



TRƯỜNG ĐẠI HỌC THỦY LỢI
Khoa Công nghệ thông tin
Bộ môn Tin học và KTTT



NHẬP MÔN LẬP TRÌNH
INTRODUCTION TO COMPUTER PROGRAMMING
CSE102



GIỚI THIỆU MÔN HỌC



- Số tín chỉ: 3/140
- Số tiết: 45 tiết (30 LT và 15 TH)
- Học phần bắt buộc cho 5 ngành thuộc Nhóm ngành CNTT
- Điều kiện dự thi kết thúc HP:
 - Nghỉ học không quá 20% số tiết và
 - Dự ít nhất 50% số bài KT và TB các bài KT đạt từ 4 trở lên
- Nội dung tóm tắt: Giới thiệu về máy tính và cách máy tính làm việc; làm quen với lập trình cơ bản mình họa bằng ngôn ngữ C. Giải quyết và cài đặt thuật giải một số vấn đề đơn giản trong Toán học, Khoa học và Kỹ thuật.
- Trọng số điểm môn học:
 - Điểm quá trình: 50%
 - Điểm thi kết thúc HP: 50%

▪ Kiểm tra đánh giá

Hình thức	Số lần lấy điểm	Mô tả	Thời gian	Trọng số
Chấm điểm chuyên cần	1 lần	- Dựa vào số buổi dự học của SV/ tổng 15 buổi và thái độ học tập của SV	Tất cả các buổi học	10%
Bài kiểm tra	2 lần	<ul style="list-style-type: none">- 50 phút/bài- B1: Vòng lặp, rẽ nhánh- B2: Mảng, Xâu, Hàm	- Kết thúc chương 1,2,3	20%
			- Kết thúc chương 4,5	20%
Tổng điểm quá trình				50%
Thi cuối kỳ	1 lần	<ul style="list-style-type: none">- 60 -90 phút- Thi trên máy	1-2 tuần sau khi kết thúc môn học	50%



GIỚI THIỆU MÔN HỌC (t)



- Thông tin giảng viên:
 - Họ và tên: Nguyễn Quỳnh Diệp
 - Chức vụ: Trưởng BM Tin học và Kỹ thuật tính toán, Trưởng ngành CNTT
 - Số điện thoại: 0904345673
 - Email: diepngq@tlu.edu.vn



NỘI DUNG CHÍNH



- Chương 1: Máy tính và Giải bài toán trên máy tính
- Chương 2: Tổng quan về ngôn ngữ lập trình C
- Chương 3: Các lệnh lựa chọn và lặp
- Chương 4: Các kiểu dữ liệu cấu trúc
- Chương 5: Hàm và truyền tham số
- Chương 6: Làm việc với tệp



CHƯƠNG 1

MÁY TÍNH VÀ GIẢI BÀI TOÁN TRÊN MÁY TÍNH



NỘI DUNG CHƯƠNG 1



- Máy tính và xử lý thông tin trên máy tính
- Lịch sử phát triển của máy tính
- Các hệ đếm, đơn vị đo thông tin
- Biểu diễn dữ liệu trong máy tính
- Các bước giải bài toán trên máy tính
- Thuật toán và các cách biểu diễn thuật toán

Thông tin (information) là gì ?

- Là khái niệm mô tả tất cả những gì đem lại cho con người sự *hiểu biết, nhận thức* về các sự vật, hiện tượng.
- Thông tin tồn tại khách quan. Có thể tạo ra, phát sinh truyền đi, lưu trữ,..
- Là ngữ cảnh trong đó dữ liệu được xem xét.

Dữ liệu (data)

- Là biểu diễn của thông tin được thể hiện bằng các tín hiệu vật lý.
- Là vật liệu thô mang tin
 - Dữ liệu sau khi được tập hợp và xử lý sẽ cho ra thông tin.

Dữ liệu trong thực tế

- Các số liệu: Dữ liệu số như trong các bảng biểu.
- Các ký hiệu quy ước; ví dụ chữ viết...
- Các tín hiệu vật lý; ví dụ như ánh sáng, âm thanh, nhiệt độ, áp suất,...

Phân biệt:

- Thông tin chứa đựng hiểu biết về sự vật, hiện tượng.
- Dữ liệu chỉ là các sự kiện không có cấu trúc và không có ý nghĩa nếu không được tổ chức và xử lý.

Ví dụ

Nhiệt độ cơ thể - dữ liệu dạng số

- 39°C \rightarrow thông tin đang bị sốt
- 36.7°C \rightarrow thông tin nhiệt độ cơ thể bình thường

▪ Tri thức (Knowledge)

- Tri thức theo nghĩa thường là thông tin ở mức trừu tượng hơn

▪ Tri thức rất đa dạng

- Có thể là sự kiện, là thông tin
- Là cách mà một người thu thập được qua kinh nghiệm hoặc qua đào tạo.
- Có thể là sự hiểu biết chung hay về một lĩnh vực cụ thể nào đó.

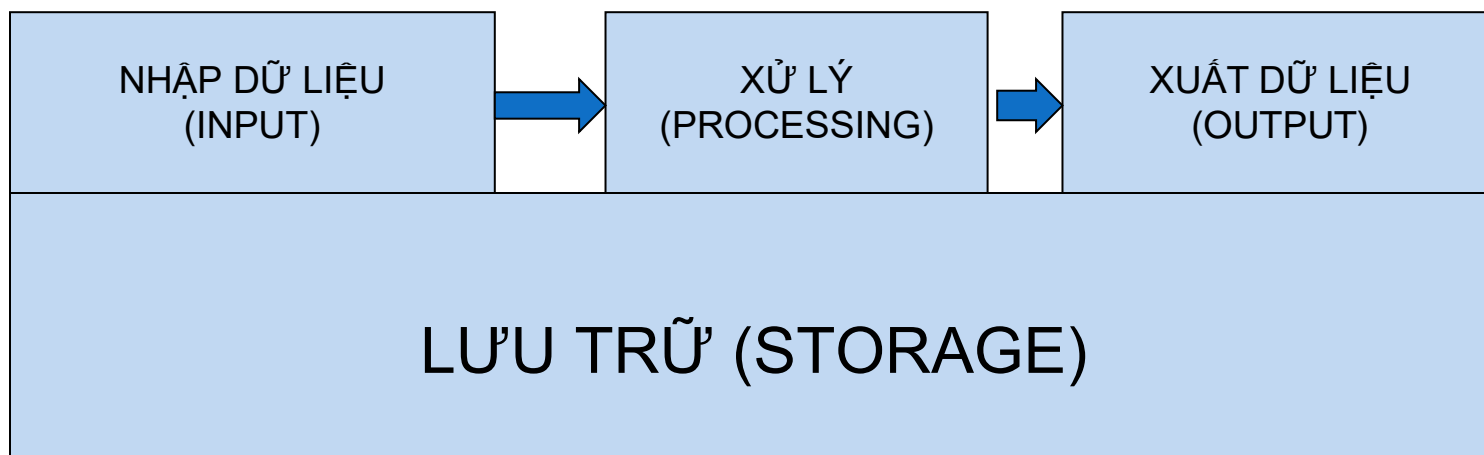
- **Hệ thống thông tin** (*information system*)

là một hệ thống thu nhận dữ liệu, xử lý chúng để tạo nên thông tin có ý nghĩa hoặc dữ liệu mới.



Xử lý thông tin

- Quy trình xử lý thông tin



- Xử lý thông tin bằng máy tính điện tử
 - Tiết kiệm rất nhiều thời gian, công sức
 - Đạt độ chính xác cao nhờ tự động một phần hay toàn bộ quá trình xử lý thông tin.

- **Máy tính** (*computer*): Là thiết bị điện tử thực hiện các công việc
 - Nhận thông tin vào
 - Xử lý thông tin theo chương trình được lưu trữ sẵn bên trong
 - Đưa thông tin ra
 - **Chương trình** (*program*): Là một dãy các lệnh trong bộ nhớ nhằm yêu cầu máy tính thực hiện công việc cụ thể.
- ⇒ Máy tính hoạt động theo chương trình.



Các thế hệ máy tính



- ✓ Thế hệ 1 (1950-1958)
- ✓ Thế hệ 2 (1958-1964)
- ✓ Thế hệ 3 (1965-1974)
- ✓ Thế hệ 4 (1974 – nay)
- ✓ Thế hệ 5 (1990 – nay)

Thể hệ 1 (1950-1958): Von Neumann Machine

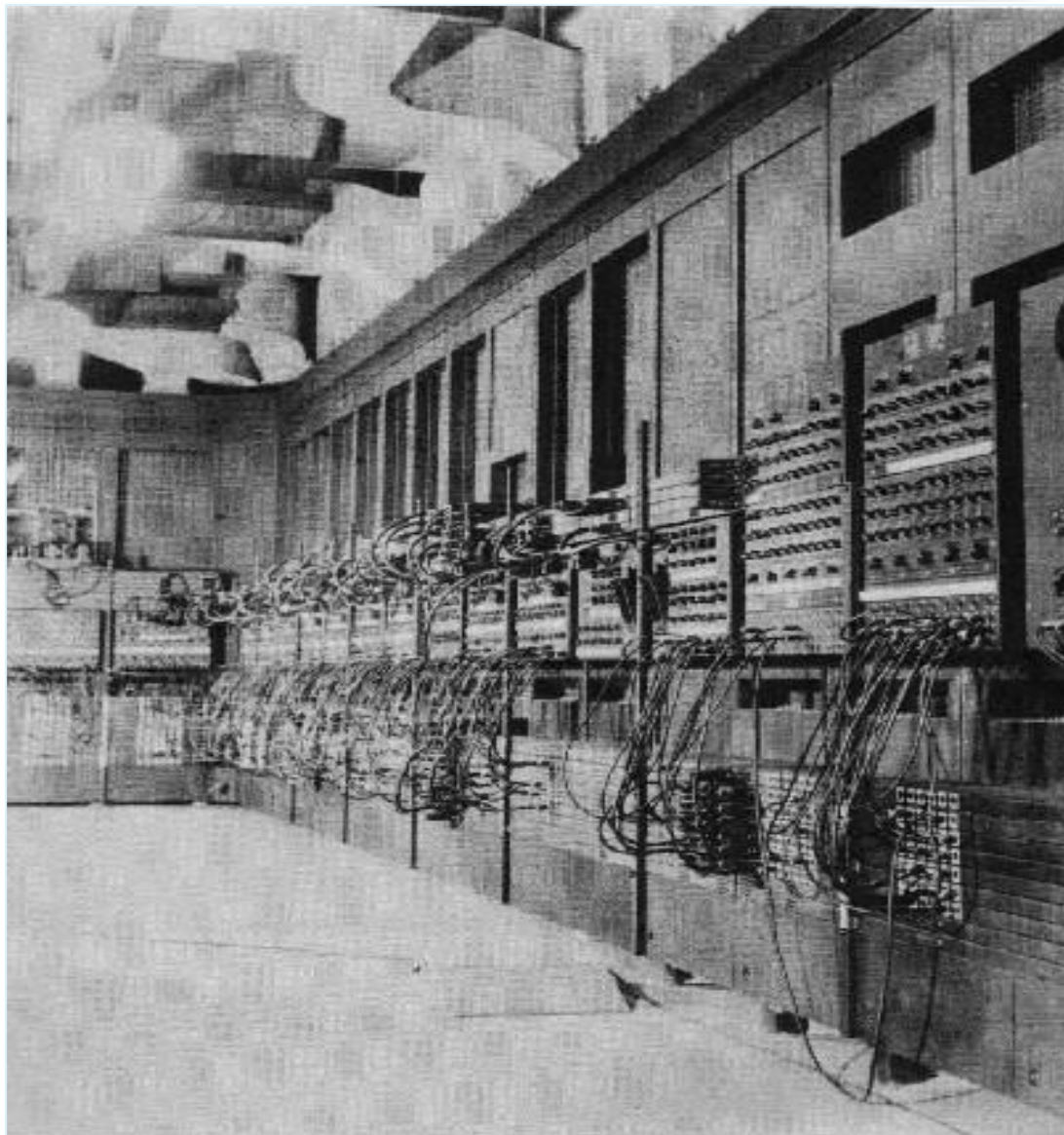
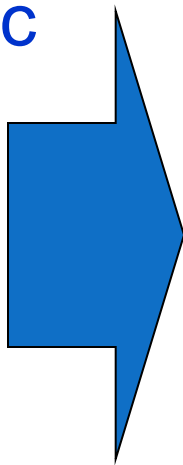
- Sử dụng các bóng đèn điện tử chân không
- Mạch riêng rẽ, vào số liệu bằng phiếu đục lỗ
- Điều khiển bằng tay, kích thước rất lớn
- Tiêu thụ năng lượng nhiều, tốc độ tính chậm khoảng 300 - 3.000 phép tính/s.

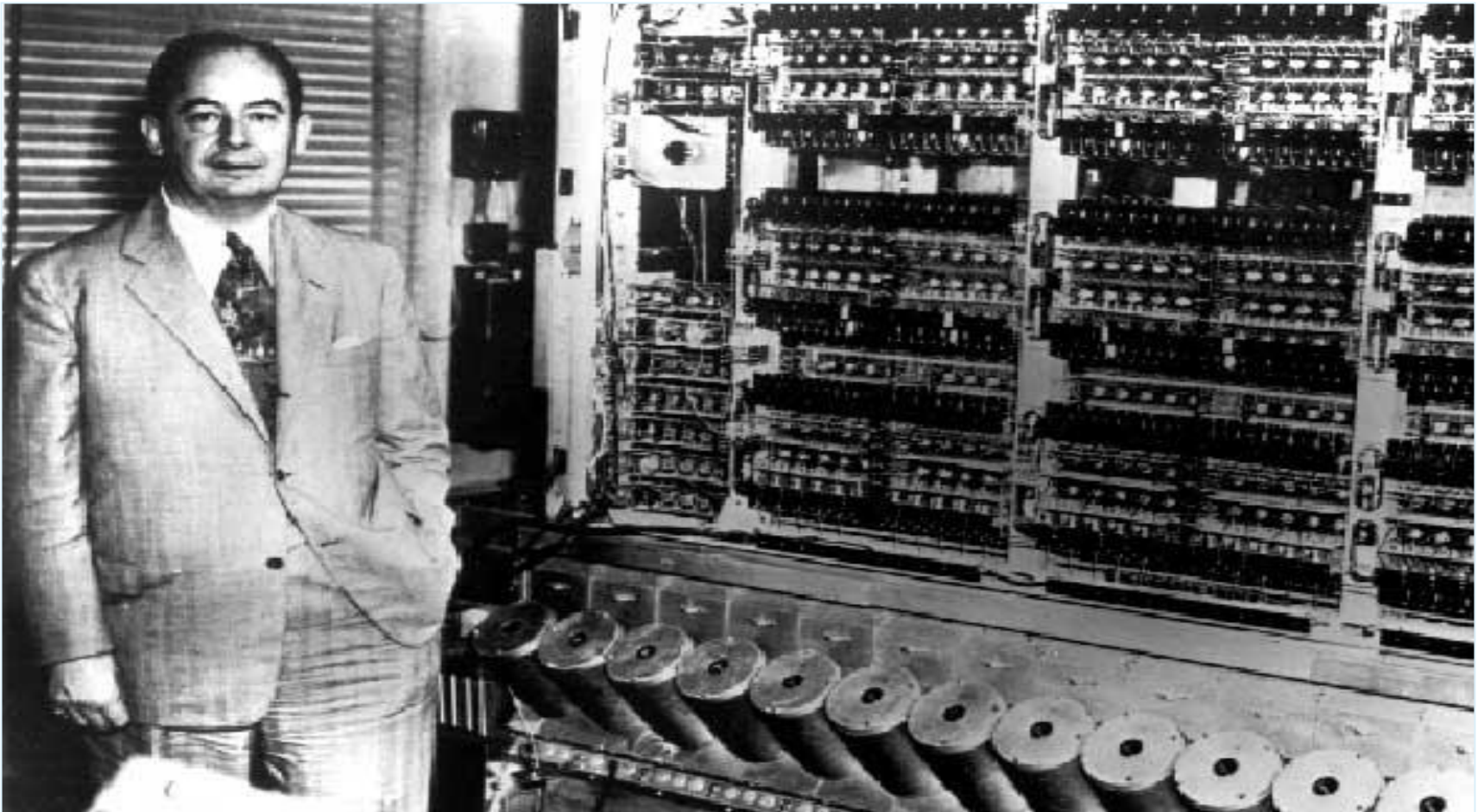
Ví dụ: EDVAC (Mỹ), BESEM (Liên xô cũ)



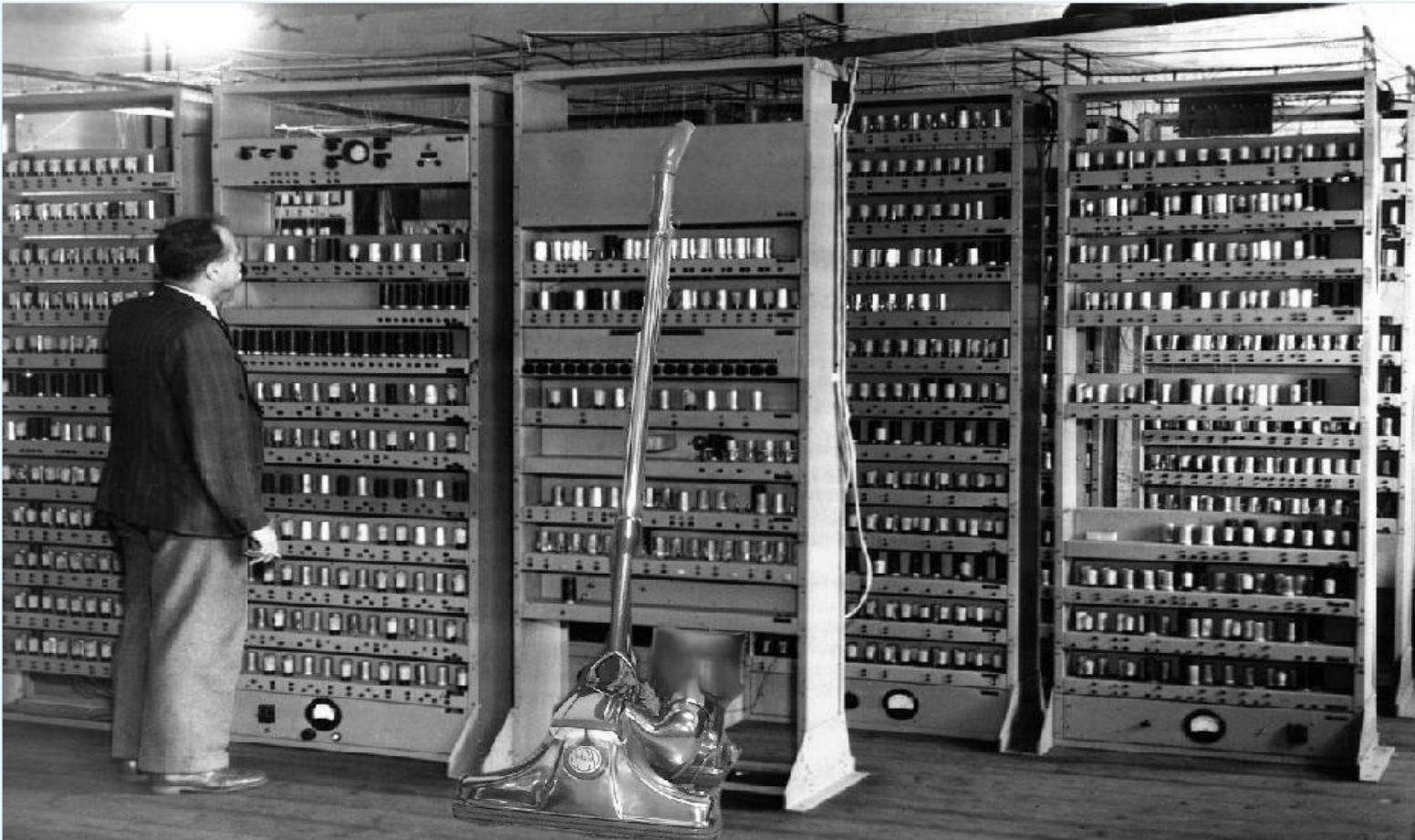
Bóng đèn chân không

Máy tính đầu tiên:
ENIAC (Electronic
Numerical
Integrator And
Computer)





Von Neumann & **UNIVAC** (Universal Automatic Computer)



EDVAC: Electronic Discrete Variable Automatic Computer

Thế hệ 2 (1958-1964): Transistors

- Sử dụng bộ xử lý bằng đèn bán dẫn, mạch in
- Đã có chương trình dịch như Cobol, Fortran và hệ điều hành đơn giản.
- Kích thước máy còn lớn
- Tốc độ tính khoảng 10.000 - 100.000 phép tính/s

Ví dụ

- IBM 7000 series (Mỹ)
- MINSK (Liên Xô cũ)



IBM 7030



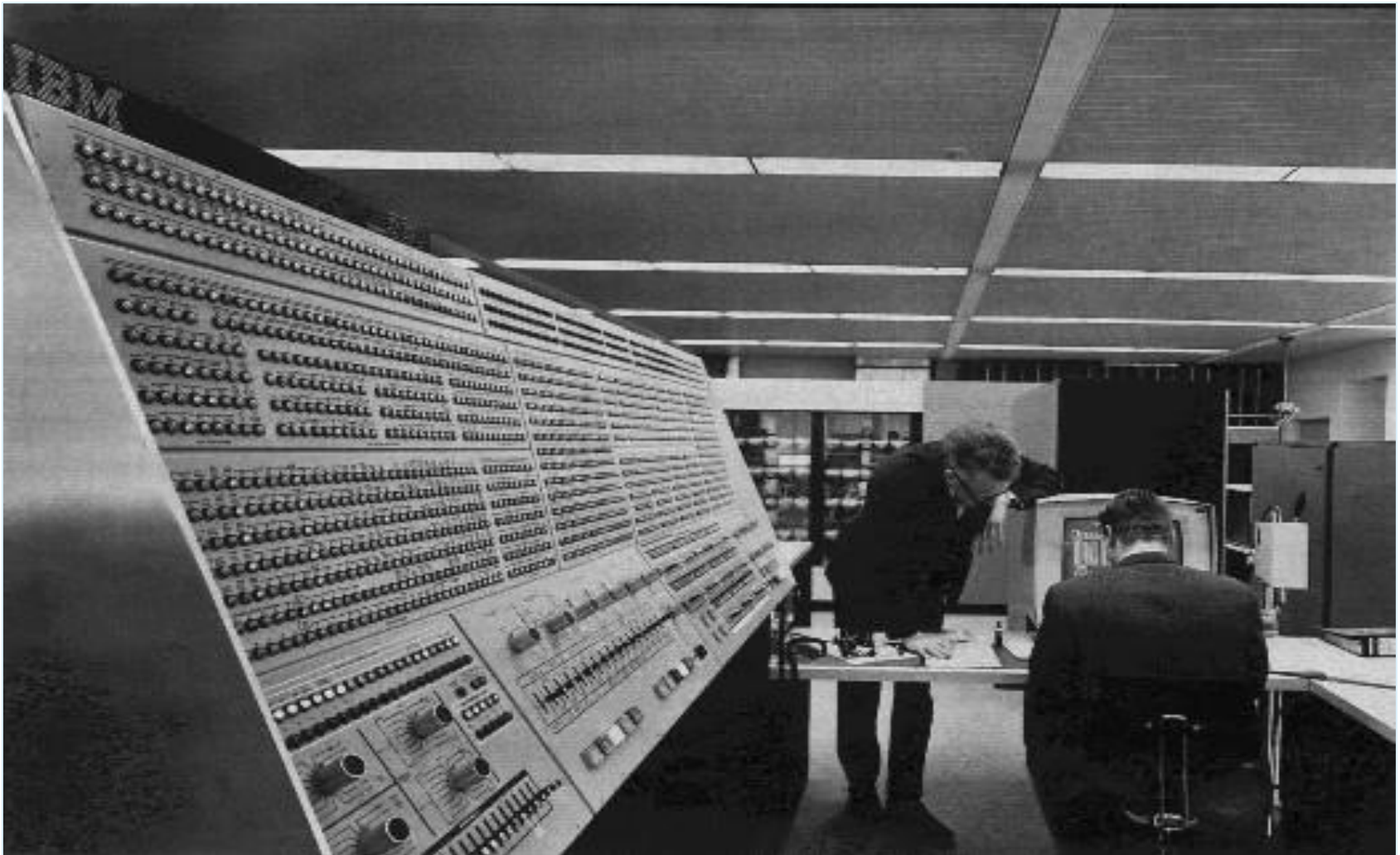
MINSK (Liên Xô cũ)

Thế hệ 3 (1965-1974): Integrated Circuits

- Các bộ vi xử lý được gắn vi mạch điện tử cỡ nhỏ
- Tốc độ tính khoảng 100.000 - 1 triệu phép tính/s.
- Có các hệ điều hành đa chương trình, nhiều người đồng thời theo kiểu phân chia thời gian.
- Kết quả từ máy tính có thể in trực tiếp từ máy in.

Ví dụ

- IBM-360 (Mỹ)
- EC (Liên xô cũ)



IBM 360/91

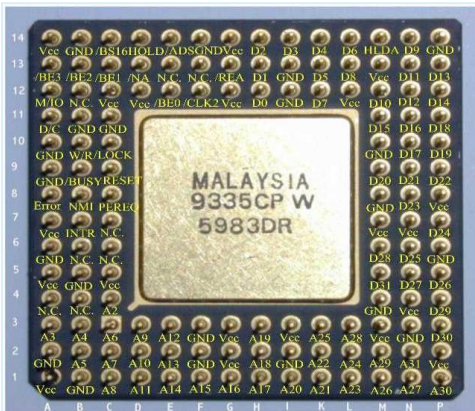
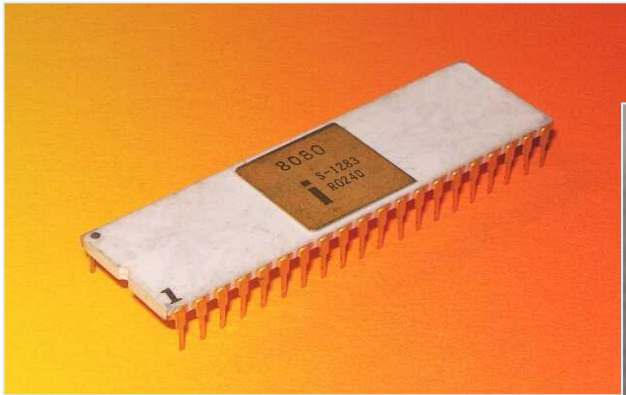
Thế hệ 4 (1974-nay):

LSI(Large Scale Integration), **Multiprocessors**:

- Máy tính có các vi mạch đa xử lý
- Tốc độ: hàng chục triệu đến hàng tỷ phép tính/s.
- Hai loại máy tính chính:
 - Máy tính cá nhân để bàn (Personal Computer - PC) hoặc xách tay (Laptop hoặc Notebook computer)
 - Các loại máy tính chuyên nghiệp thực hiện đa chương trình, đa xử lý,...
- Hình thành các hệ thống mạng máy tính (Computer Networks).
- Các ứng dụng phong phú, đa phương tiện

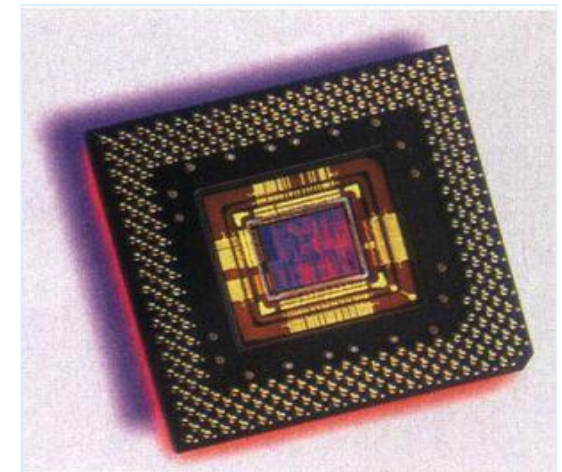
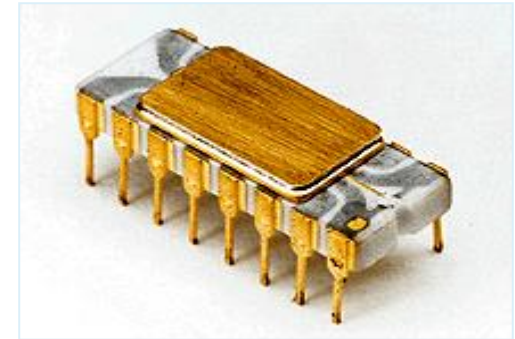
Vi mạch Intel

8080



80386

4004



Pentium

Thế hệ 5 (1990-nay): VLSI (Very Large Scale Integration), ULSI (Ultra), Artificial Intelligence (AI)

- Công nghệ vi điện tử với tốc độ tính toán cao và khả năng xử lý song song.
- Mô phỏng các hoạt động của não bộ và hành vi con người
- Có trí khôn nhân tạo với khả năng tự suy diễn phát triển các tình huống nhận được
- Hệ quản lý kiến thức cơ bản để giải quyết các bài toán đa dạng.



Tin học và các ngành liên quan



- Tin học là ngành khoa học nghiên cứu các phương pháp, công nghệ và kỹ thuật xử lý thông tin một cách tự động.
- Công cụ chủ yếu sử dụng trong tin học là máy tính điện tử và một số thiết bị truyền tin.
- Nội dung nghiên cứu của tin học gồm :
 - **Kỹ thuật phần cứng** (Hardware engineering)
 - **Kỹ thuật phần mềm** (Software engineering)

▪ Công nghệ thông tin:

- Thuật ngữ Công nghệ thông tin xuất hiện ở Việt nam vào những năm 90.
- Theo ITAA: Information Technology Association of America
 - **CNTT** là ngành nghiên cứu các hệ thống thông tin dựa vào máy tính, đặc biệt là các phần mềm ứng dụng và phần cứng máy tính.
 - **CNTT** xử lý với các máy tính điện tử và các phần mềm máy tính nhằm chuyển đổi, lưu trữ, bảo vệ, truyền tin và trích rút thông tin một cách an toàn.
- Xu hướng phát triển
 - Sử dụng "information" thay thế cho "data"
 - Mở rộng cho lĩnh vực truyền thông và CNTT trở thành **CNTT&TT**
 - *ICT: Information and Communication Technology*

▪ Ứng dụng của CNTT

- Các bài toán khoa học kỹ thuật
 - Bài toán phức tạp, cần hàng triệu phép tính/giây
- Các bài toán quản lý
 - Quản lý thông tin, CSDL, hỗ trợ quyết định
- Y tế, Giáo dục
 - Học tập trực tuyến, chẩn đoán bệnh dựa vào AI...
- Thương mại điện tử
 - Hỗ trợ mua bán, thanh toán qua mạng
- Các ứng dụng trong đời sống thường ngày
 - Máy móc, đồ điện tử...
- Giải trí: game, mạng xã hội...

➔ *CNTT là thành phần không thể thiếu của xã hội hiện đại*



Biểu diễn dữ liệu trong máy tính



- ✓ Biểu diễn số trong các hệ đếm
- ✓ Biểu diễn dữ liệu trong máy tính và đơn vị thông tin
- ✓ Biểu diễn số nguyên
- ✓ Biểu diễn số thực
- ✓ Biểu diễn ký tự

- Mỗi hệ đếm sử dụng một số ký hiệu (ký tự, chữ số,..) hữu hạn
 - Tổng số chữ số của một hệ đếm được gọi là **cơ số** (*base, radix*) của hệ đếm đó, ký hiệu là b .
 - **Ví dụ:** Hệ đếm cơ số 10,
 - 10 chữ số từ 0 đến 9.

- Trên lý thuyết, có thể biểu diễn một giá trị theo hệ đếm cơ số bất kì.
- Các hệ đếm sử dụng trong máy tính
 - **Hệ cơ số 10 (Hệ thập phân - Decimal System)**
 - Con người sử dụng
 - **Hệ cơ số 2 (Hệ nhị phân - Binary System)**
 - Máy tính sử dụng
 - **Hệ cơ số 8 (Octal System)**
 - Dùng để viết gọn số nhị phân.
 - **Hệ cơ số 16 (Hexadecimal System)**
 - Dùng để viết gọn số nhị phân

Hệ thập phân ($b = 10$)

Gồm 10 chữ số: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**

Một biểu diễn **A**: **$a_{n-1} \dots a_1 a_0 . a_{-1} a_{-2} \dots a_{-m}$** sẽ xác định giá trị:

$$\begin{aligned} A &= a_{n-1}10^{n-1} + \dots + a_110^1 + a_010^0 + a_{-1}10^{-1} + \dots + a_{-m}10^{-m} \\ &= \sum_{i=-m}^{n-1} a_i 10^i \end{aligned}$$

■ Ví dụ:

- Biểu diễn: **5246** có giá trị là

$$5246 = 5 \times 10^3 + 2 \times 10^2 + 4 \times 10^1 + 6 \times 10^0$$

- Biểu diễn: **254.68** có giá trị là

$$254.68 = 2 \times 10^2 + 5 \times 10^1 + 4 \times 10^0 + 6 \times 10^{-1} + 8 \times 10^{-2}$$

- **Điều kiện:** b nguyên, và $b \geq 2$
- Có b chữ số, giá trị từ 0 đến (b-1)
- Biểu diễn số A_b : $\mathbf{a_{n-1} \dots a_1 a_0 \cdot a_{-1} a_{-2} \dots a_{-m}}$
 - **n** chữ số biểu diễn cho phần nguyên và **m** chữ số biểu diễn cho phần lẻ
 - Cách xác định giá trị số A

$$\begin{aligned} A &= a_{n-1}b^{n-1} + \dots + a_1b^1 + a_0b^0 + a_{-1}b^{-1} + \dots + a_{-m}b^{-m} \\ &= \sum_{i=-m}^{n-1} a_i b^i \end{aligned}$$



Hệ nhị phân (binary system, $b=2$)



- Sử dụng 2 chữ số (*nhị phân*): **0,1**
- Chữ số nhị phân gọi là ***bit*** (**b**inary **d**igit)
 - bit là đơn vị thông tin nhỏ nhất
- Sử dụng n bit biểu diễn được 2^n giá trị khác nhau

$$00\dots000_2 \Leftrightarrow 0_{10}$$

....

$$11\dots111_2 \Leftrightarrow 2^n - 1_{10}$$

- Ví dụ, sử dụng 3bit biểu diễn được 8 giá trị khác nhau

000, 001, 010, 011, 100, 101, 110, 111

- Biểu diễn A : $\mathbf{a_{n-1} \dots a_1 a_0 \cdot a_{-1} a_{-2} \dots a_{-m}}$
 - $\mathbf{a_i}$ là các chữ số nhị phân (**0,1**), giá trị số A là

$$\begin{aligned} A &= a_{n-1} 2^{n-1} + \dots + a_1 2^1 + a_0 2^0 + a_{-1} 2^{-1} + \dots + a_{-m} 2^{-m} \\ &= \sum_{i=-m}^{n-1} a_i 2^i \end{aligned}$$

Ví dụ: Số nhị phân A: 1101001.1011_2 có giá trị

$$\begin{aligned} A &= 2^6 + 2^5 + 2^3 + 2^0 + 2^{-1} + 2^{-3} + 2^{-4} \\ &= 64 + 32 + 8 + 1 + 0.5 + 0.125 + 0.0625 \\ &= 105.6875_{(10)} \end{aligned}$$

- Gồm 8 chữ số: **0, 1, 2, 3, 4, 5, 6, 7**
- Biểu diễn A : **$a_{n-1} \dots a_1 a_0 . a_{-1} a_{-2} \dots a_{-m}$**
 - **a_i** là các chữ số (**0,1,...,7**)
 - Giá trị của A là

$$A = a_{n-1} 8^{n-1} + \dots + a_1 8^1 + a_0 8^0 + a_{-1} 8^{-1} + \dots + a_{-m} 8^{-m}$$
$$= \sum_{i=-m}^{n-1} a_i 8^i$$

Ví dụ: Số A: 235.64_8 có giá trị

$$A = 2 \times 8^2 + 3 \times 8^1 + 5 \times 8^0 + 6 \times 8^{-1} + 4 \times 8^{-2}$$
$$= 157.8125(10)$$

- Sử dụng 16 chữ và số:
0,1,2,3,4,5,6,7,8,9,A,B,C,
D,E,F
- Các chữ số
A, B, C, D, E, F
nhận giá trị tương ứng
(trong hệ 10) là 10, 11,
12, 13, 14, 15

Hệ thập phân	Hệ nhị phân	Hệ mười sáu
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

- Giả sử có số A được biểu diễn theo hệ 16 như sau:

$$A = a_n a_{n-1} \dots a_1 a_0 \cdot a_{-1} a_{-2} \dots a_{-m}$$

Với a_i là các chữ số trong hệ 16, khi đó giá trị của A là:

$$A = a_n 16^n + a_{n-1} 16^{n-1} + \dots + a_1 16^1 + a_0 16^0 + a_{-1} 16^{-1} + a_{-2} 16^{-2} + \dots + a_{-m} 16^{-m}$$

$$A = \sum_{i=-m}^n a_i 16^i$$



Chuyển đổi giữa các hệ đếm



- Hệ đếm bất kỳ (b) \rightarrow Hệ thập phân
- Hệ thập phân \rightarrow Hệ đếm bất kỳ (b)
- Từ hệ đếm b sang b^k và ngược lại ($b = 2$)

- Biểu diễn A_b : $\mathbf{a_{n-1} \dots a_1 a_0 \cdot a_{-1} a_{-2} \dots a_{-m}}$
 - n chữ số biểu diễn cho phần nguyên
 - m chữ số biểu diễn cho phần lẻ

Giá trị tương đương trong hệ 10 là:

$$A = a_{n-1}b^{n-1} + \dots + a_1b^1 + a_0b^0 + a_{-1}b^{-1} + \dots + a_{-m}b^{-m}$$
$$= \sum_{i=-m}^{n-1} a_i b^i$$

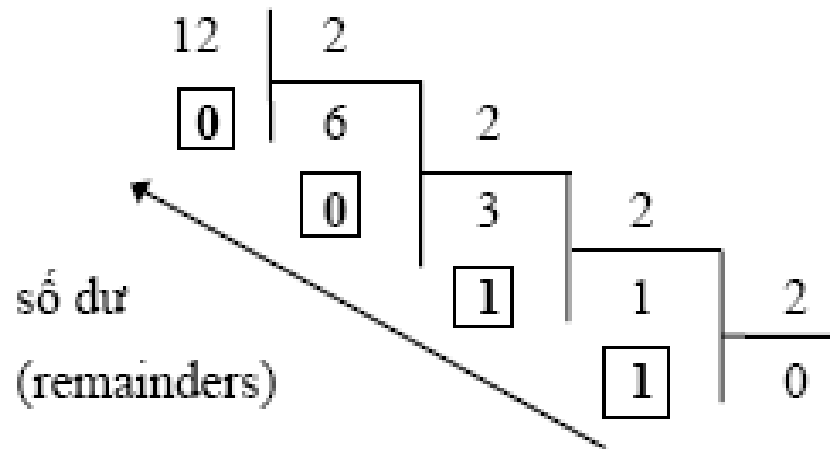
$$\text{VD: } 11010.11_2 = 2^4 + 2^3 + 2^1 + 2^{-1} + 2^{-2} = 26.75_{10}$$

$$1A.C_{16} = 1 \times 16^1 + 10 \times 16^0 + 12 \times 16^{-1} = 26.75_{10}$$

- Trường hợp tổng quát, một số N trong hệ thập phân ($N_{(10)}$) gồm phần nguyên và phần thập phân.
- Chuyển 1 số từ hệ thập phân sang 1 số ở hệ cơ số b bất kỳ gồm 2 bước:
 - Đổi phần nguyên (của số đó) từ hệ thập phân sang hệ b
 - Đổi phần thập phân (của số đó) từ hệ thập phân sang hệ cơ số b

- **Bước 1:** Lấy phần nguyên của $N_{(10)}$ chia cho b , ta được thương là T_1 số **dư d_1** .
- **Bước 2:** Nếu T_1 khác 0, Lấy T_1 chia tiếp cho b , ta được thương số là T_2 , số **dư là d_2**
- Lặp lại như vậy cho tới bước thứ n , khi ta được $T_n = 0$
- **Bước n :** Nếu T_{n-1} khác 0, lấy T_{n-1} chia cho b , ta được thương số là $T_n = 0$, **số dư là d_n**
- Kết quả ta được số $N_{(b)}$ là số tạo bởi các số dư (***được viết theo thứ tự ngược lại***) trong các bước trên
- Phần nguyên của $N_{(10)} = d_n d_{n-1} \dots d_1 (b)$

- Ví dụ: Cách chuyển phần nguyên của số $12.6875_{(10)}$ sang số trong hệ nhị phân:
 - Dùng phép chia cho 2 liên tiếp thu được một loạt các số dư như sau:



Kết quả: $12_{(10)} = 1100_{(2)}$

- Bước 1: Lấy phần thập phân của $N_{(10)}$ nhân với b , ta được một số có dạng $x_1.y_1$ (x là phần nguyên, y là phần thập phân)
- Bước 2: Nếu y_1 khác 0, tiếp tục lấy $0.y_1$ nhân với b , ta được một số có dạng $x_2.y_2$

... (Lặp lại như vậy cho đến khi $y_n=0$)

- Bước n: Nếu y_{n-1} khác 0, nhân $0.y_{n-1}$ với b , ta được $x_n.0$
- Kết quả ta được số sau khi chuyển đổi là:

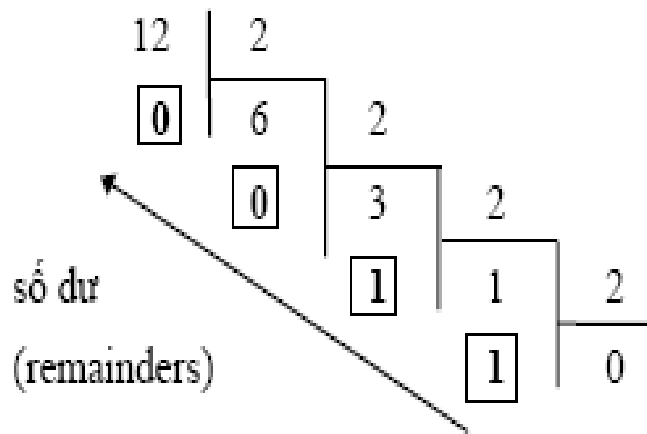
Phần thập phân của $N_{(10)} = 0.x_1x_2...x_n(b)$

- Ví dụ: Cách chuyển phần thập phân của số $12.6875_{(10)}$ sang hệ nhị phân:

$$\begin{array}{rcll} 0.6875_{(10)} = ?_{(2)} & & & \text{phần nguyên của tích} \\ 0.6875 \times 2 = 1.375 & \leftarrow & & \text{phần thập phân của tích} \\ 0.3750 \times 2 = 0.75 & & & \\ 0.75 \times 2 = 1.5 & & & \\ 0.5 \times 2 = 1.0 & & & \end{array}$$

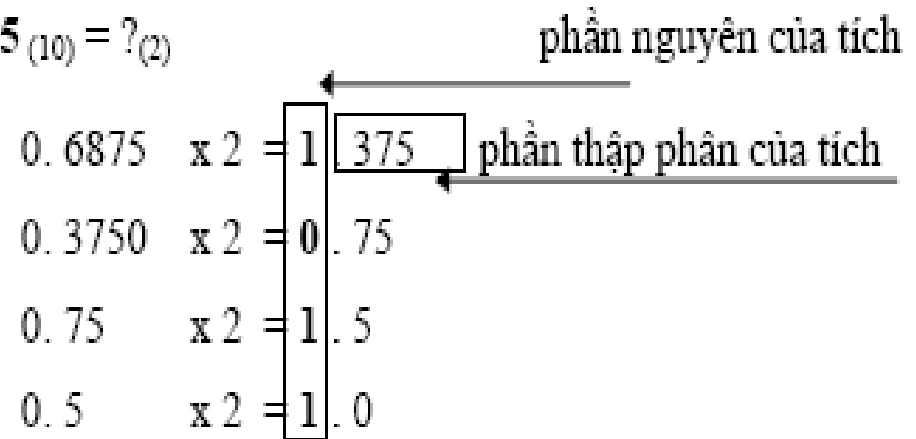
$$0.6875_{(10)} = 0.1011_{(2)}$$

▪ Ví dụ: $12.6875_{(10)} = 1100.1011_{(2)}$



Kết quả: $12_{(10)} = 1100_{(2)}$

$0.6875_{(10)} = ?_{(2)}$



$0.6875_{(10)} = 0.1011_{(2)}$

▪ $69.25_{(10)} = ?_{(2)}$



Chuyển đổi phần thập phân (t)



▪ Cách 2: Tính nhẩm

- Phân tích số đó thành tổng các lũy thừa của 2, sau đó dựa vào các số mũ để xác định dạng biểu diễn nhị phân
→ Nhanh hơn với các giá trị nhỏ
- Ví dụ: $69.25_{(10)} = 64 + 4 + 1 + \frac{1}{4}$
 $= 2^6 + 2^2 + 2^0 + 2^{-2}$
 $= 1000101.01_{(2)}$



Bài tập 1



Chuyển hệ thập phân sang hệ nhị phân

- a. 124.75
- b. 125.375
- c. 65.125

❖ Mã hóa dữ liệu

- Dữ liệu đưa vào máy tính phải được mã hóa thành số nhị phân
- Các loại dữ liệu:
 - Dữ liệu nhân tạo: Do con người quy ước
 - Dữ liệu tự nhiên:
 - Tồn tại khách quan với con người.
 - Phổ biến là các tín hiệu vật lý: âm thanh, hình ảnh,...
- Nguyên tắc mã hóa dữ liệu
 - Dữ liệu nhân tạo:
 - Dữ liệu số: Mã hóa theo các chuẩn quy ước
 - Dữ liệu ký tự: Mã hóa theo bộ mã ký tự
 - Dữ liệu tự nhiên:
 - Cần phải số hóa trước khi đưa vào máy tính



Dữ liệu trong máy tính



- Dữ liệu cơ bản
 - Số nguyên, số thực, ký tự
- Dữ liệu có cấu trúc
 - Mảng, chuỗi ký tự, tập hợp, bản ghi

- Số nguyên:
 - Không dấu: Biểu diễn theo mã nhị phân
 - Có dấu: Biểu diễn dưới dạng mã bù hai.
- Số thực:
 - Biểu diễn bằng số dấu chấm (phẩy) động.
- Ký tự:
 - Biểu diễn bằng mã ký tự trên các bộ mã ký tự.

- Là tập hợp các loại dữ liệu cơ bản được cấu thành theo một cách nào đó.
 - Ví dụ: kiểu dữ liệu mảng, kiểu xâu ký tự, kiểu tập hợp, bản ghi,...
 - *Các dữ liệu có cấu trúc sẽ được nghiên cứu cụ thể trong phần học về ngôn ngữ lập trình.*

- Bit (**BI**nary digi**T**)

- Là đơn vị thông tin nhỏ nhất
- Nhận một trong hai giá trị nhị phân 0/1

- Byte (**B**): Chuỗi 8 bit

- Các đơn vị dẫn xuất

- Kilobyte (**KB**): $1 \text{ KB} = 2^{10} \text{B} = 1024 \text{B}$
- Megabyte (**MB**): $1 \text{ MB} = 2^{10} \text{KB} = 2^{20} \text{B}$
- Gigabyte (**GB**): $1 \text{ GB} = 2^{10} \text{MB} = 2^{20} \text{KB} = 2^{30} \text{B}$
- Terabyte (**TB**): $1 \text{ TB} = 2^{10} \text{GB} = \dots = 2^{40} \text{B}$
- Petabyte (**PB**): $1 \text{ PB} = 2^{10} \text{TB} = \dots = 2^{50} \text{B}$

- Dùng n bit để biểu diễn cho một số nguyên không dấu

$$\mathbf{A: a_{n-1}a_{n-2}\dots a_2a_1a_0}$$

Trong đó $\mathbf{a_i}$ là các chữ số nhị phân (**0,1**)

- Giá trị của A được tính theo công thức

$$\begin{aligned} A &= a_{n-1}2^{n-1} + \dots + a_12^1 + a_02^0 \\ &= \sum_{i=0}^{n-1} a_i 2^i \end{aligned}$$

- Dải giá trị của A : $[0..2^n-1]$

- Ví dụ 1: Tìm biểu diễn của các số nguyên không dấu 8 bit có giá trị sau:

- **A = 45**

$$\begin{aligned} A = 45 &= 32 + 8 + 4 + 1 \\ &= 2^5 + 2^3 + 2^2 + 2^0 \end{aligned}$$

$$\rightarrow A = 0010\ 1101$$

- **B = 156**

$$\begin{aligned} B = 156 &= 128 + 16 + 8 + 4 \\ &= 2^7 + 2^4 + 2^3 + 2^2 \end{aligned}$$

$$\rightarrow B = 1001\ 1100$$



Số nguyên không dấu



Ví dụ 2: Tính giá trị các số nguyên không dấu 8bit có biểu diễn nhị phân như dưới đây

$$\begin{aligned} X = 0010\ 1011 &= 2^5 + 2^3 + 2^1 + 2^0 \\ &= 32 + 8 + 2 + 1 = 43 \end{aligned}$$

$$\begin{aligned} Y = 1001\ 0110 &= 2^7 + 2^4 + 2^2 + 2^1 \\ &= 128 + 16 + 4 + 2 = 150 \end{aligned}$$

- Với $n = 8\text{bit}$
- Dải biểu diễn là $[0, 255]$

0000 0000 = 0

0000 0001 = 1

0000 0010 = 2

0000 0011 = 3

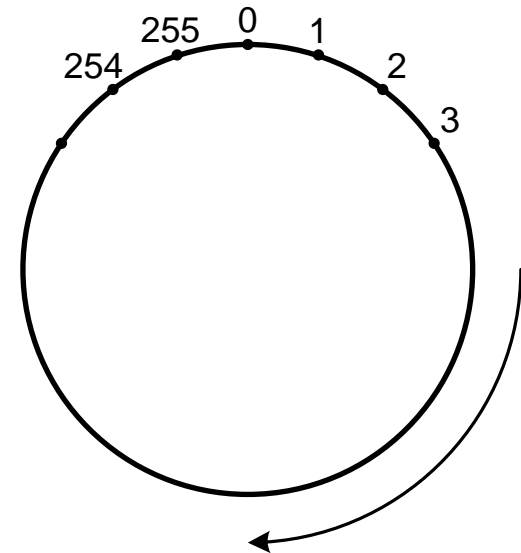
.....

1111 1111 = 255

- Trục số học tự nhiên:



- Trục số học máy tính:





Phạm vi biểu diễn



- Với $n = 16$ bit:
 - Dải biểu diễn là $[0, 65535]$
- Với $n = 32$ bit:
 - Dải biểu diễn là $[0, 2^{32}-1]$
- Với $n = 64$ bit:
 - Dải biểu diễn là $[0, 2^{64}-1]$



- Các ký tự cần được chuyển đổi thành chuỗi bit nhị phân gọi là **mã ký tự**.
- Số bit dùng cho mỗi ký tự theo các mã khác nhau là khác nhau.

Ví dụ: Bộ mã ASCII dùng 8bit cho mỗi ký tự.

ASCII = American Standard Codes for Information Interchangeable

- Do ANSI (**A**merican **N**ational **S**tandard **I**nstitute) thiết kế
- Là bộ mã được dùng để *trao đổi thông tin chuẩn của Mỹ*.
 - Lúc đầu chỉ dùng 7bit (128 ký tự) sau đó mở rộng cho 8 bit và có thể biểu diễn 256 ký tự khác nhau trong máy tính
- Bộ mã 8 bit → mã hóa được cho $2^8 = 256$ kí tự, có mã từ $00_{16} \div FF_{16}$, bao gồm:
 - 128 kí tự chuẩn có mã từ $00_{16} \div 7F_{16}$
 - 128 kí tự mở rộng có mã từ $80_{16} \div FF_{16}$

ASCII Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

 **Physical Device Controls: Format Effectors**

- **95** kí tự hiển thị được: có mã từ $20_h \div 7E_h$
 - 26 chữ cái Latin hoa 'A' \div 'Z', mã từ $41_h \div 5A_h$
 - 26 chữ cái Latin thường 'a' \div 'z', mã từ $61_h \div 7A_h$
 - 10 chữ số thập phân '0' \div '9', mã từ $30_h \div 39_h$
 - Các dấu câu: . , ? ! : ; ...
 - Các dấu phép toán: + - * / ...
 - Một số kí tự thông dụng: #, \$, &, @, ...
 - Dấu cách (mã là 20_h)
- **33** mã điều khiển: $00_h \div 1F_h$ và $7F_h$
 - Điều khiển định dạng, phân cách thông tin..

- Được định nghĩa bởi:
 - Nhà chế tạo máy tính
 - Người phát triển phần mềm

Ví dụ:

- Bộ mã ký tự mở rộng của IBM: được dùng trên máy tính IBM-PC.
- Bộ mã ký tự mở rộng của Apple: được dùng trên máy tính Macintosh.
- Các nhà phát triển phần mềm tiếng Việt cũng đã thay đổi phần này để mã hoá cho các ký tự riêng của chữ Việt, ví dụ như bộ mã TCVN 5712.



- Unicode: Mã “thống nhất”
 - Quản lý bởi Unicode Consortium gồm các hãng máy tính và phần mềm lớn (và Univ. of California, Berkeley)
 - Hiện có hơn 100.000 ký tự
- Quy định nhiều chế độ mã hóa, cho phép sử dụng 8/16/24/32bit để biểu diễn một ký tự
- Các chế độ mã hóa như UTF16, UTF32 cho phép mã hóa số lượng ký tự lớn → hỗ trợ nhiều ngôn ngữ trong cùng bảng mã



Giải quyết bài toán bằng máy tính



- ✓ Khái niệm về bài toán
- ✓ Quá trình giải quyết bài toán bằng máy tính
- ✓ Phương pháp giải quyết bài toán bằng máy tính

$A \rightarrow B$

- A: Giả thiết, điều kiện ban đầu
- B: Kết luận, mục tiêu cần thực hiện
- \rightarrow : Suy luận, giải pháp cần xác định
- 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 **B**



$$\mathbf{A \rightarrow B}$$

- A là Input
- B là Output
- \rightarrow là Chương trình cho phép biến đổi A thành B

- Chương trình
 - Cách mã hóa lại thuật toán để giải quyết vấn đề/bài toán đã cho
 - Tạo thành từ các lệnh cơ bản của máy tính
- Khó khăn
 - Tồn tại các yếu tố không xác định: A và B không đầy đủ, rõ ràng
- Giải quyết bài toán trên máy tính
 - Vấn đề tổ chức dữ liệu và thiết kế giải thuật

Cấu trúc dữ liệu + Giải thuật = Chương trình

- **Thực hiện bởi con người**

- Là cách thức chủ yếu, dựa trên
 - Những thông tin được phản ánh rõ ràng trong A, B hoặc →
 - Các tri thức của con người

- **Tự động hóa xây dựng thuật giải**

- Lĩnh vực mới, đang được nghiên cứu
- Cần phải biểu diễn nội dung và các tri thức liên quan dưới dạng tương minh và đầy đủ



- Máy tính
 - Chỉ làm được những gì được bảo.
 - **Không thông minh:** không thể tự phân tích vấn đề và đưa ra giải pháp.
 - Không thể dùng giải quyết các vấn đề liên quan đến hành động vật lý hoặc biểu thị cảm xúc
- Lập trình viên
 - Phân tích vấn đề
 - Tạo ra các chỉ dẫn để giải quyết vấn đề (xây dựng chương trình).
 - Máy tính sẽ thực hiện các chỉ dẫn này.



Các bước giải quyết bài toán



1. Xác định bài toán
2. Lựa chọn phương pháp giải
3. Xây dựng thuật toán hoặc thuật giải
4. Cài đặt chương trình
5. Hiệu chỉnh chương trình
6. Thực hiện chương trình

- Mô tả bài toán cần giải quyết
 - **Dữ liệu vào:** Danh sách các dữ kiện vào
 - **Điều kiện vào:** Ràng buộc, quan hệ giữa chúng
 - **Dữ liệu ra:** Danh sách các dữ liệu ra
 - **Điều kiện ra:** Ràng buộc, quan hệ giữa chúng
- Đánh giá, nhận định tính khả thi của bài toán
 - Thời gian, kinh phí, nguồn lực,...

Ví dụ: Bài toán tìm USCLN của 2 số nguyên dương

- **Nhập:** 2 số X, Y
- **Điều kiện nhập:** X, Y nguyên dương
- **Dữ liệu ra:** $\text{USCLN}(X, Y)$

- Có nhiều phương pháp khác nhau
 - Khác nhau về thời gian thực hiện, chi phí lưu trữ dữ liệu, độ chính xác...
- Tùy theo nhu cầu cụ thể và khả năng xử lý tự động được sử dụng để lựa chọn phương pháp thích hợp

Ví dụ: Bài toán sắp xếp dãy số

- Nổi bọt, Vun đống, Sắp xếp nhanh,...

Bước 3. Xây dựng thuật giải

1. Xác định và chính xác hóa các thao tác
 - Để đạt được kết quả cần làm gì?
2. Xác định các dữ liệu cần dùng và tính chất của chúng
 - Để thực hiện, thao tác cần dl gì và sẽ tạo ra dl gì?
3. Xác định trình tự các thao tác
 - Thao tác nào cần làm trước
 - Thao tác thực hiện 1 hay nhiều lần, thực hiện trong điều kiện nào..?
4. Diễn tả giải thuật dưới dạng chuẩn: sơ đồ khối, mã giả,..

Mã hóa giải thuật bằng một ngôn ngữ lập trình

- Thay thế các thao tác bằng các lệnh tương ứng của ngôn ngữ sử dụng
- Lựa chọn ngôn ngữ lập trình, tùy theo bài toán giải quyết
 - NNLT bậc thấp: Hợp ngữ
 - NNLT bậc cao: C, Pascal, Java,...

Chạy thử để phát hiện và điều chỉnh các sai sót có thể có ở bước 4.

- Lỗi cú pháp:
 - Viết sai cú pháp của ngôn ngữ lập trình lựa chọn
- Lỗi ngữ nghĩa
 - Mã hóa sai giải thuật
 - Giải thuật sai

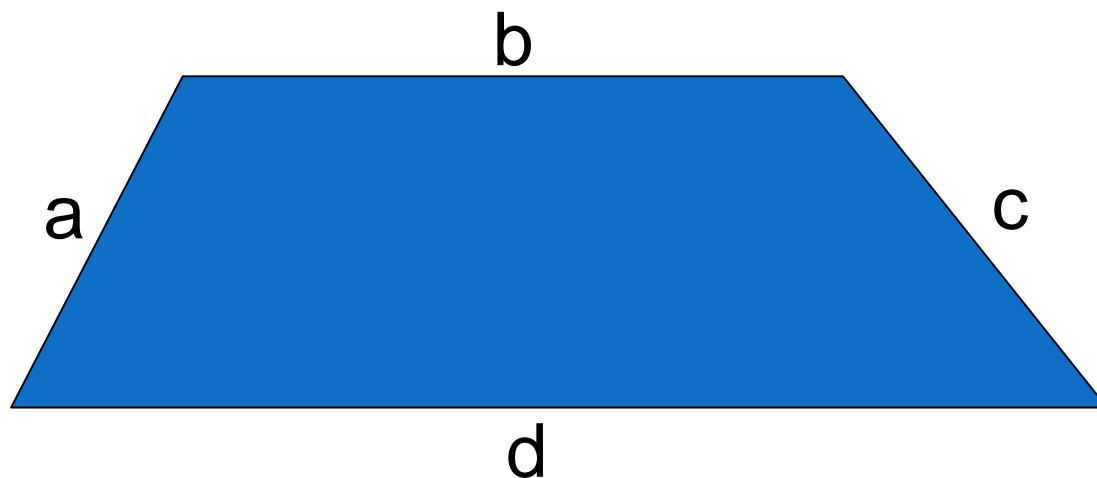


Bước 6. Thực hiện chương trình



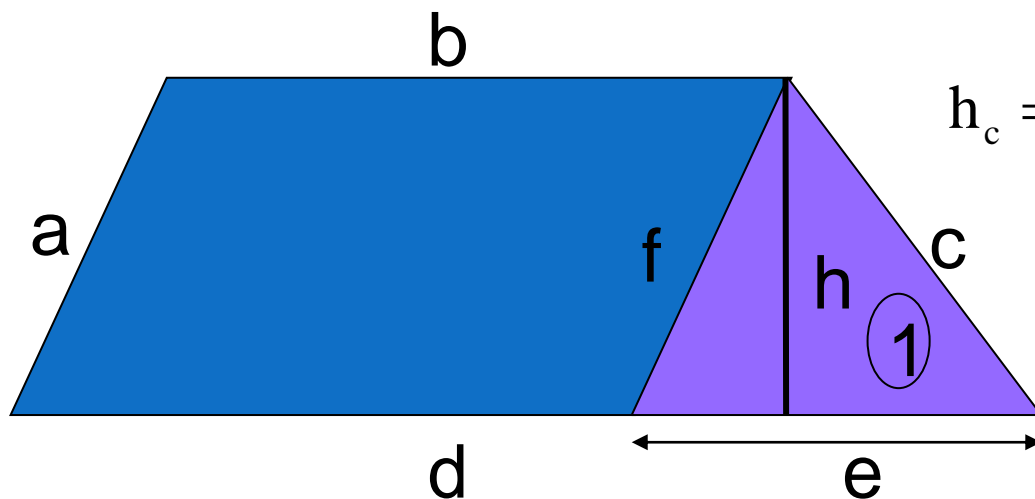
- Cho máy tính thực hiện chương trình.
- Tiến hành phân tích kết quả thu được
 - Kết quả đó có phù hợp hay không.
 - Nếu không phù hợp kiểm tra lại toàn bộ các bước.

Tính diện tích hình thang khi biết 4 cạnh



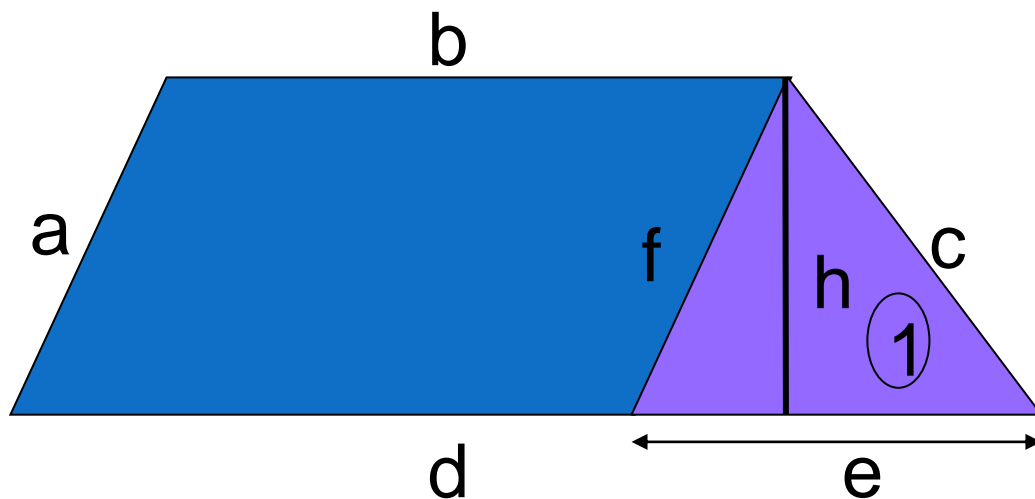
Mô tả bài toán

- Nhập: 4 cạnh a, b, c, d
- Điều kiện nhập: $a, b, c, d > 0$ và $d > b$
- Xuất: Một giá trị số tương ứng diện tích hình thang



$$h_c = \frac{2\sqrt{p(p-a)(p-b)(p-c)}}{c}$$

1. Để tính diện tích hình thang, cần tính đường cao (công thức $S = h(b+d)/2$)
2. Tính đường cao h , cần phải biết 3 cạnh của tam giác (1)
3. Cần tính cạnh tam giác (1) trước khi tính đường cao h



- Để tính cạnh của tam giác (1) cần biết các cạnh của hình thang
- Các cạnh của hình thang là dữ kiện cho biết của đề bài
→ bài toán có thể giải được

1. Nhập các số a, b, c, d
2. Tính các cạnh của tam giác (1)
 - $f \leftarrow a$
 - $e \leftarrow d - b$
 - $p \leftarrow (f + e + c) / 2$
3. Tính chiều cao của tam giác (1)

$$h = \frac{2\sqrt{p(p-e)(p-f)(p-c)}}{e}$$

4. Tính diện tích hình thang $S = h(d+b)/2$
5. In kết quả S
6. Kết thúc

Ví dụ bằng ngôn ngữ C

```
void main(){
    float a, c, b, d, S;
    float f, e, p, h;
    printf("Cho do dai cac canh cua hinh thang:\n");
    printf("a = "); scanf("%f",&a);
    printf("b = "); scanf("%f",&b);
    printf("c = "); scanf("%f",&c);
    printf("d = "); scanf("%f",&d);
    if(a>0 && b>0 && c>0 && d>0 && d>b) {
        f = a;
        e = d-b;
        p = (f+e+c)/2;
        h = 2*sqrt(p*(p-f)*(p-c)*(p-e))/e;
        S = h*(b+d)/2;
        printf("Dien tich hinh thang S = %5.3f",S);
    }
    else
        printf("So lieu khong hop le!");
}
```



- ✓ Khái niệm
- ✓ Biểu diễn thuật toán
 - ✓ Thuật toán đệ quy
 - ✓ Thuật toán ví dụ



- Thuật toán (*algorithm*) là khái niệm cơ sở của Toán học và Tin học
- Nghiên cứu thuật toán đóng vai trò quan trọng trong khoa học máy tính
 - Máy tính chỉ có khả năng thực hiện công việc theo một thuật toán.
 - Thuật toán chỉ đạo máy tính từng bước phải làm gì.



Khái niệm (t)



Thuật toán để giải một bài toán là một dãy hạn hữu các thao tác và trình tự thực hiện các thao tác đó sao cho sau khi thực hiện dãy thao tác này theo trình tự đã chỉ ra, với đầu vào (input) ta thu được kết quả đầu ra (output) mong muốn

Tìm phần tử lớn nhất trong một dãy hữu hạn các số nguyên

1. Đặt giá trị lớn nhất tạm thời (Max) bằng số nguyên đầu tiên của dãy

Max là giá trị lớn nhất ở mỗi giai đoạn thực hiện

2. Nếu tất cả số nguyên trong dãy đã được xét, thực hiện bước 5
3. So sánh số nguyên kế tiếp trong dãy với Max
 - Nếu lớn hơn Max thì thay Max bằng số nguyên này.
4. Lặp lại bước 2
5. Thông báo: Max là giá trị lớn nhất trong dãy số.



Đặc trưng của thuật toán



Khi mô tả thuật toán, cần chú ý các đặc trưng

- Nhập
- Xuất
- Tính xác định
- Tính hữu hạn
- Tính hiệu quả
- Tính tổng quát

- **Nhập (input):**

- Các giá trị "*đầu vào*" (input values) từ một tập hợp nhất định nào đó.

- **Xuất (output):**

- Những giá trị trả về (output values) thuộc một tập hợp nhất định nào đó thể hiện lời giải cho bài toán/vấn đề
- Tương ứng với tập hợp các giá trị nhập



Tính xác định



- Các bước trong thuật toán phải chính xác rõ ràng, không gây sự nhập nhằng nhầm lẫn

- Trong mọi trường hợp của dữ liệu vào, thuật toán phải cho ra kết quả sau một thời gian hữu hạn
 - Thời gian có thể phụ thuộc vào từng bài toán cụ thể, dữ liệu cụ thể các thuật toán khác nhau cho một bài toán
 - Ví dụ bài toán sắp xếp dãy số
 - Bubble sort, Quick sort,..

- Thực hiện thuật toán cần:
 - Thời gian thực thi
 - Các công cụ hỗ trợ (bộ nhớ,..) để lưu kết quả trung gian
- Độ phức tạp thuật toán: Thời gian và các công cụ hỗ trợ
 - Thuật toán càng hiệu quả độ phức tạp càng nhỏ
 - Trong máy tính, thường quan tâm tới
 - Thời gian thực hiện: số thao tác cơ bản cần thực hiện
 - Nhu cầu bộ nhớ mà thuật toán sử dụng



Tính tổng quát



Thuật toán có tính tổng quát cao nếu có thể giải bất kỳ bài toán nào trong một lớp lớn các bài toán

Ví dụ

Thuật toán giải phương trình $ax^2+bx+c=0$ tổng quát hơn thuật toán giải phương trình $x^2+5x+6=0$



Các cách biểu diễn thuật toán



1. Ngôn ngữ tự nhiên: người - người
2. Lưu đồ thuật toán: sử dụng ký pháp
3. Mã giả: mô phỏng ngôn ngữ lập trình
4. Ngôn ngữ lập trình: chương trình máy tính



Biểu diễn bằng Ngôn ngữ tự nhiên



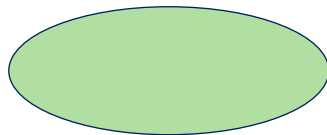
- Nguyên tắc:
 - Sử dụng ngôn ngữ tự nhiên để liệt kê các bước của thuật toán
- Đặc điểm
 - Không yêu cầu phải có một số kiến thức đặc biệt
 - Dài dòng,
 - Không làm nổi bật cấu trúc của thuật toán.

- **Bài toán:** giải phương trình bậc nhất

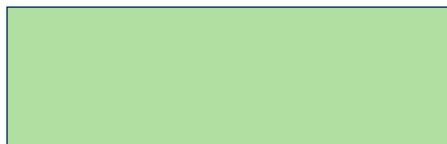
$$ax + b = 0$$

- **B1:** Nhập a và b.
- **B2:** Nếu $a \neq 0$ thì hiển thị "Phương trình có 1 nghiệm duy nhất $x = -b/a$ ". Kết thúc.
- **B3:** ($a=0$) Nếu $b \neq 0$ thì hiển thị "Phương trình vô nghiệm". Kết thúc.
- **B4:** ($a=0$)($b=0$) Hiển thị "Phương trình vô số nghiệm". Kết thúc.

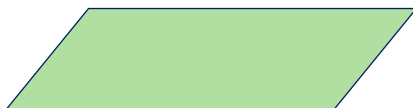
■ Ký pháp



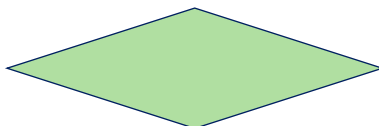
Bắt đầu hoặc kết thúc



Thao tác tính toán



Lệnh vào, lệnh ra



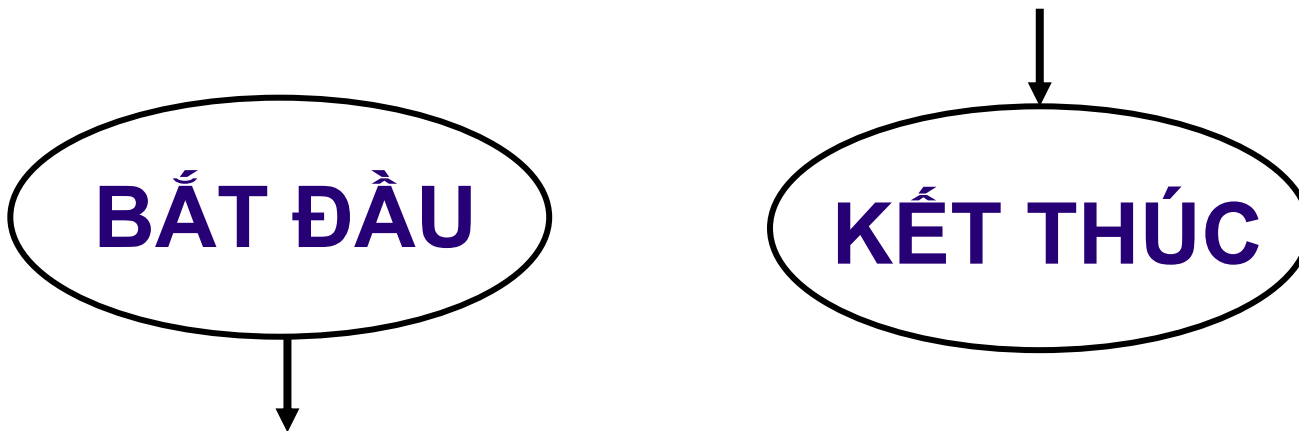
Kiểm tra điều kiện, rẽ nhánh



Luồng thực hiện

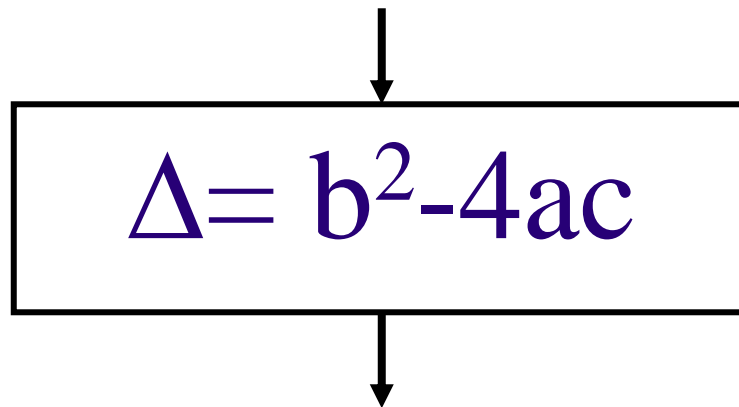
▪ Khối giới hạn:

- 2 loại khối giới hạn: đầu và cuối
- Ghi rõ điểm bắt đầu và kết thúc (dừng) của thuật toán



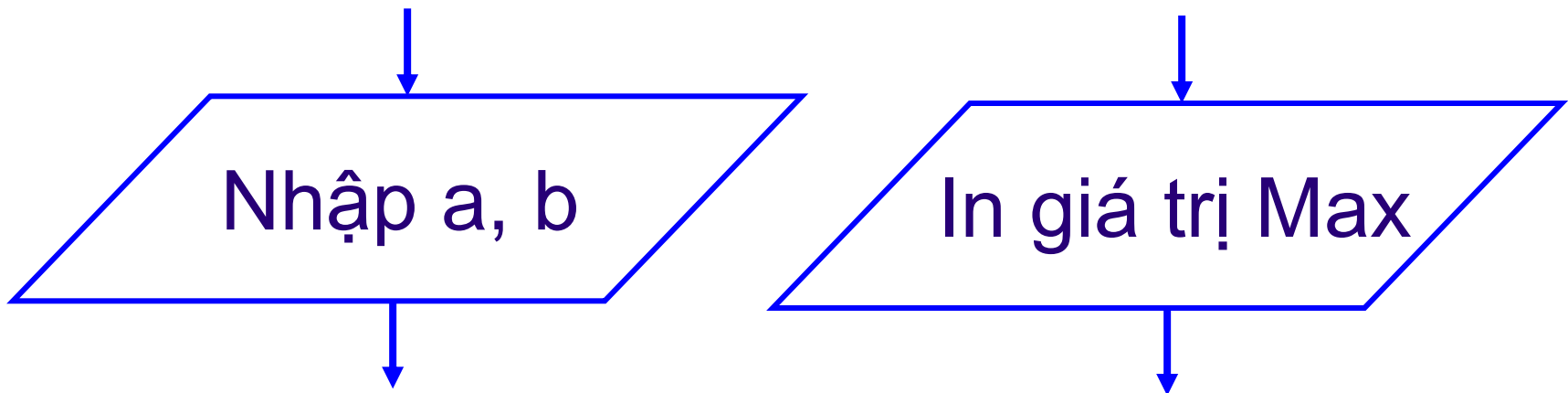
▪ Khối thao tác, tính toán

Hình chữ nhật chứa dãy các lệnh xử lý dữ liệu



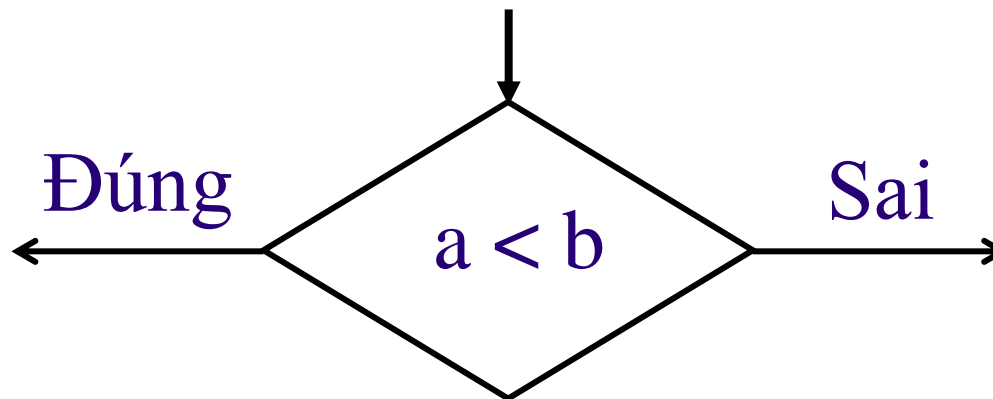
▪ Khối vào/ra dữ liệu

Hình bình hành chứa thao tác nhập/ xuất dữ liệu



▪ Khôi điều kiện

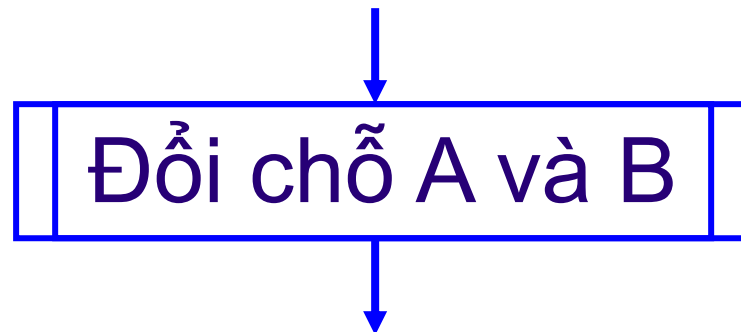
Là hình thoi chứa một điều kiện/biểu thức logic cần kiểm tra, một luồng vào, và hai luồng ra.



▪ Khối gọi chương trình con

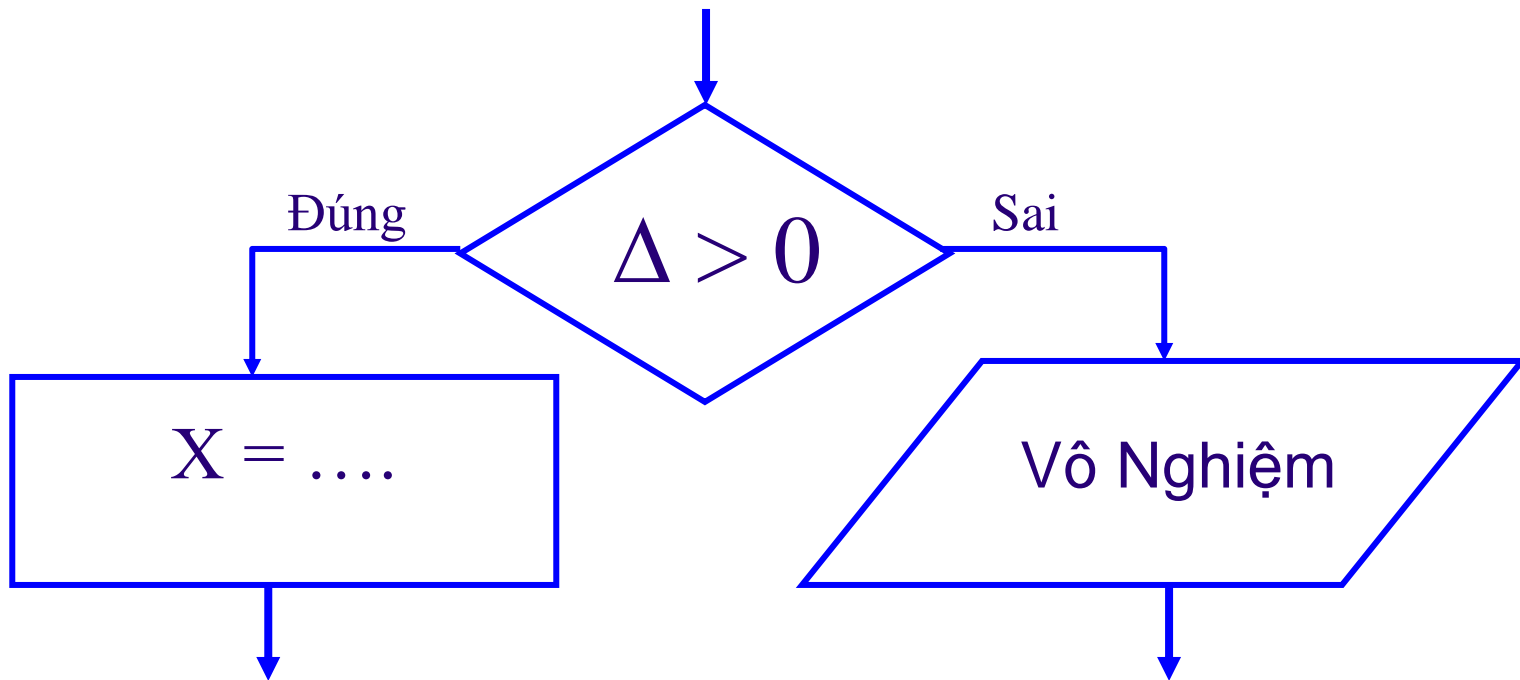
Là hình chữ nhật, cạnh kép chứa tên một chương trình con cần thực hiện

- Chương trình con: Thuật toán đã biết
- Nhằm giảm độ phức tạp, tăng tính cấu trúc của sơ đồ



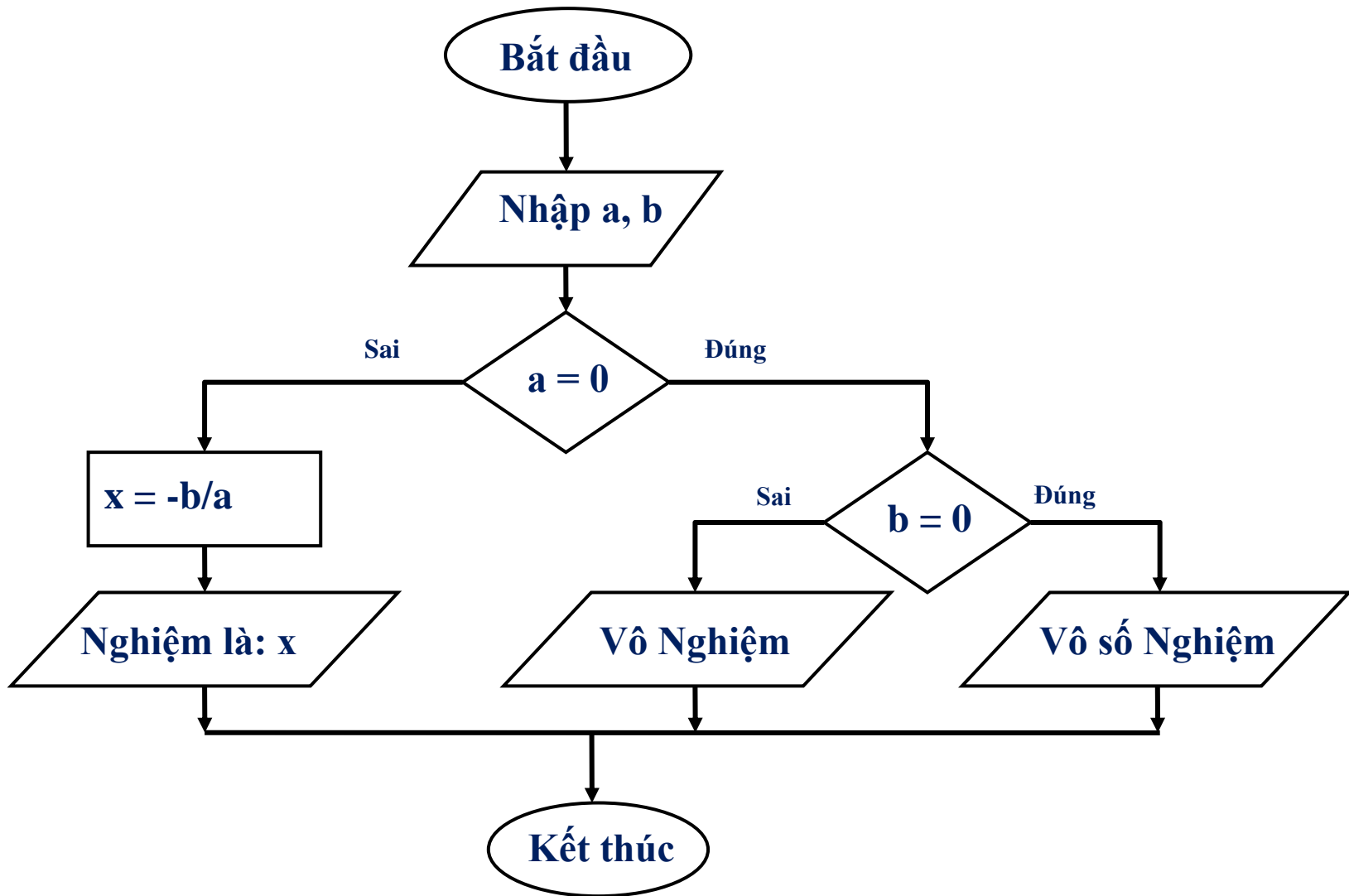
▪ Cung:

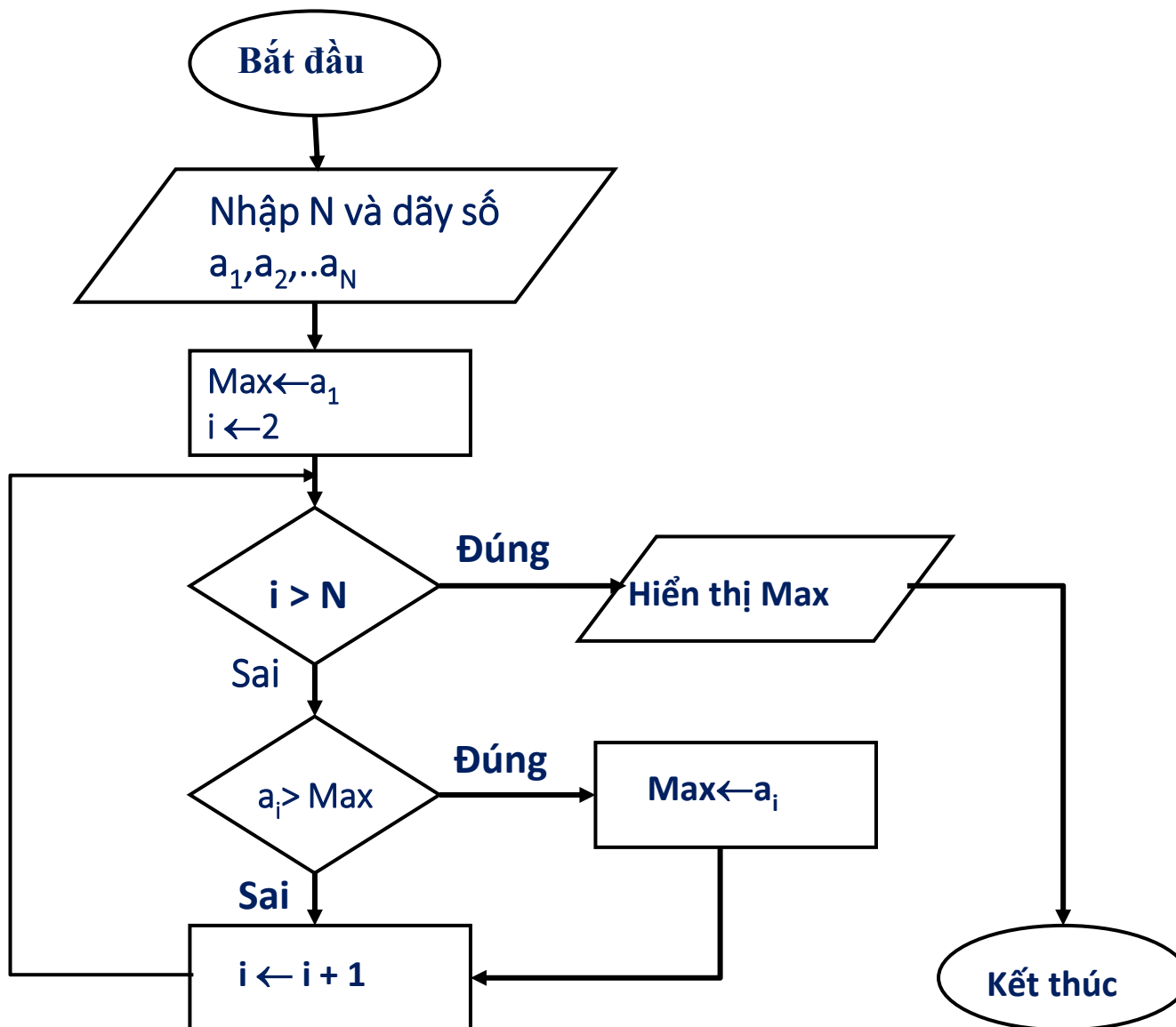
Là các đường nối từ nút này đến nút khác của lưu đồ



▪ Hoạt động

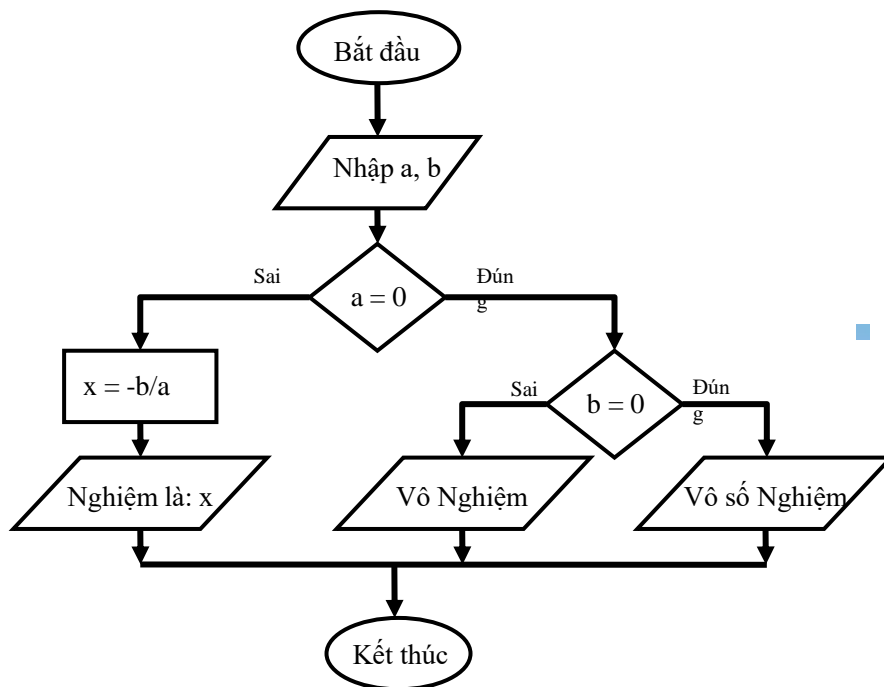
- Bắt đầu từ khối giới hạn đầu tiên (bắt đầu).
- Đi theo luồng thực thi (mũi tên).
- Thực hiện các thao tác được ghi trong nút
- Nếu là nút điều kiện: kiểm tra điều kiện rồi đi theo luồng thực thi tương ứng.
- Thuật toán sẽ dừng khi gặp nút kết thúc





- Mô tả thuật toán theo ngôn ngữ phỏng theo ngôn ngữ lập trình
 - Sử dụng các mệnh đề có cấu trúc chuẩn hóa
 - Vẫn có thể dùng ngôn ngữ tự nhiên.
 - Có thể sử dụng các ký hiệu toán học
 - Có thể sử dụng cấu trúc kiểu thủ tục để trình bày thuật toán đệ quy/ thuật toán phức tạp..
- Đặc điểm:
 - Tiện lợi, đơn giản, và dễ hiểu.
- Các cấu trúc thường gặp:
 - Gán, lựa chọn, lặp, nhảy, gọi hàm

Giải phương trình $ax + b = 0$



```
#include <stdio.h>
int main(){
    float a,b;
    printf("Cho a = "); scanf("%f",&a);
    printf("Cho b = "); scanf("%f",&b);
    if(a==0)
        if (b==0)
            printf("Vo so nghiem");
        else
            printf("Vo nghiem");
    else
        printf("Nghiem %5.2f",-b/a);
    return 0;
}
```

➔ Sẽ học trong nội dung tiếp theo của học phần

Mô tả các thuật toán (sơ đồ khối) sau:

1. Đếm số chẵn của một dãy N số nguyên
2. Đưa ra trung bình cộng các số dương của một dãy gồm N số
3. Tìm số lớn nhất trong 1 dãy gồm N số nguyên, và đếm số phần tử có giá trị lớn nhất.
4. Giải phương trình bậc 2 (xét cả bậc 1)
5. Tính tổng n số nguyên liên tiếp. $A=1+2+\dots+n$