

Cơ sở lập trình

Bài 2. Cơ bản Lập trình máy tính (3 tiết)

Biên soạn TS. Trần Minh Thái

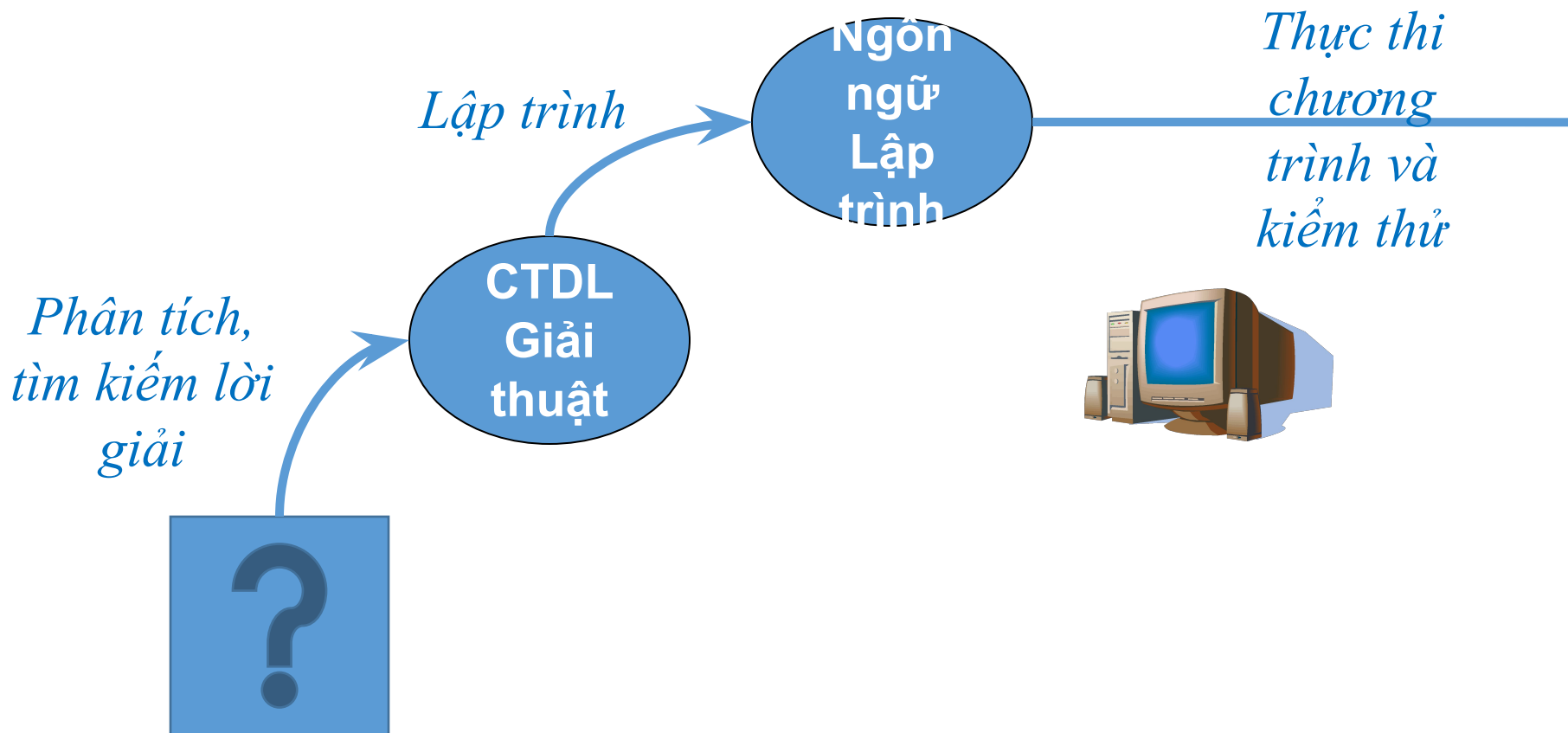
Giảng viên: Lê Thị Minh Nguyễn

Email: nguyenltm@huflit.edu.vn

Mục tiêu

- Hiểu được quá trình tổng quát để viết một chương trình trên máy tính
- Nắm được khái niệm cơ bản về tổ chức dữ liệu và giải thuật
- Mô tả giải thuật bằng Flowchart
- Ngôn ngữ lập trình

Chương trình máy tính?



Các đặc điểm cần có của chương trình

- Đúng đắn, chính xác (*correctness*)
- Chắc chắn (*robustness*)
- Thân thiện (*user friendliness*)
- Khả năng thích nghi (*adapability*): Chương trình có khả năng để phát triển tiến hóa theo yêu cầu
- Tính tái sử dụng (*reuseability*): Chương trình có thể dùng để làm một phần trong một chương trình lớn khác

Các đặc điểm cần có của chương trình

- Tính hiệu quả (*efficiency*)
- Tính khả chuyển (*porability*): Khả năng chuyển đổi giữa các môi trường
- Tính an toàn (*security*)
- Tính dừng (*halt*)

Các ngôn ngữ lập trình

- Fortran
- Pascal
- Java
- C
- C++
- C#
- F#
- VB.Net
- Python
-

Các môi trường hỗ trợ lập trình (IDE)

- Borland C++
- Dev C
- Free C
- Microsoft Visual C++
- Jbuidar
- Visual .Net
- ...

Xác định bài toán

Input -> Process -> Output



- **Input:** Giả thiết, thông tin được cung cấp?
- **Process:** Giải quyết vấn đề gì? Giải quyết như thế nào?
- **Output:** Đạt được những yêu cầu nào?

Kết quả có đưa vào xử lý tiếp hay không???

Xác định cấu trúc dữ liệu

- Phải biểu diễn đầy đủ được thông tin nhập và xuất của bài toán
- Phù hợp với giải thuật (cách giải) được chọn
- Có thể cài đặt được trên ngôn ngữ lập trình cụ thể

Tìm giải thuật

- Tập hợp hữu hạn của các chỉ thị hay phương cách được định nghĩa rõ ràng cho việc hoàn tất một số sự việc từ một trạng thái ban đầu cho trước; khi các chỉ thị này được áp dụng triệt để thì sẽ dẫn đến kết quả sau cùng như đã dự đoán
- Có thể là công thức/ các bước cần phải thực hiện

Tính chất quan trọng của giải thuật

- **Tính chính xác:** để đảm bảo kết quả tính toán hay các thao tác mà máy tính thực hiện được là chính xác
- **Tính rõ ràng:** giải thuật phải được thể hiện bằng các câu lệnh minh bạch; các câu lệnh được sắp xếp theo thứ tự nhất định
- **Tính khách quan:** Một giải thuật dù được viết bởi nhiều người trên nhiều máy tính vẫn phải cho kết quả như nhau

Tính chất quan trọng của giải thuật

- **Tính phổ dụng:** giải thuật không chỉ áp dụng cho một bài toán nhất định mà có thể áp dụng cho một lớp các bài toán có đầu vào tương tự nhau
- **Tính kết thúc:** giải thuật phải gồm một số hữu hạn các bước tính toán

Các loại giải thuật

- Tìm kiếm
- Sắp xếp
- Đệ quy
- Xử lý chuỗi ký tự
- Xử lý file
- Đồ họa
- Đồ thị
- v.v...

Các phương pháp chính mô tả giải thuật

- Mã tự nhiên
- Pseudocode (mã giả)
- Flowchart (lưu đồ)

Khi mô tả giải thuật phải bao gồm:

- *Input - Đầu vào*
- *Output - Đầu ra / kết quả*
- *Process - Mô tả xử lý của giải thuật*

Ví dụ: Tìm ước số chung lớn nhất (USCLN) của 2 số nguyên dương a và b

- **Đầu vào:** 2 số nguyên dương a và b
- **Đầu ra:** USCLN của a và b

Cách 1: Dùng mã tự nhiên

Bước 1: Nếu $a = b$ thì kết luận a là USCLN và kết thúc

Bước 2: Nếu $a > b$ thì $a = a - b$;

Ngược lại thì $b = b - a$;

Bước 3: Quay trở lại Bước 1

Ví dụ: Tìm ước số chung lớn nhất (USCLN) của 2 số nguyên dương a và b

Cách 2: Dùng mã giả (Pseudocode)

WHILE $a \neq b$ DO

IF $a > b$ THEN

$a = a - b$

ELSE

$b = b - a$

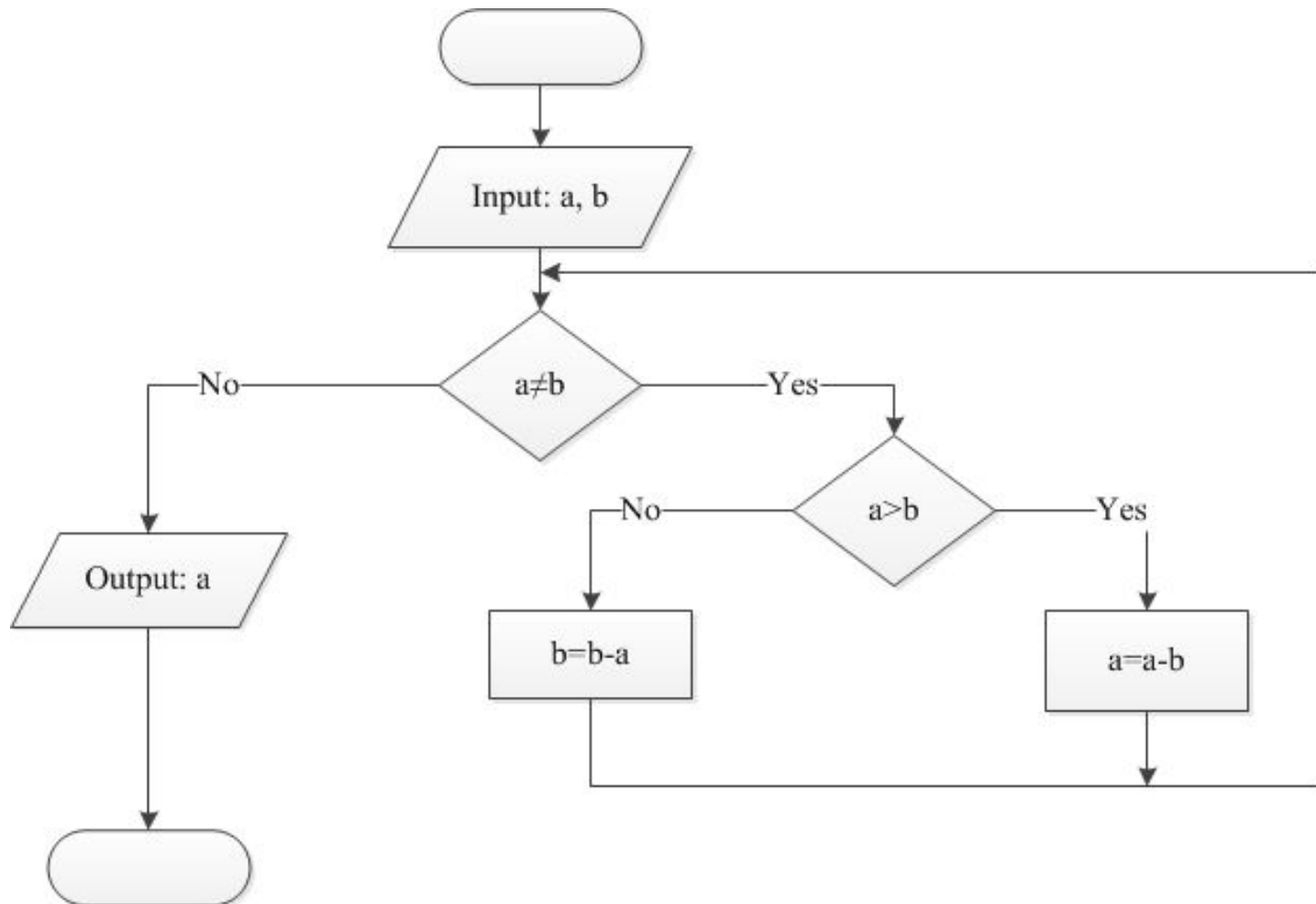
ENDIF

ENDWHILE

Là sự lai ghép giữa ngôn ngữ lập trình và ngôn ngữ tự nhiên

Ví dụ: Tìm ước số chung lớn nhất (USCLN) của 2 số nguyên dương a và b

Cách 3: Dùng lưu đồ (flowchart)



Mô tả giải thuật bằng pseudocode

- Dễ hiểu, không chi tiết đến các kỹ thuật lập trình
- Ở cấp độ hết sức tổng quát: gần ngôn ngữ tự nhiên
- Hoặc chi tiết: như dùng ngôn ngữ tựa Pascal, C, ...

IF <Điều kiện> THEN ...ENDIF

IF <Điều kiện> THEN ... ELSE ... ENDIF

WHILE <Điều kiện> DO ... ENDWHILE

DO ... UNTIL <Điều kiện>

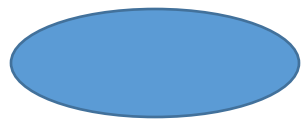
DISPLAY ...

RETURN ...

Mô tả giải thuật bằng lưu đồ (flowchart)

- Lưu đồ thuật toán là công cụ dùng để **biểu diễn thuật toán**, việc mô tả **nhập** (input), dữ liệu **xuất** (output) và luồng xử lý thông qua các **ký hiệu hình học**
- **Phương pháp duyệt lưu đồ**
 - Duyệt từ trên xuống
 - Duyệt từ trái sang phải

Các ký hiệu flowchart



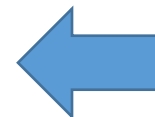
Bắt đầu/ kết thúc



Rẽ nhánh



Nhập/ Xuất



Giá trị trả về



Luồng xử lý



Khối xử lý



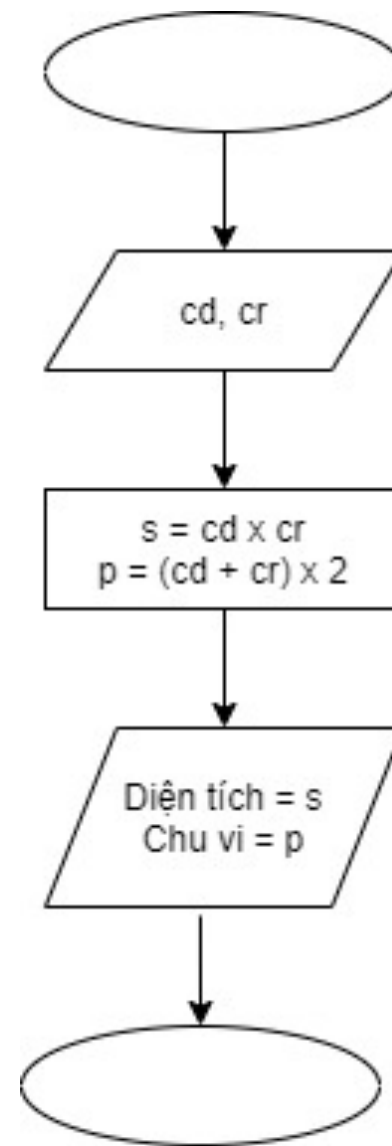
Điểm nối

VD Mô tả lưu đồ giải thuật

1. Tính diện tích và chu vi hình chữ nhật
2. Tính giá trị tuyệt đối của số nguyên n
3. Giải và biện luận phương trình bậc I: $ax+b=0$

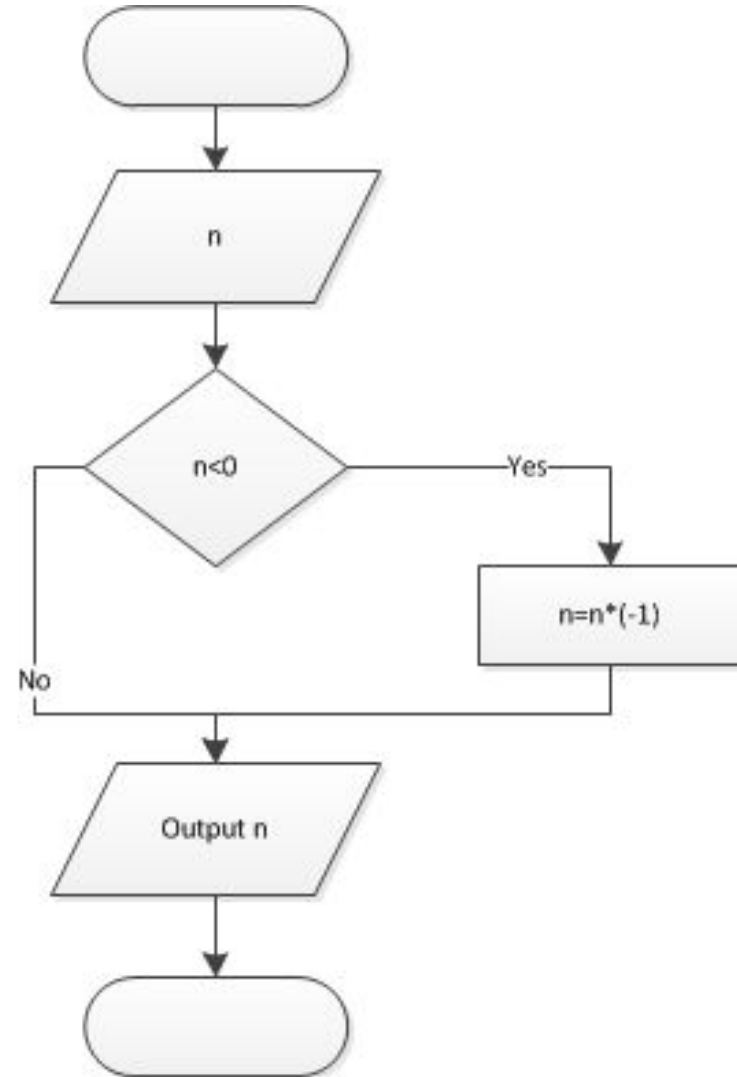
Tính diện tích và chu vi hình chữ nhật

- Đầu vào: Chiều dài: cd , Chiều rộng: cr
- Đầu ra: Diện tích: s , Chu vi: p



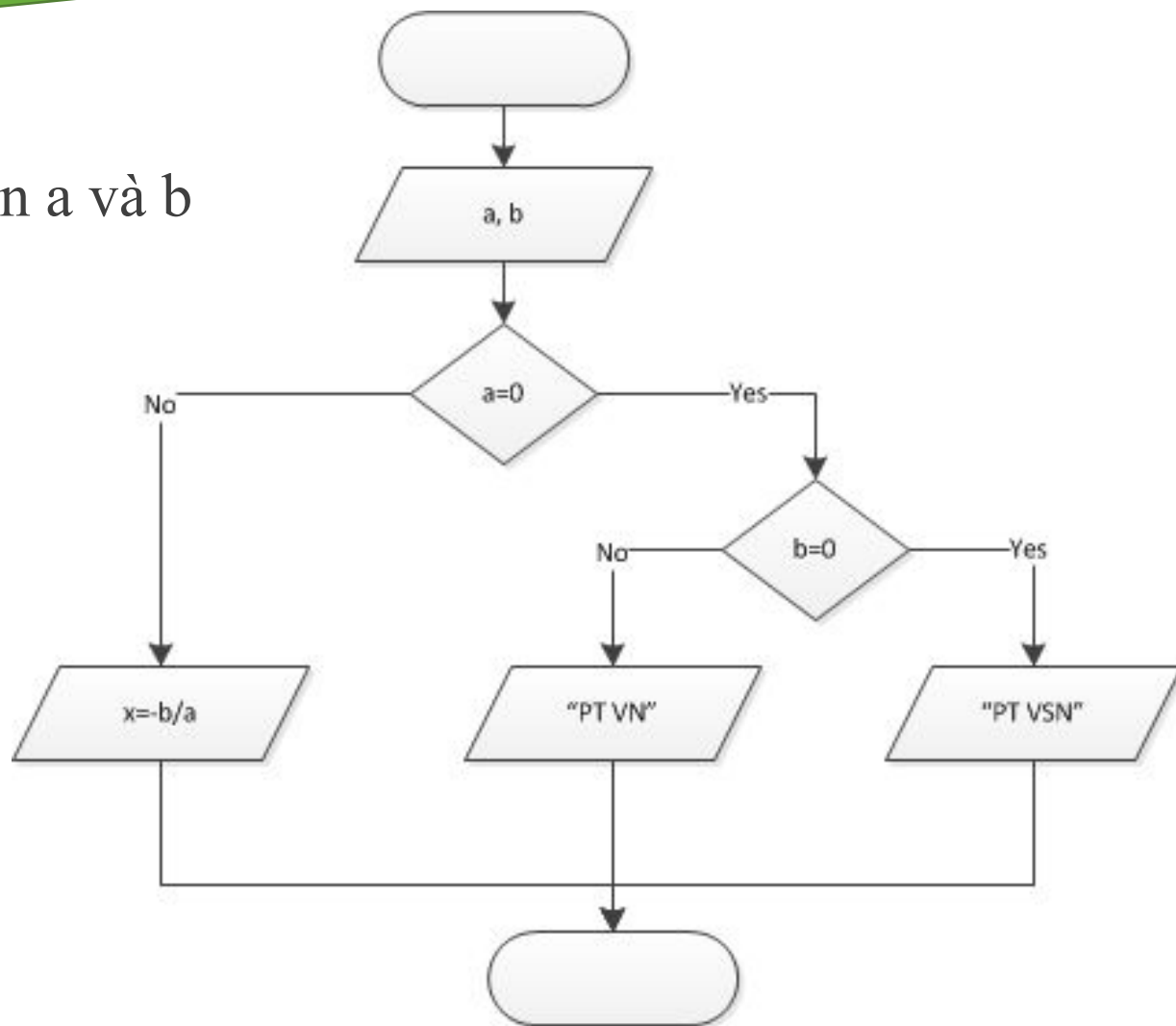
Cho số nguyên n . Tính trị tuyệt đối của n

- Đầu vào: Số nguyên n
- Đầu ra: $|n|$



Giải và biện luận phương trình bậc I: $ax+b=0$

- Đầu vào: Hai số nguyên a và b
- Đầu ra: Nghiệm của pt



Bài tập

Hãy mô tả lưu đồ giải thuật:

1. Tìm số lớn nhất trong 3 số nguyên khác biệt a, b, c
2. Nhập vào độ dài của a, b và c . Kiểm tra xem a, b, c có tạo thành 3 cạnh của tam giác không?
3. Nhập vào số nguyên k ($k > 0$), Xuất ra màn hình k dòng chữ “Xin chào”
4. Tính tổng: $S = 1 + 2 + 3 + \dots + n$ với $n > 0$
5. Tính tổng: $S(n) = 1 - 2 + 3 - 4 + \dots + (-1)^{n+1} n$ với $n > 0$

Bài tập mô tả lưu đồ giải thuật

Giá bán lẻ nước sinh hoạt	Giá bán (đồng/m³)
Đến 4m ³ /người/tháng	6,000
Từ 5m ³ đến 6m ³ /người/tháng	11,500
Từ 7m ³ /người/tháng	12,800

Bài tập mô tả lưu đồ giải thuật

Giá bán lẻ điện sinh hoạt	Giá bán (đồng/kWh)
Bậc 1: Cho kWh từ 1 – 50	1.549
Bậc 2: Cho kWh từ 51 – 100	1.600
Bậc 3: Cho kWh từ 101 – 200	1.858
Bậc 4: Cho kWh từ 201 – 300	2.340
Bậc 5: Cho kWh từ 301 – 400	2.615
Bậc 6: Cho kWh từ 401 trở lên	2.701

Bài tập mô tả lưu đồ giải thuật

1. In ra các ước số của số nguyên dương n
2. Kiểm tra số nguyên n có phải là số nguyên tố không?

Q&A

