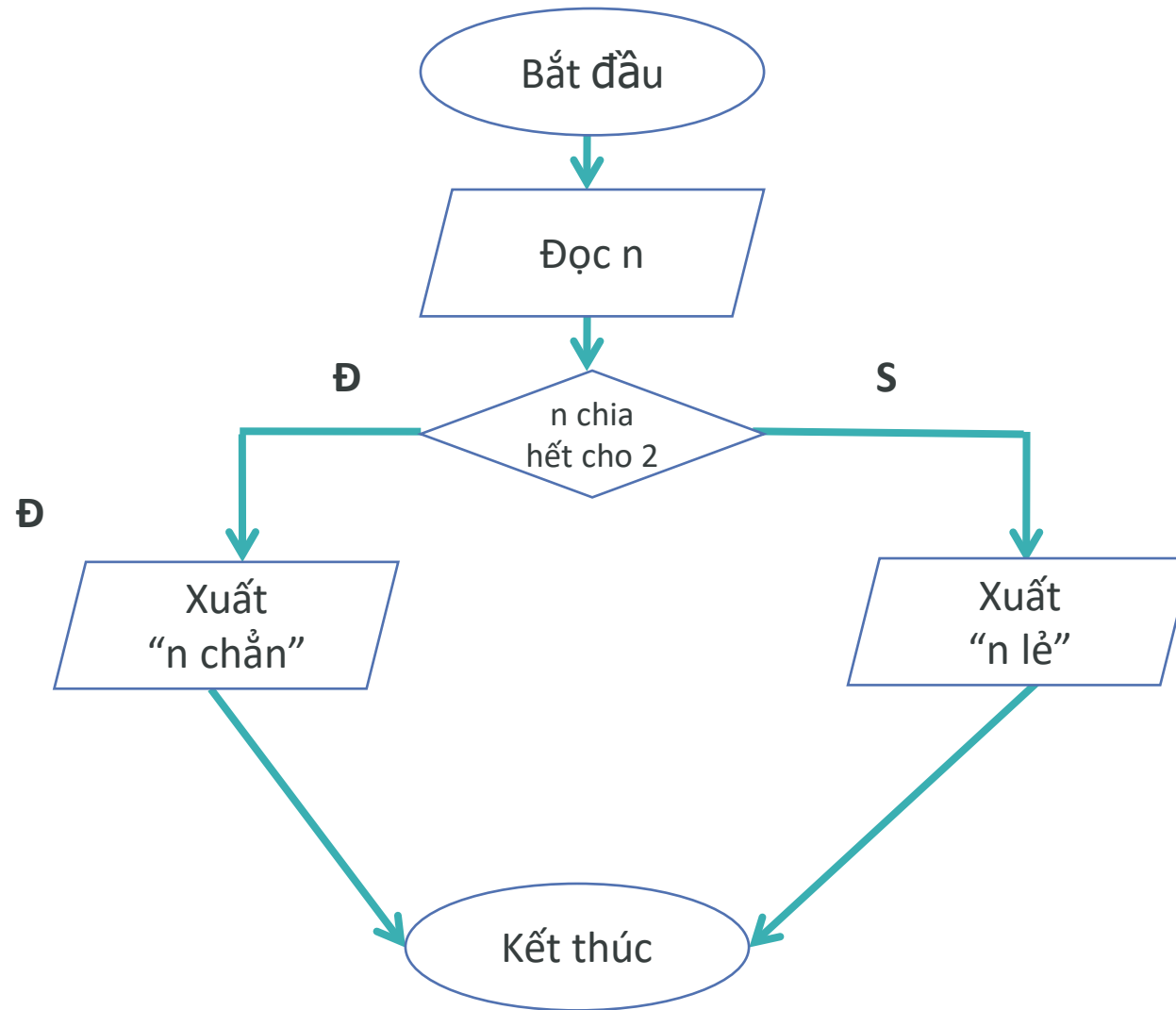
int main() {
    int a, b;
    printf("Nhap a, b: ");
    scanf("%d%d", &a, &b);
    if (a == 0)
        if (b == 0)
            printf("Phương trình VSN");
        else
            printf("Phương trình VN");
    else
        printf("x = %.2f", -float(b)/a);
}
```



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 .

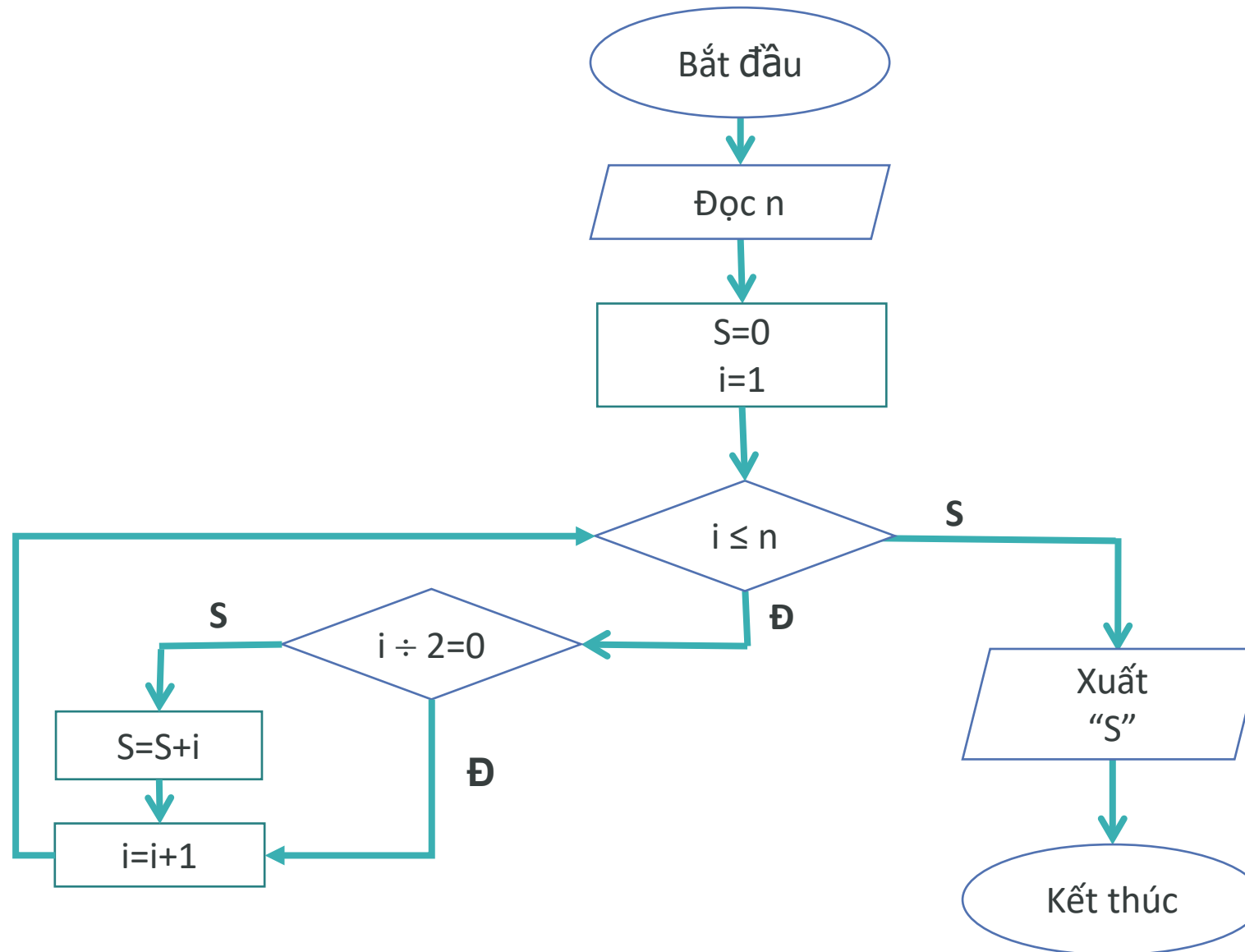
Ví dụ 1: Kiểm tra tính chẵn lẻ của một số nguyên



Ví dụ 2: Tính tổng các số nguyên dương lẻ từ 1 -> n



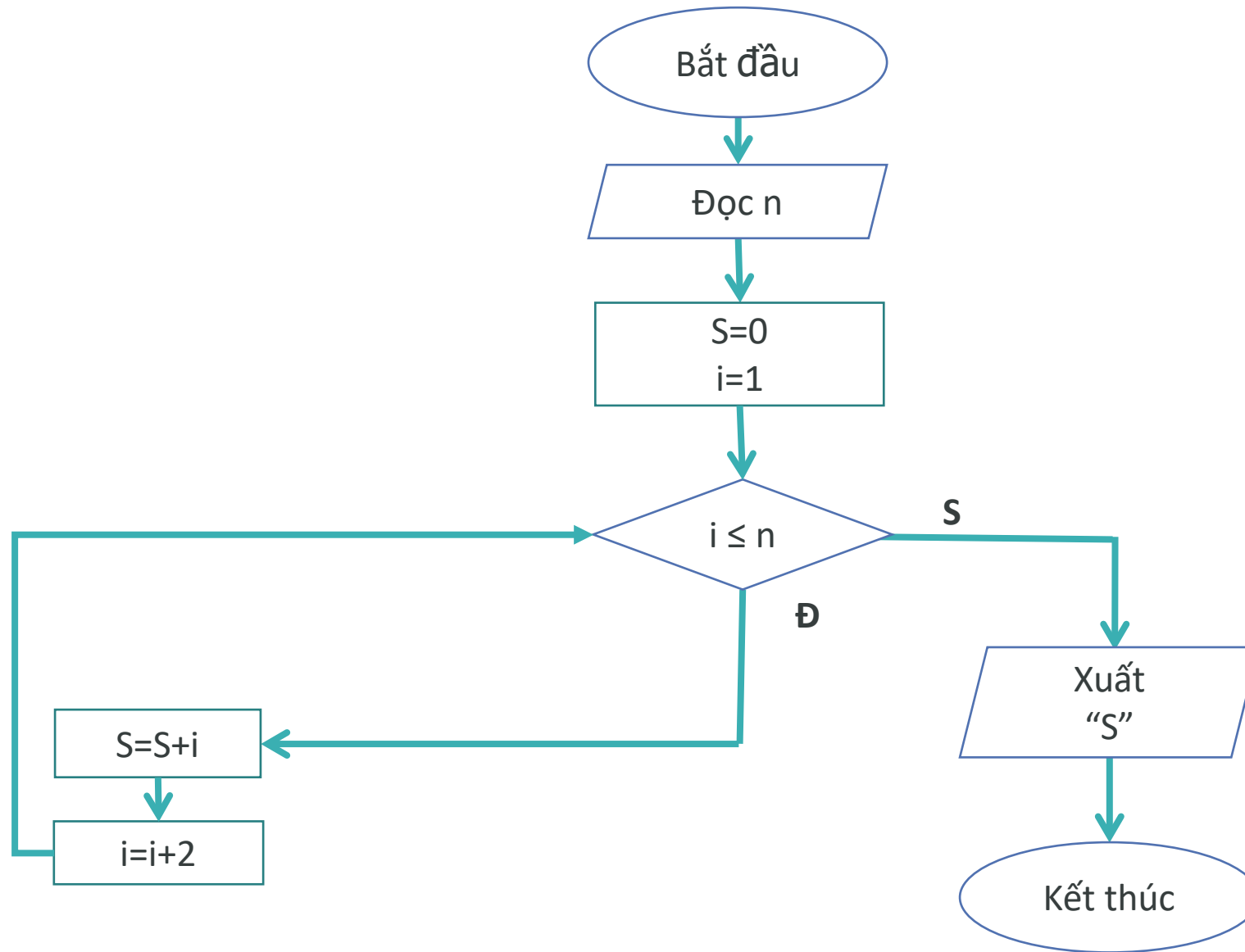
- Cách 1: $i=i+1$



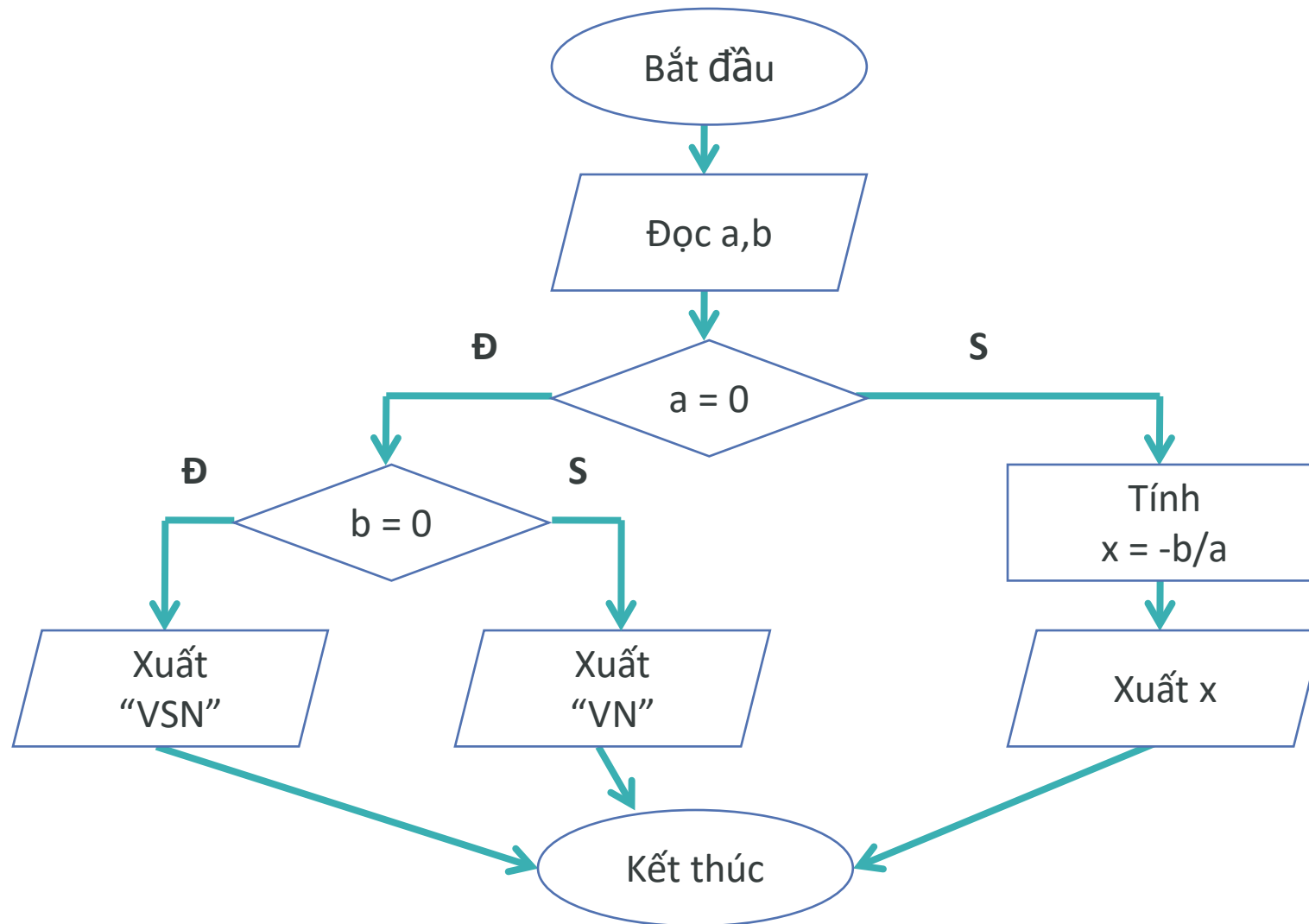
Ví dụ 2: Tính tổng các số nguyên dương lẻ từ 1 -> n



- Cách 2: $i=i+2$



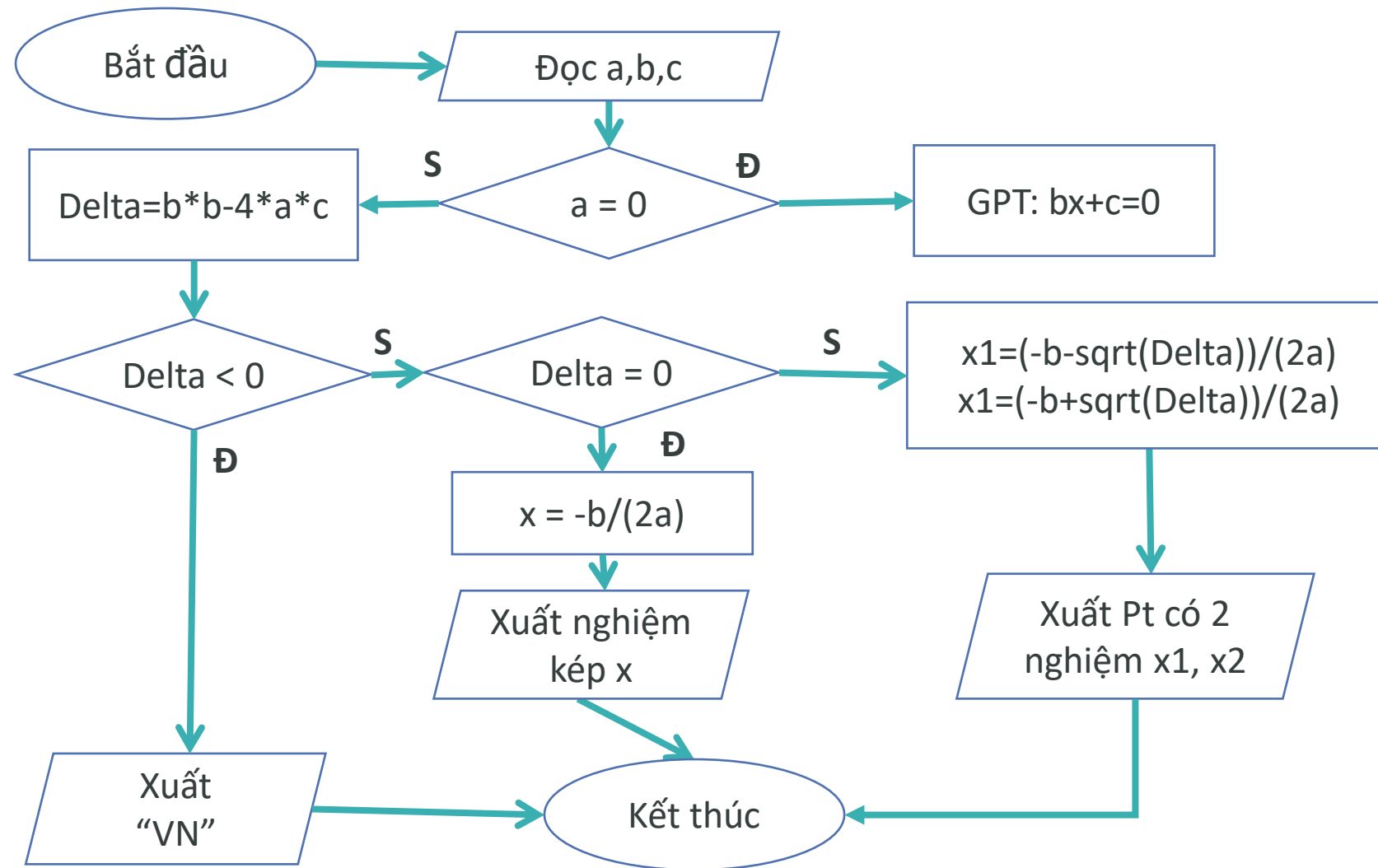
Ví dụ 3: Tìm nghiệm của phương trình bậc nhất một ẩn





Ví dụ 3: Tìm nghiệm của phương trình bậc hai một ẩn

- Giải pt: $ax^2+bx+c=0$



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





9. Độ phức tạp thuật toán

- 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 3 tiêu chuẩn:
 - Đơn giản, dễ hiểu, dễ lập trình.
 - Tính đúng đắn
 - Tính hiệu quả: Thời gian thực hiện nhanh, dùng ít tài nguyên máy tính
- Đánh giá độ phức tạp của giải thuật là đánh giá thời gian thực hiện giải thuật đó.

ĐÁNH GIÁ BẰNG CÁCH NÀO?



1. Thuật toán là gì? Trình bày các tính chất quan trọng của một thuật toán?
2. Các bước xây dựng chương trình?
3. Các cách biểu diễn thuật toán? Ưu và khuyết điểm của từng phương pháp? Cho ví dụ minh họa.



Vẽ lưu đồ thuật toán cho các bài toán sau:

1. Viết chương trình nhập và xuất ra tên của chính bạn.
2. Viết chương trình nhập vào một số nguyên. In ra số đối xứng. VD: Nhập 1, in ra -1; nhập -2, in ra 2.
3. Viết chương trình chuyển độ F sang độ C và ngược lại, biết công thức chuyển là:
$$^{\circ}\text{F} = (^{\circ}\text{C} \times 1.8) + 32$$
$$^{\circ}\text{C} = (^{\circ}\text{F} - 32) / 1.8$$
4. Liệt kê các số chẵn từ 1 đến n.
5. Nhập vào 2 số nguyên a, b. Tính tổng hiệu tích thương.
6. Viết chương trình nhập vào ba số nguyên. Tìm số lớn nhất và số nhỏ nhất.



6. Tìm ước số lớn nhất và nhỏ nhất của một số nguyên dương n mà không phải là chính số đó và số 1.

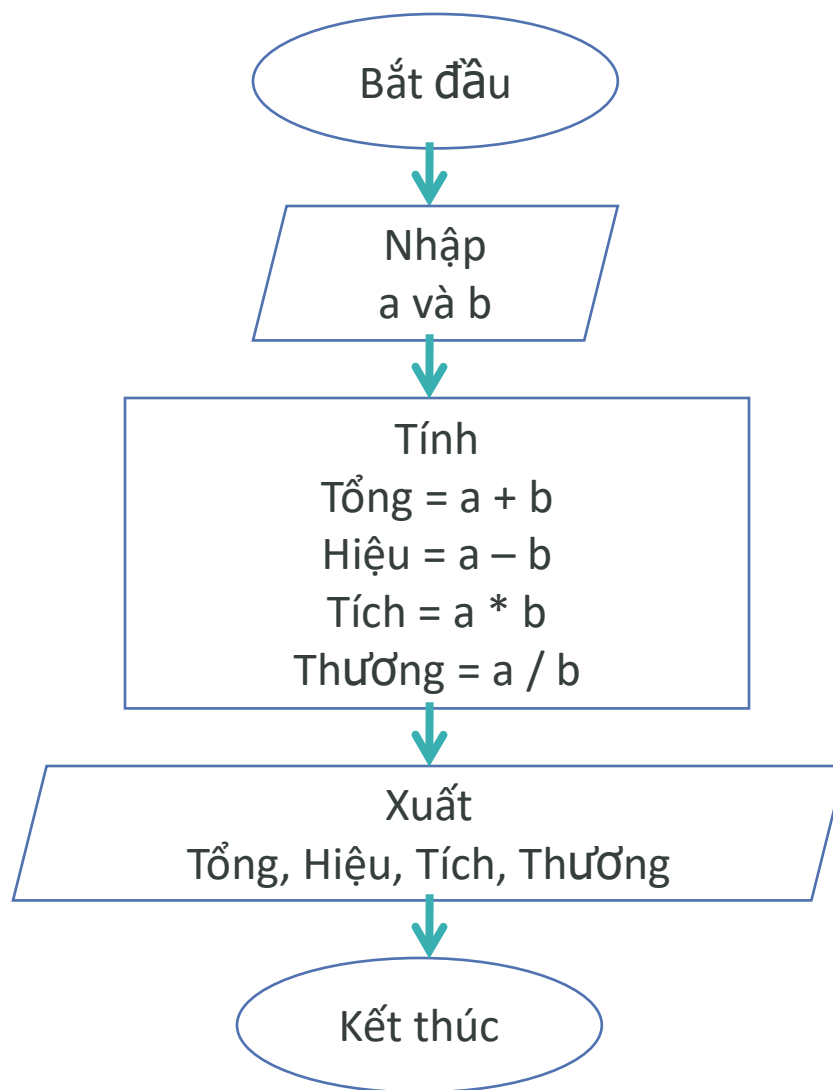
7. Tính giá trị biểu thức:

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$
$$S = 1 + 2^2 + 3^3 + \dots + n^n$$

8. Liệt kê tất cả ước số của số nguyên dương n .

9. Nhập vào số nguyên dương n . Đếm số chữ số của số đó.

10. Nhập vào số xe (gồm 4 chữ số) của bạn. Cho biết số xe của bạn được mấy nút?





Chúc các em học tốt!

