



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

NHẬP MÔN LẬP TRÌNH

INTRODUCTION TO COMPUTER PROGRAMMING

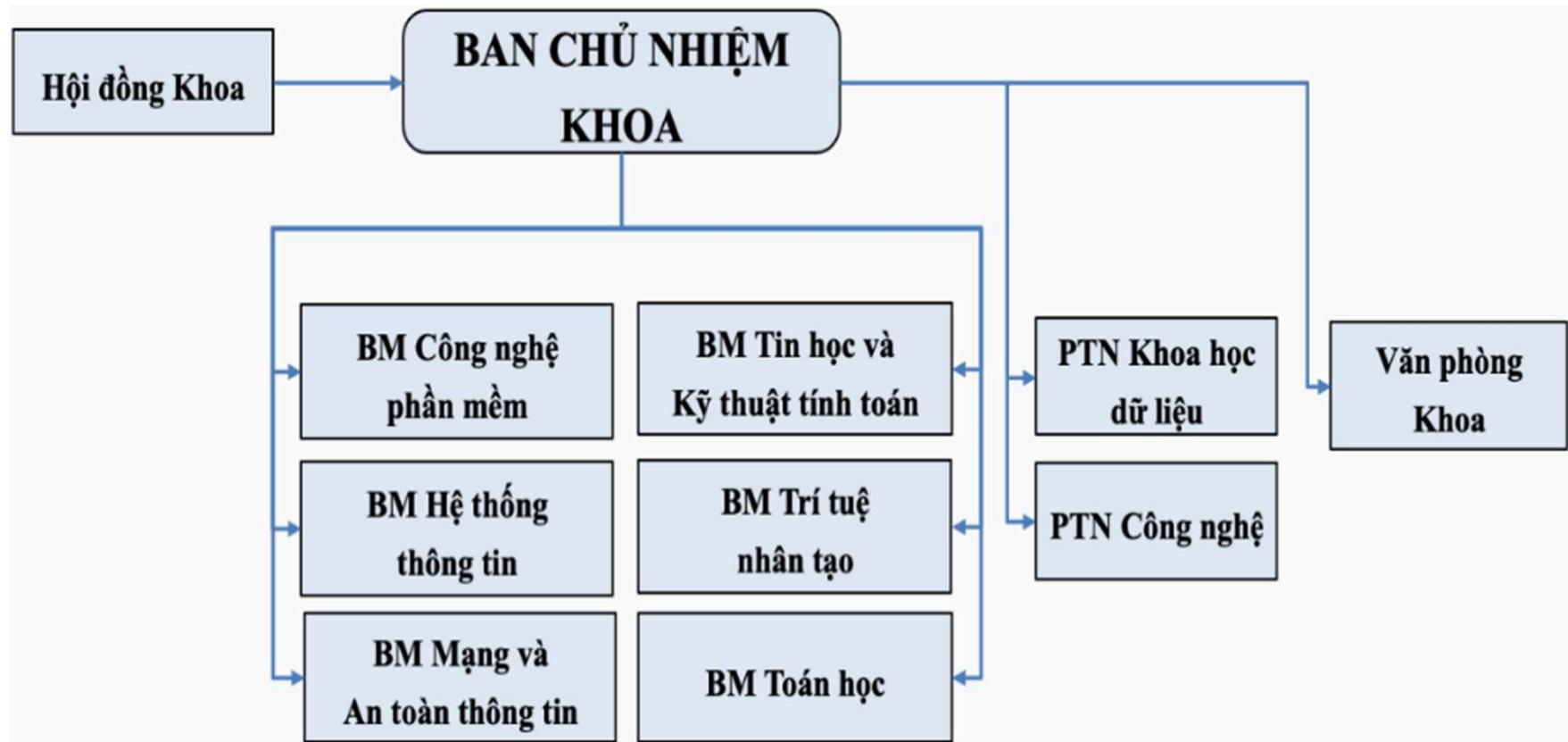
Giảng viên: TS.GVC Bùi Thị Thanh Xuân

Email: xuanbtt@tlu.edu.vn

Điện thoại: 0902001581



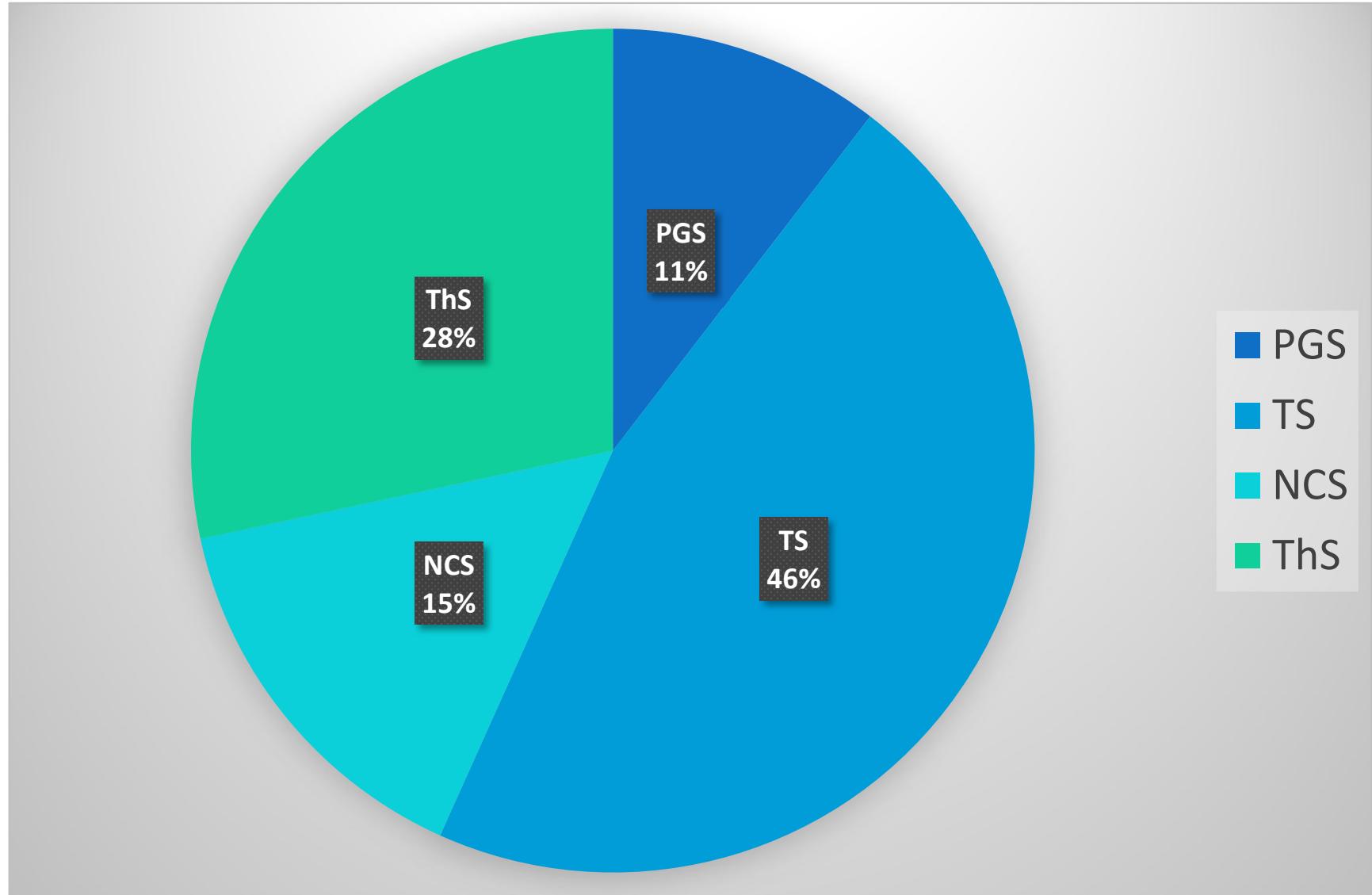
Giới thiệu về Khoa CNTT



Văn phòng Khoa: P201-Tầng 2 Nhà C1
Website: <https://cse.tlu.edu.vn/>



Giới thiệu đội ngũ Khoa CNTT





Các ngành đào tạo của Khoa CNTT



Ngành Công nghệ thông tin



Ngành Kỹ thuật phần mềm



Ngành Hệ thống thông tin



Ngành Trí tuệ nhân tạo và KHDL



Ngành An ninh Mạng

DỊCH VỤ IT NỔI BẬT CỦA VIỆT NAM



Thương Mại Điện tử
(E-Commerce)



Công nghệ Tài Chính
(Fintech)



Gọi xe/
thức ăn



Xuất bản
phần mềm



Truyền thông trực tuyến,
Nội dung số



Du lịch
trực tuyến



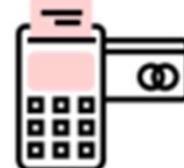
Vận chuyển
hàng hóa



Business
Process Outsourcing
(BPO)



High-tech
(AI/ML, IoT,
Blockchain...)



Business service
(SaaS,...)



Edtech



Healthcare

Theo một báo cáo mới nhất về Nền kinh tế số Đông Nam Á năm 2019 Dự kiến nền kinh tế số của khu vực sẽ vượt mức 100 tỷ USD trong năm nay và sẽ sớm tăng gấp ba vào năm 2025. Đông Nam Á có khả năng trở thành một trong những thị trường phát triển nhanh nhất về thương mại điện tử nhờ dân số am hiểu công nghệ, đặc biệt là sử dụng smartphone ngày càng gia tăng. Theo khảo sát của TopDev, trong năm 2020, thị trường tại Việt Nam có sự phản hồi tốt và phát triển mạnh trong 12 lĩnh vực chính như: E-commerce, Fintech, Gọi xe/thức ăn, Edtech, Healthcare, v.v...



Tình hình Công nghệ thông tin



SỐ LIỆU NGÀNH ICT VIỆT NAM

XUẤT KHẨU ICT

91,5 ^(*) TỶ USD
(ƯỚC TÍNH 2019)

KIM NGẠCH XUẤT KHẨU PHẦN MỀM

3,5 ^(**) TỶ USD (2018)

DOANH NGHIỆP HOẠT ĐỘNG TRONG ICT

50.000 DOANH NGHIỆP
(2018)

VỐN ĐẦU TƯ NƯỚC NGOÀI 2018

LĨNH VỰC KHOA HỌC CÔNG NGHỆ

2,1 ^(*) TỶ USD
(ƯỚC TÍNH 2019)

TỔNG DOANH THU CÔNG NGHIỆP ICT ^(*)

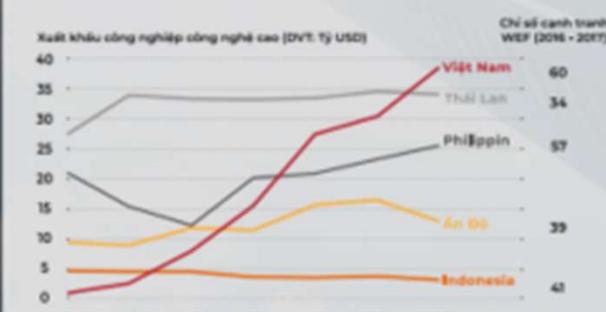
112,35 ^(*) TỶ USD
(ƯỚC TÍNH 2019)

DOANH THU CÔNG NGHIỆP PHẦN MỀM

5,0 ^(*) TỶ USD
(ƯỚC TÍNH 2019)

TỐC ĐỘ TĂNG TRƯỞNG CÔNG NGHIỆP PHẦN MỀM

13,8% ^(*)



Theo báo cáo tổng kết năm 2019 và định hướng năm 2020 của Bộ TT&TT, tổng doanh thu lĩnh vực công nghiệp ICT 2019 ước đạt 112,350 tỷ USD, trong đó xuất khẩu ICT chiếm 81,5%. Cũng theo đó, Bộ TT&TT cũng công bố rằng, doanh thu công nghiệp phần mềm đạt 5 tỷ USD, tăng 500 triệu USD so với năm 2018. Tổng số tiền nộp ngân sách nhà nước từ công nghiệp ICT năm 2019 là 54.000 tỷ đồng, tăng 2000 tỷ so với năm 2018. Tuy nhiên, với ngành công nghiệp nội dung số, hiện doanh thu của nó chỉ đang chiếm một phần rất nhỏ trong doanh thu ngành CNTT (chiếm 0,76% doanh thu ngành CNTT). Riêng ngành viễn thông tăng trưởng gần 19%, với sự đóng góp của 50.000 doanh nghiệp công nghệ. Công nghiệp CNTT duy trì tăng trưởng 10%.

Về những định hướng lớn mà ngành TT&TT tập trung triển khai trong năm 2020, Bộ trưởng Bộ TT&TT Nguyễn Mạnh Hùng cho biết: Năm 2020 ngành TT&TT xác định là năm chuyển đổi số quốc gia, là năm khởi động mạnh mẽ để tiến tới một Việt Nam số. Đây sẽ là một sự chuyển đổi sâu sắc và toàn diện, mà đầu tiên là sự chuyển đổi về phương thức, quy trình vận hành, về cách làm việc trong mọi lĩnh vực. Đây cũng được xem là một dấu hiệu đáng mừng, giúp Việt Nam tiến gần hơn đến việc trở thành một quốc gia IT có tầm vóc trong khu vực.



Tình hình Công nghệ thông tin



VIỆT NAM
QUỐC GIA IT

BÁO CÁO THỊ TRƯỜNG IT 2020

TOPDev

6

NHÂN LỰC IT
400.000

SINH VIÊN CNTT
TỐT NGHIỆP MỖI NĂM
50.000

TỪ HƠN 153 TRƯỜNG ĐÀO TẠO CNTT

THỨ HẠNG
LẬP TRÌNH VIÊN VIỆT NAM

SKILLVALUE
HẠNG 29

TRÊN TOÀN THẾ GIỚI TRONG BẢNG XẾP HẠNG
KỸ NĂNG LẬP TRÌNH VIÊN
CỦA SKILLVALUE REPORT (2018)

HACKERRANK
HẠNG 23

TRÊN TOÀN THẾ GIỚI TRONG BẢNG XẾP HẠNG
VỀ KỸ NĂNG LẬP TRÌNH VIÊN
CỦA HACKERRANK (2016)

TOPCODER
TOP 6

TRÊN TOÀN THẾ GIỚI TRONG BẢNG XẾP HẠNG
VỀ KỸ NĂNG LẬP TRÌNH VIÊN CỦA TOPCODER
(2016)

CÔNG BỐ 2 NGHIÊN CỨU KHOA HỌC
VỀ AI ĐẦU TIÊN

TRI NEURIPS - HỘI NGHỊ SỐ 1 TRÊN THẾ GIỚI VỀ
TRÍ TUỆ NHÂN TẠO TẠI CANADA TÙ
VIEN NGHIEN CUU TRI TUỆ NHÂN TẠO
VINAI RESEARCH

NỀN TẢNG NHÂN LỰC

VIỆT NAM
TOP 8

TRÊN THỦ HẠNG OECD'S PISA (2015)
VỀ KHÁ NĂNG CỦA HỌC SINH 15 TUỔI VỀ TOÁN, KHOA HỌC VÀ ĐỌC HIỂU
TRÊN CÁC NỀN KINH TẾ PHÁT TRIỂN NHƯ HÀN QUỐC, TRUNG QUỐC, HỒNG KÔNG

DỰ ÁN
150.000

HỌC SINH, SINH VIÊN VIỆT NAM HỌC LẬP TRÌNH MIỄN PHÍ
TRONG DỰ ÁN "LẬP TRÌNH TƯƠNG LAI CÙNG GOOGLE"

VIỆT NAM ĐẨY MẠNH GIÁO DỤC
S.T.E.M

SCIENCE - KHOA HỌC | TECHNOLOGY - CÔNG NGHỆ
ENGINEERING - KỸ THUẬT | MATHEMATICS - TOÁN HỌC

NỘI DUNG LẬP TRÌNH & THUẬT TOÁN ĐƯỢC DẠY TỪ
LỚP 8

TRONG MÔN TIN HỌC CHƯƠNG TRÌNH COPT MỚI

LÀN SÓNG ĐẦU TƯ CÔNG NGHỆ VÀ IT MẠNH MẼ

Trong thời gian sắp tới, Việt Nam sẽ là điểm đến của nhiều công ty tập đoàn công nghệ lớn trên thế giới. Phó chủ tịch phụ trách chuỗi cung ứng của Apple, ông Rory Sexton cho biết Apple sẵn sàng gặp gỡ, tìm hiểu cơ hội hợp tác với các doanh nghiệp ICT Việt Nam có mong muốn tham gia chuỗi cung ứng của Apple. Ngoài Samsung là một trong những tập đoàn vốn nước ngoài giúp đẩy mạnh nền kinh tế của Việt Nam, sự tham gia của Apple trong thời gian tới có thể sẽ còn tiếp thêm sức mạnh cho làn sóng đầu tư nước ngoài tại Việt Nam trong năm 2020. Riêng Samsung trong năm 2020, họ cho biết sẽ đầu tư thêm 300 triệu đô vào R&D tại khu vực Hà Nội, cần thêm 4,000 nhân lực cho lĩnh vực công nghệ trong thời gian tới.

Hindustan Computers Limited (HCL), một trong 3 công ty IT lớn nhất Ấn Độ (Top 5 công ty outsource trên thế giới) đã phát triển trung tâm của mình tại TP.HCM, cần thêm 10,000 kỹ sư trong 5 năm nữa. Công ty Axon Enterprise, một trong những công ty phát triển công nghệ cho việc hành pháp hàng đầu tại Mỹ cũng đã tập trung cơ sở phát triển công nghệ của mình tại TP.HCM Việt Nam.



Tình hình Công nghệ thông tin



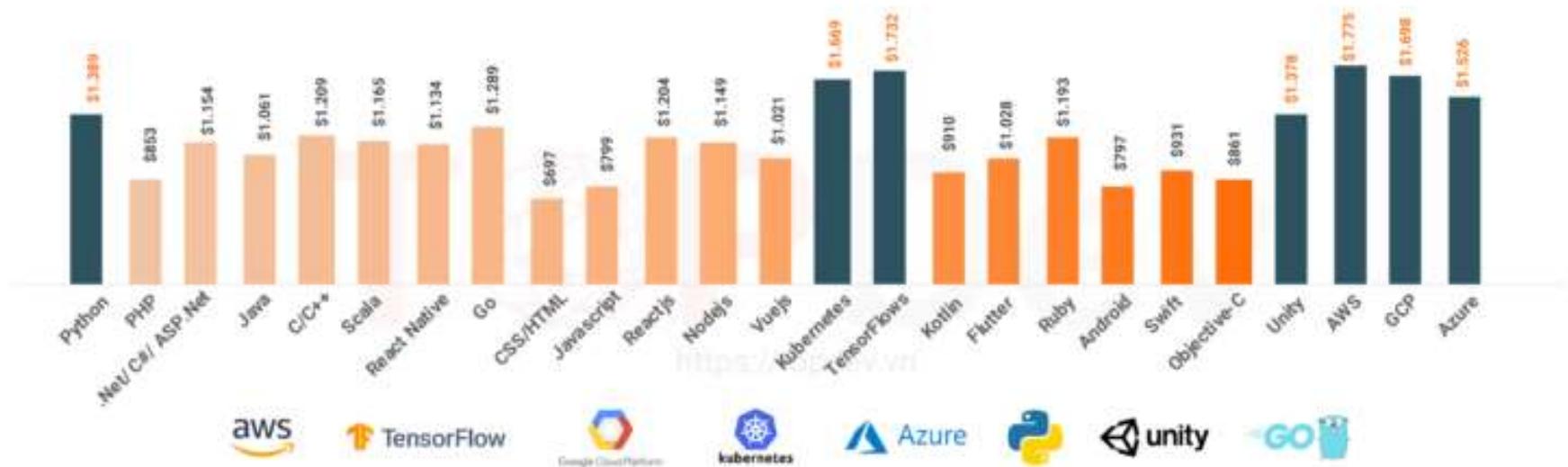
Mức lương lập trình viên @CÔNG NGHỆ

VIETNAM IT MARKET REPORT
TECH HIRING 2022

Mức lương lập trình viên theo công nghệ

(một cách tương đối dành cho đối tượng có tối đa 3 năm kinh nghiệm làm việc)

[đơn vị: USD]
Lưu ý: Tất cả dữ liệu tiền lương được nêu trong trang này để cập
đến tổng lương hàng tháng trước thuế (Gross) và không bao gồm
các lợi ích khác như làm thêm giờ, tiền thưởng, v.v.



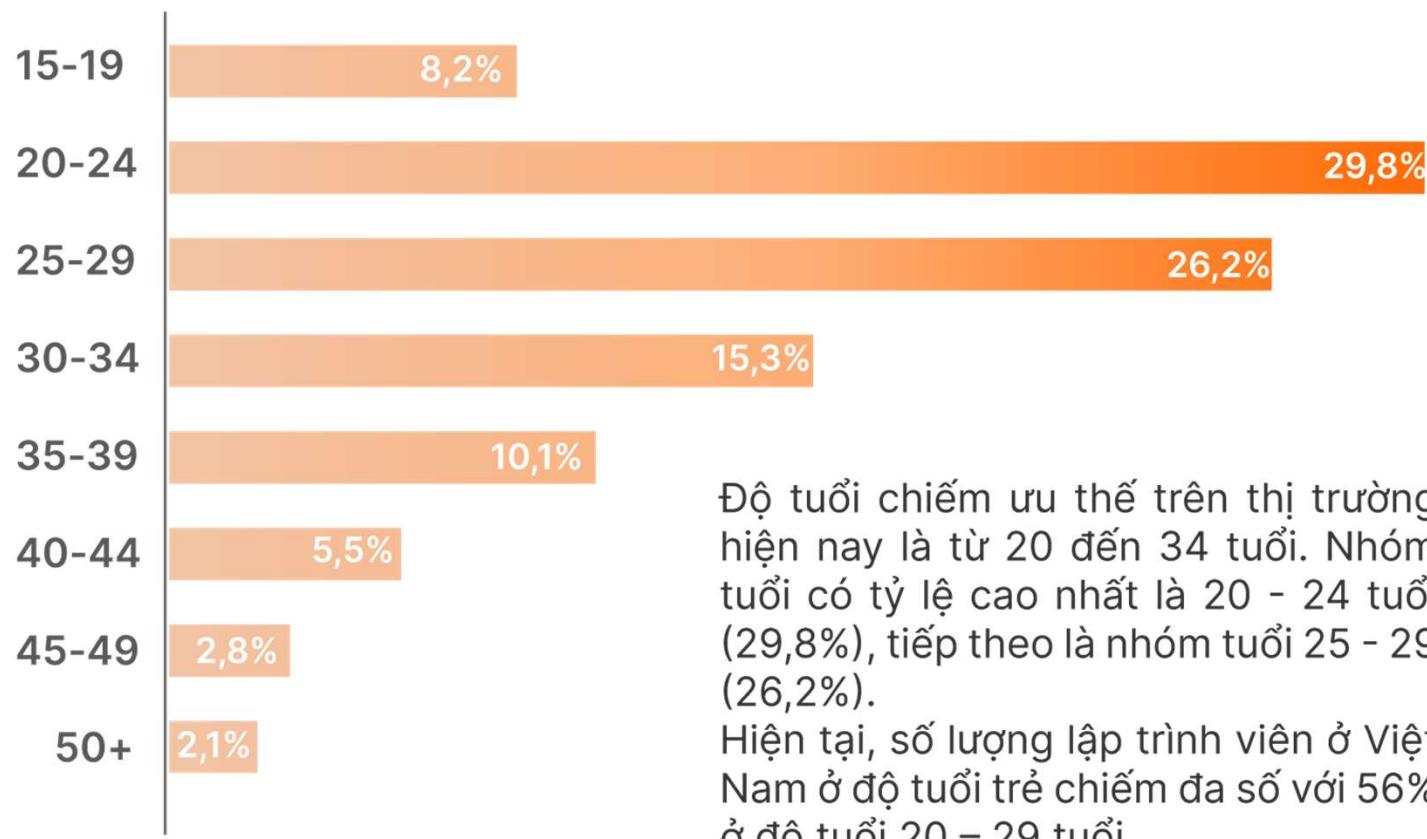
Top công nghệ được trả lương cao nhất trong danh sách có thể được chia thành 2 nhóm lớn: High-tech liên quan đến xu hướng AI / ML (Kubernetes, TensorFlow, Python) và Điện toán đám mây (AWS, GCP, Azure); công nghệ nối lên Unity với xu hướng play-to-earn & NFT ngày càng tăng. Chuyển đổi kỹ thuật số / tiền điện tử trên toàn thế giới cũng như tác động đáng kể từ Covid-19 đã tạo nên những con số. Theo sau danh sách hàng đầu, các công nghệ cơ bản liên quan đến phát triển web, hệ thống và thiết bị di động vẫn có thứ hạng cao đáng kể. Mặc dù có sự sụt giảm đáng kể về các kỹ năng kiểu cũ trên thiết bị di động so với đầu năm 2010 với làn sóng Mobile, các nền tảng mới dành cho Phát triển di động vẫn khẳng định được giá trị của mình với mức lương cao hơn so với các nền tảng công nghệ thông thường khác.



Tình hình Công nghệ thông tin 2023



TUỔI





Tình hình Công nghệ thông tin 2023



Bối cảnh tài năng công nghệ

VIETNAM
TECH TALENTS REPORT
2023 Vietnam IT Market Report By TopDev
- EXECUTIVE SUMMARY VERSION -

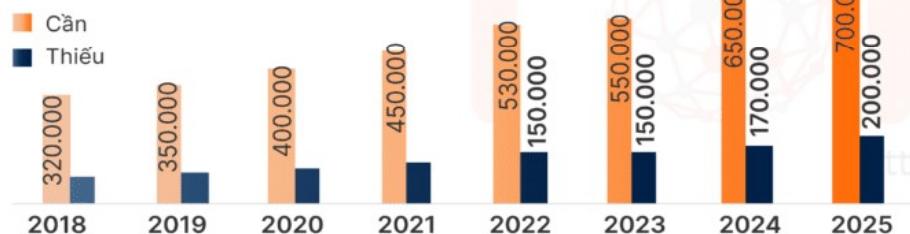
**LỰC LƯỢNG TECH/ DIGITAL
LÀM VIỆC TẠI
CÁC LĨNH VỰC LIÊN QUAN
ĐẾN CNTT**
530.000

Dựa trên tỷ lệ ước tính (1,0%+) của lực lượng lao động các ngành liên quan đến Công nghệ & CNTT trên tổng lực lượng lao động từ 15 tuổi trở lên.

**SINH VIÊN CNTT
NHẬP HỌC
HÀNG NĂM
(CHUYÊN NGÀNH PHẦN MỀM)**
50.000 - 57.000

Dựa trên báo cáo hàng năm của Bộ GD & ĐT từ 2019 - 2022 về tỷ lệ các chuyên ngành liên quan đến Công nghệ thông tin trên chỉ tiêu tuyển sinh đại học/cao đẳng phổ thông hàng năm.

NHU CẦU NHÂN LỰC IT TẠI VIỆT NAM 2018 - 2025 (NGÀNH PHẦN MỀM)



Sự thiếu hụt nhân sự CNTT luôn là vấn đề khó khăn nhất đối với thị trường CNTT. Mặc dù mức lương và tiền thưởng của ngành này đang tăng lên đáng kể nhưng dự đoán từ năm 2023 - 2025, Việt Nam vẫn sẽ thiếu hụt từ 150.000 đến 200.000 lập trình viên/kỹ sư hàng năm.

NĂNG LỰC LẬP TRÌNH VIÊN VIỆT NAM

SKILLVALUE

TOP 10 Sau Mỹ, Anh, Pháp...trong Xếp hạng Thế giới Khảo sát lập trình viên tốt nhất

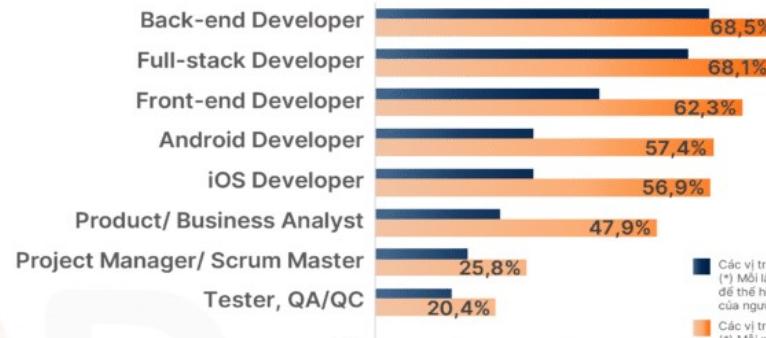
TOP 2 Chỉ sau Hoa Kỳ trong xếp hạng Thế giới về Khảo sát Freelancer tốt nhất

TopDev
Top IT Jobs For Developers

HACKERRANK
23RD

Bảng xếp hạng Developer Skills Charts of HackerRank's Report 2016

TOP VỊ TRÍ IT TUYỂN NHIỀU

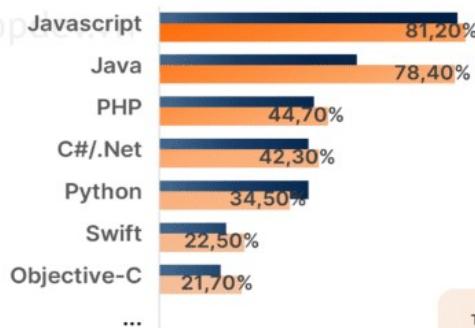


Các vị trí phổ biến của lập trình viên
(*) Mỗi lập trình viên có thể chọn nhiều phương án để thể hiện tốt nhất bản chất công việc của người được hỏi.

Các vị trí phổ biến để đăng tuyển
(*) Mỗi nhà tuyển dụng/ nhân sự có thể lựa chọn nhiều phương án để có thể hiện thực hóa tốt nhất bản chất của bản mô tả công việc.

Đa số các nhà tuyển dụng cho biết họ hầu như luôn cần các lập trình viên Back-end, Lập Trình Viên Full-stack và Lập Trình Viên Front-end.

TOP CÔNG NGHỆ TUYỂN NHIỀU



Các vị trí phổ biến của lập trình viên
(*) Mỗi lập trình viên có thể chọn nhiều phương án để thể hiện tốt nhất bản chất công việc

Các vị trí phổ biến để đăng tuyển
(*) Mỗi nhà tuyển dụng/ nhân sự có thể lựa chọn nhiều phương án để có thể hiện thực hóa tốt nhất bản chất của bản mô tả công việc.

Top 5 kỹ năng CNTT hàng đầu mà các công ty đang tìm kiếm vẫn giống như những năm trước: Javascript, Java, PHP, C#/.Net & Python. Gần 80% công ty đang tìm kiếm lập trình viên JavaScript.

Thông tin thêm trong phiên bản đầy đủ:
Công cụ làm việc của Lập trình viên
Hành vi và sở thích của Lập trình viên
Bảng xếp hạng các Lập trình viên Việt Nam
Tình hình lực lượng lao động công nghệ và kỹ thuật số

Scan để tải
bản đầy đủ





Tình hình Công nghệ thông tin 2023



Mức lương Lập trình viên

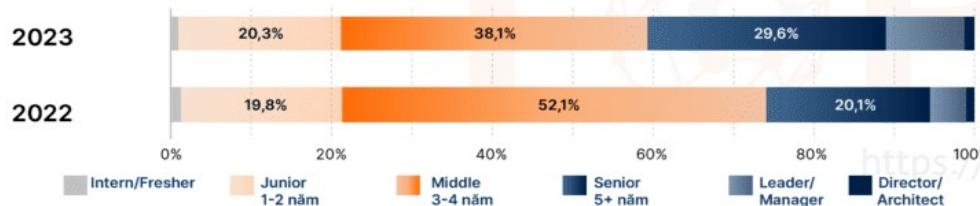
MỨC LƯƠNG THEO ĐỊA ĐIỂM

[đơn vị: tỷ lệ JD trên mỗi địa điểm %]

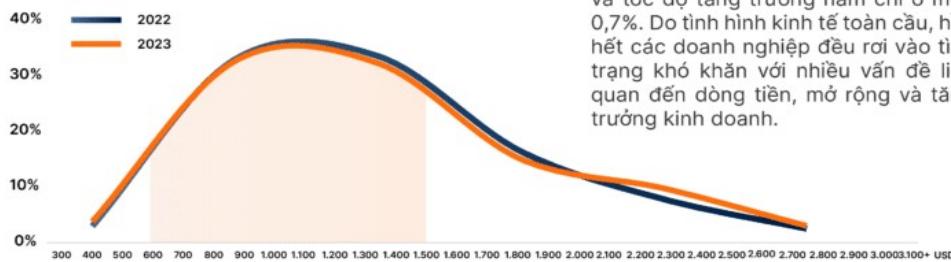


SỰ CHUYỂN DỊCH VỀ YÊU CẦU TRÌNH ĐỘ CHUYÊN MÔN

[đơn vị: tỷ lệ JD trên mỗi trình độ %]



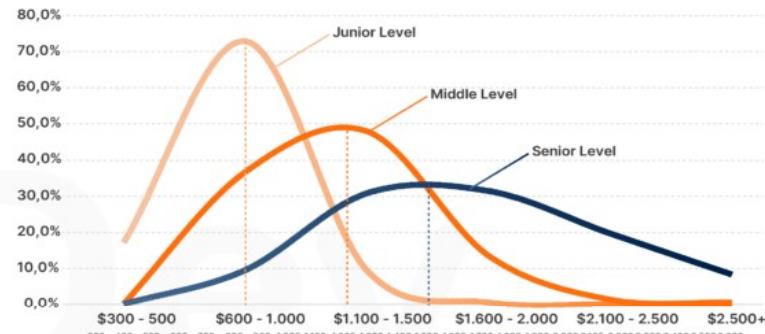
MỨC LƯƠNG TRUNG BÌNH



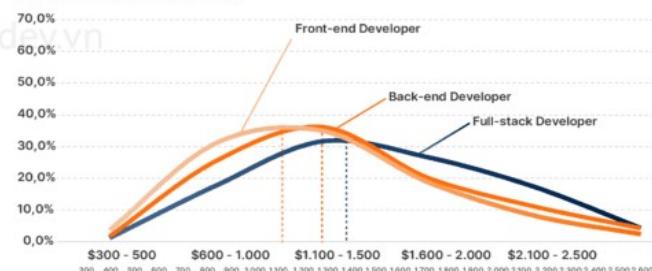
Rõ ràng, mức lương bình quân năm nay không thay đổi nhiều so với năm ngoái và tốc độ tăng trưởng năm chỉ ở mức 0,7%. Do tình hình kinh tế toàn cầu, hầu hết các doanh nghiệp đều rơi vào tình trạng khó khăn với nhiều vấn đề liên quan đến dòng tiền, mở rộng và tăng trưởng kinh doanh.

PHÂN BỐ LƯƠNG LẬP TRÌNH VIÊN THEO CẤP CHÍNH JUNIOR, MIDDLE & SENIOR

Hơn 70% vị trí Junior được cung cấp trong khoảng \$600 - \$1.000. Phạm vi chính cho cấp độ Trung bình là \$1.100 - \$1.500 với mức độ phổ biến gần 50%. Cấp độ cao cấp bao gồm phạm vi lương rộng hơn, hầu hết bắt đầu từ \$1.100 đến \$2.000. Khoảng 10% vị trí cấp cao đưa ra mức lương từ 2.500 USD.



WEB DEVELOPER



Thông tin thêm trong phiên bản đầy đủ:
 Lương lập trình viên theo công nghệ
 Lương lập trình viên theo cấp độ
 Lương lập trình viên theo vị trí công việc
 Phân bố lương lập trình viên của các vị trí phổ biến
 Top Việc Làm IT Phát Triển Nhanh Nhất 2023 - 2025

Scan để tải
bản đầy đủ





Một số xu hướng ngôn ngữ lập trình theo các định hướng



Best Programming Languages for **Mobile App Development**

- JAVA
- Kotlin
- Swift
- C++
- C#
- Objective-C
- Python
- HTML5
- JavaScript

Programming Languages used in **Artificial Intelligence**

- Python
- C++
- R
- Lisp
- Prolog
- Java
- Julia
- Haskell

Programming Languages for **Robotics**

- Python
- C/C++
- Java
- C#/.NET
- MATLAB
- LISP
- BASIC / Pascal
- Assembly Language

Programming Languages for **Machine Learning**

- Python
- C++
- JavaScript
- Java
- C#
- Julia
- Shell
- R
- TypeScript
- Scala

Programming Languages Used in **Game Development**

- C++
- C#
- JAVA
- Python
- Objective-C

Languages Used in **Web Development**

- HTML
- CSS
- JavaScript
- PHP
- ASP.NET
- Python
- Java
- Ruby
- SQL
- XML

Programming Languages Used in **Software Development**

- C
- C++
- C#
- JAVA
- Ruby
- VB.SCRIPT
- Python
- SQL

Lập trình C,
C++, Python
được cung
cấp cho SV
năm 1 là rất
cần thiết.



Giới thiệu môn học



- ❖ Số tín chỉ: 3 (2-0-1)
- ❖ Số tiết: 45 tiết (30 LT và 15 TH)
- ❖ Học phần bắt buộc cho ngành: CNTT, HTTT, KTPM, ANM, TTNT
- ❖ Điều kiện dự thi kết thúc HP: Nghỉ học không quá 20% số tiết (Nghỉ không quá 3 buổi)
- ❖ Trọng số điểm môn học:
 - Điểm quá trình: 50%
 - Điểm thi kết thúc HP: 50%



Giới thiệu môn học

❖ Trọng số điểm môn học

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 trên máy	2 lần	- 60 phút/bài (chấm code tự động trên máy) - Bài kiểm tra số 1 - Bài kiểm tra số 2		20% 20%
Tổng điểm quá trình				50%
Thi cuối kỳ	1 lần	- 60 phút - Thi chấm code tự động trên máy	1-2 tuần sau khi kết thúc môn học	50%



Công cụ hỗ trợ môn học



1. Hệ thống quản lý lớp học trực tuyến: Thực hiện điểm danh, cung cấp tài liệu môn học, nộp bài tập,...

The screenshot shows a web browser window titled "Elearning TLU". The address bar says "Not secure | lms.tlu.edu.vn". The main content area displays the "Hệ thong LMS_TLU" homepage, featuring a photo of two students and the text "Tự hào truyền thống - Kết nối sáng tạo - Vững bước tương lai". Below the header, there's a banner for "TRƯỜNG ĐẠI HỌC THỦ ĐỨC". The page includes sections for "Courses" and "Tài liệu tập huấn". Under "Tài liệu tập huấn", there are two items: "Tài liệu tập huấn sử dụng LMS_TLU cho giảng viên" and "Tài liệu hướng dẫn sử dụng LMS_TLU cho sinh viên".

Truy cập website:
<http://lms.tlu.edu.vn/>

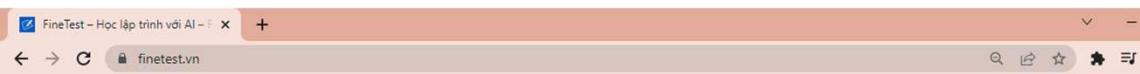
Sinh viên đăng nhập vào
hệ thống theo:
username: mã sinh viên
Pass: Tlu@mã sinh viên



Công cụ hỗ trợ môn học



2. Phần mềm thực hành/thi trực tuyến



Phần mềm FineTest: Phiên bản 4

Bản cài đặt mới nhất

Tải về và chạy để cài đặt, chú ý tắt trình diệt virus (bao gồm cả Windows Defender) trước khi thực hiện cài đặt

[PHIÊN BẢN CHO WINDOWS 8 TRỞ LÊN \(67 MB\)](#)

[PHIÊN BẢN CHO WINDOWS 7 SP1 \(66 MB\)](#)

.NET Framework 4.8

Cần trong trường hợp máy tính cần nâng cấp phiên bản .NET (Windows 7, Windows 8, Windows 8.1)

[DOWNLOAD \(116 MB\)](#)

Phiên bản khả chuyển

(dành cho những bạn có chuyên môn) Phần mềm portable, đang nén zip, người dùng tải về và bung file nén ở thư mục tùy chọn

[PHIÊN BẢN CHO WINDOWS 8 TRỞ LÊN \(125 MB\)](#)

[PHIÊN BẢN CHO WINDOWS 7 SP1 \(126 MB\)](#)

Xem hướng dẫn cài đặt

Video hướng dẫn cài đặt, lưu ý các bước thực hiện, nếu bạn không thực hiện chính xác, có thể gặp các lỗi tiềm ẩn khi sử dụng

[MỞ YOUTUBE](#)

Truy cập website:
<https://finetest.vn/>

Để tải phần mềm về
máy và cài đặt



Công cụ hỗ trợ môn học



3. Phần mềm lập trình Dev-C++

The screenshot shows a web browser window with the URL <https://sourceforge.net/projects/orwelldevcpp/>. The page is titled "Dev-C++ download | SourceForge". The SourceForge header includes links for "Open Source Software", "Business Software", and "Resources". Below the header, the project page for "Dev-C++" is displayed. It features the Dev-C++ logo, a brief description ("A free, portable, fast and simple C/C++ IDE"), and credit to "Brought to you by: orwelldevcpp". It shows a 5-star rating with 138 reviews, weekly downloads of 156,608, and download links for Windows and BSD. There are also buttons for "Get Updates" and "Share This".

Truy cập website:
<https://sourceforge.net/projects/orwelldevcpp/>

Để tải phần mềm về
máy và cài đặt



Nội dung chính của Học phần



- ❖ Chương 1: Máy tính và chương trình 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 chương trình máy tính



Chương 1. Máy tính và chương trình



- ❖ Các thể hệ máy tính
- ❖ Các hệ đếm, đơn vị đo thông tin
- ❖ Giải bài toán trên máy tính
- ❖ Thuật toán và các cách biểu diễn thuật toán



Máy tính và xử lý thông tin trên máy tính



1. Các khái niệm cơ bản về thông tin và tin học

- Thông tin và xử lý thông tin
- Máy tính điện tử và phân loại
- Tin học và các ngành liên quan

2. Biểu diễn dữ liệu trong máy tính

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



Thuật ngữ tin học



- ❖ Nguồn gốc từ tiếng Đức
 - Năm 1957 Karl Steinbuch đề xướng trong bài báo *Informatik: Automatische Informationsverarbeitung (Informatics: automatic information processing)*.
- ❖ Năm 1962, Philippe Dreyfus người Pháp gọi là “*informatique*”, tiếp theo là Walter F.Bauer cũng sử dụng tên này.
- ❖ Phần lớn các nước Tây Âu, đều chấp nhận.
 - Tại Anh, sử dụng thuật ngữ ‘**computer science**’, hay ‘**computing science**’,
- ❖ Nga cũng chấp nhận tên *informatika* (1966).



Tin học (informatic)



- 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)



- ❖ 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 Công nghệ thông tin



- ❖ 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, chuẩn 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*



Thông tin và xử lý thông tin



- ❖ Thông tin
- ❖ Dữ liệu
- ❖ Tri thức
- ❖ Hệ thống thông tin
- ❖ Xử lý thông tin



Thông tin



Thông tin (information) là gì ?

- Là khái niệm trừu tượng 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



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,...



Dữ liệu



- ❖ 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 → thông tin đang bị sốt
- 36.7°C → thông tin nhiệt độ cơ thể bình thường



Tri thức



▪ 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.



Đơn vị đo thông tin trong máy tính



❖ BIT (**B**I_nary digiT)

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

❖ Byte (**B**): $1B = 8$ bit

❖ Các đơn vị dẫn xuất

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



Mã hóa thông tin



❖ Phân loại thông tin

- Thông tin số: số nguyên, số thực
- Thông tin phi số: văn bản, hình ảnh, âm thanh, video

❖ **Mã hóa:** Mọi thông tin đưa vào máy tính đều được mã hóa bằng một chuỗi các ký hiệu 0 và 1 được gọi là dữ liệu nhị phân

- Các số → mã hóa → dữ liệu nhị phân
- Các ký tự → mã hóa → dữ liệu nhị phân
- Âm thanh → mã hóa → dữ liệu nhị phân
- Hình ảnh → mã hóa → dữ liệu nhị phân

❖ Các loại dữ liệu khác nhau được mã hóa theo các chuẩn qui ước



Hệ thống thông tin



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.

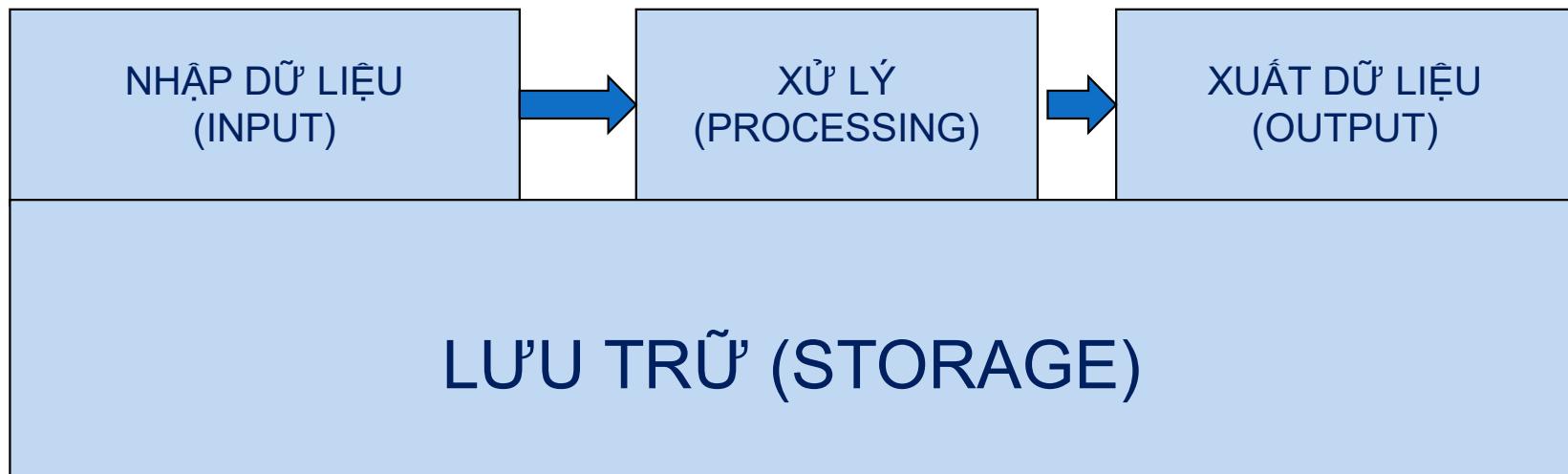




Xử lý thông tin



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 và phân loại



❖ Khái niệm

- Máy tính và chương trình máy tính
- Các thể hệ của máy tính

❖ Phân loại máy tính



Khái niệm

- **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.



Lịch sử hình thành và phát triển



Các thế hệ của máy tính điện tử

- ❖ 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



Von Neumann Machine (1950-1958)

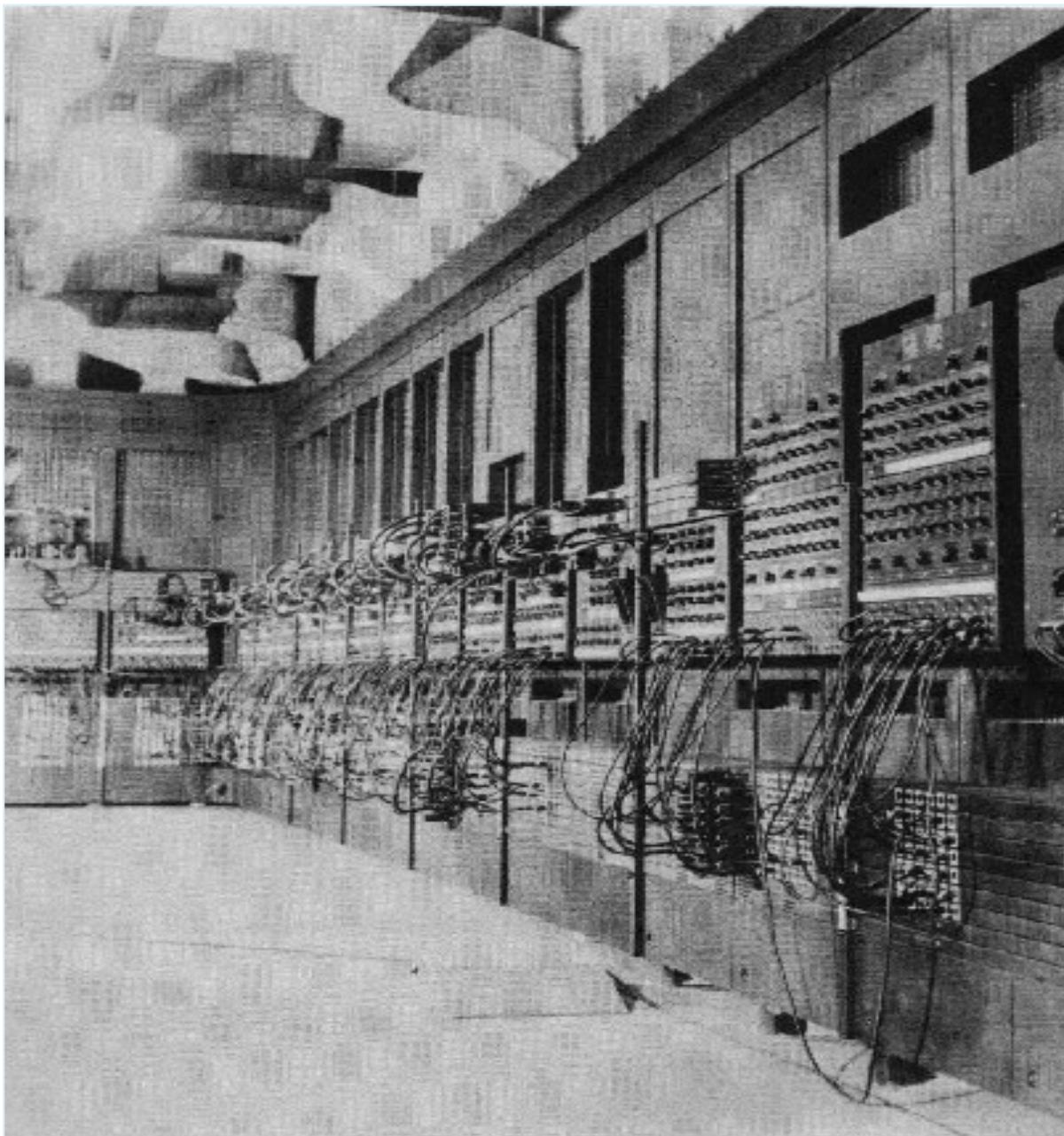
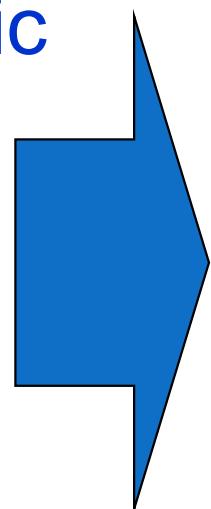
- Sử dụng các bóng đèn điện tử chân không
- Mạch riêng rẽ, vào số liệu bằng phiếu đục lỗ
- Điều khiển bằng tay, kích thước rất lớn
- Tiêu thụ năng lượng nhiều, tốc độ tính chậm khoảng 300 - 3.000 phép tính/s.

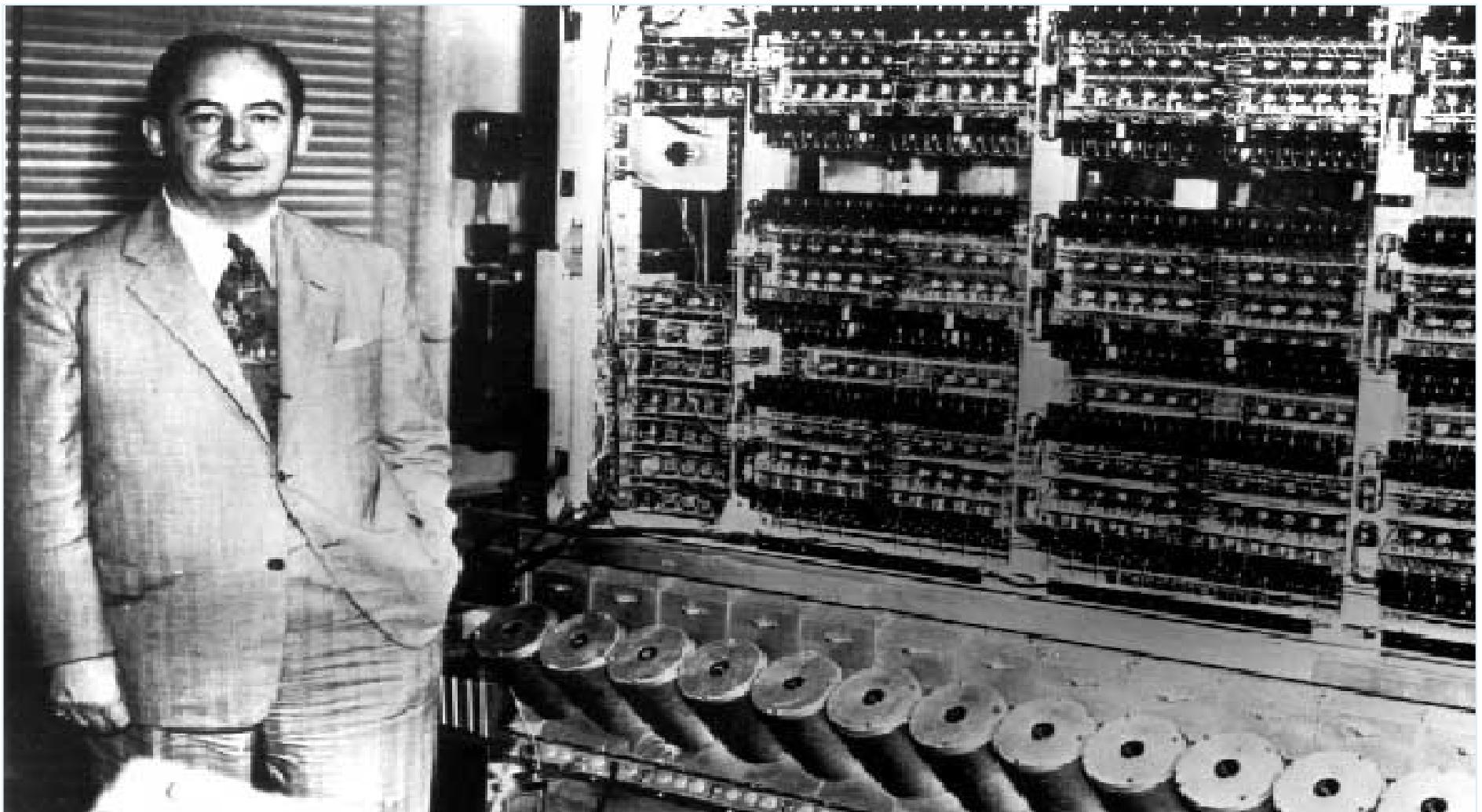
Ví dụ: EDVAC (Mỹ), BESEM (Liên xô cũ)



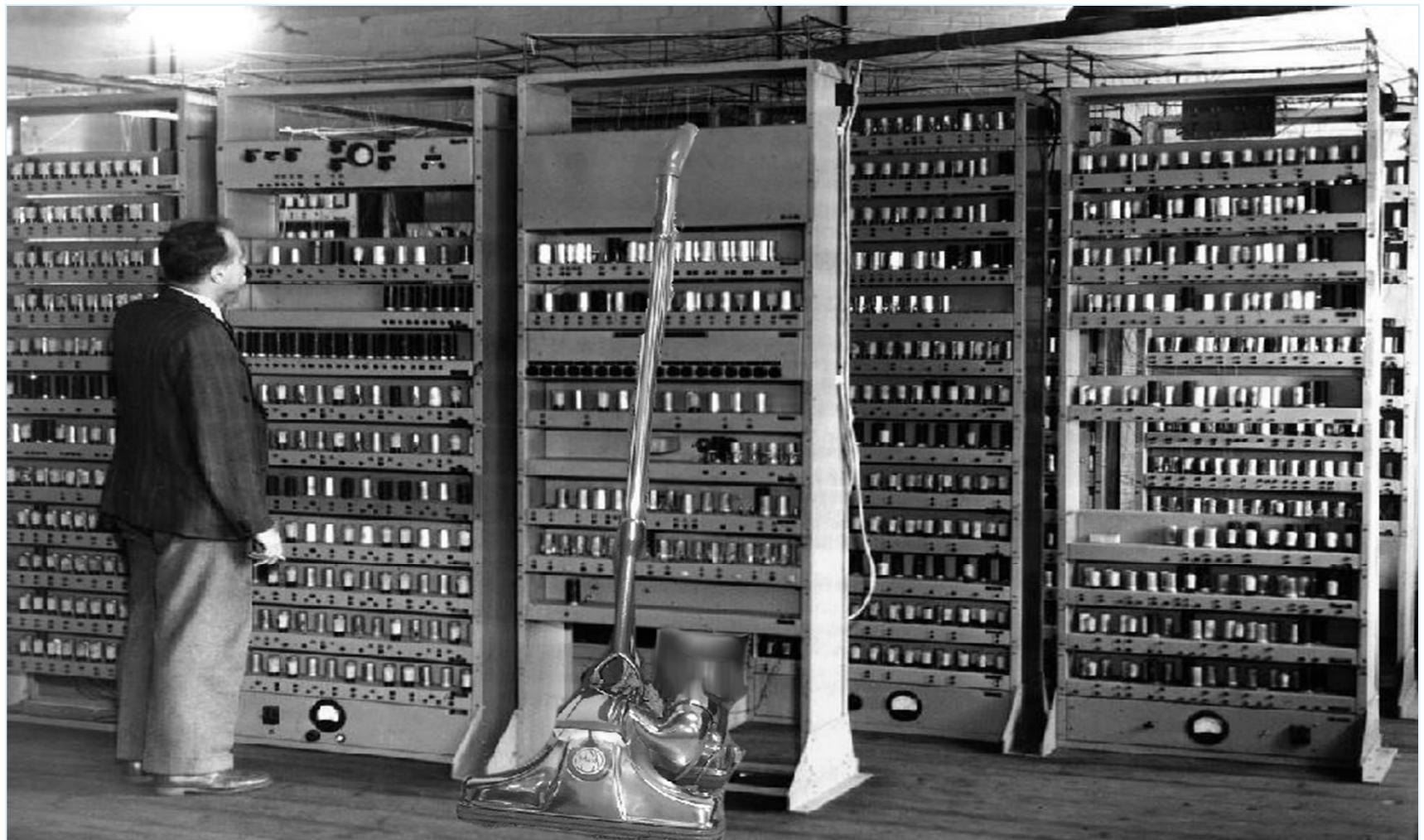
Bóng đèn chân không

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



Transistors (1958 - 1964):

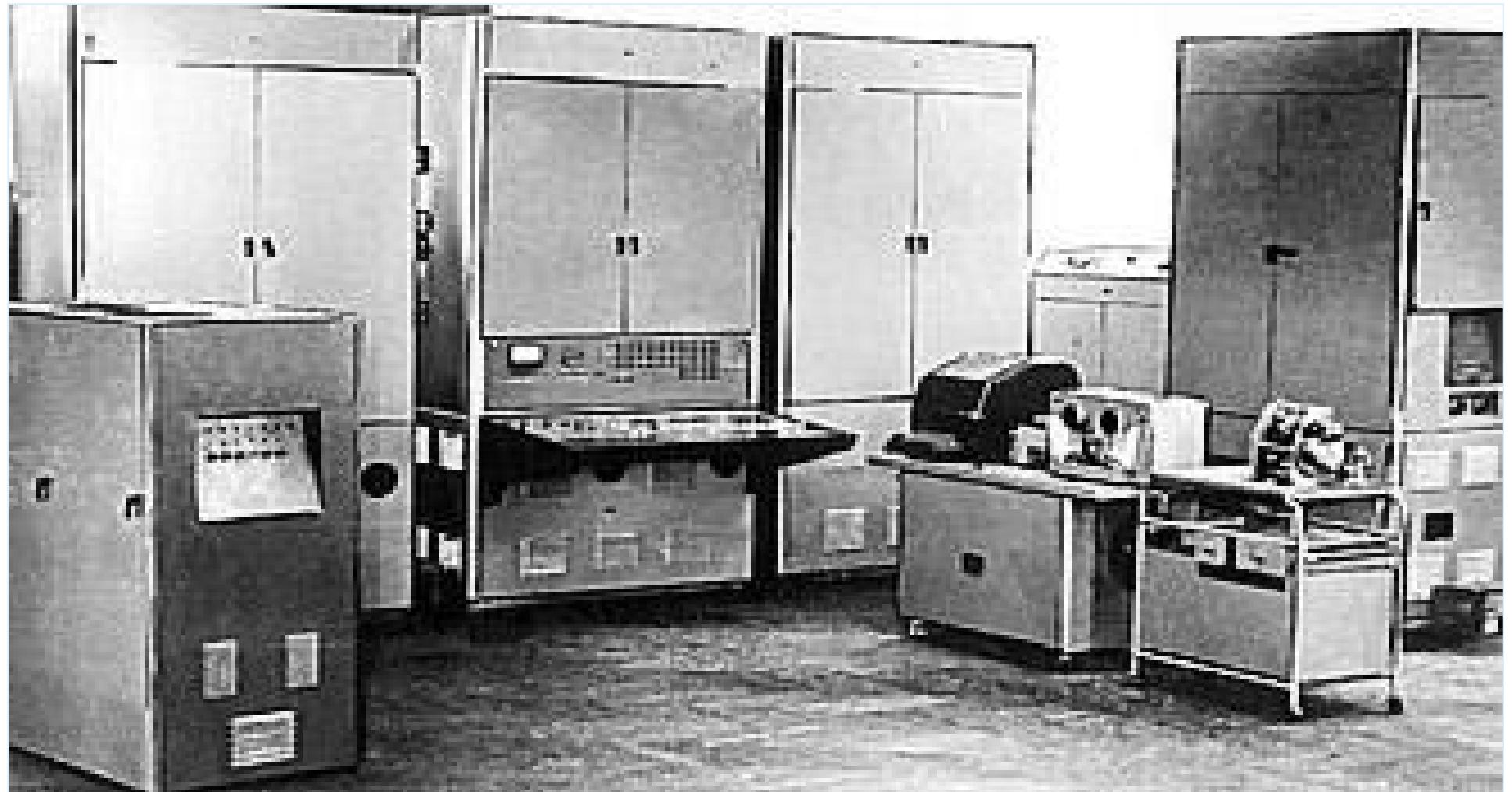
- Sử dụng **bộ xử lý bằng đèn bán dẫn, mạch in**
- Đã có chương trình dịch như Cobol, Fortran và hệ điều hành đơn giản.
- Kích thước máy còn lớn
- Tốc độ tính khoảng 10.000 - 100.000 phép tính/s

Ví dụ

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



IBM 7030



MINSK (Liên Xô cũ)



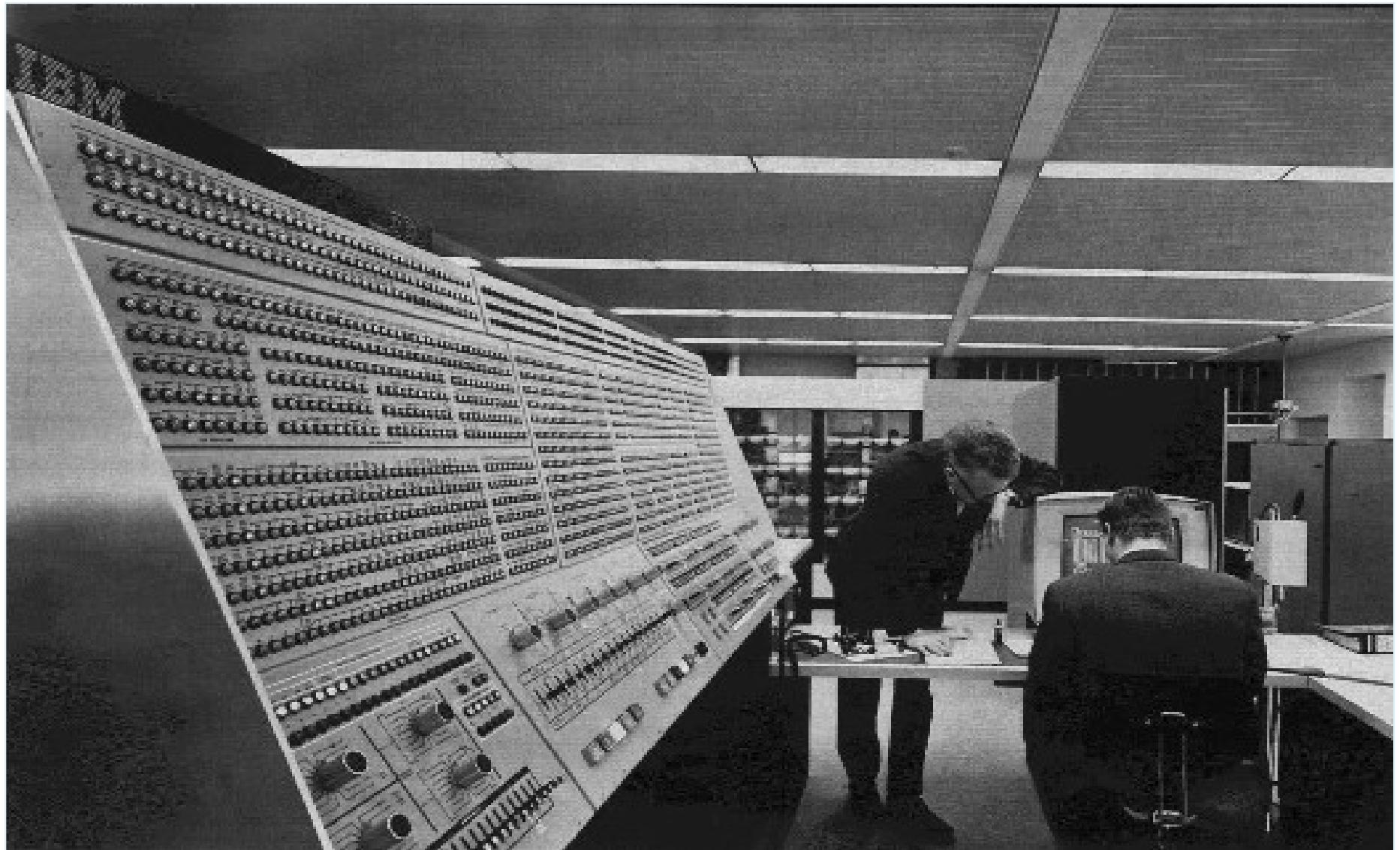
Thế hệ 3

Integrated Circuits (1965 - 1974):

- 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



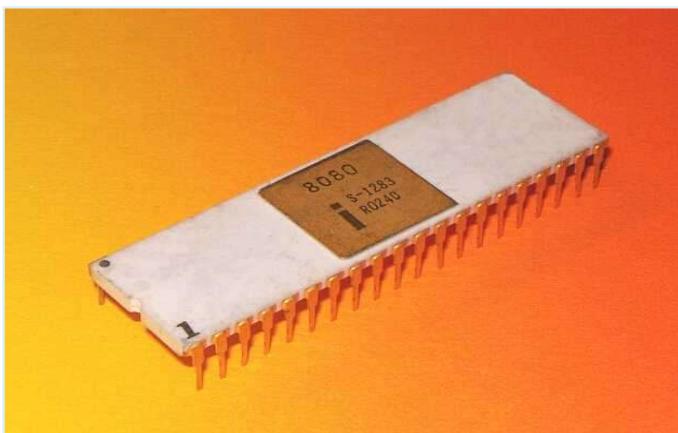
Thế hệ 4



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

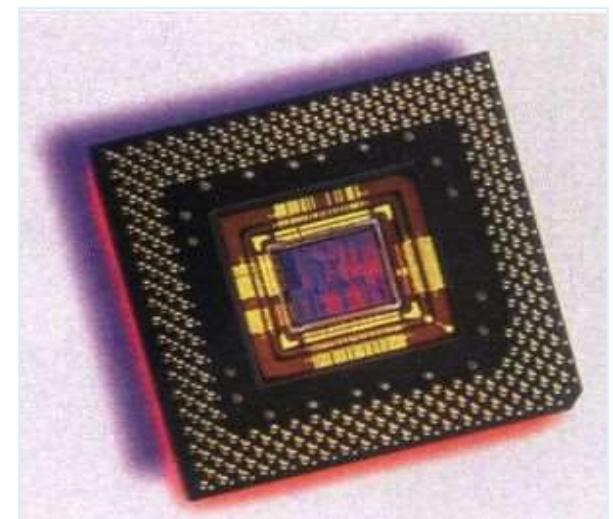
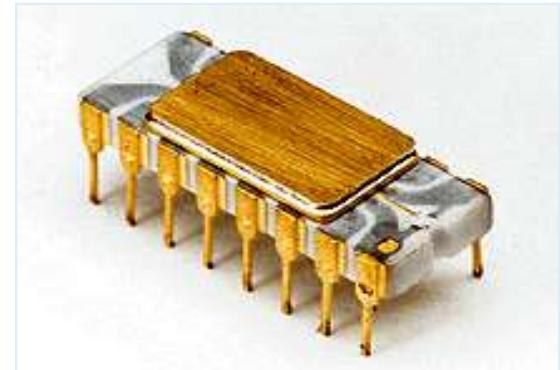
- Máy tính có các vi mạch đa xử lý
- Tốc độ: hàng chục triệu đến hàng tỷ phép tính/s.
- Hai loại máy tính chính:
 - Máy tính cá nhân để bàn (Personal Computer - PC) hoặc xách tay (Laptop hoặc Notebook computer)
 - Các loại máy tính chuyên nghiệp thực hiện đa chương trình, đa xử lý,...
- Hình thành các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



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.



Phân loại máy tính



❖ Cách phân loại truyền thống

- Máy vi tính (Micro-Computer)
- Máy tính nhỏ (Mini-Computer)
- Máy tính lớn (Mainframe Computer)
- Siêu máy tính (Super-Computer)

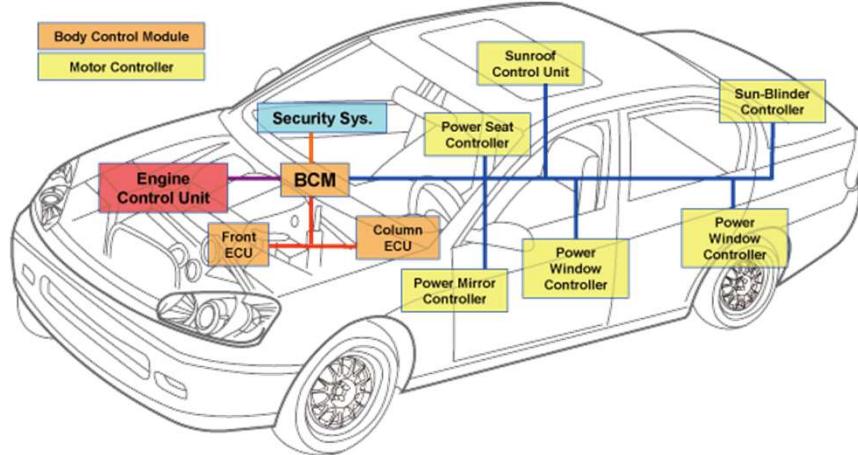


Phân loại máy tính hiện đại



- ❖ Thiết bị di động cá nhân (Personal Mobile Devices):
 - Smartphones, Tablet
- ❖ Máy tính cá nhân, máy trạm (PC, Workstation)
 - Desktop computers, Laptop computers
- ❖ Máy chủ (Servers)
 - Thực chất là Máy phục vụ
 - Dùng trong mạng theo mô hình Client/Server
- ❖ Máy tính cụm/máy tính qui mô lớn
 - Sử dụng tại các trung tâm tính toán, trung tâm dữ liệu
- ❖ Siêu máy tính
- ❖ Máy tính nhúng (Embedded Computers)
 - Đặt ẩn trong thiết bị khác
 - Được thiết kế chuyên dụng

Một số loại máy tính



Máy tính nhúng



PC



Server

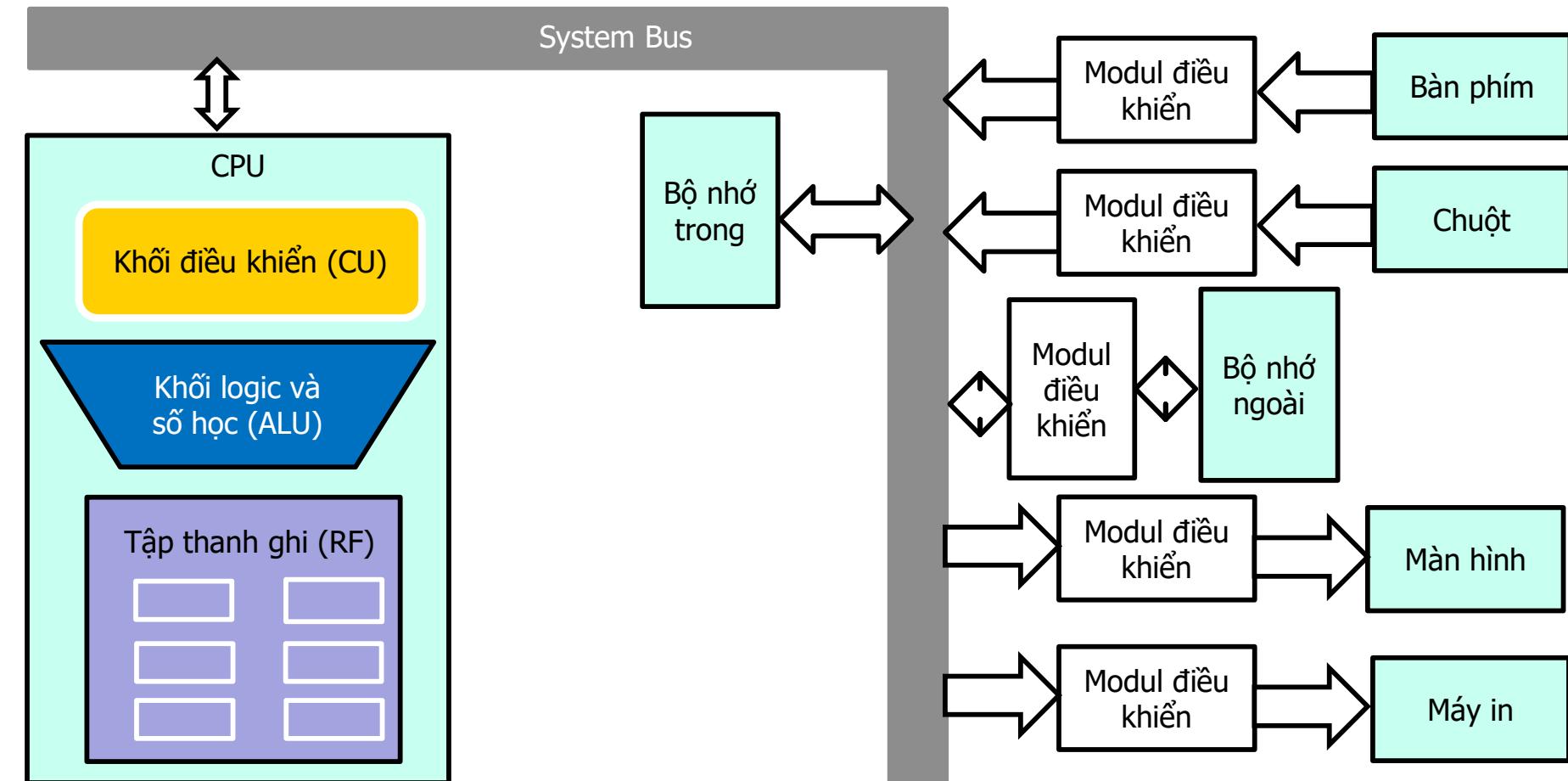


Super computer



Kiến trúc máy tính

Thiết bị đầu vào



Thiết bị đầu ra



Hoạt động của máy tính

- ❖ Hoạt động cơ bản của máy tính là thực hiện chương trình.
- ❖ Chương trình gồm một tập các lệnh được lưu trữ trong bộ nhớ.
- ❖ Việc thực hiện chương trình là lặp đi lặp lại chu trình lệnh bao gồm các bước:
 - CPU nhận lệnh từ bộ nhớ chính
 - CPU thực hiện lệnh đã nhận



Phần cứng máy tính



❖ Chức năng của các thành phần

➤ CPU

- Xử lý dữ liệu (Data processing)
- Điều khiển (Control)

➤ Bộ nhớ: Lưu trữ dữ liệu (Data storage)

- Bộ nhớ trong: trao đổi dữ liệu trực tiếp với CPU, gồm RAM và ROM
- Bộ nhớ ngoài: trao đổi dữ liệu gián tiếp với CPU thông qua module điều khiển vào ra, gồm đĩa cứng Hard Disk, CD-ROM, ...

➤ Thiết bị vào/ ra: Trao đổi thông tin giữa bên ngoài và bên trong máy tính

➤ System bus: vận chuyển dữ liệu (Data movement) giữa các thành phần



Bộ xử lý trung tâm - CPU



❖ Là bộ não của hệ thống

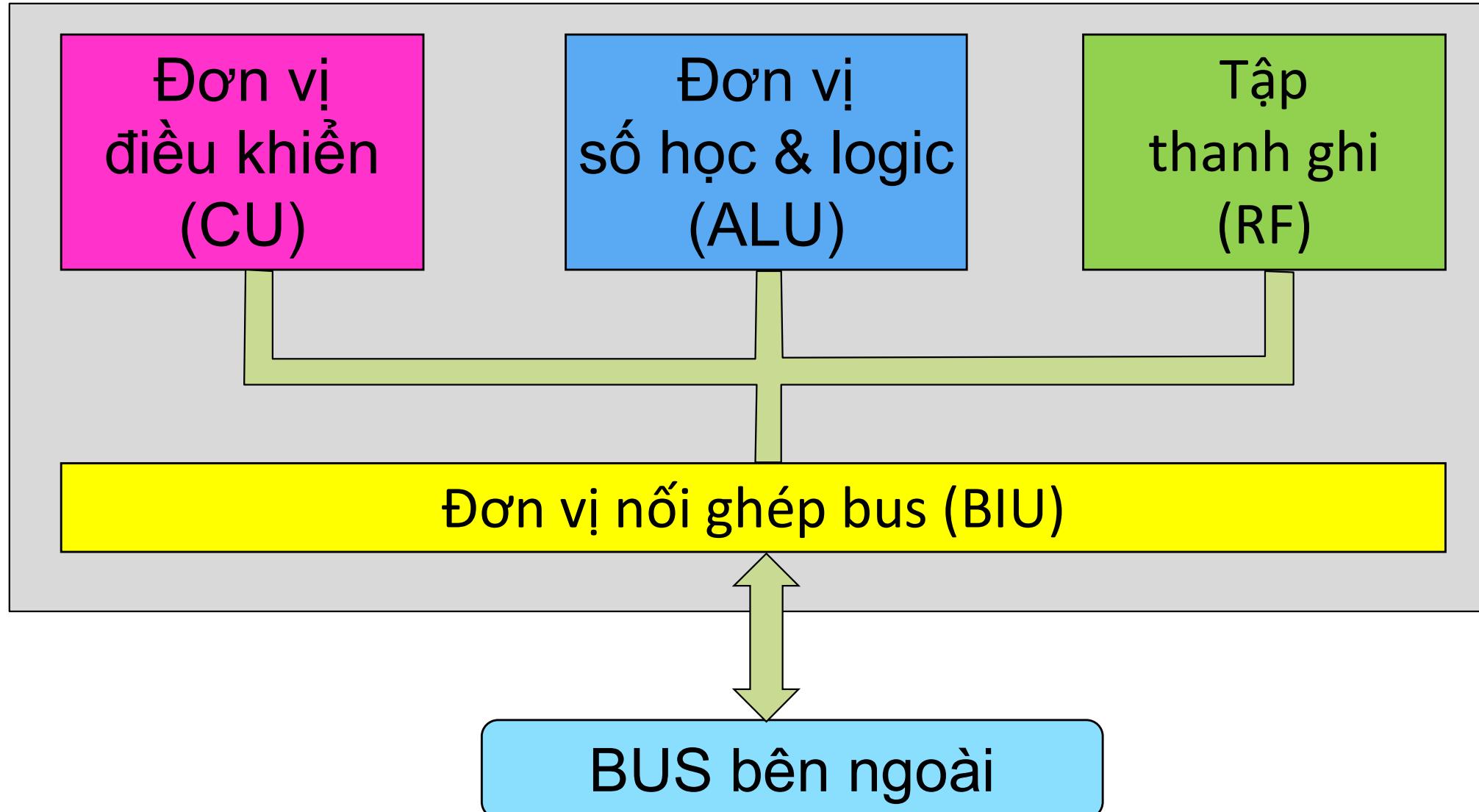
- Điều khiển hoạt động của toàn bộ hệ thống máy tính
- Xử lý dữ liệu

❖ Thực hiện đầy đủ chu trình lệnh của máy tính

- Nhận lệnh từ bộ nhớ chính
- Giải mã lệnh
- Nhận toán hạng
- Thực hiện lệnh
- Ghi kết quả



Cấu trúc cơ bản của CPU





Bộ xử lý trung tâm - CPU



- ❖ Khối điều khiển (*Control Unit – CU*): Điều khiển các thành phần còn lại của máy tính để thực hiện chu trình lệnh
- ❖ Khối tính toán số học và logic (*Arithmetic and Logic Unit - ALU*): Thực hiện các phép toán số học và logic
- ❖ Tập thanh ghi (*Register File - RF*): Lưu trữ các thông tin tạm thời phục vụ cho hoạt động của CPU
- ❖ Bus bên trong (Internal Bus):
 - Kết nối các thành phần bên trong CPU với nhau
 - Kết nối các lõi xử lý trong một bộ xử lý đa lõi
- ❖ Đơn vị ghép nối bus (Bus Interface Unit – BIU): Kết nối bộ xử lý với các bus bên ngoài để trao đổi thông tin giữa bộ xử lý với các thành phần còn lại (bộ nhớ, hệ thống vào ra).



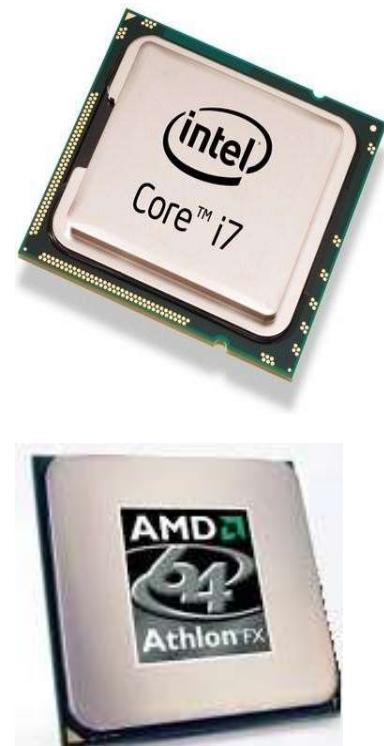
CPU- Central Processing Unit



- ❖ Tốc độ của CPU được đánh giá gián tiếp thông qua tần số xung nhịp cấp cho CPU

Bảng 1-2. Đơn vị đo tốc độ xung nhịp.

Kí hiệu	Tên gọi	Giá trị
Hz	Hertz	
KHz	Kilohertz	1000 Hz
MHz	Megahertz	1000 KHz
GHz	Gigahertz	1000 MHz
THz	Terahertz	1000 GHz



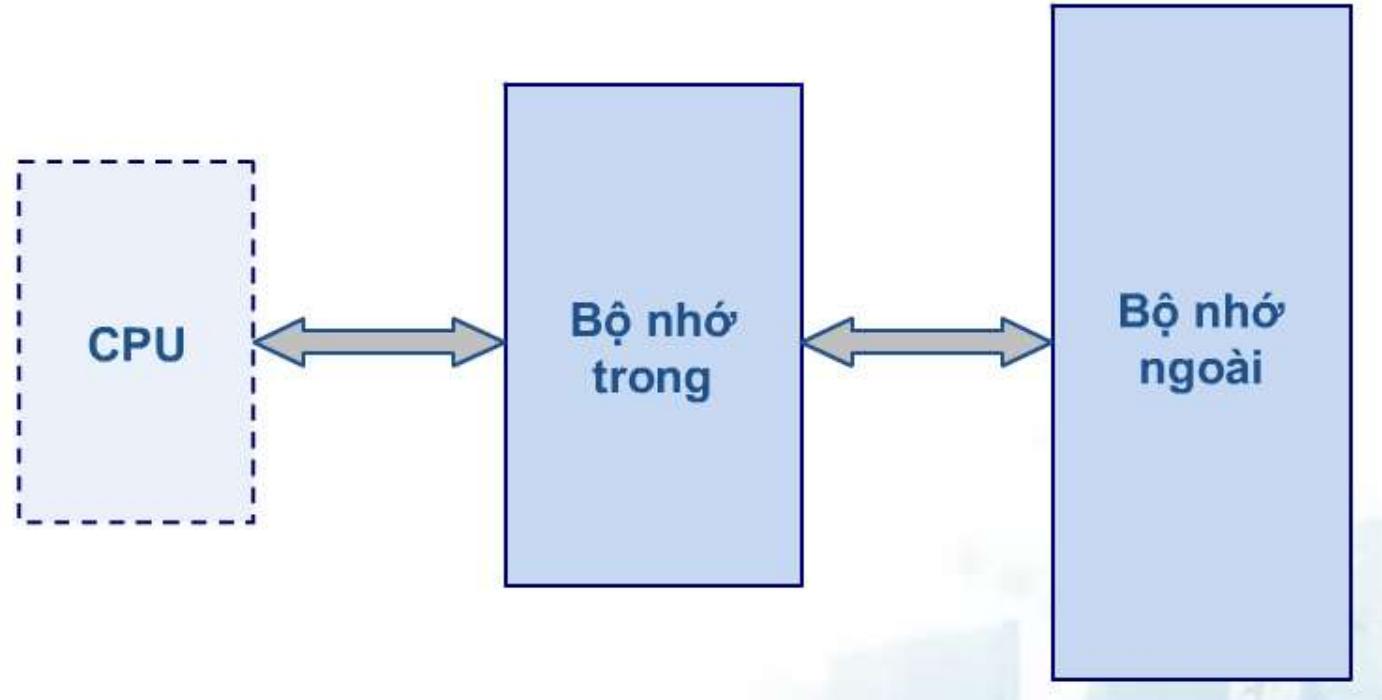


Bộ nhớ



- ❖ Chức năng: Lưu trữ chương trình và dữ liệu
- ❖ Các thao tác cơ bản với bộ nhớ:
 - Thao tác đọc (read)
 - Thao tác ghi (write)
- ❖ Các thành phần chính
 - Bộ nhớ trong (Internal Memory)
 - Bộ nhớ ngoài (External Memory)

Các thành phần bộ nhớ máy tính



Phân cấp hệ thống nhớ trong máy tính



Bộ nhớ trong



❖ Chức năng và đặc điểm

- Chứa các thông tin mà CPU có thể trao đổi trực tiếp
- Tốc độ rất nhanh
- Dung lượng không lớn
- Sử dụng bộ nhớ bán dẫn: ROM và RAM

❖ Các loại bộ nhớ trong:

- Bộ nhớ chính
- Bộ nhớ cache (bộ nhớ đệm)



Bộ nhớ chính

- ❖ Là thành phần nhớ tồn tại trên mọi hệ thống máy tính
- ❖ Chứa các chương trình và dữ liệu đang được CPU sử dụng
- ❖ Tổ chức thành các ngăn nhớ được đánh địa chỉ
- ❖ Ngăn nhớ thường được tổ chức theo Byte
- ❖ Nội dung của ngăn nhớ có thể thay đổi, song địa chỉ vật lý của ngăn nhớ luôn cố định
- ❖ Thông thường, bộ nhớ chính gồm 2 phần:
 - ROM: chỉ đọc
 - RAM: cho phép đọc/ghi, mất dữ liệu khi mất điện

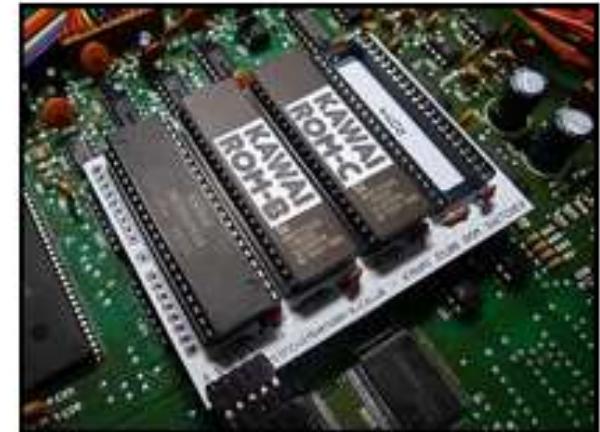
Nội dung	Địa chỉ
00101011	0000
11010101	0001
00001010	0010
01011000	0011
11111011	0100
00001000	0101
11101010	0110
00000000	0111
10011101	1000
00101010	1001
11101011	1010
00000010	1011
00101011	1100
00101011	1101
11111111	1110
10101010	1111

❖ **ROM** (Read Only Memory)

- Bộ nhớ chỉ đọc
- Bộ nhớ không khả biến

❖ **RAM** (Random Access Memory)

- Bộ nhớ đọc, ghi
- Bộ nhớ khả biến



ROM



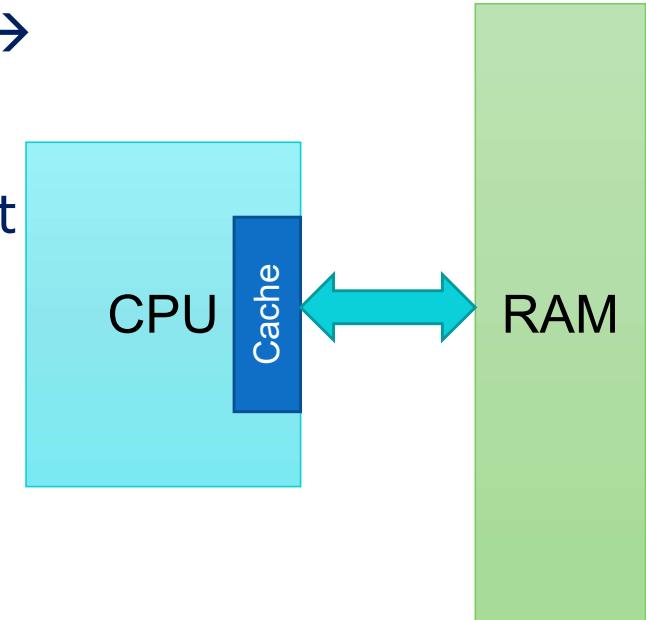
RAM



Bộ nhớ đệm (Cache memory)



- Vai trò: tăng tốc độ trao đổi dữ liệu giữa CPU và bộ nhớ chính
- Tốc độ xử lý CPU lớn hơn nhiều so với tốc độ của RAM → CPU mất thời gian chờ dữ liệu từ RAM
- Cache: dung lượng nhỏ, nhưng tốc độ cao hơn RAM, đặt ngay trong CPU → CPU đọc/ghi rất nhanh
- Sử dụng bộ nhớ Cache:
 - Dữ liệu từ RAM được nạp vào cache theo từng khối khi CPU cần
 - CPU thao tác với dữ liệu trong cache thay vì trên RAM
 - Nếu CPU truy cập nhiều lần dữ liệu thuộc một khối đã nằm trong cache, thì tốc độ sẽ tăng cao
- Hiện tại cache được tích hợp ngay trên CPU, và được chia thành nhiều mức: L1, L2, L3





Bộ nhớ ngoài



❖ Chức năng và đặc điểm

- Lưu giữ tài nguyên phần mềm của máy tính, bao gồm hệ điều hành, các chương trình và dữ liệu.
- Bộ nhớ ngoài được kết nối với hệ thống dưới dạng các thiết bị vào ra
- Dung lượng lớn
- Tốc độ chậm

Bộ nhớ ngoài

❖ Các loại bộ nhớ ngoài:

- Bộ nhớ từ: Đĩa cứng, đĩa mềm
- Bộ nhớ quang: Đĩa CD, DVD,...
- Bộ nhớ bán dẫn: Flash disk, memory card, SSD



Bộ nhớ ngoài



Ổ đĩa cứng



Đĩa mềm, đĩa CD/DVD

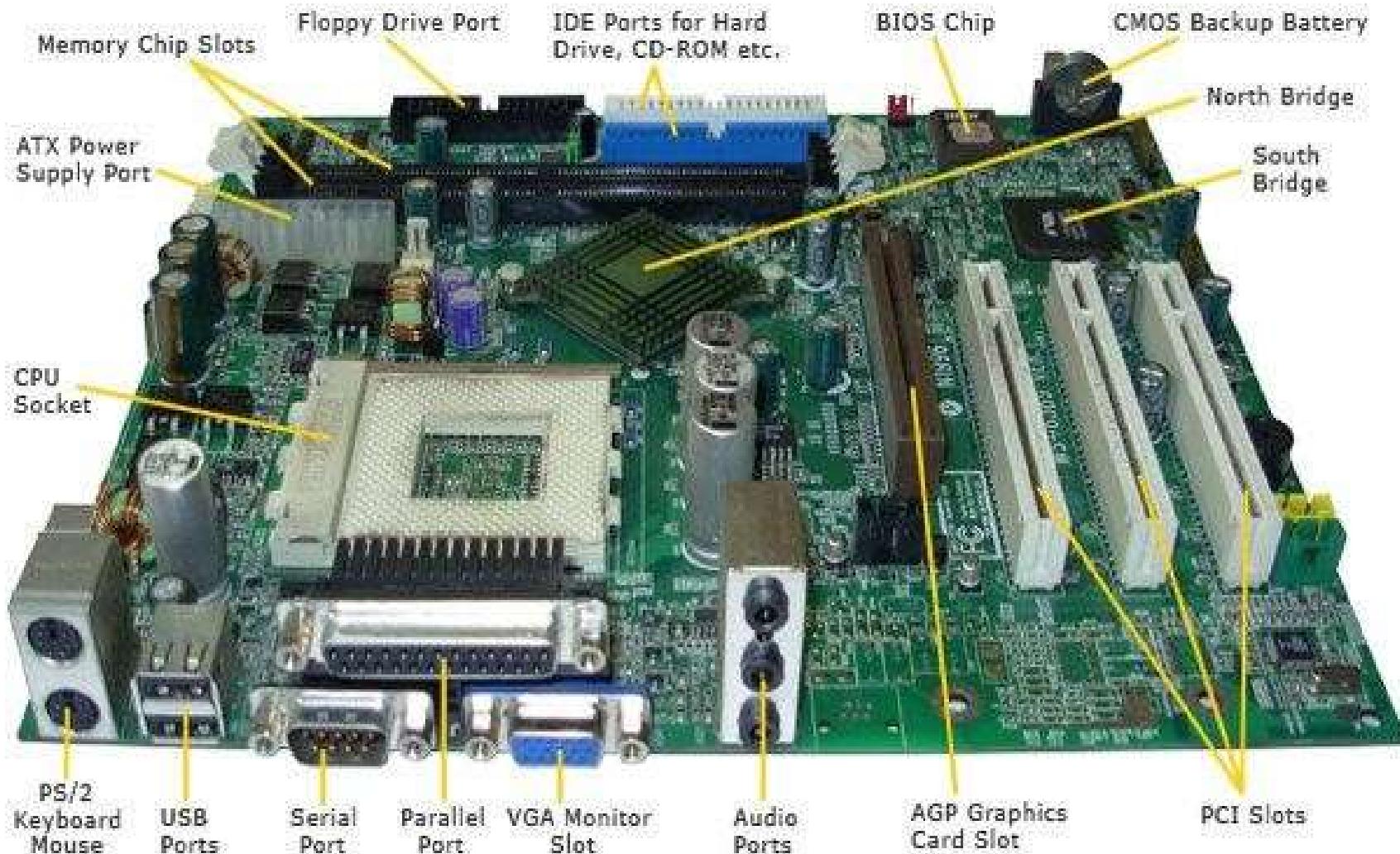


USB



Đĩa ngoài

Mainboard – bo mạch chủ

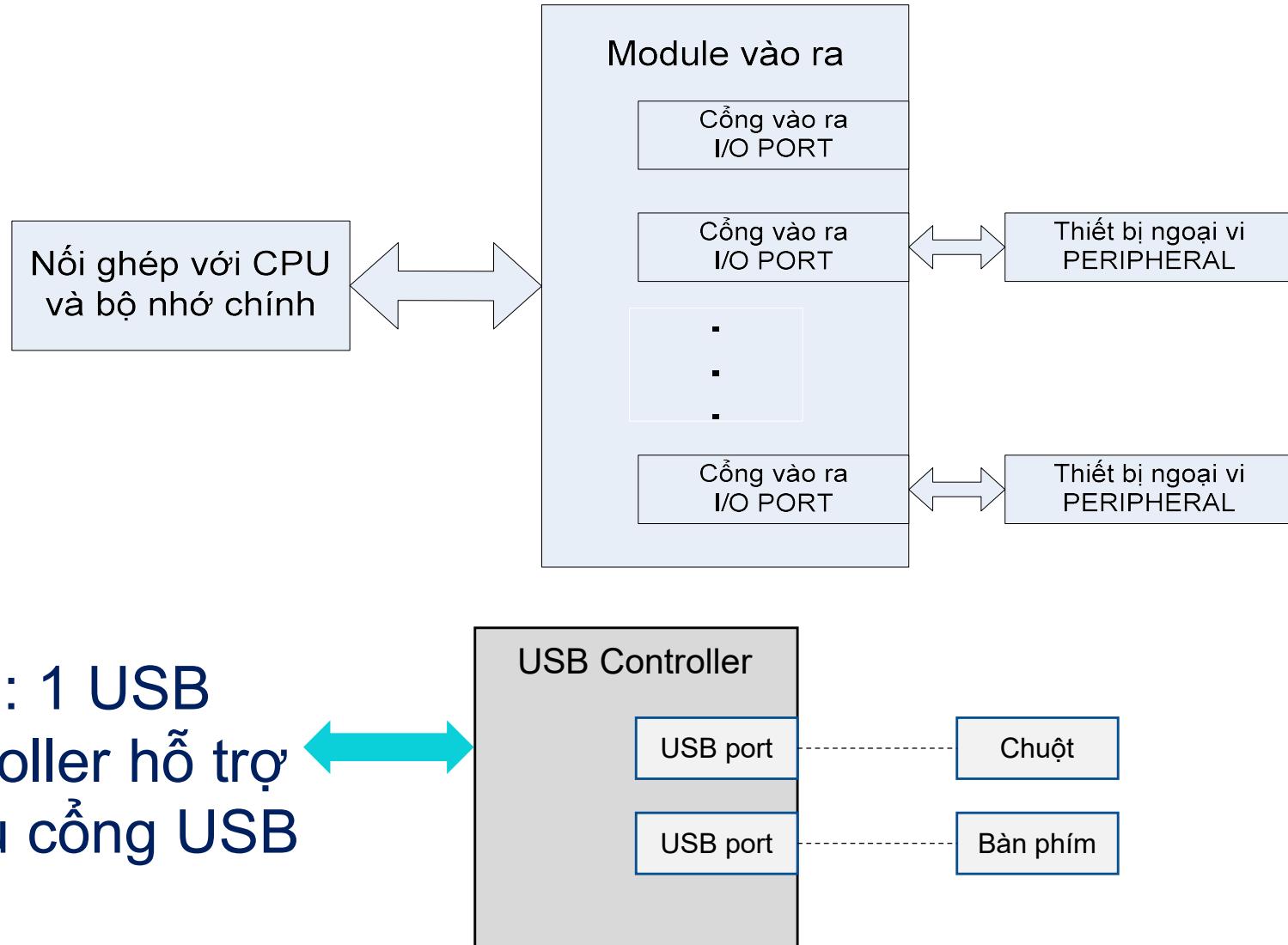




Hệ thống vào-ra

- ❖ Chức năng: Trao đổi thông tin giữa máy tính với thế giới bên ngoài.
- ❖ Các thao tác cơ bản
 - Vào dữ liệu (Input)
 - Ra dữ liệu (Output)
- ❖ Các thành phần chính:
 - Các **thiết bị vào-ra** hay còn gọi là thiết bị ngoại vi (Peripheral devices)
 - Các **mô-đun ghép nối vào-ra**

Cấu trúc cơ bản của hệ thống vào-ra





Thiết bị vào ra



- ❖ Chức năng: chuyển đổi dữ liệu giữa dạng vật lý (bên ngoài máy tính) và dữ liệu số để lưu trữ, xử lý trong máy tính
- ❖ Các thiết bị ngoại vi cơ bản:
 - Thiết bị vào: Bàn phím, chuột, máy quét,...
 - Thiết bị ra: Màn hình, máy in,...
 - Thiết bị nhớ: Các ổ đĩa,...
 - Thiết bị truyền thông: Modem,...



Mô-đun ghép nối vào ra



- ❖ Các thiết bị vào ra không kết nối trực tiếp với CPU mà được kết nối thông qua các mô-đun ghép nối vào-ra.
- ❖ Trong các mô đun ghép nối vào-ra có các cổng vào-ra (IO Port)
- ❖ Các cổng này được quản lý bởi CPU.
- ❖ Mỗi thiết bị vào-ra kết nối với CPU thông qua một số cổng có địa chỉ xác định.



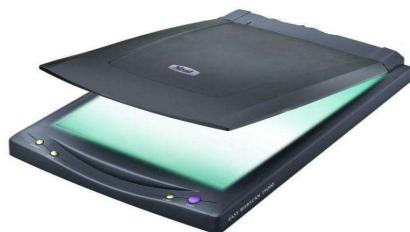
Thiết bị vào



Bàn phím



Chuột



Máy quét



Webcam



Microphone



Thiết bị ra



Màn hình



Máy in



Máy
chiếu



Loa



Các cổng kết nối



Cổng mạng



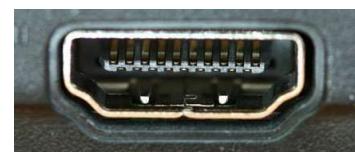
Cổng Audio



Cổng song song



**Cổng nối tiếp
(VGA)**



Cổng HDMI



**Cổng nối tiếp vạn năng
(USB)**



Phần mềm máy tính



- ❖ Máy tính hoạt động theo chương trình (program). Mỗi nhiệm vụ/bài toán cần một chương trình tương ứng.
- ❖ Phần mềm máy tính (Computer Software): khái niệm tương đương chương trình song thường mang ý nghĩa rộng hơn
- ❖ Phần mềm của máy tính có thể thay đổi linh hoạt hơn phần cứng → “**mềm**”.



Phân loại phần mềm

❖ Phần mềm hệ thống:

- Điều khiển hoạt động bên trong của máy tính và cung cấp môi trường giao tiếp giữa người dùng và máy tính nhằm khai thác hiệu quả phần cứng phục vụ cho nhu cầu sử dụng.
- Đòi hỏi tính ổn định, tính an toàn cao.
- Ví dụ: Hệ điều hành, BIOS, phần mềm format ổ đĩa...

❖ Phần mềm ứng dụng:

- Phục vụ các hoạt động của con người như quản lý, kế toán, soạn thảo văn bản,...
- Nhu cầu về phần mềm ứng dụng ngày càng tăng và đa dạng.



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



Hệ đếm



❖ 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.



Hệ đếm



- ❖ 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 (cơ số b =10)



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

Dùng n chữ số thập phân có thể biểu diễn được 10^n giá trị khác nhau:

$$00\dots000 = 0$$

....

$$99\dots999 = 10^n - 1$$

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

$$A = a_{n-1} 10^{n-1} + \dots + a_1 10^1 + a_0 10^0 + a_{-1} 10^{-1} + \dots + a_{-m} 10^{-m}$$

$$= \sum_{i=-m}^{n-1} a_i 10^i$$



Hệ thập phân (cơ số b =10)



❖ 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

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



Hệ đếm cơ số b



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

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

$$= \sum_{i=-m}^{n-1} a_i b^i$$



Hệ đếm nhị phân (binary system, b= 2)



- ❖ Sử dụng 2 chữ số (*nhi phân*): **0,1**
- ❖ Chữ số nhị phân gọi là **bit (binary digit)**
 - 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



Hệ đếm nhị phân (binary system, b = 2)



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

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

$$= \sum_{i=-m}^{n-1} a_i 2^i$$

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

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



Hệ cơ số 8 (Octal system, b=8)



- ❖ Gồm 8 chữ số: **0, 1, 2, 3, 4, 5, 6, 7**
- ❖ Dùng **n** chữ số có thể biểu diễn được **8^n** giá trị khác nhau:

$00\dots000 = 0$ (giá trị trong hệ 10)

....

$77\dots777 = 8^n - 1$ (giá trị trong hệ 10)



Hệ cơ số 8 (Octal system, b=8)



- Biểu diễn A : $a_{n-1} \dots a_1 a_0 \cdot a_{-1} a_{-2} \dots a_{-m}$
 - a_i là các chữ số (0,1,...,7), giá trị của A là

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

Ví dụ: Số A: 235.64_8 có giá trị

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



Hệ cơ số 16 (Hexadecimal, b=16)



- ❖ Sử dụng 16 chữ 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



Hệ cơ số 16 (Hexadecimal, b=16)



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



Chuyển đổi hệ cơ số b → hệ thập phân



➤ Biểu diễn A_b : $a_{n-1} \dots a_1 a_0 \cdot a_{-1} a_{-2} \dots a_{-m}$

- n chữ số biểu diễn cho phần nguyên
- m chữ số biểu diễn cho phần lẻ

Giá trị tương đương trong hệ 10 là:

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

$$= \sum_{i=-m}^{n-1} a_i b^i$$

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

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



Chuyển từ hệ thập phân sang hệ cơ số b



- ❖ 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



Chuyển đổi phần nguyên



- ❖ **Bước 1:** Lấy phần nguyên của $N_{(10)}$ chia cho b, ta được thương là T_1 số **d₁**.
- ❖ **Bước 2:** Nếu T_1 khác 0, Lấy T_1 chia tiếp cho b, ta được thương số là T_2 , số **d₂**
- (Cứ làm 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 $N_{(10)} = d_nd_{n-1}...d_1$ (b)

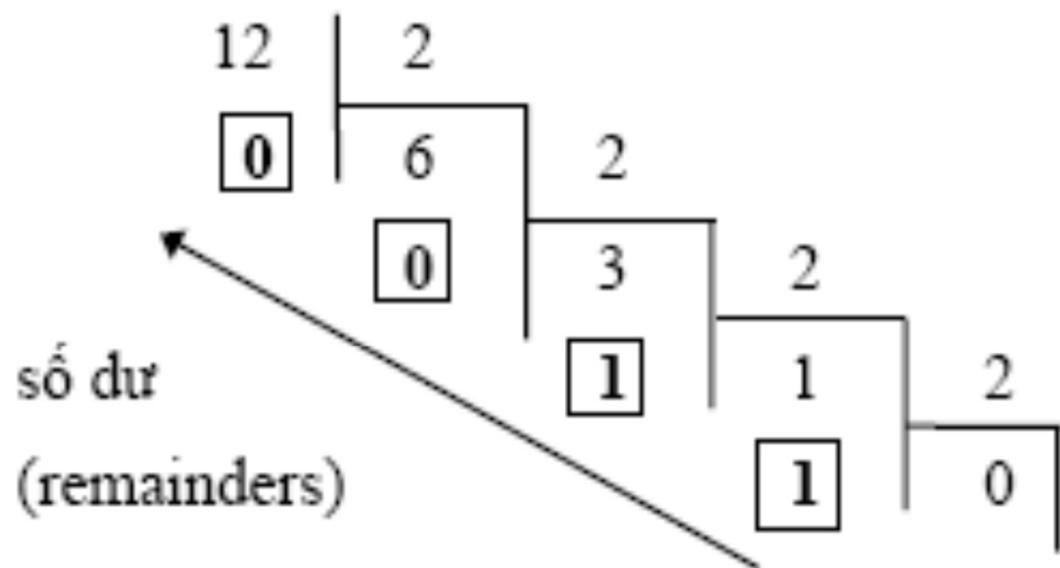


Chuyển đổi phần nguyên (2)



❖ 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: $12_{(10)} = 1100_{(2)}$



Chuyển đổi phần thập phân



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

... (*cứ làm như vậy cho đến khi $y_n=0$*)

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

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



Chuyển đổi phần thập phân (2)



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

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

phần nguyên của tích

$$\begin{array}{r} 0.6875 \times 2 = 1 \boxed{375} \\ 0.3750 \times 2 = 0.75 \\ 0.75 \times 2 = 1.5 \\ 0.5 \times 2 = 1.0 \end{array}$$

phần thập phân của tích

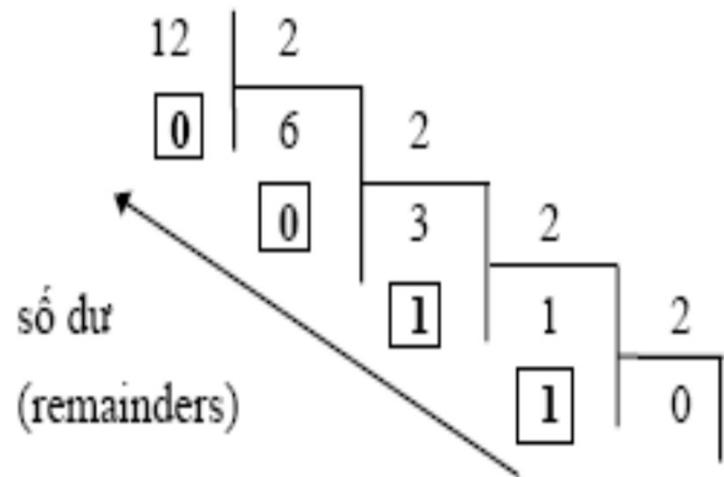
$$0.6875_{(10)} = 0.1011_{(2)}$$



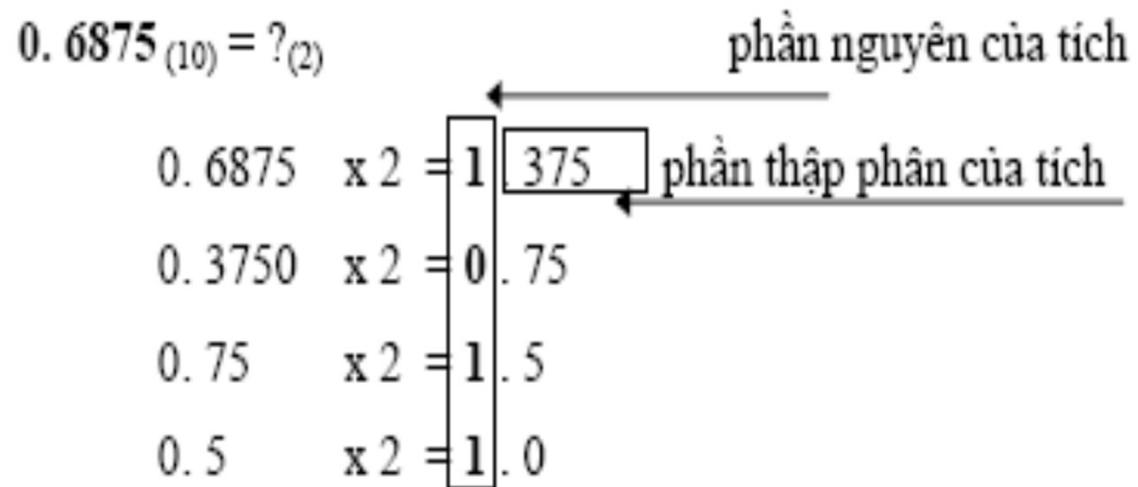
Ví dụ: Chuyển từ thập phân sang nhị phân



$$\diamond 12.6875_{(10)} = 1100.1011_{(2)}$$



Kết quả: $12_{(10)} = 1100_{(2)}$



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

$$\diamond 69.25_{(10)} = ?_{(2)}$$



Cách 2: Tính nhẩm khi chuyển về hệ 2



- ❖ 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: $69.25_{(10)} = 64 + 4 + 1 + \frac{1}{4}$
 $= 2^6 + 2^2 + 2^0 + 2^{-2}$
 $= 1000101.01_{(2)}$



Chuyển giữa hệ 2, 8, và 16



- ❖ Chuyển về số trung gian ở hệ cơ số 2
- ❖ Mỗi chữ số trong hệ 8 hoặc 16 tương đương một nhôm gồm 3 hoặc 4 chữ số hệ 2

Ví dụ:

- Nhị phân → Hexa: $11\ 1011\ 1110\ 0110_{(2)} = ?$
 $0011\ 1011\ 1110\ 0110_{(2)} = 3BE6_{(16)}$
- Hexa → Nhị phân: $AB7_{(16)} = ?$
 $AB7_{(16)} = 1010\ 1011\ 0111_{(2)}$



Chuyển cơ số Hexa → Thập phân



Ví dụ: Chuyển $3A8C_{(16)}$ → ? $_{(10)}$

$$\begin{aligned}3A8C_{(16)} &= 3 \times 16^3 + 10 \times 16^2 + 8 \times 16^1 + 12 \times 16^0 \\&= 12288 + 2560 + 128 + 12 \\&= 14988_{(10)}\end{aligned}$$



Chuyển số thập phân 14988 sang Hệ Hexa (16)



$$\begin{array}{rcl} 14988 : 16 & = & 936 \text{ dư } 12 \text{ tức là C} \\ 936 : 16 & = & 58 \text{ dư } 8 \\ 58 : 16 & = & 3 \text{ dư } 10 \text{ tức là A} \\ 3 : 16 & = & 0 \text{ dư } 3 \end{array}$$

Như vậy, ta có: $14988_{(10)} = 3A8C_{(16)}$

Số thập phân	Chia cho 16		
	Thương	Số dư	Ký hiệu
14988	936	12	C
936	58	8	8
58	3	10	A
3	0	3	3

Vậy
 $14988_{(10)} = 3A8C_{(16)}$



Bài tập 1



Chuyển hệ thập phân sang hệ nhị phân

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



Ví dụ: Chuyển 124.75 sang hệ 2



Phân nguyên	Chia cho 2	
	Thương	Số dư
124	62	0
62	31	0
31	15	1
15	7	1
7	3	1
3	1	1
1	0	1

Phân thập phân	Nhân 2	Phân nguyên	Phân thập phân
0.75	1.5	1	0.5
0.5	1	1	0

Vậy

$$124.75_{(10)} = 1111100.11_{(2)}$$



Ví dụ chuyển 125.375 sang hệ 2



Phân nguyên	Chia cho 2	
	Thương	Số dư
125	62	1
62	31	0
31	15	1
15	7	1
7	3	1
3	1	1
1	0	1

Phân thập phân	Nhân 2	Phân nguyên	Phân thập phân
0.375	0.75	0	0.75
0.75	1.5	1	0.5
0.5	1	1	0

Vậy

$$125.375_{(10)} = 1111101.011_{(2)}$$



Bài tập 2



Chuyển hệ thập phân sang hệ Hexa

- a. 124.375 → ?
- b. 125.75 → ?
- c. 65.125 → ?



VD: chuyển 124.375 sang hệ 16



Phân nguyên	Chia cho 16			Phân thập phân	Nhân 16	Phân nguyên	Phân thập phân	Quy đổi
	Thương	Số dư	Quy đổi					
124	7	12	C	0.375	6	6	0	6
7	0	7	7					

Vậy

$$124.375_{(10)} = 7C.6_{(16)}$$



VD: chuyển 125.75 sang hệ 16



Phân nguyên	Chia cho 16		
	Thương	Số dư	Quy đổi
125	7	13	D
7	0	7	7

Phân thập phân	Nhân 16	Phân nguyên	Phân thập phân
0.75	12	12	0

Vậy

$$125.75_{(10)} = 7D.C_{(16)}$$



Biểu diễn dữ liệu trong máy tính



- ❖ 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

- Sô nguyên, số thực, ký tự

❖ Dữ liệu có cấu trúc

- Mảng, xâu ký tự, tập hợp, bản ghi



Dữ liệu cơ bản



❖ 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.

Độ dài từ dữ liệu:

- Số bit được sử dụng để mã hóa loại dữ liệu tương ứng
- Thực tế, độ dài từ dữ liệu thường là bội số của 8.



Dữ liệu có cấu trúc



- ❖ 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.*



Đơn vị đo thông tin



❖ Bit (**B**inary digiT)

- 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**): $1\text{ PB} = 2^{10}\text{TB} = \dots = 2^{50}\text{B}$



Số nguyên không dấu



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

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

Trong đó 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 8bit có giá trị sau

❖ **A = 45**

$$\begin{aligned} A = 45 &= 32 + 8 + 4 + 1 \\ &= 2^5 + 2^3 + 2^2 + 2^0 \\ \rightarrow A &= 0010\ 1101 \end{aligned}$$

❖ **B = 156**

$$\begin{aligned} B = 156 &= 128 + 16 + 8 + 4 \\ &= 2^7 + 2^4 + 2^3 + 2^2 \\ \rightarrow B &= 1001\ 1100 \end{aligned}$$



Ví dụ 2



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

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

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

- Dải biểu diễn là $[0, 255]$

$$0000\ 0000 = 0$$

$$0000\ 0001 = 1$$

$$0000\ 0010 = 2$$

$$0000\ 0011 = 3$$

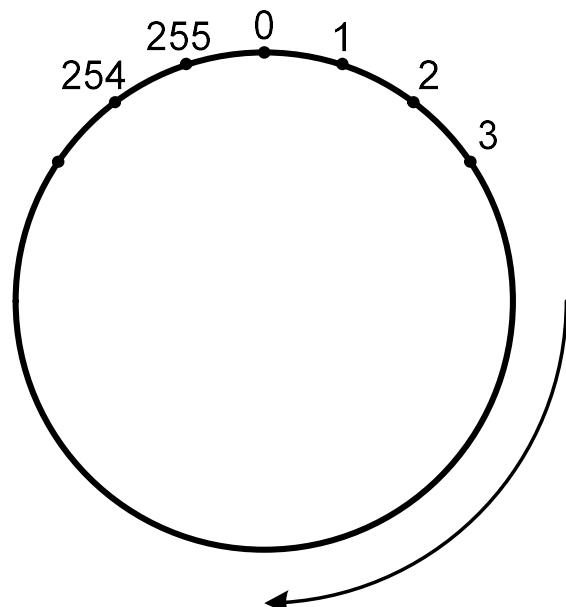
.....

$$1111\ 1111 = 255$$

- Trục số học tự nhiên:



- Trục số học máy tính:





Phạm vi biểu diễn



❖ n = 16 bit:

➤ Dải biểu diễn là [0, 65535]

❖ n = 32 bit:

➤ Dải biểu diễn là [0, $2^{32}-1$]

❖ n = 64 bit:

➤ Dải biểu diễn là [0, $2^{64}-1$]



Số nguyên có dấu



- ❖ Số nguyên có dấu được dùng để biểu diễn số nguyên với cả giá trị âm và dương
- ❖ Sử dụng bit đầu tiên làm bit dấu:
 - Bit 0 cho số dương
 - Bit 1 cho số âm
- ❖ Biểu diễn số âm bằng số bù 2 của số dương tương ứng



Số bù một và số bù hai



❖ Giả sử có một số nhị phân được biểu diễn bởi n bit.

❖ Định nghĩa:

$$\text{➢ Số bù một của } A = (2^n - 1) - A$$

$$\text{➢ Số bù hai của } A = 2^n - A$$

$$\text{➢ Nhận xét: Số bù hai} = \text{Số bù một} + 1$$

❖ Ví dụ

$$\text{➢ Xét } n = 4\text{bit}, A = 0110$$

$$\text{➢ Số bù một của } A = (2^4 - 1) - 0110 = 1001$$

$$\text{➢ Số bù hai của } A = 2^4 - 0110 = 1010$$



Nhận xét



- ❖ Xét $n = 4$ bit, $A = 0110$
 - ❖ Số bù một của $A = (2^4 - 1) - 0110 = 1001$
 - ❖ Số bù hai của $A = 2^4 - 0110 = 1010$
- Có thể tìm số bù một của A bằng cách đảo ngược tất cả các bit
- Số bù hai = Số bù một + 1
- $A + \text{Số bù hai của } A = 0$ nếu bỏ qua bit nhỡ ra khỏi bit cao nhất



Biểu diễn số nguyên có dấu



- ❖ Biểu diễn số âm bằng số bù 2
- ❖ Ví dụ: Biểu diễn số nguyên có dấu sau đây bằng 8bit nhị phân:

$$A = -70_{(10)}$$

Biểu diễn 70 = 0100 0110

Bù 1: 1011 1001 (nghịch đảo các bit)

$$\begin{array}{r} + \\ \hline \end{array}$$

Bù 2: 1011 1010

Vậy: A = 1011 1010₍₂₎



Biểu diễn số nguyên có dấu (tiếp)



- Dạng tổng quát của số nguyên có dấu A:

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

- Giá trị của A được xác định như sau:

$$A = -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i$$

- Dải biểu diễn: $[-2^{n-1}, 2^{n-1}-1]$



Ví dụ



❖ Xác định giá trị của các số nguyên có dấu 8bit sau đây:

$$A = 0101\ 0110$$

$$B = 1101\ 0010$$

Giải:

$$A = 2^6 + 2^4 + 2^2 + 2^1 = 64 + 16 + 4 + 2 = +86$$

$$B = -2^7 + 2^6 + 2^4 + 2^1$$

$$= -128 + 64 + 16 + 2 = -46$$

Phạm vi biểu diễn: với $n = 8$ bit

❖ Dải biểu diễn là $[-128, 127]$

$$0000\ 0000 = 0$$

$$0000\ 0001 = +1$$

$$0000\ 0010 = +2$$

.....

$$01111111 = +127$$

$$10000000 = -128$$

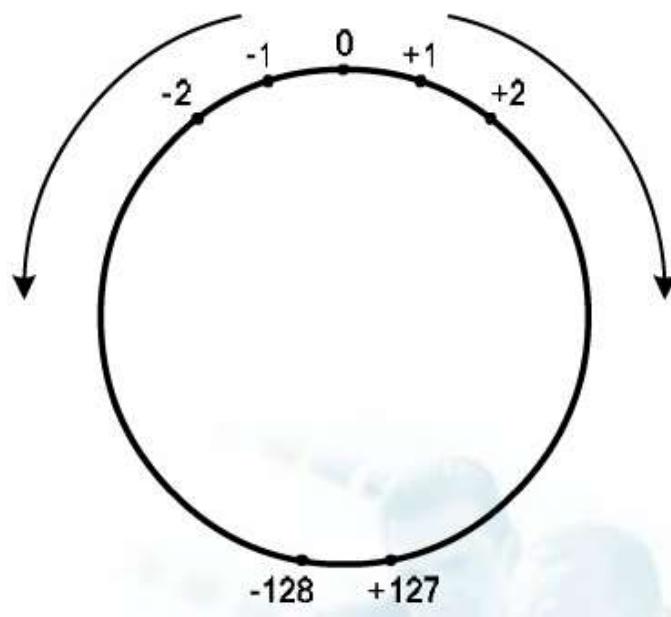
$$10000001 = -127$$

.....

$$1111\ 1110 = -2$$

$$1111\ 1111 = -1$$

❖ Trục số học máy tính





Tính toán số học với số nguyên



- Cộng trừ số nguyên không dấu
- Cộng/trừ số nguyên có dấu
- Nhân chia các số nguyên không dấu
- Các phép toán logic với số nhị phân



Tính toán trong hệ nhị phân



❖ Phép cộng:

$$1+0=0+1=1;$$

$$0+0=0;$$

$$1+1=10;$$

$$110$$

$$+ 111$$

$$1101$$

❖ Phép trừ:

$$0-1=1; \text{ (vay 1)}$$

$$1-1=0;$$

$$0-0=0;$$

$$1-0=1$$

$$1101$$

$$- 111$$

$$110$$



Cộng/trừ số nguyên không dấu



- ❖ Tiến hành cộng/trừ lần lượt từng bít từ phải qua trái.
- ❖ Khi cộng/trừ hai số nguyên không dấu n bit ta thu được một số nguyên không dấu n bit.
 - Nếu tổng của hai số đó lớn hơn $2^n - 1$ thì đó sẽ tràn số và kết quả sẽ là sai.
 - Trừ số không dấu thì ta chỉ trừ được số lớn cho số nhỏ. Trường hợp ngược lại sẽ sai



Ví dụ



- ❖ Thực hiện phép toán với nguyên không dấu 8 bit
- ❖ Trường hợp không xảy ra tràn số (carry-out):

$$X = 1001\ 0110 = 150$$

$$Y = 0001\ 0011 = 19$$

$$S = 1010\ 1001 = 169$$

$$\text{Cout} = 0$$

- ❖ Trường hợp có xảy ra tràn số (carry-out):

$$X = 1100\ 0101 = 197$$

$$Y = 0100\ 0110 = 70$$

$$S = 0000\ 1011 \neq 267$$

$$\text{Cout} = 1 \rightarrow \text{carry-out}$$

➤ (KQ sai = $2^3 + 2^1 + 2^0 = 11$)



Cộng số nguyên có dấu



- Cộng lần lượt các cặp bit từ phải qua trái, bỏ qua bit nhớ cao nhất (nếu có).
- Cộng hai số khác dấu: kết quả luôn đúng
- Cộng hai số cùng dấu:
 - Nếu tổng nhận được cùng dấu với 2 số hạng thì kết quả là đúng
 - Nếu tổng nhận được khác dấu với 2 số hạng thì đã xảy ra hiện tượng *tràn số học* (overflow) và kết quả nhận được là sai



Ví dụ: Cộng/trừ số nguyên có dấu



❖ VD: không tràn số

$$\begin{array}{r} X = 0100 \ 0110 = +70 \\ + \underline{Y = 0010 \ 1010 = +42} \\ S = 0111 \ 0000 = +112 \end{array}$$

$$\begin{array}{r} X = 0110 \ 0001 = +97 \\ + \underline{Y = 1100 \ 1100 = -52} \\ S = 0010 \ 1101 = +45 \\ C_{out} = 1 \rightarrow bỏ qua \end{array}$$

$$\begin{array}{r} X = 1010 \ 0110 = -90 \\ + \underline{Y = 0010 \ 0100 = +36} \\ S = 1100 \ 1010 = -54 \end{array}$$

$$\begin{array}{r} X = 1011 \ 0110 = -74 \\ + \underline{Y = 1110 \ 0010 = -30} \\ S = 1001 \ 1000 = -104 \\ C_{out} = 1 \rightarrow bỏ qua \end{array}$$



Ví dụ: Cộng/trừ số nguyên có dấu



❖ Có xảy ra tràn số:

$$\begin{array}{r} X = 0100 \ 1011 = +75 \\ + Y = 0101 \ 0001 = +81 \\ \hline S = 1001 \ 1100 \neq +156 \\ (S = -2^7 + 2^4 + 2^3 + 2^2 = -100) \end{array}$$

$$\begin{array}{r} X = 1001 \ 1000 = -104 \\ + Y = 1011 \ 0110 = -74 \\ \hline S = 0100 \ 1110 \neq -178 \\ C_{out} = 1 \rightarrow bỏ qua \\ (S = 2^6 + 2^3 + 2^2 + 2^1 = 78) \end{array}$$



Trừ số nguyên có dấu



❖ Nhận xét

$$X - Y = X + (-Y)$$

❖ Kết quả chính là tổng X với bù 2 của Y

❖ Ví dụ:

$$\begin{array}{r} 97 - 52 = 97 + (-52) \\ + \quad \quad \quad \quad = \quad 0110\ 0001 \\ -52 \quad \quad \quad \quad = \quad 1100\ 1100 \\ \hline \end{array}$$

$$\begin{array}{r} 45 = \textcircled{1} 0010\ 1101 \\ \text{Bỏ qua} \end{array}$$



Nhân/chia số nguyên không dấu



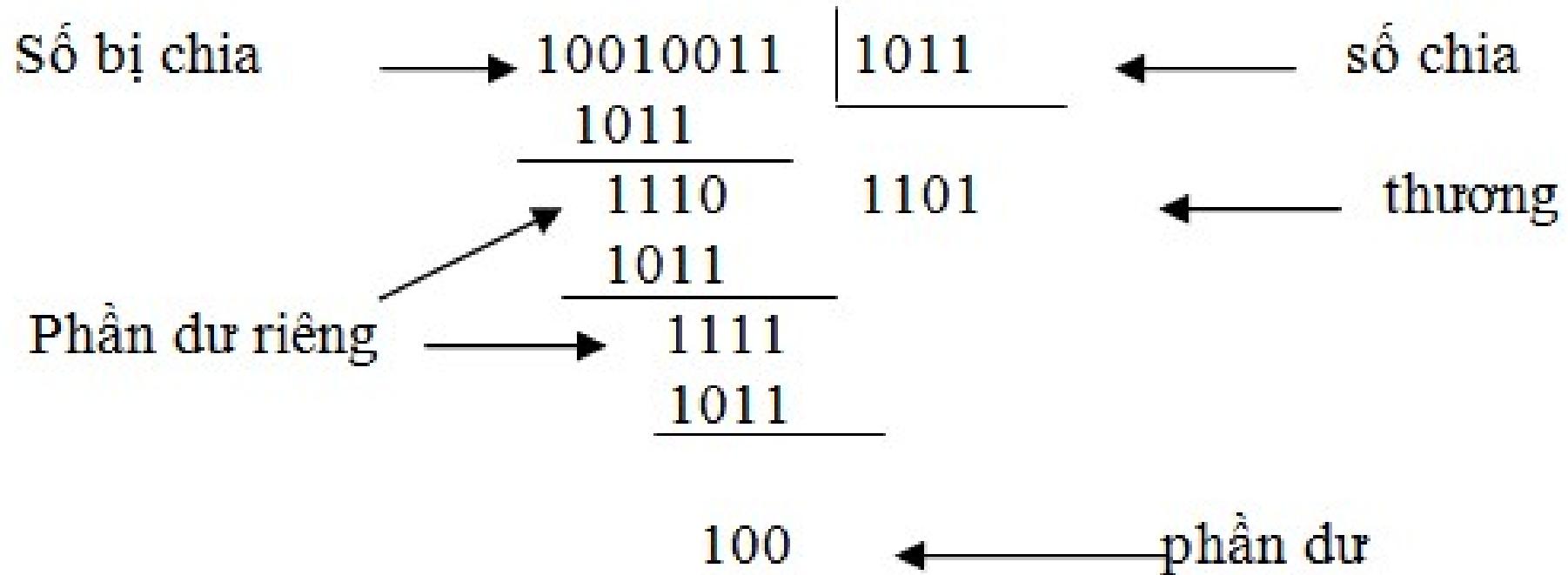
❖ Các bước thực hiện như trong hệ 10

❖ Ví dụ: Phép nhân

$$\begin{array}{r} 1011 & (11 \text{ cơ số } 10) \\ \times & \\ 1101 & (13 \text{ cơ số } 10) \\ \hline 1011 \\ 0000 \\ 1011 \\ 1011 \\ \hline 10001111 & (143 \text{ cơ số } 10) \end{array}$$



Phép chia số nguyên không dấu





Nhân/chia số nguyên có dấu



- ❖ Bước 1: Chuyển đổi thành số dương tương ứng
- ❖ Bước 2: Thực hiện nhân/chia với số nguyên không dấu
- ❖ Bước 3: Hiệu chỉnh dấu của kết quả.



Các phép toán logic với số nhị phân



		AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

	NOT
0	1
1	0

- Thực hiện các phép toán logic với 2 số nhị phân:
 - Thực hiện các phép toán logic với từng cặp bit của 2 số
 - Các phép toán logic chỉ tác động lên từng cặp bit, không ảnh hưởng đến bit khác.
 - Kết quả là 1 số nhị phân



Ví dụ



		NOT
A	1010 1010	01010101
B	0000 1111	11110000
AND	00001010	
OR	10101111	
XOR	10100101	

- Có thể dùng phép **AND** để xoá một số bit và giữ nguyên các bit còn lại (Ví dụ: **AND 11011011**)
- Có thể dùng phép **OR** để thiết lập 1 số bit và giữ nguyên các bit khác. (Ví dụ: **OR 00100100**)



Biểu diễn số thực



- ❖ Số thực trong máy tính được biểu diễn thông qua ký pháp dấu phẩy động (*Floating Point Number*)
- ❖ Một số thực V được biểu diễn theo dạng dấu phẩy động như sau: $V = M * R^E$

Trong đó:

- **R** là cơ số (*Radix*) thường là 2 hoặc 10.
 - **M** là phần định trị (*Mantissa*) thỏa mãn $-R < M < R$
 - **E** là phần mũ (*Exponent*)
- ❖ Nhận xét:
 - Nếu R cố định, để lưu trữ V chỉ cần lưu trữ M và E (*dưới dạng số nguyên*)



Ví dụ



Với cơ số R = 10

$$V = -52.5 = -5.2500 \times 10^1$$

- ❖ Phần định trị: M = -5.2500 / Phần số mũ: E = 1

Ghi chú

- ❖ Để chuẩn hóa, phần định trị luôn nằm trong khoảng (-R, R) và số đầu tiên là khác 0 (*nếu được*)
 - -52.5×10^0 và -0.5250×10^2 tương đương với -5.25×10^1 nhưng không phải số chuẩn hóa
 - Việc chuẩn hóa nhằm duy trì độ chính xác và biểu diễn các số rất lớn/bé
 - 0.000162 biểu diễn với 4 chữ số thập phân $\Rightarrow 0.0001$
 - $\pm 9.999 \times 10^9 = \pm 9\ 999\ 000\ 000$
 - $\pm 0.001 \times 10^{-9} = \pm 0.000\ 000\ 000\ 001$



Phép toán với dấu phẩy động



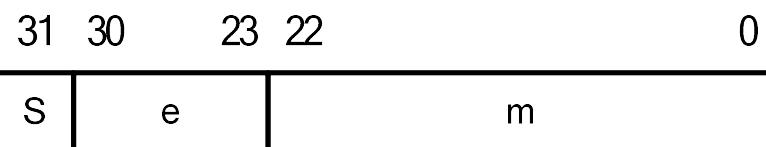
- ❖ Được tiến hành trên cơ sở các giá trị của phần định trị và phần mũ
- ❖ Ví dụ, cho R_1 và R_2 là số dấu phẩy động:
 - $R_1 = M_1 \times R^{E_1}$ và $R_2 = M_2 \times R^{E_2}$
- ❖ Các phép toán số học thực hiện như sau:
 - $R_1 \pm R_2 = (M_1 \times R^{E_1 - E_2} \pm M_2) \times R^{E_2}$,
 - Giả thiết $E_1 \geq E_2$
 - $R_1 \times R_2 = (M_1 \times M_2) \times R^{E_1 + E_2}$
 - $R_1 / R_2 = (M_1 / M_2) \times R^{E_1 - E_2}$



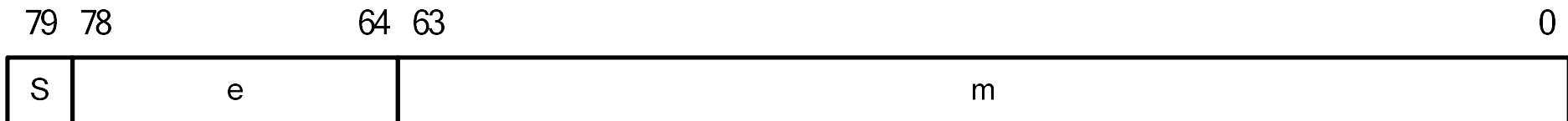
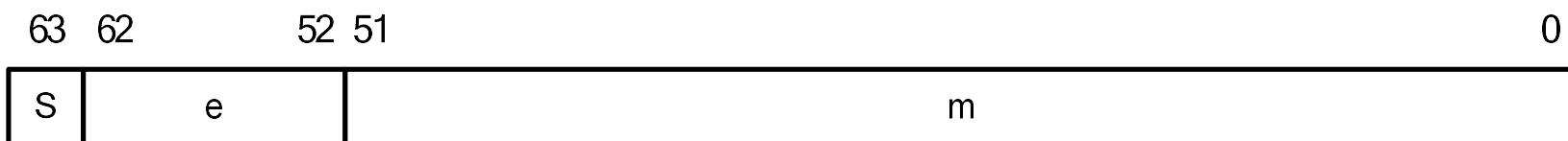
Chuẩn IEEE 754/85



- ❖ Là chuẩn mã hóa số dấu phẩy động
- ❖ Cơ số $R = 2$
- ❖ Các dạng cơ bản:
 - Dạng có độ chính xác đơn, 32-bit
 - Dạng có độ chính xác kép, 64-bit
 - Dạng có độ chính xác kép mở rộng, 80-bit



Khuôn dạng mã hóa





Chuẩn IEEE 754/85



- ❖ **S** là bit dấu,
 - $S = 0$ đó là số dương/ $S = 1$ đó là số âm.
- ❖ **e** là mã lệch (excess) của phần mũ E
$$E = e - b$$
Trong đó b là độ lệch (bias):
 - Dạng 32-bit : $b = 127$, hay $E = e - 127$
 - Dạng 64-bit : $b = 1023$, hay $E = e - 1023$
 - Dạng 80-bit : $b = 16383$, hay $E = e - 16383$
- ❖ **m** là các bit phần lẻ của phần định trị M
 - Phần định trị được ngầm định: $M = 1.m$



Chuẩn IEEE 754/85



R =

s	e	m
---	---	---

$$R = (-1)^s \times 1.m \times 2^{e-b}$$



ví dụ



- ❖ Tìm giá trị số thực R biết biểu diễn của R theo chuẩn IEEE 754 dạng 32bit như sau

1100 0001 0101 0110 0000 0000 0000 0000

$S = 1 \rightarrow R$ là số âm

- $e = 1000\ 0010 = 130$
 - Số 32 bit: $b = 127 \Rightarrow E = 130 - 127$
- $m = 10101100...00 \Rightarrow M = 1.m$
- Vậy $R = (-1)^1 \times 1.10101100...00 \times 2^{130-127}$ $= -1.101011 \times 2^3 = -1101.011 = -13.375$



Ví dụ



- ❖ Xác định giá trị thập phân của số thực X có dạng biểu diễn theo chuẩn IEEE 754 dạng 32bit như sau:

0011 1111 1000 0000 0000 0000 0000 0000

- ❖ Giải:

- S = 0 → X là số dương
- e = 0111 1111 = 127
- m = 000000...00
- Vậy $X = (-1)^0 \times 1.0000...00 \times 2^{127-127}$
 $= 1.0 \times 2^0 = 1$



Ví dụ



❖ Biểu diễn số thực $X = 9.6875$ về dạng số dấu chấm động theo chuẩn IEEE 754 dạng 32 bit

❖ Giải:

$$X = 9.6875_{(10)} = 1001.1011_{(2)} = 1.0011011 \times 2^3$$

Theo chuẩn IEEE754:

➤ $S = 0$ vì đây là số dương

$$\begin{aligned} \text{➤ } E &= e - 127 \text{ nên } e = 127 + 3 = 130_{(10)} \\ &= 1000\ 0010_{(2)} \end{aligned}$$

➤ $m = 001101100...00$ (23 bit)

$$X = 0100\ 0001\ 0001\ 1011\ 0000\ 0000\ 0000\ 0000$$

$$= 411B0000_{(16)}$$



Một số quy ước đặc biệt



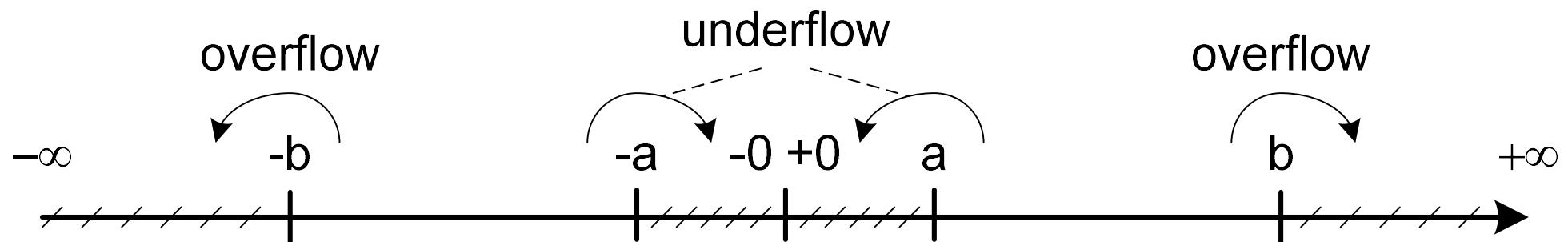
- ❖ Nếu tất cả các bit của e đều bằng 0, các bit của m đều bằng 0, thì $X = \pm 0$
- ❖ Nếu tất cả các bit của e đều bằng 1, các bit của m đều bằng 0, thì $X = \pm \infty$
- ❖ Nếu tất cả các bit của e đều bằng 1, m có ít nhất một bit bằng 1, thì X không phải là số (not a number - NaN)



Phạm vi biểu diễn



- ❖ Dạng 32 bit: $a = 2^{-127} \approx 10^{-38}$ $b = 2^{+127} \approx 10^{+38}$
- ❖ Dạng 64 bit: $a = 2^{-1023} \approx 10^{-308}$ $b = 2^{+1023} \approx 10^{+308}$
- ❖ Dạng 80 bit: $a = 2^{-16383} \approx 10^{-4932}$ $b = 2^{+16383} \approx 10^{+4932}$



$$X = (-1)^S \times 1.m \times 2^{e-127}$$



Chú ý



❖ Khoảng cách giữa 2 số thực liên tiếp ?

➤ Sai số máy (Machine epsilon)

❖ Sai số làm tròn

➤ $0.7 = ?$

■ $0.1011001100 = 0.69921875$

■ $0.10110011001100 = 0.69999980926..$

■ $0.1011001101 = 0.700195312$



Biểu diễn ký tự



- ❖ 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



Bộ mã ASCII



- ❖ Do ANSI (**American National Standard Institute**) 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 $2^8 = 256$ ký tự, có mã từ $00_{16} \div FF_{16}$, bao gồm:
 - 128 ký tự chuẩn có mã từ $00_{16} \div 7F_{16}$
 - 128 ký 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	U	Ш	Х	Ү	Z	[\]	^	_
6	~	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	u	ш	х	ү	z	{		}	~	DEL

Physical Device Controls: Format Effectors



Bộ mã ASCII



❖ **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..



Các ký tự mở rộng của bảng mã ASCII



❖ 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 hóa cho các ký tự riêng của chữ Việt, ví dụ như bộ mã TCVN 5712.



Bộ mã Unicode



- ❖ 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 100000 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



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



Khái niệm bài toán



A → B

- ❖ A: Giả thiết, điều kiện ban đầu
- ❖ B: Kết luận, mục tiêu cần thực hiện
- ❖ →: Suy luận, giải pháp căn 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**



Trong khoa học máy tính



A → B

- ❖ A là Input
- ❖ B là Output
- ❖ → là Chương trình cho phép biến đổi A thành B



Chương trình



❖ 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



Thiết kế thuật giải

❖ 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 & Lập trình viên



❖ 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



Bước 1: Xác định bài toán



- ❖ 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 ƯSCLN của 2 số nguyên dương

- Nhập: 2 số X, Y
- Điều kiện nhập: X, Y nguyên dương
- Dữ liệu ra: Z
- Điều kiện ra: Z là ƯSCLN(X,Y)



Bước 2: Lựa chọn phương pháp



- ❖ Tồn tại 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



- ❖ Xây dựng mô hình chặt chẽ, chính xác và chi tiết cho phương pháp đã lựa chọn
- ❖ Lắp liên tiếp các bước sau để thuật toán ngày càng hoàn chỉnh hơn (*quá trình tinh chỉnh từng bước*)
 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 gì và sẽ tạo ra 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..?



Bước 3: Xây dựng thuật giải (tiếp)



- ❖ *Quá trình tinh chỉnh từng bước dừng lại khi*
 - Yêu cầu cho biết 1 hay nhiều đại lượng
 - Tính một đại lượng theo công thức đã biết rõ
 - Thông báo một hay nhiều kết quả đã xử lý
- ❖ Sau khi tinh chỉnh cần phải diễn tả giải thuật dưới dạng chuẩn
 - Ngôn ngữ liệt kê các hành động
 - Sơ đồ khôi...



Bước 4: Cài đặt chương trình



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
 - **Thao tác:** In ra một thông báo
 - **Câu lệnh:** printf("....")/ write(".....");
- ❖ Lựa chọn ngôn ngữ lập trình, tùy theo bài toán giải quyết
 - NNLT bậc thấp: Hợp ngữ
 - NNLT bậc cao: C, Pascal, Java,..



Bước 5: Hiệu chỉnh chương trình



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



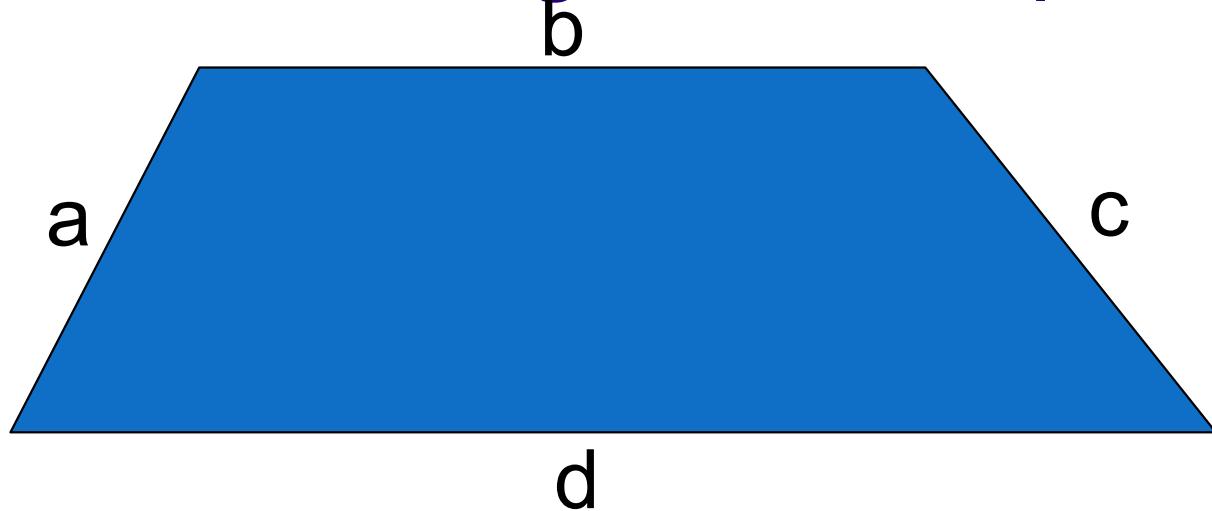
Bước 6: Thực hiện chương trình



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

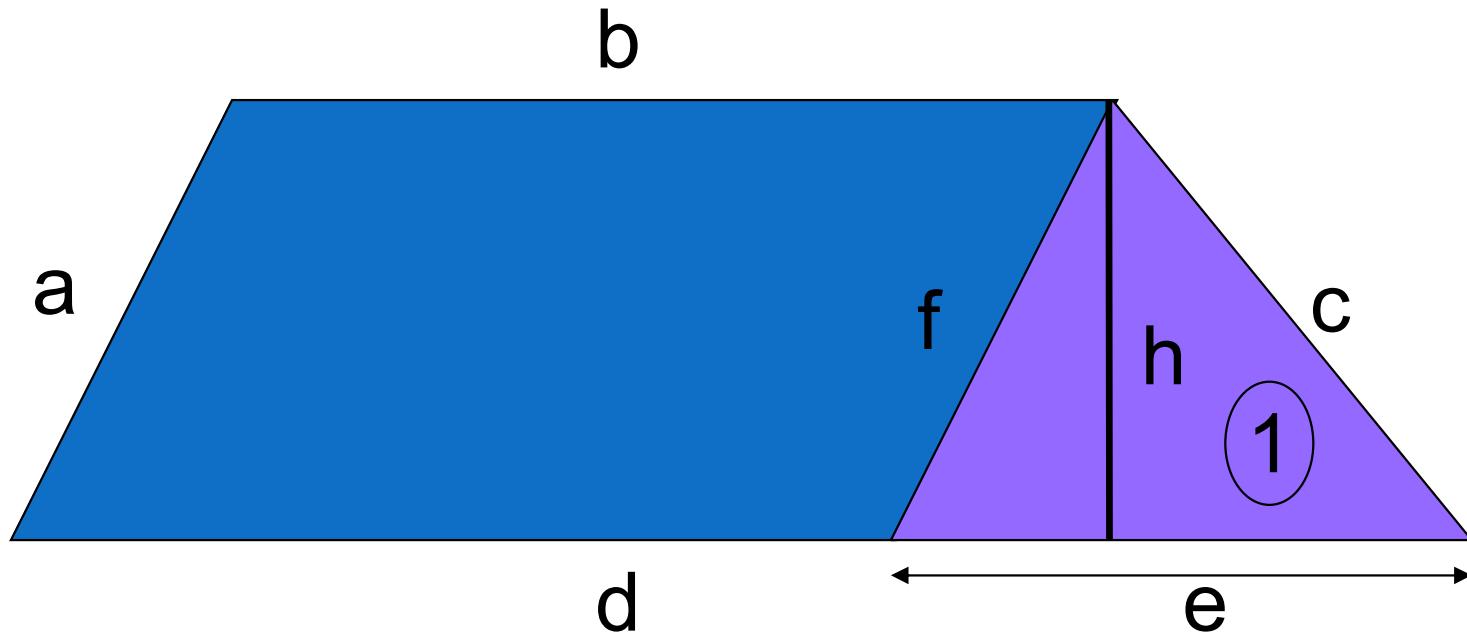
Ví dụ

Tính diện tích hình thang khi biết 4 cạnh



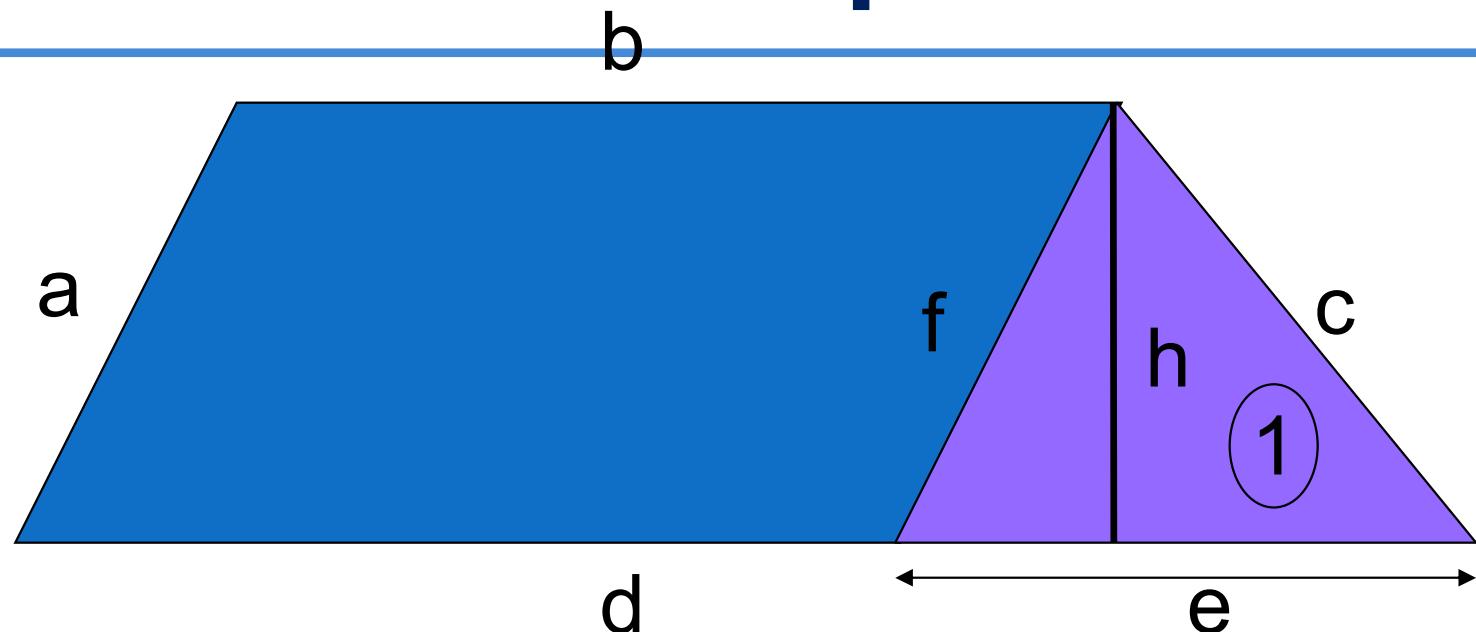
Mô tả bài toán

- 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



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

Ví dụ



- ❖ 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



Chuẩn hóa thuật toán



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!");
}
```



Phương pháp giải quyết bài toán bằng máy tính



- ❖ Xác định trực tiếp lời giải
- ❖ Tìm kiếm lời giải



Xác định trực tiếp lời giải



- ❖ Bài toán thông dụng, đơn giản, đã có lời giải từ trước.
- ❖ Xác định trực tiếp được lời giải qua
 - Các thủ tục tính toán theo công thức, hệ thức, định luật...
 - Các thủ tục bao gồm một số hữu hạn các thao tác sơ cấp, có thể chuyển thành các thuật toán và chương trình chạy trên máy tính.



Hướng xác định trực tiếp lời giải



❖ Trường hợp dùng các công thức lắp để tính gần đúng nghiệm của bài toán.

- Lời giải xác định bởi các công thức lắp có thể xấp xỉ lời giải thật sự của bài toán với độ chính xác tăng theo quá trình lắp.
 - Ví dụ: giải phương trình $f(x)=0$ bằng PP chia đôi
- Đây là hạn chế khi tính toán thủ công nhưng là thế mạnh của máy tính.
- Được xem là cách xác định trực tiếp lời giải



Hướng tìm kiếm lời giải



- ❖ Cách tiếp cận dựa theo nguyên lý “**thử - sai**”.
- ❖ Ứng dụng hiệu quả cho một số bài toán - vấn đề phức tạp
- ❖ Có nhiều phương pháp, cách làm đã được đề xuất



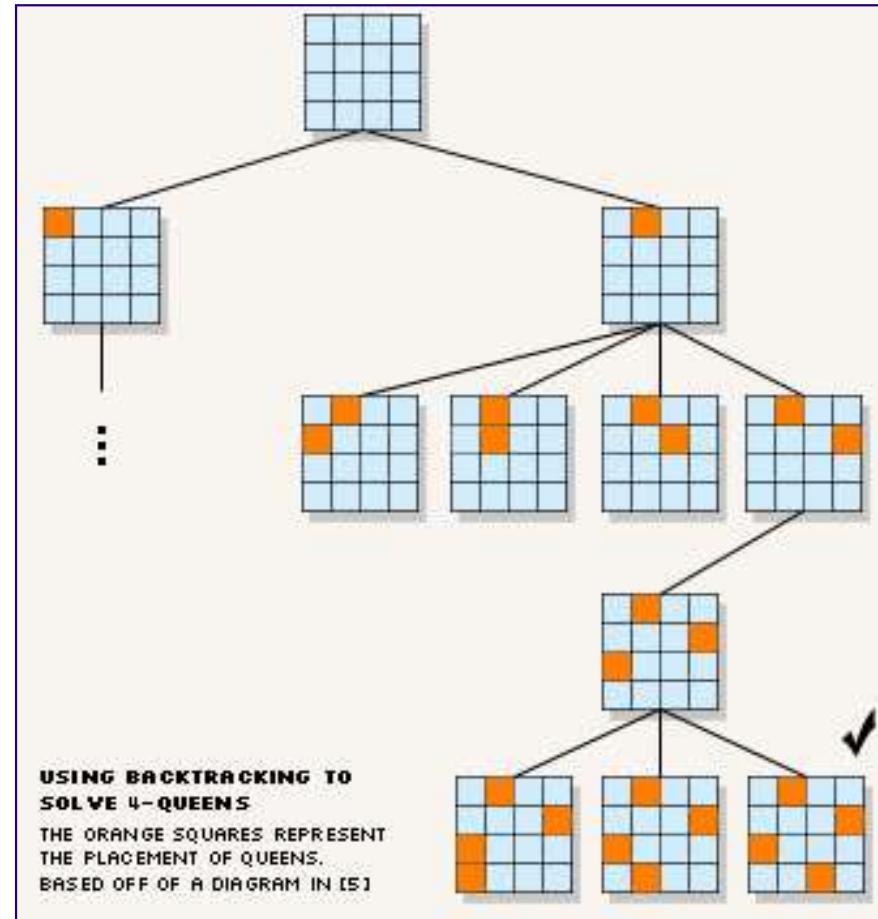
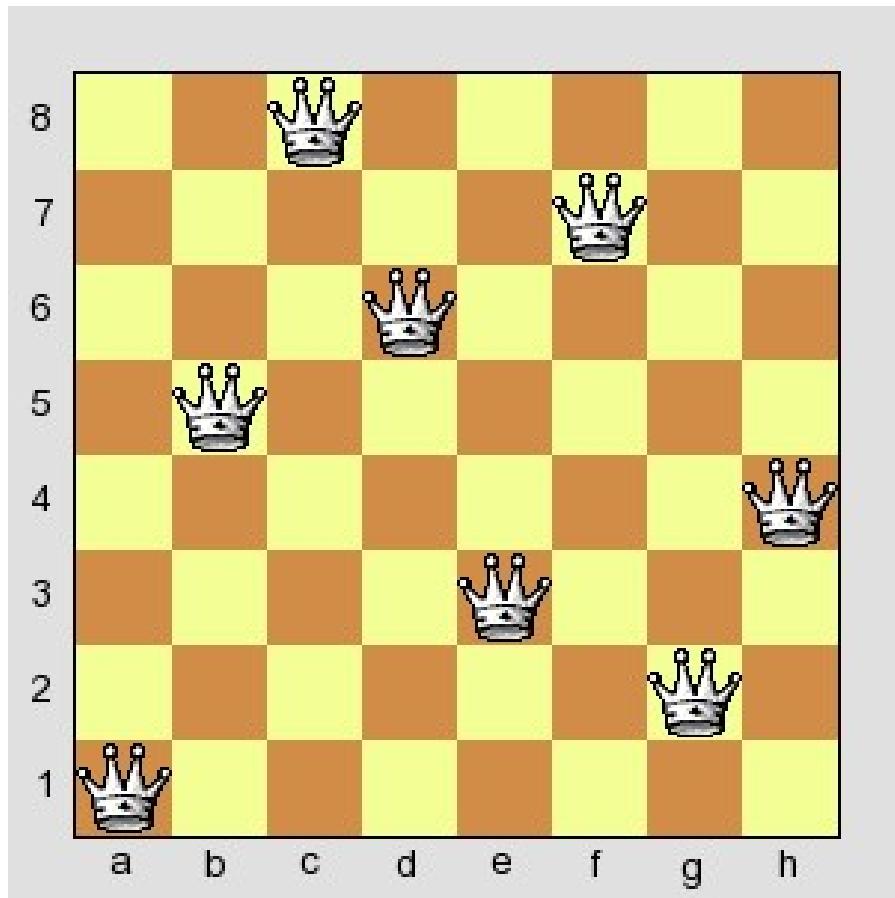
Hướng tìm kiếm lời giải → Một số phương pháp



- ❖ Phương pháp liệt kê hay vét cạn:
 - Xác định tập các khả năng chứa các lời giải và cách thức liệt kê của từng khả năng để thử, không bỏ sót một khả năng nào.
- ❖ Phương pháp thử ngẫu nhiên:
 - Thủ một số khả năng được chọn ngẫu nhiên trong tập (*rất lớn*) các khả năng.
 - Khả năng thành công tùy theo chiến lược chọn ngẫu nhiên và một số điều kiện cụ thể của bài toán.
- ❖ Chia bài toán thành bài toán con:
 - Chia cho tới khi bài toán ban đầu được quy thành bài toán con có lời giải
- ❖ Phương pháp quay lui:
 - Đánh dấu các thử nghiệm thất bại và thử khả năng mới (quay lui tìm đường khác).



Hướng tìm kiếm lời giải → Ví dụ bài toán 8 hậu





Thuật toán



- ❖ Khái niệm
- ❖ Biểu diễn thuật toán
- ❖ Một số thuật toán ví dụ



Khái niệm



- ❖ 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



- ❖ Một tập các lệnh hay chỉ thị nhằm hướng dẫn việc thực hiện một công việc nào đó
- ❖ Bao gồm một dãy hữu hạn các chỉ thị rõ ràng và có thể thi hành được, được bố trí theo một trình tự nhất định, cần thực hiện trên những dữ liệu vào sao cho sau một số hữu hạn bước ta thu được kết quả của bài toán cho trước
- ❖ Thuật toán là sự thể hiện của một phương pháp để giải quyết một vấn đề



Ví dụ



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.



Định nghĩa trong Khoa học máy tính



Thuật toán để giải một bài toán là một dãy hữu hạn 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



Thao tác/lệnh



- ❖ Là hành động cần được thực hiện bởi cơ chế của thuật toán
- ❖ Các thao tác (*lệnh*) sẽ biến đổi bài toán từ trạng thái trước tới trạng thái sau
- ❖ Dãy các thao tác cần thiết sẽ biến đổi bài toán từ trạng thái ban đầu đến kết quả
- ❖ Các thao tác có thể phân tích thành thao tác khác nhau
- ❖ Thứ tự thao tác là quan trọng
 - Cùng tập thao tác, thứ tự khác nhau dẫn đến kết quả khác nhau
- ❖ Cơ cấu thể hiện trình tự thực hiện các thao tác gọi là **Cấu trúc điều khiển**
 - Có 3 loại cơ bản: Tuần tự, Lặp, Rẽ nhánh



Các 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/Xuất



❖ 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
- ❖ Cùng một điều kiện nhập, cùng một giải thuật thì 2 bộ VXL (người, máy) phải cho ra cùng một kết quả



Tính hữu hạn



- ❖ 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,..



Tính hiệu quả



❖ Thực hiện thuật toán cần

- Thời gian thực thi
- Các công cụ hỗ trợ (giấy, 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 $x^2+5x+6=0$



Thuật toán



- ❖ Khái niệm
- ❖ Biểu diễn thuật toán
- ❖ Thuật toán đệ quy
- ❖ Một số thuật toán ví dụ



Biểu diễn thuật toán



- ❖ Mô tả thuật toán một cách tường minh
 - Truyền đạt thuật toán cho người khác
 - “*Truyền đạt*” thuật toán cho máy tính
- ❖ Phương pháp:
 1. Ngôn ngữ tự nhiên: người - người
 2. Lưu đồ thuật toán (Sơ đồ TT): sử dụng ký pháp
 3. Mã giả: mô phỏng ngôn ngữ lập trình
 4. Ngôn ngữ lập trình: chương trình máy tính



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



Biểu diễn bằng ngôn ngữ tự nhiên



❖ **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.



Ví dụ



Tìm giá trị lớn nhất của một dãy N số nguyên

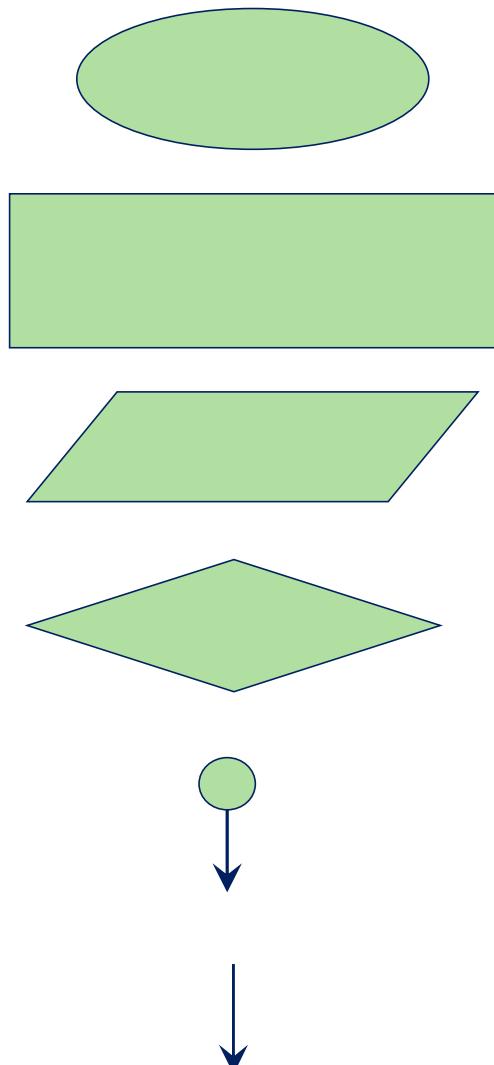
- ❖ B1: Nhập N
- ❖ B2: Nhập dãy số a_i gồm N số.
- ❖ B3: Gán giá trị a_1 cho Max, 2 cho biến i ($i \leftarrow 2$)
- ❖ B4: Nếu $i > N$, thực hiện bước 8
- ❖ B5: Nếu $a_i > Max$, gán giá trị a_i cho Max.
- ❖ B6: Tăng i lên 1 đơn vị.
- ❖ B7: Quay lên B4.
- ❖ B8: Thông báo: Max là giá trị lớn nhất dãy
- ❖ B9: Kết thúc.



Lưu đồ thuật toán



❖ Ký pháp



Bắt đầu hoặc kết thúc

Thao tác tính toán

Lệnh vào, lệnh ra

Kiểm tra điều kiện, rẽ nhánh

Nối tiếp đoạn lệnh

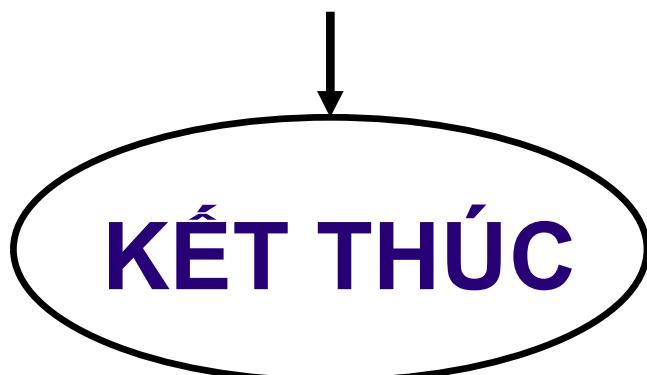
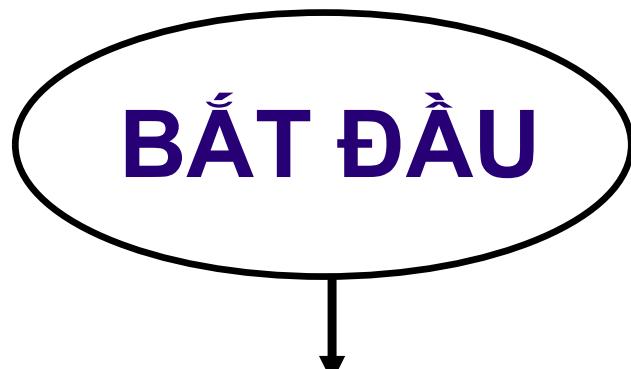
Luồng thực hiện



Khối giới hạn



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

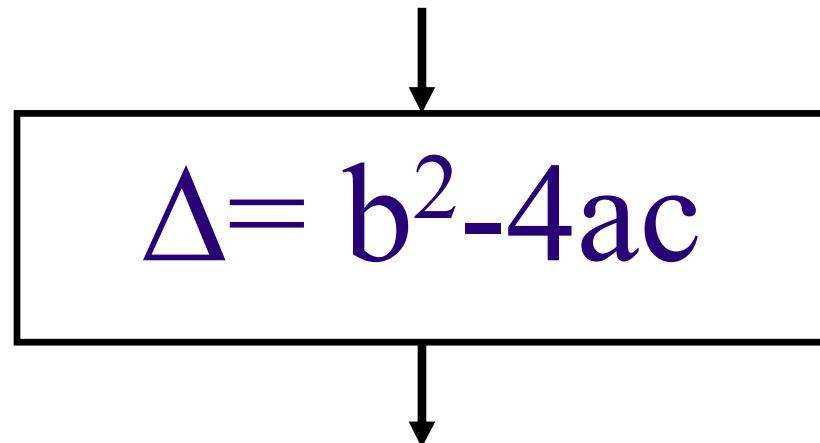




Khối thao tác

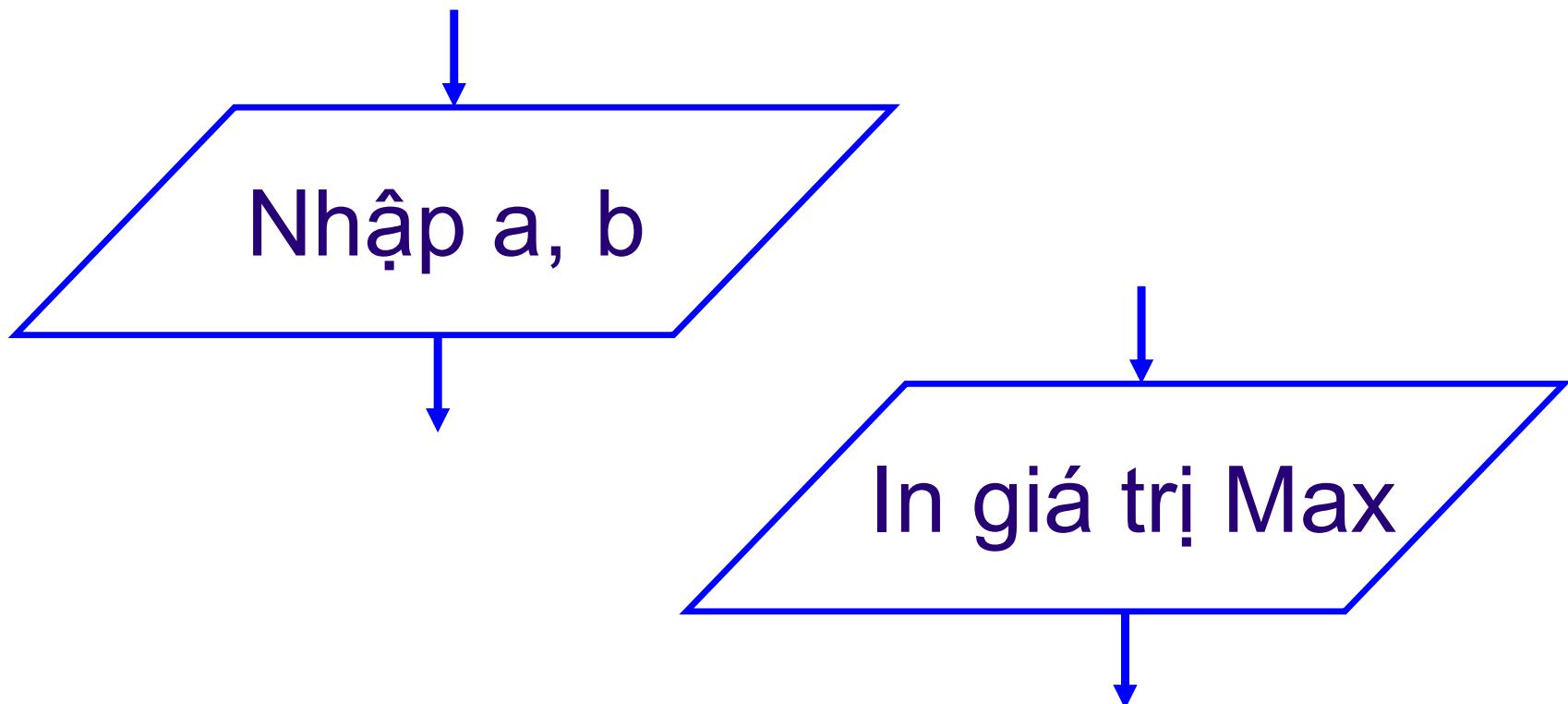


Hình chữ nhật chứa dãy các lệnh xử lý dữ liệu



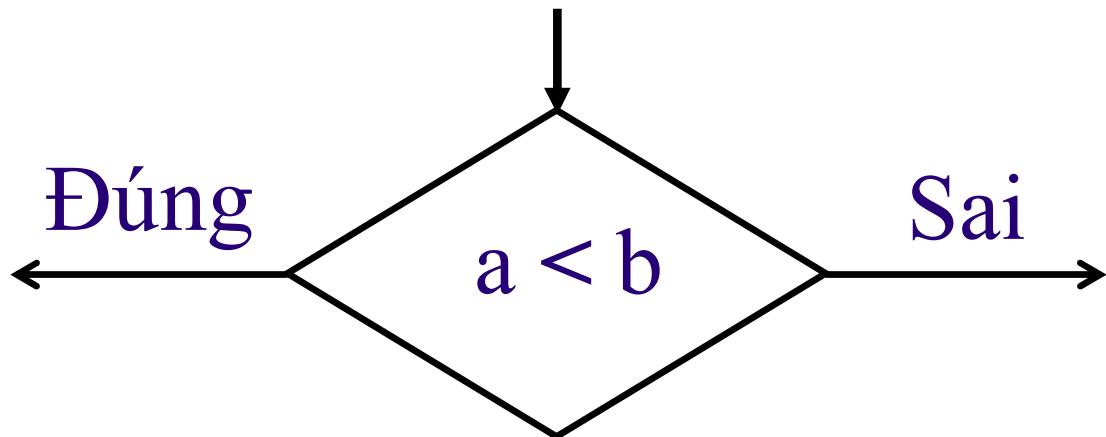
Khôi vào/ra dữ liệu

Hình bình hành chứa thao tác nhập/ xuất dữ liệu



Khối điều kiện

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



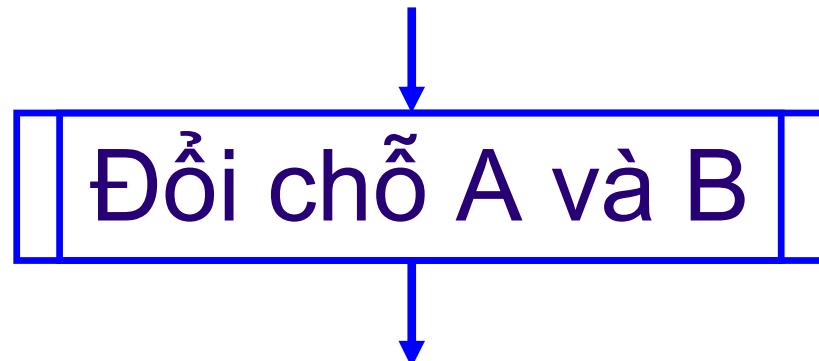


Khôi gợi chương trình con

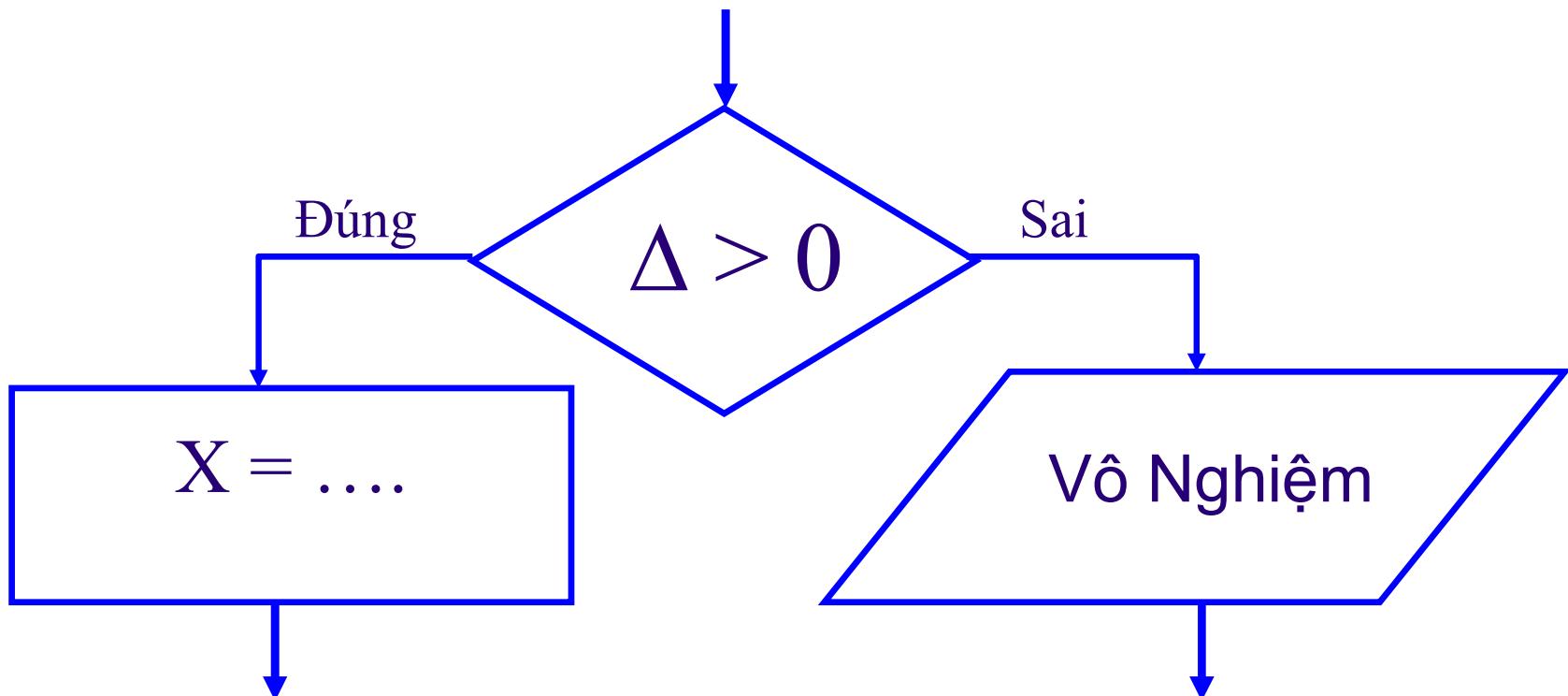


Là hình chữ nhật, cạnh kép chứa tên một chương trình con cần thực hiện

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



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





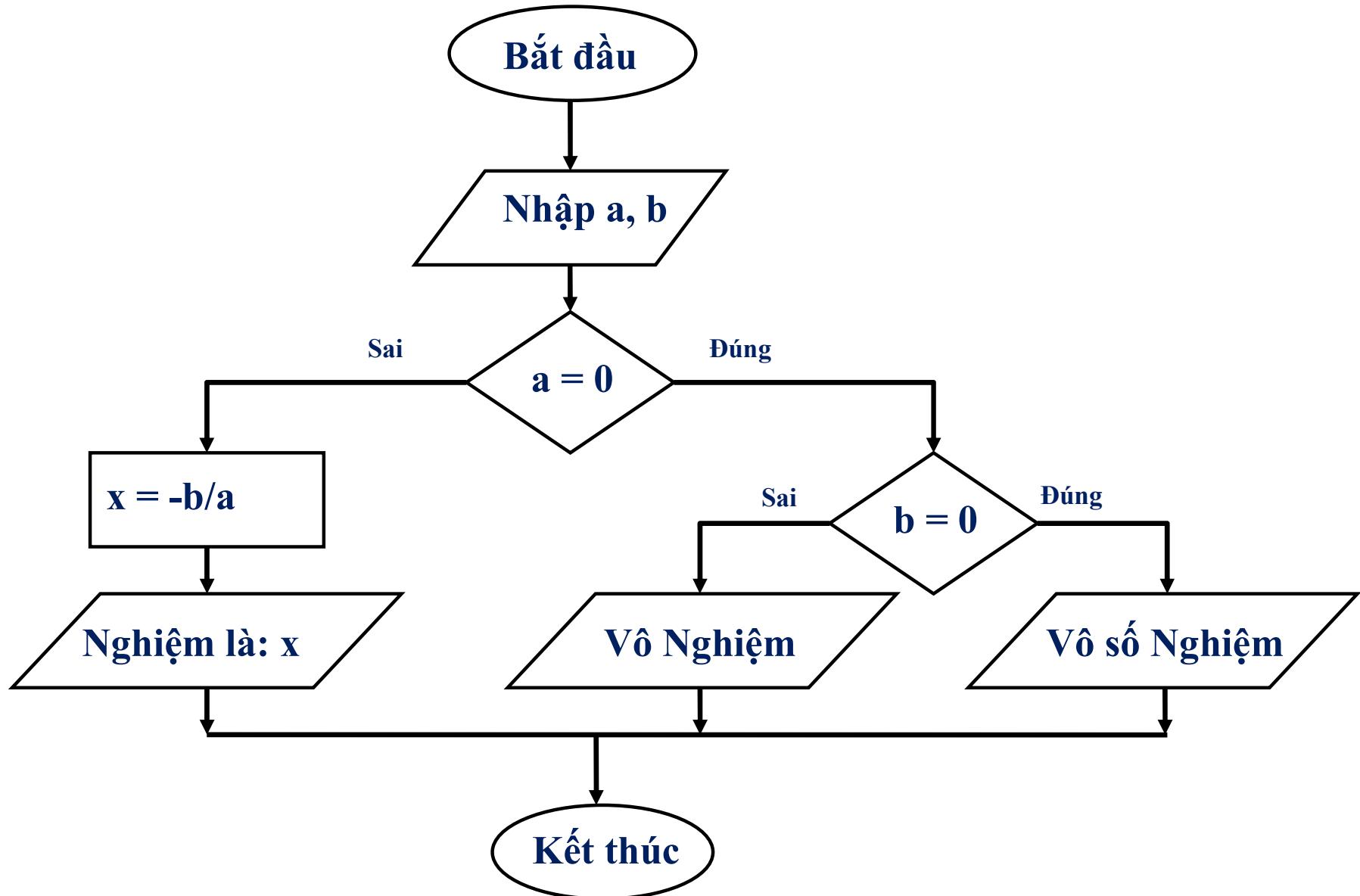
Hoạt động



- ❖ Bắt đầu từ khõi 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

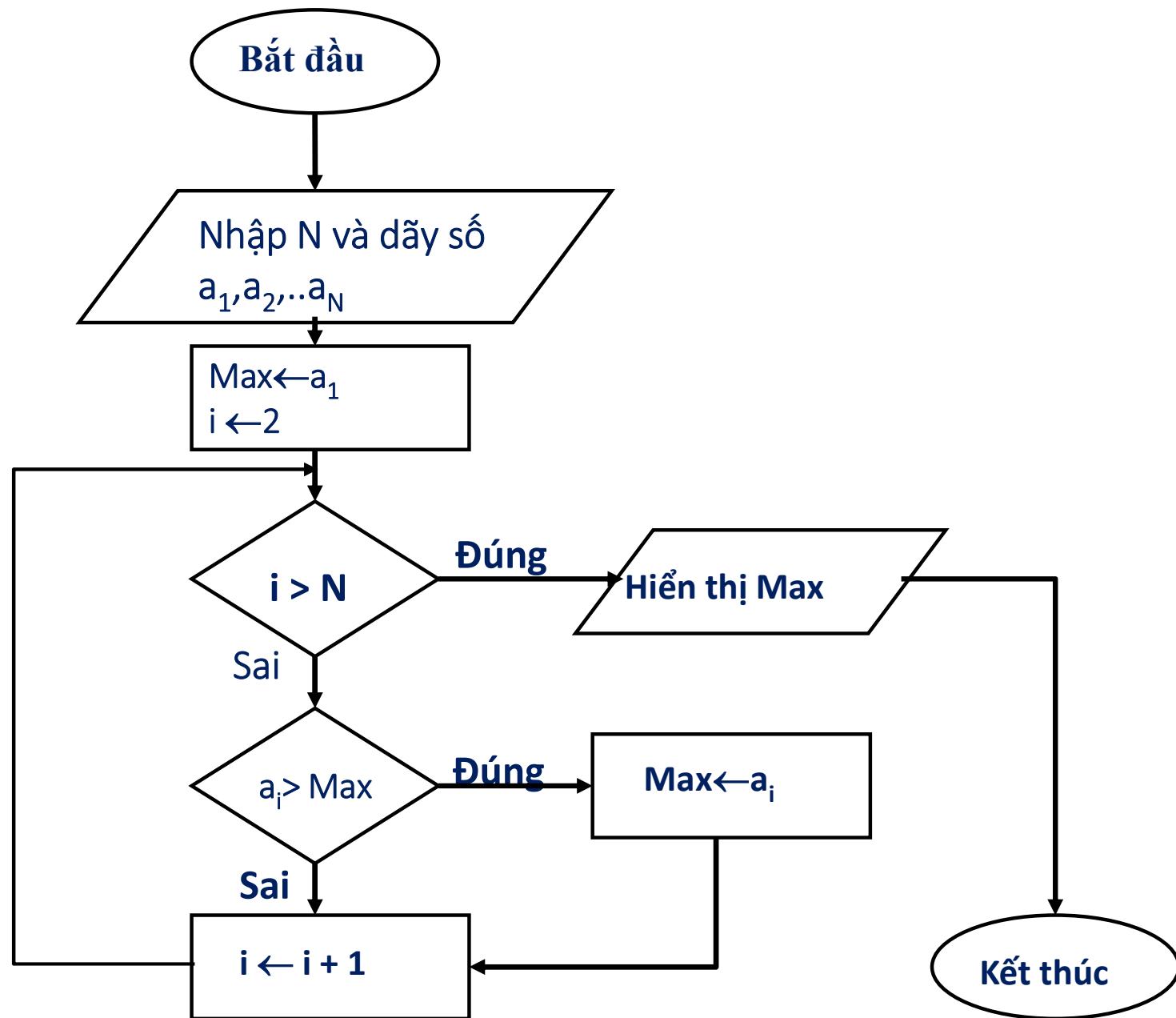


VD: Biểu diễn sơ đồ khôi





VD: Biểu diễn sơ đồ khối





Mã giả



❖ Mô tả thuật toán theo ngôn ngữ phỏng theo ngôn ngữ lập trình

- Sử dụng các mệnh đề có cấu trúc chuẩn hóa
- Vẫn 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

Mô tả tựa Pascal



Thao tác gán giá trị



❖ Mục đích:

- Đặt giá trị cho một biến

❖ Biểu diễn:

Max := a₁

Max ← a₁

n ← n + 1



Cấu trúc lựa chọn/rẽ nhánh (Mô tả tựa Pascal)



if <điều kiện> **then** <hành động>

endif

Hoặc là

if <điều kiện> **then** <hành động>

else <hành động>

endif



Cấu trúc lặp (Mô tả tựa Pascal)



```
while <điều kiện> do  
    <hành động>  
end while
```

```
repeat  
    <hành động>  
until <điều kiện>
```

```
for biến←giá trị đầu to giá trị cuối do  
    <hành động>  
end for  
  
for biến ←giá trị đầu downto giá trị cuối  
do  
    < hành động>  
end for
```



Lệnh nhảy không điều kiện



Nhảy đến vị trí có nhãn là L
goto L

Chú ý: không nên dùng!



Hàm/Thủ tục



❖ Khai báo hàm

Function <Tên hàm>(<Các tham số>)

Hành động với các tham số

return <Giá trị>

End Function

❖ Gọi hàm

[**Call**] <Tên hàm>(<Các tham số>)



Ví dụ: tìm số lớn nhất của dãy



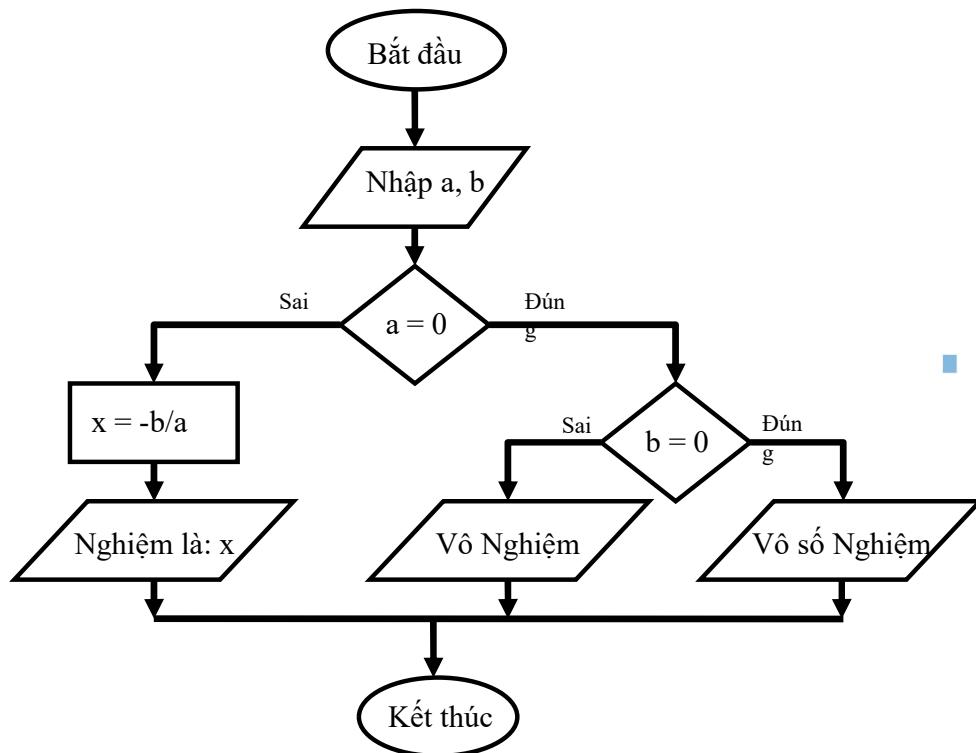
- 1. Begin**
- 2. Input N**
- 3. Input a_1, a_2, \dots, a_N**
4. $\text{Max} \leftarrow a_1$
5. $i \leftarrow 2$
- 6. While $i \leq N$ do**
 7. **If $a_i > \text{Max}$ Then**
 8. $\text{Max} \leftarrow a_i$
 9. **End if**
 10. $i \leftarrow i+1$
- 11. End while**
- 12. Output Max**
- 13. End**



Ngôn ngữ lập trình



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("Vô số nghiệm");
        else
            printf("Vô nghiệm");
    else
        printf("Nghiệm %5.2f",-b/a);
    return 0;
}
```

→ Sẽ học trong nội dung tiếp theo của học phần



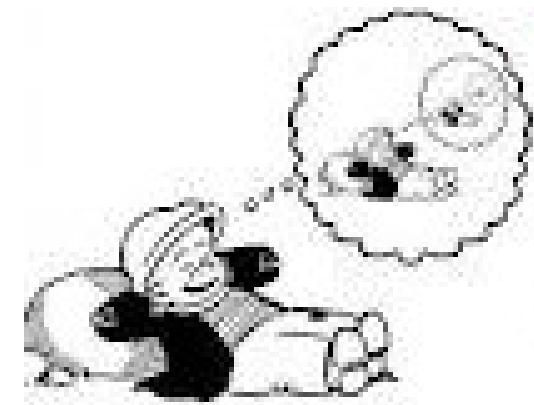
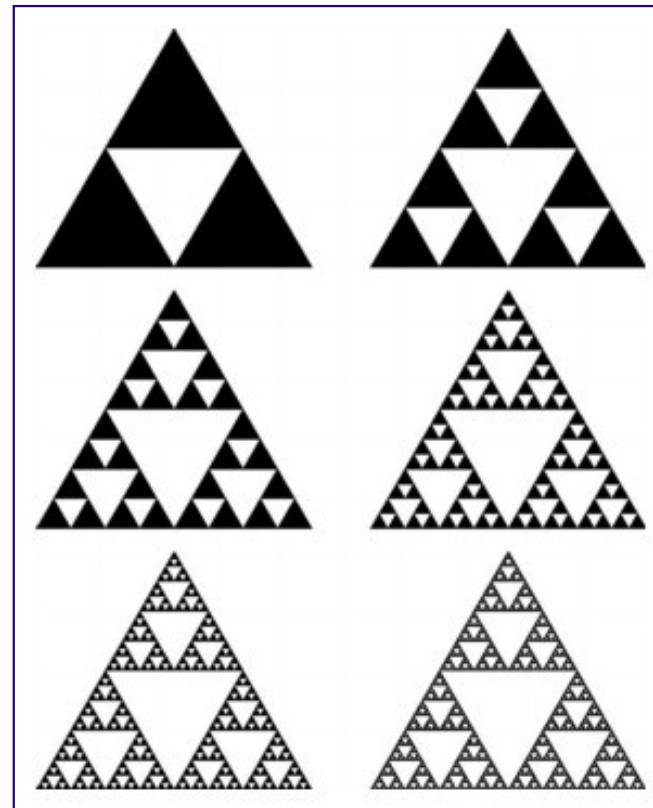
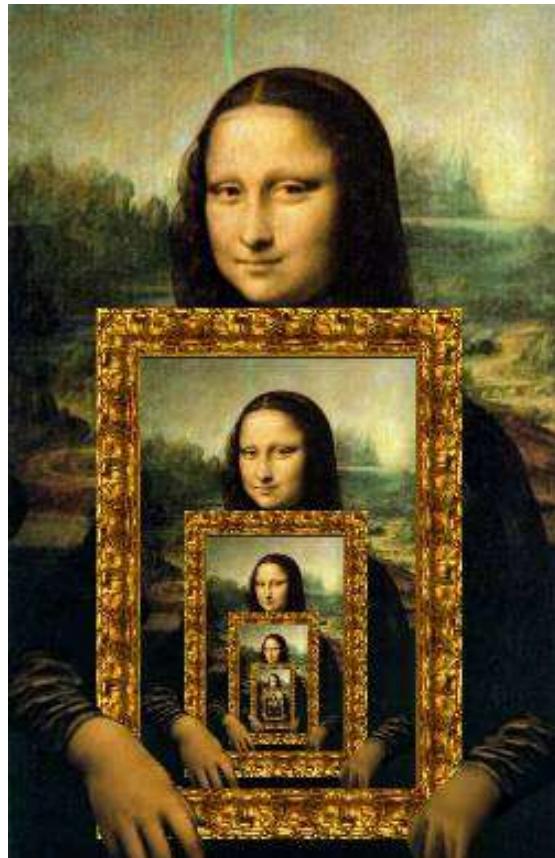
Thuật toán



- ❖ Khái niệm
- ❖ Biểu diễn thuật toán
- ❖ Thuật toán đệ quy
- ❖ Một số thuật toán ví dụ



Ví dụ đệ quy





Khái niệm thuật toán đệ quy



- ❖ Bài toán có thể được phân tích và đưa tới việc giải một bài toán cùng loại nhưng cấp độ thấp hơn,
 - Độ lớn dữ liệu vào nhỏ hơn
 - Giá trị cần tính toán nhỏ hơn
- ❖ **Ví dụ:** Giai thừa của một số tự nhiên n , ký hiệu là $n!$, được định nghĩa bằng cách quy nạp như sau:
 - $0! = 1$,
 - $n! = (n-1)! * n$, với mọi $n > 0$
- ❖ **Thuật toán đệ qui:** mỗi bước của thuật toán tái hiện lại chính thuật toán đó với đầu vào nhỏ hơn



VD: Tính giai thừa của một số tự nhiên



- ❖ Input: số tự nhiên n
- ❖ Output: $F(n)=n!$

❖ Thuật toán:

Function F(n)

If $n=0$ **then**

Return 1

Else

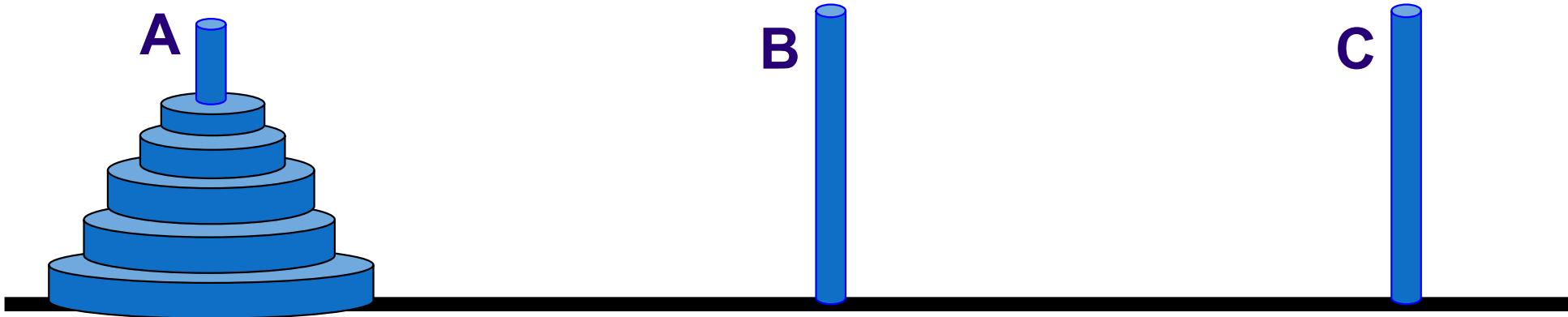
Return $n * F(n-1)$

End If

End Function

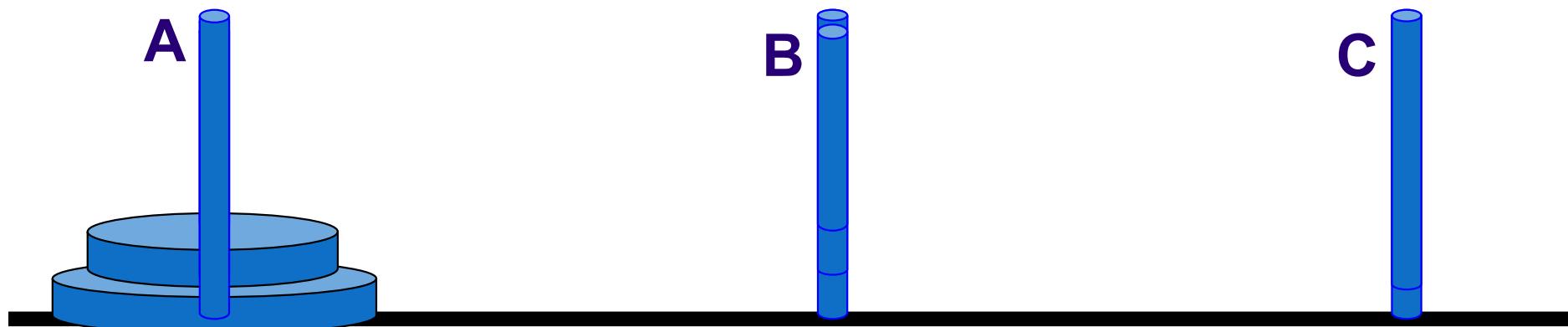


Ví dụ: Bài toán tháp Hà Nội



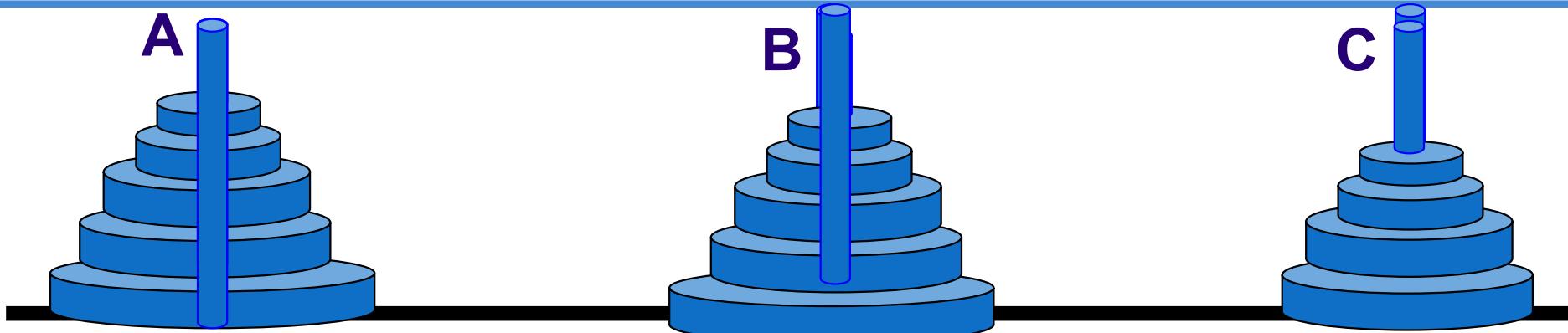
Yêu cầu: Dịch chuyển N đĩa từ cọc A sang B với trung gian là cọc C

$N=2$:





Trường hợp tổng quát với $N > 2$:



```
PROCEDURE ThapHanoi (N,A,B,C)
  IF N = 2 Then Print(A→C, A→B, C→B)
  ELSE
    ThapHanoi(N-1,A,C,B)
    Print(A →B)
    ThapHaNoi(N-1,C,B,A)
  ENDIF
END
```



Lưu ý



- ❖ Thuật toán đệ qui gồm 2 phần
 - **Phân cơ sở:** không cần thực hiện lại thuật toán
 - Không có yêu cầu gọi đệ qui.
 - Là điều kiện dừng thuật toán
 - **Phân đệ qui:** có yêu cầu gọi đệ qui
 - Yêu cầu thực hiện lại thuật toán
 - Đặt trong một điều kiện kiểm tra việc gọi đệ qui.
- ❖ Đệ qui gọi rất nhiều chương trình con
 - dễ gây tràn bộ nhớ (stack)
- ❖ Nếu có thể, nên khử đệ quy (vd: dùng vòng lặp)



Thuật toán



- ❖ Khái niệm
- ❖ Biểu diễn thuật toán
- ❖ Thuật toán đệ quy
- ❖ Một số thuật toán ví dụ



Các bài toán

1. Thuật toán số học

- Hoán đổi giá trị
- Số nguyên tố, phân tích ra thừa số nguyên tố...
- Tìm ước số chung, phân số tối giản
- Số hoàn hảo

2. Thuật toán trên dãy

- Vào/ra dãy
- Tìm Max, Min
- Sắp xếp
- Tìm phần tử; Đếm phần tử
- Tính toán trên các phần tử..
 - Trung bình cộng, tính tổng,...
- Chèn phần tử/Xóa phần tử (*liên quan tới kiểu mảng*)



Hoán đổi giá trị 2 biến X, Y



Nguyên tắc: Dùng một biến trung gian T

Function swap(X,Y)

T←X

X ←Y

Y←T

End Function



Kiểm tra số nguyên tố



Begin

- 1. Input P**
- 2. Flag \leftarrow FALSE**
- 3. If $P=1$ Then Goto Bước 6**
- 4. flag \leftarrow TRUE (*gán cho cờ hiệu "flag" giá trị true*)**
- 5. For $k:=2$ to $p-1$ do**
 If (k là ước số của P) **Then**
 flag \leftarrow FALSE;
 Goto B6
 Endif
End for
- 6. If flag=TRUE Then Output: P là số nguyên tố**
Else Output: P không là số nguyên tố
Endif
End