



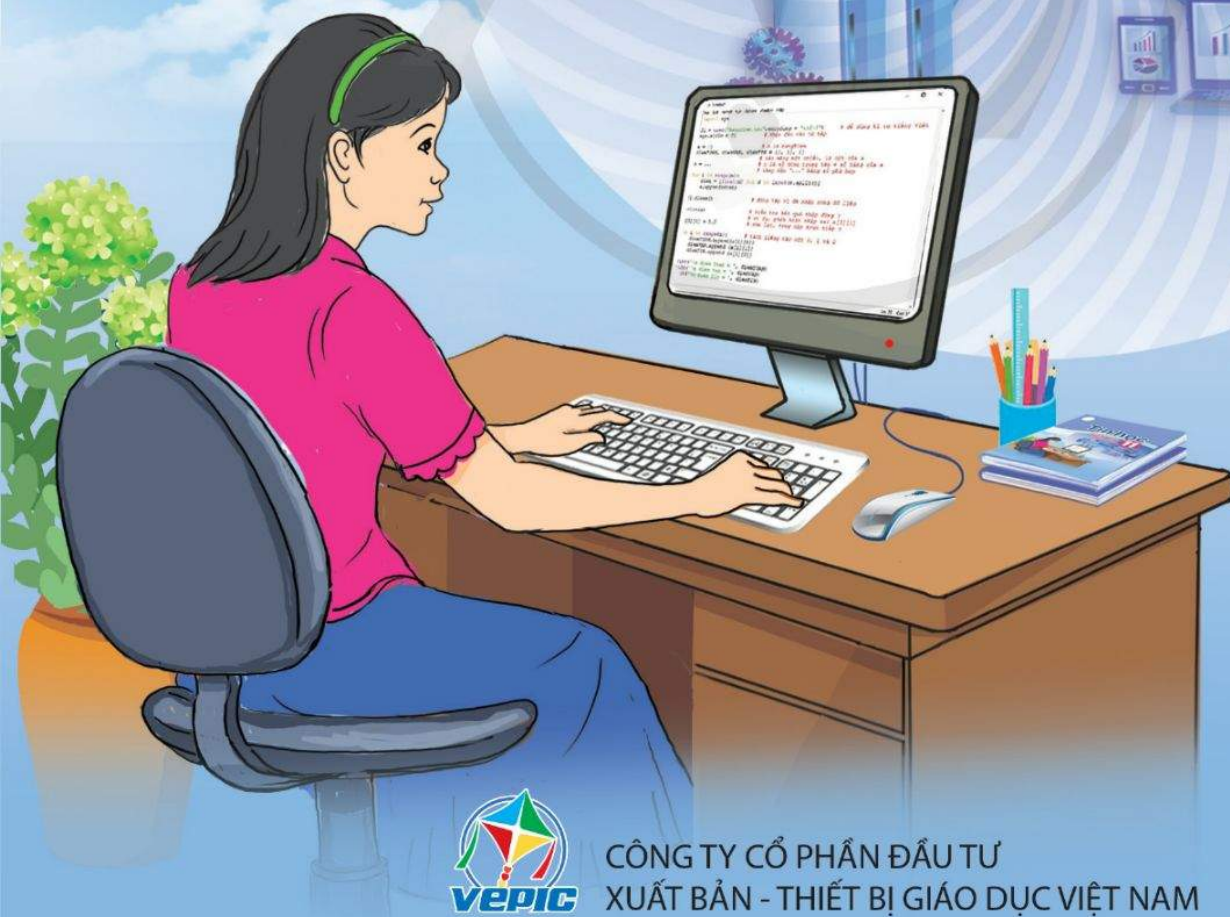
HỒ SĨ ĐÀM (Tổng Chủ biên) – NGUYỄN ĐÌNH HOÀ (Chủ biên)
HOÀNG VÂN ĐÔNG – HỒ CẨM HÀ – LÊ MINH HOÀNG
PHẠM THỊ ANH LÊ – NGUYỄN THANH TÙNG

Tin học

KHOA HỌC
MÁY TÍNH

11

BẢN MẪU



CÔNG TY CỔ PHẦN ĐẦU TƯ
XUẤT BẢN - THIẾT BỊ GIÁO DỤC VIỆT NAM

HỘI ĐỒNG QUỐC GIA THẨM ĐỊNH SÁCH GIÁO KHOA
Môn: Tin học – Lớp 11

*(Kèm theo Quyết định số 2026/QĐ-BGDĐT ngày 21 tháng 7 năm 2022
của Bộ trưởng Bộ Giáo dục và Đào tạo)*

Họ và tên	Chức vụ Hội đồng
Ông Lê Hoài Bắc	Chủ tịch
Ông Trần Đăng Hưng	Phó Chủ tịch
Ông Hồ Vĩnh Thắng	Ủy viên, Thư kí
Ông Đỗ Năng Toàn	Ủy viên
Ông Nguyễn Trung Trực	Ủy viên
Ông Nguyễn Đức Nhuận	Ủy viên
Bà Lưu Thị Bích Hương	Ủy viên
Bà Nguyễn Thị Vân Khánh	Ủy viên
Bà Phan Thị May	Ủy viên
Bà Nguyễn Thị Hồng Thái	Ủy viên
Ông Hoàng Văn Quyến	Ủy viên

HỒ SĨ ĐÀM (Tổng Chủ biên) – NGUYỄN ĐÌNH HOÁ (Chủ biên)
HOÀNG VÂN ĐÔNG – HỒ CẨM HÀ – LÊ MINH HOÀNG
PHẠM THỊ ANH LÊ – NGUYỄN THANH TÙNG

Tin học

KHOA HỌC MÁY TÍNH

11

BẢN MẪU



CÔNG TY CỔ PHẦN ĐẦU TƯ
XUẤT BẢN - THIẾT BỊ GIÁO DỤC VIỆT NAM

CÁC CHỦ ĐỀ

CHỦ ĐỀ A

Máy tính và xã hội tri thức

Thế giới thiết bị số – Hệ điều hành và phần mềm ứng dụng

CHỦ ĐỀ C

Tổ chức lưu trữ, tìm kiếm và trao đổi thông tin

Tìm kiếm và trao đổi thông tin trên mạng

CHỦ ĐỀ D

Đạo đức, pháp luật và văn hoá trong môi trường số

Ứng xử văn hoá và an toàn trên mạng

CHỦ ĐỀ F

Giải quyết vấn đề với sự trợ giúp của máy tính

Giới thiệu các hệ cơ sở dữ liệu

CHỦ ĐỀ G

Hướng nghiệp với tin học

Giới thiệu nghề quản trị cơ sở dữ liệu

CHỦ ĐỀ F^{CS}

Giải quyết vấn đề với sự trợ giúp của máy tính

Kĩ thuật lập trình

KÍ HIỆU DÙNG TRONG SÁCH



Khởi động



Hoạt động



Luyện tập



Vận dụng



Câu hỏi tự kiểm tra



Thuật ngữ

Các em giữ gìn sách cẩn thận, không viết vào sách để sử dụng được lâu dài.

LỜI NÓI ĐẦU

Thực hiện định hướng nghề nghiệp ở cấp trung học phổ thông, nội dung cốt lõi và chuyên đề học tập ở môn Tin học được phân hoá theo hai định hướng là Tin học ứng dụng (ICT) và Khoa học máy tính (CS). Học sinh được lựa chọn học theo một trong hai định hướng đó tùy theo nghề em muốn làm trong tương lai.

Bộ sách giáo khoa Tin học Cánh Diều ở lớp 11 gồm bốn quyển: quyển *Tin học 11 Khoa học máy tính*, quyển *Tin học 11 Tin học ứng dụng* và hai quyển Chuyên đề học tập dành cho hai định hướng.

Quyển sách *Tin học 11 Khoa học máy tính* là nội dung cốt lõi của định hướng CS. Các chủ đề A, C, D, F, G với thời lượng 34 tiết học là chung và giống nhau cho cả hai định hướng. Chủ đề F^{CS} về “Kỹ thuật lập trình” chỉ dành riêng cho định hướng CS với thời lượng 30 tiết.

Mỗi chủ đề gồm một số bài học. Ở đầu mỗi bài học đều nêu những yêu cầu mà các em cần đạt được sau khi hoàn thành bài học đó. Tiếp theo là hoạt động *Khởi động* và các nội dung kiến thức mới; sau đó có những câu hỏi, bài tập để các em *Luyện tập*, *Vận dụng*. Cuối mỗi bài học là *Câu hỏi tự kiểm tra* và *Tóm tắt bài học*. Một số bài học còn có *Bài tìm hiểu thêm*. Các em thực hiện mỗi bài thực hành theo hướng dẫn trong sách và dưới sự hướng dẫn của thầy, cô giáo.

Tin học luôn đồng hành cùng các em trong cuộc sống mỗi ngày, trong học tập và làm việc. Tin học sẽ nâng cánh cho các em bay cao, bay xa và thành công trong cuộc sống.

Các tác giả

BẢNG GIẢI THÍCH THUẬT NGỮ

Thuật ngữ	Giải thích	Trang
liên kết hai bảng	kết nối các bản ghi của hai bảng theo một biểu thức logic (gọi là điều kiện kết nối) được thoả mãn	72
kiểu dữ liệu có cấu trúc	được tạo thành từ nhiều mục dữ liệu nguyên thuỷ theo một quy cách định trước, ví dụ như mảng, danh sách,...	145
liên kết khoá chính và khoá ngoài	thiết lập việc tham chiếu giá trị của trường (cột) dữ liệu là khoá ngoài trong một bảng với giá trị của trường khoá chính trong bảng khác	59
lược đồ tiến trình	một mô tả trình tự thực hiện các bước giải quyết bài toán	127
nén dữ liệu	làm giảm dung lượng lưu trữ của dữ liệu số, chuyển dữ liệu gốc sang dạng được biểu diễn dùng dung lượng lưu trữ nhỏ hơn	83
NFC	(Near-Field Communications) công nghệ kết nối không dây tầm ngắn trong khoảng cách 4 cm. Công nghệ này sử dụng cảm ứng từ trường để thực hiện kết nối giữa các thiết bị (smartphone, tablet, loa, tai nghe,...) khi có sự tiếp xúc trực tiếp (chạm)	37
tham số tùy chọn/bắt buộc	tham biến đầu vào của một hàm mà ta có thể bỏ qua khi viết lời gọi hàm và chương trình sẽ gán cho nó một giá trị mặc định trước, đối lập với tham số bắt buộc phải cho giá trị khi viết lời gọi hàm	100
truy cập dữ liệu	việc lấy ra, sửa đổi, sao chép hay di chuyển những mục dữ liệu trong CSDL	31
truy cập ngẫu nhiên	việc truy cập trực tiếp tới một vị trí bất kì trong bộ nhớ mà không phải dò tìm tuần tự từ một điểm bắt đầu nào đó	90
truy xuất thông tin	tìm và lấy ra thông tin cụ thể từ dữ liệu được lưu trữ	67

BÀI 1

BÊN TRONG MÁY TÍNH

Học xong bài này, em sẽ:

- ✓ Nhận biết được sơ đồ của các mạch logic AND, OR, NOT; giải thích được vai trò của các mạch logic trong thực hiện các tính toán nhị phân.
- ✓ Nêu được tên, nhận diện được hình dạng, mô tả được chức năng và giải thích được đơn vị đo hiệu năng của các bộ phận chính bên trong máy tính.



Em hãy cho biết CPU là gì và làm nhiệm vụ gì trong máy tính?

1 Các cổng logic và tính toán nhị phân

a) Cổng logic

Trong máy tính, một bóng bán dẫn chỉ thực hiện được chức năng bật hoặc tắt mạch đơn giản, tương ứng với hai giá trị 0 và 1. Mỗi cách kết hợp các bóng bán dẫn tạo ra một cổng logic. Các cổng logic là thành phần cơ bản thực hiện mọi tính toán trong máy tính.

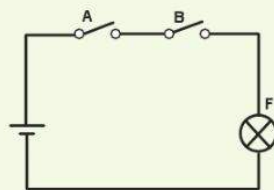


1

Quan sát mạch điện ở *Hình 1*. Mạch có hai công tắc A và B phối hợp để điều khiển đèn F. Đèn chỉ sáng khi cả hai công tắc cùng đóng.

Nếu quy ước: công tắc mở tương ứng với mức “0”, công tắc đóng tương ứng với mức “1”; đèn tắt tương ứng với mức “0”, đèn sáng tương ứng với mức “1”. Em hãy:

- 1) Nêu giá trị đúng tại dấu ? cho mỗi hàng của đầu ra F.
- 2) Nhận xét về hoạt động của mạch điện.



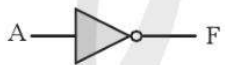
A	B	F
0	0	?
0	1	?
1	0	?
1	1	?

Hình 1. Mạch điều khiển đèn và bảng hoạt động tương ứng

Ta thấy rằng: Để đèn F sáng thì cả công tắc A và công tắc B đồng thời phải đóng, nếu một trong hai công tắc mở thì đèn F tắt. Hoạt động của mạch điện minh họa chức năng của cổng logic AND và bảng hoạt động tương ứng của mạch điện được gọi là bảng chân lí. Cổng AND thực hiện chức năng nhân logic.

Để thực hiện các phép toán logic khác, ta cần có thêm nhiều loại cổng logic. Dựa trên quan hệ giữa đầu ra và đầu vào, các cổng logic được đặt tên tương ứng là cổng AND, cổng OR, cổng NOT, cổng XOR,... *Bảng 1* dưới đây liệt kê một số loại cổng logic thông dụng.

Bảng 1. Một số cổng logic thông dụng

Cổng logic	Kí hiệu	Biểu thức logic	Bảng chân lí	Đặc điểm															
AND		$F = A \text{ AND } B = A \cdot B$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1	Đầu ra bằng 1, khi tất cả đầu vào bằng 1.
A	B	F																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
OR		$F = A \text{ OR } B = A + B$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1	Đầu ra bằng 1, khi hoặc một trong các đầu vào bằng 1.
A	B	F																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
NOT		$F = \text{NOT } A = \overline{A}$	<table><tr><th>A</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	F	0	1	1	0	Đầu ra có giá trị đảo lại giá trị đầu vào.									
A	F																		
0	1																		
1	0																		
XOR		$F = A \text{ XOR } B = A \oplus B$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0	Đầu ra bằng 1 khi hai đầu vào khác nhau.
A	B	F																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	

b) Thực hiện phép toán nhị phân với mạch logic

Các phép toán trên hệ nhị phân cũng có nguyên tắc thực hiện giống như trên hệ thập phân. Ví dụ nguyên tắc cơ bản để cộng hai số nhị phân như ở *Hình 2*.

Giả sử ta cộng hai số nhị phân 1 bit là A với B được tổng là S và nhớ là C. Vì là các số nhị phân nên A và B chỉ nhận các giá trị là 0 hoặc 1, lập bảng các trường hợp có thể xảy ra với các đầu vào A, B và điền giá trị đầu ra S, C tương ứng, ta có bảng chân lí mạch cộng hai số nhị phân 1 bit (*Bảng 2*).

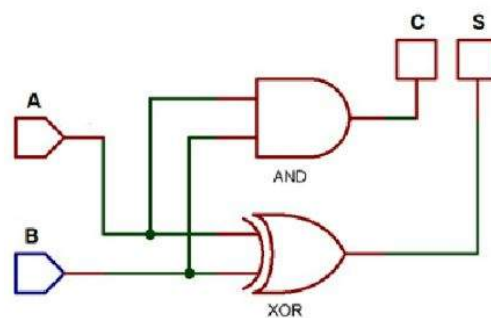
$0 + 0 = 0$ (Bằng 0, nhớ 0)
 $1 + 0 = 1$ (Bằng 1, nhớ 0)
 $0 + 1 = 1$ (Bằng 1, nhớ 0)
 $1 + 1 = 10$ (Bằng 10 (Bằng 0, nhớ 1))

Hình 2. Phép cộng hai bit trong hệ nhị phân

Bảng 2. Bảng chân lí mạch cộng hai số nhị phân 1 bit

Đầu vào		Đầu ra	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

So sánh *Bảng 2* với bảng chân lí của các cổng logic trong *Bảng 1*, để thấy tổng $S = A \text{ XOR } B$ và nhớ $C = A \text{ AND } B$. Từ đó, ta lập được sơ đồ mạch logic để thực hiện phép cộng hai số nhị phân 1 bit (*Hình 3*). Phép cộng hai số nhị phân dài nhiều bit thực hiện bằng cách cộng lần lượt từng cặp bit từ phải sang trái và có bit nhớ (C_{in}) mang sang cột kế bên trái như ở *Bảng 3*.



Hình 3. Mạch cộng 1 bit

Bảng 3. Minh họa phép cộng hai số nhị phân dài hơn 1 bit

C_{in}	1101	Giá trị thập phân
x	101	5
y	1101	13
$x + y$	10010	18

Mạch cộng đầy đủ (FA – Full Adder) có ba đầu vào là A, B và bit nhớ mang sang C_{in} , có hai đầu ra là bit tổng S và bit nhớ C_{out} để phân biệt với C_{in} đầu vào. Mạch cộng đầy đủ là ghép nối hai mạch cộng 1 bit.

Như vậy, bằng cách kết hợp các cổng logic AND, XOR, máy tính có thể thực hiện được phép tính cộng nhị phân. Tương tự, các cổng logic cơ bản cũng có thể kết hợp để tạo thành các mạch logic thực hiện tất cả các tính toán nhị phân khác.

2 Những bộ phận chính bên trong máy tính

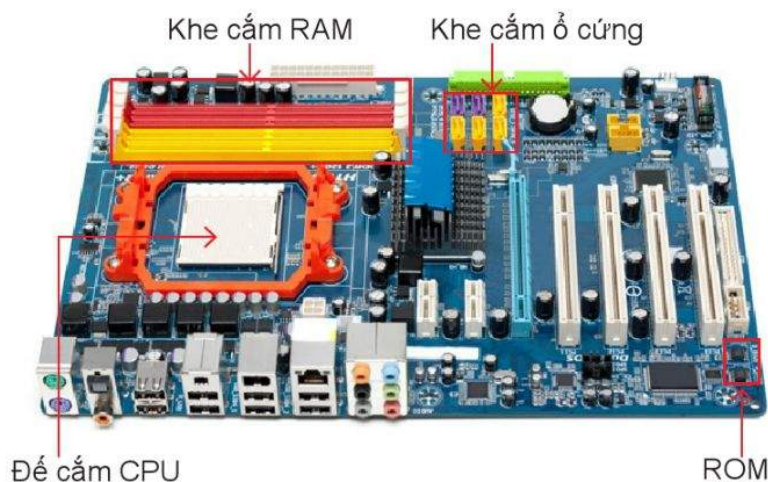


2

Theo em, bộ phận nào của máy tính là quan trọng nhất?

Máy tính có nhiều loại như: máy tính để bàn, máy tính xách tay, máy tính bảng. Bên trong thân máy tính được cấu thành từ các bộ phận chính gồm: bảng mạch chính, CPU, RAM, ROM, thiết bị lưu trữ. Tốc độ và dung lượng của chúng ảnh hưởng lớn tới hiệu năng của máy.

Bảng mạch chính (*Hình 4*) có để cắm CPU, ROM, các khe cắm RAM, các khe cắm ổ cứng và một số khe cắm khác. Bảng mạch chính đóng vai trò làm nền giao tiếp giữa CPU, RAM và các linh kiện điện tử khác phục vụ cho việc kết nối với các thiết bị ngoại vi.



Hình 4. Bảng mạch chính

CPU (Central Processing Unit – bộ xử lý trung tâm) (Hình 5) đóng vai trò bộ não của máy tính; đảm nhiệm công việc tìm nạp lệnh, giải mã lệnh và thực thi lệnh cho máy tính.



Hình 5. CPU

RAM (Random Access Memory – bộ nhớ truy cập ngẫu nhiên) (Hình 6) lưu trữ dữ liệu tạm thời trong quá trình tính toán của máy tính. Dữ liệu sẽ bị mất khi máy tính bị mất điện hoặc khởi động lại.



Hình 6. RAM

ROM (Read Only Memory – bộ nhớ chỉ đọc) (Hình 4) lưu trữ chương trình giúp khởi động các chức năng cơ bản của máy tính.

Thiết bị lưu trữ (Hình 7) dùng để lưu trữ dữ liệu lâu dài và không bị mất đi khi máy tính tắt nguồn. Ngày nay, máy tính thường sử dụng ổ cứng HDD, ổ cứng SSD hoặc ổ USB để lưu trữ dữ liệu.



Hình 7. Cấu tạo bên trong ổ cứng SSD của máy tính

Dung lượng lưu trữ dữ liệu của máy tính là tổng dung lượng của ổ cứng HDD, ổ cứng SSD gắn sẵn bên trong máy tính, không bao gồm dung lượng lưu trữ của RAM. Hiện nay, dung lượng lưu trữ của máy tính có thể lên tới hàng TB.

3 Hiệu năng của máy tính

Hiệu năng của máy tính phụ thuộc vào thông số kỹ thuật của từng bộ phận và sự đồng bộ giữa chúng. Có thể đánh giá nhanh hiệu năng của máy thông qua tốc độ CPU, dung lượng bộ nhớ RAM.

Thông số kỹ thuật cần quan tâm của CPU gồm:

– Tốc độ của CPU: đo bằng Hz (Hertz), biểu thị số chu kỳ xử lý mỗi giây mà CPU có thể thực hiện được. Tốc độ này càng cao thì máy tính chạy càng nhanh. Hiện nay, CPU có tốc độ hàng GHz ($1 \text{ GHz} = 10^9 \text{ Hz}$).

– Số lượng nhân hay lõi (core): CPU có cấu tạo gồm một hoặc nhiều nhân (còn gọi là lõi) vật lý. Với cùng một công nghệ sản xuất, CPU có nhiều nhân hơn thì hiệu năng, khả năng đa nhiệm và tốc độ xử lý tốt hơn.

Thông số kỹ thuật cần quan tâm của RAM là dung lượng. Dung lượng của RAM được đo bằng đơn vị Byte. Hiện nay, máy tính có RAM với dung lượng hàng GB ($1 \text{ GB} = 2^{30} \text{ Byte}$). Máy tính có RAM với dung lượng lớn hơn thì hiệu năng cao hơn.



Câu 1. Em hãy nêu giá trị thích hợp tại dấu ? cho hai cột S và C_{out} để hoàn thành bảng chân lí cho mạch cộng đầy đủ (Bảng 4).

Bảng 4. Bảng chân lí mạch cộng đầy đủ

Đầu vào			Đầu ra	
A	B	C_{in}	S	C_{out}
0	0	0	?	?
0	0	1	?	?
0	1	0	?	?
0	1	1	?	?
1	0	0	?	?
1	0	1	?	?
1	1	0	?	?
1	1	1	?	?

Câu 2. Hãy nêu tên một số thành phần chính bên trong máy tính và cho biết chức năng của nó.



Em hãy sắp xếp thứ tự ưu tiên khi chọn mua máy tính:

a) Ổ cứng dung lượng lớn. b) RAM dung lượng lớn. c) CPU tốc độ cao.



Trong các câu sau, những câu nào đúng?

- a) CPU có tốc độ càng cao thì máy tính có hiệu năng càng cao.
- b) Dung lượng ổ cứng đo bằng GHz.
- c) Các bộ nhớ RAM ngày nay có dung lượng hàng TB.
- d) Dung lượng RAM có ảnh hưởng tới hiệu năng của máy tính.

Tóm tắt bài học

- ✓ Bằng cách kết hợp các cổng logic cơ bản để tạo thành các mạch logic, máy tính có thể thực hiện được các tính toán nhị phân.
- ✓ Các bộ phận chính bên trong thân máy tính gồm: bảng mạch chính, CPU, RAM, ROM, thiết bị lưu trữ.
- ✓ Hiệu năng của máy tính được quyết định bởi hiệu năng của từng thành phần, trong đó CPU, RAM có vai trò quan trọng nhất. Ngày nay, CPU có tốc độ hàng GHz, bộ nhớ RAM có dung lượng hàng GB, ổ cứng có dung lượng hàng TB.

BÀI 2

KHÁM PHÁ THẾ GIỚI THIẾT BỊ SỐ THÔNG MINH

Học xong bài này, em sẽ:

- ✓ Đọc hiểu được một số điểm chính trong tài liệu hướng dẫn về thiết bị số thông dụng. Thực hiện được một số chỉ dẫn trong tài liệu đó.
- ✓ Đọc hiểu và giải thích được một vài thông số cơ bản của các thiết bị số thông dụng.



Em đã sử dụng các thiết bị số của mình như thế nào? Theo em, sử dụng như thế đã đúng cách chưa?

1 Sử dụng đúng cách các thiết bị số

Hiện nay các loại thiết bị kỹ thuật số mới xuất hiện thêm mỗi ngày. Các thiết bị đã phổ biến cũng liên tục được đổi mới, thế hệ sau có nhiều tính năng hiện đại hơn thế hệ trước. Để biết sử dụng thiết bị số kiểu mới, thế hệ mới đúng cách, không làm hỏng và khai thác tốt nhất các tính năng thiết bị, cần đọc kỹ và thực hiện theo đúng hướng dẫn sử dụng ngay từ bước lắp đặt, thiết lập ban đầu và tiếp tục trong quá trình sử dụng sau này.



1

Quan sát *Hình 1*, em hãy:

- 1) Phân biệt mục đích của thông điệp CẢNH BÁO và THẬN TRỌNG.
- 2) Thực hiện theo các bước của hướng dẫn.

Quy trình vệ sinh

Thực hiện theo các quy trình trong phần này để vệ sinh máy tính của bạn một cách an toàn.

⚠ **CẢNH BÁO:** Để ngăn ngừa điện giật hoặc hư hỏng cho các linh kiện, không cố gắng làm sạch máy tính của bạn khi máy đang bật.

1. Tắt máy tính.
2. Ngắt kết nối nguồn AC.
3. Ngắt kết nối với tất cả các thiết bị đang chạy bằng điện bên ngoài.

⚠ **THẬN TRỌNG:** Để tránh gây hư hỏng cho các cấu phần bên trong, không phun chất tẩy rửa hoặc chất lỏng trực tiếp lên bất kỳ bề mặt nào của máy tính. Chất lỏng nhỏ giọt trên bề mặt có thể gây tổn hại vĩnh viễn cho các cấu phần bên trong.

Làm sạch màn hình

Nhẹ nhàng lau sạch màn hình bằng vải mềm không có xơ vải và được làm ẩm bằng nước lau kính không chứa cồn. Đảm bảo rằng màn hình đã khô trước khi đóng máy tính lại.

Hình 1. Một nội dung trong hướng dẫn sử dụng máy tính

Tài liệu hướng dẫn sử dụng thiết bị số thường có các mục sau:

- Hướng dẫn an toàn (Safety): nhằm mục đích ngăn chặn các rủi ro hoặc hư hỏng không thể lường trước khi vận hành sản phẩm không đúng cách.
- Lắp đặt/thiết đặt (Setup): hướng dẫn lắp ráp hoặc thiết đặt thông số ban đầu cho thiết bị.
- Vận hành (Operation): hướng dẫn sử dụng các tính năng chính của thiết bị.
- Bảo trì (Maintenance): hướng dẫn vệ sinh, chăm sóc kỹ thuật,... nhằm đảm bảo sự hoạt động bình thường của thiết bị.
- Xử lý sự cố (Troubleshooting): hướng dẫn chẩn đoán và xử lý sơ bộ các lỗi thường gặp của thiết bị.
- Thông tin về nơi để tìm thêm sự trợ giúp và chi tiết liên hệ (Support).

2) Thông số kỹ thuật của thiết bị số



2

Quan sát các thiết bị trong Hình 2, em thấy chúng có các bộ phận nào giống nhau?



Hình 2. Một số thiết bị số thông dụng

Các thiết bị số rất đa dạng, các bộ phận xử lý dữ liệu số của chúng cũng tương tự như của máy tính. Các thông số kỹ thuật quan trọng là tốc độ bộ vi xử lý CPU, dung lượng RAM, dung lượng lưu trữ. Các thông số kỹ thuật khác nhau tùy vào chức năng của thiết bị (Bảng 1).

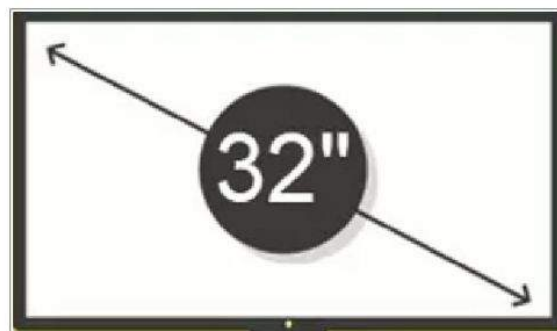
Bảng 1. Một số thông số kỹ thuật quan trọng của các thiết bị số điển hình

	Máy tính	Điện thoại	Ti vi
Tốc độ CPU	2 GHz – 5 GHz	1,8 GHz – 2,8 GHz	1,5 GHz – 1,9 GHz
Dung lượng RAM	1 GB – 64 GB	1 GB – 18 GB	1 GB – 2,5 GB
Dung lượng lưu trữ	Hàng TB	Hàng trăm GB	Hàng GB

Ngoài bộ phận xử lý dữ liệu số tích hợp sẵn (nếu có), các thiết bị số để nhập dữ liệu hay xuất thông tin cho con người sử dụng như: máy in, máy chiếu, màn hình, loa, micro, camera,... có những thông số kỹ thuật quan trọng khác tùy theo chức năng.

Kích thước màn hình

Màn hình hiển thị hình ảnh hình chữ nhật thường được quy định tiêu chuẩn hoá tỉ lệ giữa chiều dài và chiều rộng, ví dụ 4:3, 16:9, 21:9. Do đó, kích thước màn hình thường được thể hiện bằng độ dài đường chéo của nó, đơn vị đo là inch (1 inch tương đương 2.54 cm). Hiện nay, màn hình điện thoại thường có kích thước 4 inch – 6.5 inch, màn hình laptop thường có kích thước 13.3 inch – 17 inch, còn tivi thường có kích thước 40 inch – 65 inch.



Hình 3. Minh họa kích thước màn hình

Độ phân giải hình ảnh

Hình ảnh số hoá được tạo nên từ các điểm ảnh rất nhỏ gọi là pixel (picture element). Độ phân giải điểm ảnh thể hiện bằng cặp hai số đếm điểm ảnh theo chiều ngang và theo chiều cao. Tích hai số này là số điểm ảnh của hình ảnh. Một triệu điểm ảnh là một megapixel. Hình ảnh càng nhiều điểm ảnh thì càng rõ nét. Điện thoại thông minh hiện nay có camera với độ phân giải lên đến vài chục megapixel.



Em hãy tính số đo bằng centimet theo chiều dài và chiều rộng của màn hình máy tính có kích thước 24", 27", 32" tương ứng với tỉ lệ 16:9 và 21:9.



Tìm hiểu cấu hình của một điện thoại thông minh. Em hãy cho biết kích thước màn hình, tốc độ CPU, dung lượng RAM, dung lượng lưu trữ, độ phân giải camera của điện thoại đó.



Câu 1. Trong tài liệu hướng dẫn sử dụng thiết bị số:

a) Mục “Hướng dẫn an toàn” nhằm mục đích gì?

b) Mục “Xử lý sự cố” thường hướng dẫn những gì?

Câu 2. Tại sao kích thước màn hình của các thiết bị điện tử thường được thể hiện bằng số đo độ dài đường chéo?

Câu 3. Nói “Camera có độ phân giải 12 megapixel” nghĩa là gì?

Tóm tắt bài học

- ✓ Tài liệu hướng dẫn sử dụng thiết bị số giúp ta sử dụng an toàn và đúng cách thường có nội dung gồm các mục: hướng dẫn an toàn, lắp đặt/thiết đặt, vận hành, bảo trì, xử lý sự cố, thông tin hỗ trợ khách hàng.
- ✓ Các thông số kĩ thuật quan trọng về xử lý dữ liệu số: tốc độ CPU, dung lượng RAM, dung lượng lưu trữ.
- ✓ Các thông số kĩ thuật quan trọng về hình ảnh kĩ thuật số: độ dài đường chéo màn hình, độ phân giải hình ảnh.

BÀI 3

KHÁI QUÁT VỀ HỆ ĐIỀU HÀNH

Học xong bài này, em sẽ:

- ✓ Trình bày được một cách khái quát mối quan hệ giữa phần cứng, hệ điều hành và phần mềm ứng dụng.
- ✓ Nêu được sơ lược lịch sử phát triển, vai trò và chức năng cơ bản của hai hệ điều hành thông dụng.
- ✓ Trình bày được sơ lược về một số hệ điều hành tiêu biểu.
- ✓ Sử dụng được một số tiện ích có sẵn của hệ điều hành để nâng cao hiệu suất sử dụng máy tính.



Khi mua một máy tính mới, máy tính bảng hay điện thoại thông minh, trước khi bắt đầu sử dụng cần kích hoạt chế độ cài đặt. Tại sao cần làm việc này và những gì sẽ được cài đặt vào máy?

1 Hệ điều hành, vai trò và chức năng của hệ điều hành

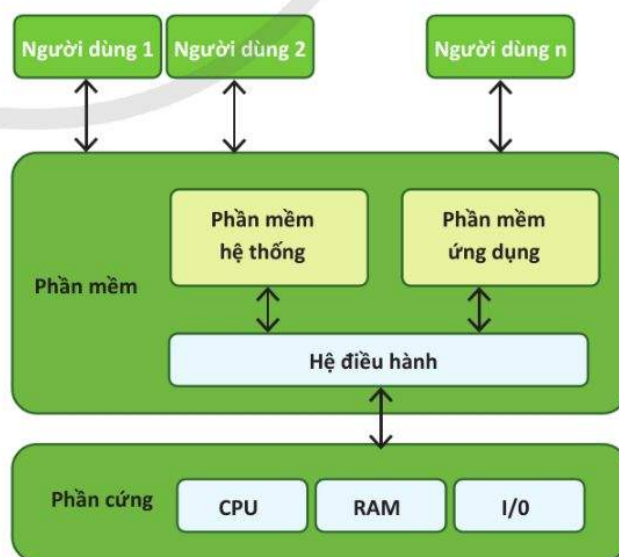


1

Khi bật máy tính, ta phải chờ một lúc rồi mới có thể bắt đầu công việc. Với điện thoại thông minh có khác biệt gì không? Em hãy trả lời và giải thích rõ thêm.

Hệ điều hành (OS – *Operating System*) là tập các chương trình điều khiển và xử lý tạo giao diện trung gian giữa các thiết bị của hệ thống với phần mềm ứng dụng, đồng thời quản lý các thiết bị của hệ thống, phân phối tài nguyên và điều khiển các quá trình xử lý trong hệ thống.

Hình 1 minh họa mối quan hệ giữa phần cứng, hệ điều hành, phần mềm ứng dụng trong máy tính và những người dùng máy tính.



Hình 1. Mối quan hệ giữa hệ điều hành, phần cứng và phần mềm của máy tính

Các phần mềm để soạn thảo văn bản, duyệt web, xử lý hình ảnh, viết chương trình bằng ngôn ngữ Python,... Đây là các phần mềm ứng dụng. Các phần mềm ứng dụng chạy trên nền tảng OS nào cần phù hợp với OS đó. Ví dụ, phần mềm xử lý ảnh trên nền tảng Windows khác với phần mềm cùng chức năng trên nền tảng Android hay iOS.

Các phần mềm thiết kế cho việc vận hành và điều khiển phần cứng máy tính là các phần mềm hệ thống, ví dụ như các trình điều khiển thiết bị.

OS điều phối tất cả các thiết bị, làm trung gian giữa phần mềm ứng dụng và phần cứng. OS cũng là trung gian giữa người dùng máy tính và thiết bị phần cứng, giúp dễ dàng sử dụng thiết bị mà không cần biết sâu về kỹ thuật công nghệ. OS giúp người dùng dễ dàng cài đặt, gỡ bỏ phần mềm ứng dụng theo nhu cầu.

Các chức năng cơ bản của hệ điều hành:

- Quản lý tệp: Tạo và tổ chức lưu trữ các tệp trên bộ nhớ ngoài, cung cấp các công cụ để tìm kiếm và truy cập các tệp, chia sẻ và bảo vệ tệp.

- Quản lý, khai thác các thiết bị của hệ thống: OS tự nhận biết có thiết bị ngoại vi mới được kết nối với máy tính qua các cổng vào – ra như: USB, HDMI, Datamini Port, Bluetooth,... và tự động bổ sung các chương trình điều khiển vào hệ thống. Người dùng có thể sử dụng các thiết bị đó ngay sau khi thiết bị kết nối với hệ thống. OS sẽ tự động ngắt kết nối khi tháo thiết bị khỏi hệ thống.

- Quản lý tiến trình: Các phần mềm ứng dụng xử lý dữ liệu thông qua nhiều tiến trình, mỗi tiến trình làm một việc nhất định. OS tạo ra các tiến trình, điều khiển giao tiếp giữa các tiến trình để phối hợp nhịp nhàng hoàn thành nhiệm vụ. OS huỷ bỏ tiến trình khi nó kết thúc công việc.

- OS cung cấp phương thức giao tiếp để người dùng điều khiển máy tính bằng câu lệnh hoặc qua giao diện đồ hoạ hay dùng tiếng nói.

- Bảo vệ hệ thống: OS có cơ chế nhằm bảo vệ hệ thống và thông tin lưu trữ, hạn chế tối đa ảnh hưởng của các sai lầm do vô tình hay cố ý.

Sơ lược lịch sử phát triển của hệ điều hành qua các thế hệ máy tính

Máy tính thế hệ thứ nhất không có hệ điều hành

Ở giai đoạn này, các chương trình viết bằng ngôn ngữ máy. Việc điều khiển máy tính được thực hiện bằng cách nối dây trên các bảng cắm nối. Phần mềm hỗ trợ người dùng chỉ là thư viện các chương trình mẫu và một số chương trình phục vụ.

Hệ điều hành của các máy tính thế hệ thứ hai

Máy tính thế hệ này bắt đầu có hệ điều hành, tại mỗi thời điểm chỉ cho phép thực

hiện một chương trình của người dùng. Hệ thống phần mềm được bổ sung các chương trình phục vụ như nạp, dịch và thực hiện chương trình ứng dụng, đồng thời hỗ trợ công việc liên quan tới thiết bị ngoại vi.

Hệ điều hành của máy tính thế hệ thứ ba

Hệ điều hành của máy tính thứ ba theo chế độ đa nhiệm, cho phép tại mỗi thời điểm có nhiều chương trình được thực hiện. Ví dụ, trong khi một chương trình đang sử dụng CPU thì chương trình thứ hai có thể sử dụng máy in để in kết quả ra. Máy tính chỉ có một CPU nhưng mỗi chương trình được OS cấp thời gian để CPU xử lý theo cách luân phiên. Đó là cơ chế phân chia thời gian. OS IBM 360/370 là tiêu biểu cho giai đoạn này. Ngoài ra, OS cũng có khả năng quản lý giao tiếp với nhiều người dùng. Vào những năm 70 của thế kỉ XX, OS bắt đầu có thêm khả năng điều hành mạng để khai thác hiệu quả mạng cục bộ và mạng diện rộng.

Hệ điều hành của máy tính thế hệ thứ tư

Ở giai đoạn này, có hai khuynh hướng phát triển máy tính: máy tính cá nhân và siêu máy tính, với mỗi loại máy tính có loại OS tương ứng.

Một số hệ điều hành tiêu biểu

**2**

Ngoài hệ điều hành Windows, em có biết hệ điều hành nào khác không?

a) Hệ điều hành cho máy tính cá nhân

Một số OS thương mại tiêu biểu là: MS DOS, Windows cho dòng máy tính với CPU Intel.

MS DOS là OS đơn chương trình, tổ chức thông tin theo đơn vị quản lý là file, theo cấu trúc thư mục phân cấp dạng cây. Giao tiếp giữa người và máy tính thông qua lệnh. Năm 1981, công ty Microsoft đã đưa ra thị trường MS DOS 2.0. Nhiều năm liên tiếp phát triển thành phiên bản MS DOS 5.0 và được bình chọn là OS tốt nhất cho máy tính cá nhân.

Windows các phiên bản đầu tiên chạy trên nền tảng của MS DOS sử dụng giao diện đồ họa rất đẹp, thân thiện. Cho đến nay, kiểu giao tiếp với các biểu tượng (icon) và cơ chế chỉ định bằng chuột đã trở thành chuẩn.

Từ năm 1995, với sự phổ biến rộng rãi của máy tính cá nhân có cấu hình mạnh, hai loại OS được sử dụng chủ yếu, rộng rãi là Windows cho các máy tính của hãng IBM và MacOS cho các máy tính của hãng Apple (ra đời sớm hơn, từ năm 1985). Cuối thập kỉ XX, có các OS tiêu biểu như Windows 95/98/NT. Được phát hành năm 1995,

Windows 95 là một cột mốc phát triển OS với giao diện đẹp, có nhiều công cụ tiện ích như menu **Start**, thanh trạng thái **Taskbar**, biểu tượng lối tắt **Shortcut**. Windows 2000 Server có nhiều công cụ để quản trị mạng, cung cấp nhiều dịch vụ cho mạng cục bộ kết nối với Internet.

Năm 2001, Windows XP được phát hành với nâng cấp để chạy trên các bộ xử lý tiên tiến 64 bit thế hệ mới. Sau đó là các phiên bản Windows 7 (năm 2009), Windows 8 (năm 2012). Windows 10 (năm 2015) đang được sử dụng phổ biến vì tính hiệu quả, có sẵn các hỗ trợ phòng chống virus, an toàn dữ liệu,... và hoạt động ổn định, đáng tin cậy. Windows 11 (năm 2021) là thế hệ mới nhất sẽ dần dần thay thế các phiên bản Windows trước đó.

Hệ điều hành cho máy tính bảng và điện thoại thông minh: có các công cụ quản lý thông tin cá nhân, xử lý âm thanh và đồ họa được đặc biệt chú ý nhiều hơn cả để đảm bảo chất lượng cao trong vai trò của công cụ giải trí, thư giãn.

b) Hệ điều hành cho máy tính lớn

OS UNIX xuất hiện từ thế hệ máy tính thứ ba, do Ken Thompson xây dựng, được sử dụng chủ đạo cho các máy tính lớn, siêu máy tính. UNIX là OS đa nhiệm, nhiều người dùng dựa trên cơ chế phân chia thời gian, kiểm soát người dùng rất nghiêm ngặt, đảm bảo an toàn cho các chương trình cùng thực hiện đồng thời trên một máy tính. UNIX được viết bằng ngôn ngữ lập trình C, cung cấp các lệnh thao tác với file, thư mục, các phương tiện lập trình, quản trị hệ thống. UNIX sử dụng giao thức mạng TCP/IP phục vụ truyền thông tốt. Nhờ có chế độ vận hành bộ nhớ ảo nên UNIX cho phép máy tính thực hiện các chương trình lớn hơn bộ nhớ của nó.

4) Hệ điều hành nguồn mở

a) Hệ điều hành LINUX

Năm 1991, Linus Benedict Torvalds (*Hình 2*), một sinh viên ngành khoa học máy tính tại Đại học Helsinki (Phần Lan), bắt đầu một dự án mà kết quả sau đó là hạt nhân (phần cốt lõi) của OS LINUX.

LINUX là OS nguồn mở, theo kiểu UNIX, viết trên ngôn ngữ C và được cung cấp miễn phí toàn bộ mã nguồn các chương trình hệ thống. Đầu tiên, LINUX được phát hành với giấy phép riêng, hạn chế sử dụng cho hoạt động thương mại. Năm 1992, Torvalds đề nghị phát hành hạt nhân LINUX theo giấy phép công cộng



Hình 2. Linus Torvalds – tác giả của nhân LINUX

GNU, đặt cơ sở để các nhà phát triển LINUX tạo ra một OS miễn phí có đầy đủ chức năng.

Nhờ đó, mọi người đều có thể sửa đổi, nâng cấp không vi phạm bản quyền. Điều này tạo thuận lợi cho việc có thể bản địa hoá LINUX, tạo giao diện theo tiếng địa phương, ví dụ bằng tiếng Việt.

Sau đây là một số mốc phát triển của OS LINUX:

- Năm 1994: Torvalds đánh giá tất cả các thành phần của hạt nhân đã được hoàn thiện và phiên bản 1.0 của LINUX được phát hành.

- Năm 1996: Phiên bản 2.0 của OS LINUX ra đời, có thể phục vụ nhiều bộ vi xử lý cùng lúc.

Những năm tiếp theo nhiều công ty lớn như IBM, Compaq và Oracle tuyên bố hỗ trợ LINUX.

- Năm 1998: LINUX lần đầu tiên xuất hiện trong danh sách Top 500 siêu máy tính nhanh nhất và đến năm 2017 tất cả Top 500 siêu máy tính đều chạy LINUX.

- Các phiên bản 3.0 (năm 2011), 4.0 (năm 2015) và 5.0 (năm 2019) của nhân LINUX lần lượt được phát hành.

b) Hệ điều hành Android

Android là OS nguồn mở, dựa trên nền tảng của LINUX dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh, máy tính bảng.

Năm 2003, OS Android được bắt đầu phát triển. Cuối năm 2008, hơn một năm sau khi iPhone của Apple xuất hiện, điện thoại thông minh HTC Dream (T-Mobile G1) chạy OS Android 1.0 ra đời, được coi là điện thoại dùng OS Android đầu tiên.

Từ năm 2015, Google đã đưa ra phiên bản OS Android cài đặt cho ô tô và ti vi. Tháng 8 năm 2019, Google quyết định không dùng các icon bánh kẹo nữa (Hình 3) mà chuyển sang đánh số thứ tự. Android 10, được phát hành vào tháng 9 năm 2019. Android 11 ra mắt vào tháng 6 năm 2020. Android 12 được công bố lần đầu tiên vào tháng 2 năm 2021. Android 13 được phát hành cho công chúng vào ngày 15 tháng 8 năm 2022.



Hình 3. Một số icon các phiên bản Android trước năm 2019

5 Thực hành tìm hiểu về hệ điều hành

Tìm hiểu các khả năng của máy tính và sử dụng một số tiện ích có sẵn của hệ điều hành để nâng cao hiệu suất sử dụng máy tính.

Nhiệm vụ 1. Tìm hiểu các khả năng của máy tính hay điện thoại (ưu tiên tìm hiểu OS Android hay iOS).

- a) Khả năng phát âm thanh và video.
- b) Thử nghiệm chụp ảnh ở chế độ chụp ảnh toàn cảnh, ghi ảnh, xem lại và chia sẻ cho người khác.

Nhiệm vụ 2. Một số tổ hợp phím tắt của OS Windows cho phép người dùng thao tác nhanh hơn khi dùng chuột. Hãy khám phá tác dụng của một số phím tắt dưới đây và mô tả các bước thao tác bằng chuột để có kết quả tương tự.

- a) **Ctrl + Win + O**: bật/tắt bàn phím ảo trên màn hình.
- b) **Alt + Tab**: chuyển cửa sổ đang hoạt động.
- c) **Win + D**: chuyển sang màn hình nền.
- d) **Ctrl + Shift**: chuyển chế độ gõ bàn phím.
- e) **Win + H**: bật/tắt micro.
- g) **Win + .** (hoặc **;**): bật/tắt cửa sổ chứa các biểu tượng cảm xúc.



Tìm hiểu xem điện thoại thông minh của em dùng hệ điều hành gì?
Nó có phải là hệ điều hành nguồn mở hay không?



Câu 1. Hệ điều hành có phải là phần mềm duy nhất trong máy tính, máy tính bảng hoặc điện thoại thông minh hay không? Vì sao?

Câu 2. Nêu tên một số hệ điều hành thương mại thường gặp.

Câu 3. Nêu tên một số hệ điều hành nguồn mở thường gặp.

Tóm tắt bài học

- ✓ Hệ điều hành tạo môi trường để người dùng khai thác máy tính và các thiết bị ngoại vi một cách tối ưu và đơn giản.
- ✓ Hệ điều hành cung cấp các dịch vụ để tổ chức và quản lý tệp, thực hiện các chương trình ứng dụng.
- ✓ Hệ điều hành có nhiều loại: thương mại và nguồn mở; dành cho máy tính và dành cho điện thoại thông minh.
- ✓ Hệ điều hành có lịch sử phát triển gắn với các thế hệ máy tính và ngày càng tiện lợi hơn, hỗ trợ tốt hơn cho người dùng.

BÀI 4**THỰC HÀNH VỚI CÁC THIẾT BỊ SỐ**

Học xong bài này, em sẽ:

- ✓ Kết nối được các bộ phận thân máy, bàn phím, chuột, màn hình của máy tính với nhau.
- ✓ Kết nối được PC với các thiết bị số thông dụng như máy in, điện thoại thông minh, máy ảnh số,...
- ✓ Tùy chỉnh được một vài chức năng cơ bản của máy tính và các thiết bị vào – ra thông dụng để phù hợp với nhu cầu sử dụng và đạt hiệu quả tốt hơn.

Nhiệm vụ 1. Lắp ráp các bộ phận của máy tính

Yêu cầu:

Kết nối thân máy, bàn phím, chuột, màn hình.

Hướng dẫn thực hiện:

Bước 1. Chuẩn bị

- Chuẩn bị các bộ phận cần thiết: thân máy tính, màn hình, bàn phím, chuột.
- Xác định các cổng kết nối có trên máy tính và các thiết bị (*Bảng 1*).

Bảng 1. Một số cổng giao tiếp của máy tính và cáp kết nối tương ứng

Cổng (Port)	Hình ảnh của cổng	Cáp kết nối (Connector)	Sử dụng
USB			Kết nối với các loại thiết bị có giao tiếp USB như: chuột, bàn phím, điện thoại,...
VGA			Kết nối với màn hình, máy chiếu,...
HDMI			Kết nối với màn hình, máy chiếu,...
Cổng tròn 3.5 mm			Kết nối với các thiết bị âm thanh: loa, tai nghe, micro.

Bước 2. Xếp đặt vị trí thiết bị

Lựa chọn vị trí đặt các thiết bị sao cho:

- Đảm bảo thiết bị có thể cắm được vào nguồn điện.
- Đặt bàn phím, chuột và màn hình trên bàn nơi phù hợp với vị trí lựa chọn.
- Lựa chọn vị trí đặt thân máy tính để dễ dàng kết nối với các thiết bị ngoại vi.

Bước 3. Kết nối các bộ phận với nhau (Hình 1)

- Kết nối màn hình với thân máy.
- Kết nối chuột, bàn phím với thân máy.
- Kết nối cáp VGA (thường là màu xanh lam) với màn hình, vặn chặt các vít.
- Kết nối dây nguồn của màn hình và thân máy với ổ điện.



Hình 1. Kết nối thân máy, bàn phím, chuột, màn hình của máy tính với nhau

Bước 4. Kiểm tra kết quả

- Bật nguồn màn hình, bật nguồn máy tính.
- Kiểm tra màn hình đã hiển thị bình thường chưa.
- Di chuyển chuột và quan sát màn hình để kiểm tra hoạt động của chuột.
- Gõ phím và quan sát màn hình để kiểm tra hoạt động của bàn phím.

Nhiệm vụ 2. Kết nối máy tính với các thiết bị số thông dụng

Nhiều thiết bị số có thể kết nối với máy tính, trở thành thiết bị ngoại vi trong một phiên làm việc và ngắt kết nối khi xong việc. Điện thoại thông minh, máy in, máy chiếu,... là các ví dụ. Có thể kết nối thiết bị ngoại vi với máy tính qua cáp nối (kết nối có dây) hoặc qua Bluetooth, qua Wi-Fi (kết nối không dây). Có những thiết bị số

có thể kết nối với máy tính bằng cả hai cách (có dây hay không dây) tùy ý người sử dụng.

Kết nối có dây dễ thực hiện vì thường sử dụng dây cáp với hai đầu cắm phù hợp để cắm vào cổng trên thiết bị và cổng trên máy tính. Kết nối Bluetooth bằng phương thức ghép đôi phải thao tác theo hướng dẫn từng bước được hiển thị trên máy tính và trên thiết bị. Nhiệm vụ thực hành này sử dụng hai thiết bị thường dùng là máy in và điện thoại thông minh, thực hiện một kết nối có dây và một kết nối không dây.

a) Kết nối máy tính với máy in

Yêu cầu:

Kết nối máy tính với máy in thông qua cổng USB, sau đó in thử một tài liệu.

Hướng dẫn thực hiện:

Bước 1. Bật nguồn cho máy tính và máy in.

Bước 2. Kết nối máy in với máy tính bằng kết nối USB.

Bước 3. Trên máy tính, tìm cài đặt máy in (Printer settings). Với máy dùng Windows ta chọn **Control Panel**, với máy dùng MacOS ta chọn **System Preferences**.

Bước 4. Tìm tùy chọn **Add a printer** (Hình 2) để cài đặt máy in, sau đó làm theo hướng dẫn. Biểu tượng máy in mới xuất hiện.



Hình 2. Các bước cài đặt máy in mới kết nối

Bước 5. Mở tài liệu và lựa chọn máy in vừa cài đặt để in thử.

b) Kết nối máy tính với điện thoại thông minh

Yêu cầu 1:

Sử dụng cáp USB kết nối máy tính với điện thoại thông minh và sao chép một số dữ liệu từ điện thoại sang máy tính và ngược lại.

Hướng dẫn thực hiện:

Bước 1. Kết nối máy tính với điện thoại thông qua cáp USB.

Sử dụng dây cáp USB, một đầu cắm vào điện thoại, đầu còn lại cắm vào cổng USB trên máy tính (Hình 3).



Hình 3. Kết nối máy tính với điện thoại thông qua cáp USB

Bước 2. Chọn chế độ kết nối.

Thông thường sẽ có các chế độ: sạc pin, truyền tệp, truyền ảnh. Ta chọn chế độ truyền tệp trao đổi dữ liệu giữa máy tính và điện thoại.

Bước 3. Truy cập ổ đĩa bộ nhớ điện thoại và thực hiện việc chuyển/sao chép dữ liệu qua lại giữa máy tính và điện thoại.

Yêu cầu 2:

Kết nối máy tính dùng Windows 10 với điện thoại thông minh dùng Android thông qua Bluetooth và sao chép một số dữ liệu từ điện thoại sang máy tính và ngược lại.

Hướng dẫn thực hiện:

Bước 1. Bật Bluetooth trên điện thoại.

Vào **Settings** (Cài đặt), chọn **Bluetooth** và gạt công tắc sang chế độ **ON**.

Bước 2. Bật Bluetooth trên máy tính.

Chọn **Windows Settings**, sau đó chọn **Devices**. Tại mục Bluetooth, kéo chuột để gạt thanh công tắc sang phải bật kết nối.

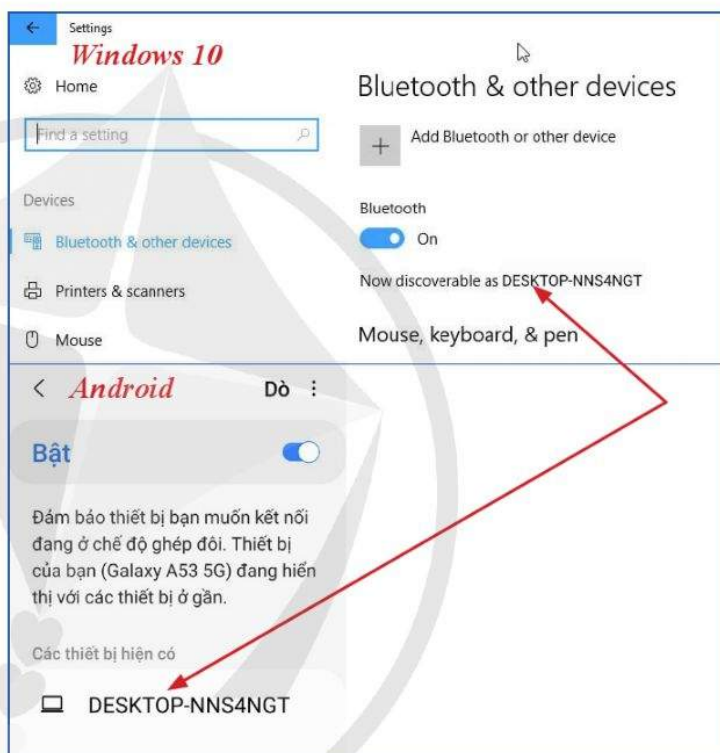
Bước 3. Dò tìm và kết nối máy tính với điện thoại (Hình 4).

– Yêu cầu kết nối: Để thực hiện yêu cầu kết nối từ điện thoại, trên điện thoại ta chọn vào thiết bị muốn kết nối (máy tính). Khi đó trên máy tính nhận được yêu cầu kết nối hiện ở góc thông báo của Windows.

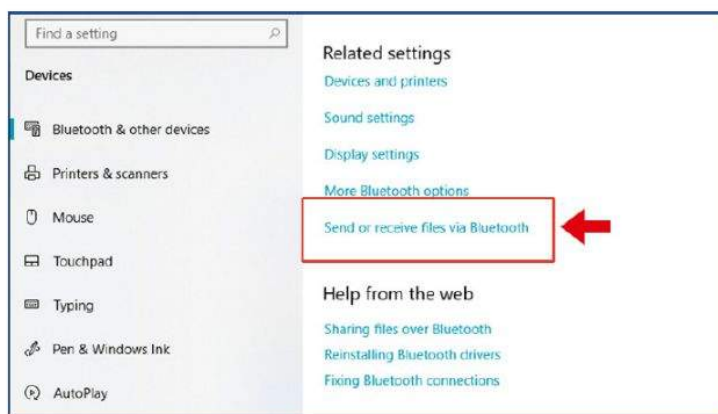
– Chọn **Connect\OK** trên cả hai thiết bị.

Bước 4. Gửi/nhận file (Hình 5).

Tại thanh công cụ, nhấp chuột phải vào biểu tượng Bluetooth. Chọn **Send a file** để gửi file đi hoặc **Receive a file** để nhận file.



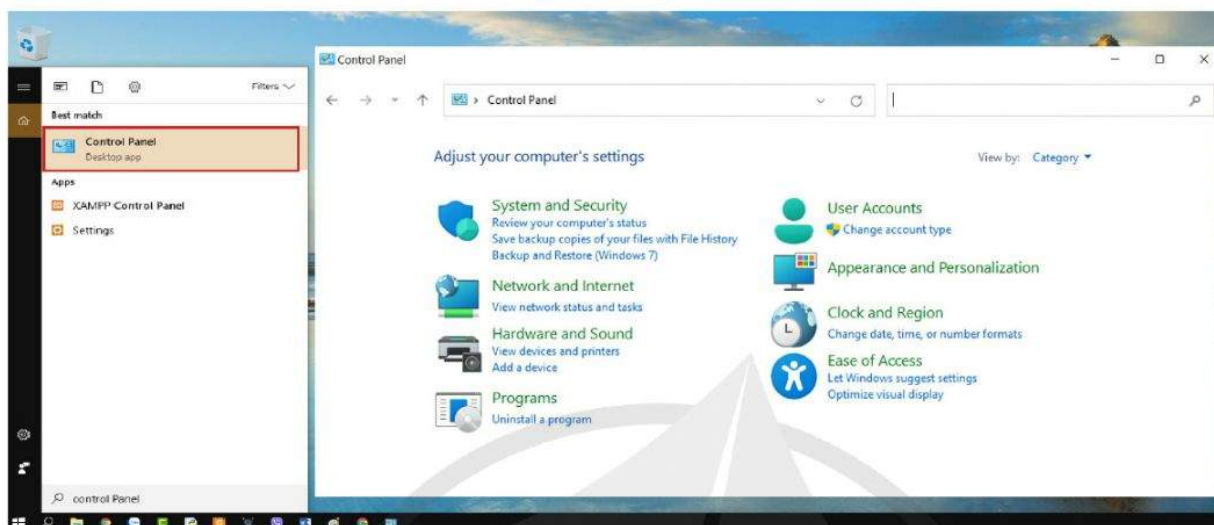
Hình 4. Kết nối máy tính với điện thoại thông qua Bluetooth



Hình 5. Gửi/nhận file

Nhiệm vụ 3. Cá nhân hoá máy tính

Có thể tùy chỉnh máy tính theo nhu cầu sử dụng hoặc sở thích cá nhân. OS thường có sẵn một số công cụ tiện ích hỗ trợ việc này.



Hình 6. Mở bảng điều khiển Control Panel để thực hiện tùy chỉnh cho máy tính

Yêu cầu:

Với sự hướng dẫn của giáo viên, em hãy thực hiện các tùy chỉnh sau:

- 1) Lựa chọn ảnh nền, chỉnh độ sáng, độ phân giải cho màn hình.
- 2) Thay đổi kích thước, màu sắc, lựa chọn nút nhấn chính của chuột.
- 3) Chỉnh sửa thời gian cho đồng hồ về đúng thời gian hiện tại.

Hướng dẫn thực hiện:

- Máy tính sử dụng OS Windows: vào mục **Control Panel** (Hình 6).
- Máy tính sử dụng OS iOS: nhấn vào biểu tượng Apple, sau đó chọn **System**

Preferences.

Dưới đây là một số mục tùy chỉnh (của OS Windows). Hãy nhấn chuột để mở các lựa chọn chi tiết hơn.

- **Appearance and Personalization**: cách hiển thị các mục trên màn hình và cá nhân hoá.
- **Easy of Access**: cho phép thay đổi cách hoạt động của chuột, của bàn phím,...
- **Clock and Region**: thay đổi cách hiển thị ngày, tháng và các số.



Kết nối điện thoại với máy tính bằng phần mềm để gửi tệp tin từ điện thoại tới máy tính và ngược lại.

Gợi ý: Sử dụng AirDroid cho điện thoại Android và AirDrop cho điện thoại iPhone.

BÀI 5

PHẦN MỀM ỨNG DỤNG VÀ DỊCH VỤ PHẦN MỀM

Học xong bài này, em sẽ:

- ✓ Biết vấn đề bản quyền trong sử dụng phần mềm nguồn mở.
- ✓ Biết cách khai thác các mặt mạnh của phần mềm trực tuyến, sử dụng các phần mềm này trong học tập và công việc.
- ✓ Hiểu được vai trò của phần mềm nguồn mở và phần mềm thương mại đối với sự phát triển của ICT.
- ✓ Nêu được tên một số phần mềm soạn thảo văn bản, phần mềm trình chiếu và phần mềm bảng tính nguồn mở trong bộ OpenOffice.



Những phần mềm đang có trong máy tính đã đáp ứng đầy đủ mọi nhu cầu làm việc của em hay chưa?

1 Một số loại phần mềm



Lần gần đây nhất, em đã tải về và cài thêm phần mềm mới nào trên điện thoại thông minh của em? Tại sao em cần thêm phần mềm đó?

Có thể phân loại phần mềm ứng dụng theo nhiều góc độ khác nhau. Dưới đây là một số cách phân biệt thường thấy:

- Phần mềm thương mại và phần mềm miễn phí.
- Phần mềm nguồn mở và phần mềm nguồn đóng.
- Phần mềm khai thác trực tuyến (online) và phần mềm cài trên máy tính cá nhân.

Phần mềm thương mại là phần mềm phải trả tiền mua để sử dụng dù đó là phần mềm ứng dụng hay OS. Phần lớn phần mềm ứng dụng là phần mềm thương mại.

Đối lập với phần mềm thương mại là phần mềm miễn phí. Người dùng không phải trả chi phí và có thể cài đặt trên máy để sử dụng. Các phần mềm phục vụ học tập thường được cung cấp miễn phí và có chất lượng cao (ví dụ: phần mềm nguồn mở Codeblocks, Dev C++ cho hệ thống lập trình C++, Python, Java,...).

Phần mềm nguồn đóng được cung cấp dưới dạng các mô đun chương trình viết trên ngôn ngữ máy. Phần mềm thương mại thường được cung cấp dưới dạng nguồn đóng.

Phần mềm nguồn mở được cung cấp dưới dạng các mô đun chương trình viết trên một ngôn ngữ lập trình bậc cao. Ví dụ, bộ phần mềm OpenOffice là phần mềm nguồn mở, có các khả năng như một bộ phần mềm văn phòng điển hình. Nó gồm Writer (tương tự với Microsoft Word), Calc (tương tự với Microsoft Excel), Impress (tương tự với Microsoft PowerPoint),...

Phần mềm khai thác trực tuyến chỉ có thể sử dụng trên môi trường web, có thể miễn phí hoặc phải trả tiền cho từng phiên sử dụng.

Các phần mềm khai thác trực tuyến đóng vai trò quan trọng trong cuộc sống ngày nay. Các phần mềm tra cứu bản đồ và chỉ dẫn đường đi (maps.google.com), dịch văn bản (translate.google.com), hệ thống văn phòng trực tuyến (docs.google.com), hệ thống phục vụ dạy và học trực tuyến (meet.google.com),... ngày càng trở nên quen thuộc với người dùng. Các phần mềm này được cung cấp dưới hai dạng: miễn phí (bị hạn chế một số tính năng) và có trả phí, nếu không muốn bị hạn chế sử dụng những tính năng phong phú khác. Phần mềm khai thác trực tuyến có trả phí được cung cấp theo cách linh hoạt, như một dịch vụ, tùy theo nhu cầu của người dùng.

Giấy phép phần mềm công cộng viết tắt là GNU GPL hay ngắn gọn hơn là GPL (General Public License). GPL là giấy phép phần mềm phổ biến nhất trong lĩnh vực phần mềm tự do nguồn mở. Nó đảm bảo cho người dùng được tự do khai thác, nghiên cứu, sửa đổi và chia sẻ phần mềm.

Để có thể tự do nghiên cứu và sửa đổi phần mềm thì mã nguồn phải mở. Tính mở và tính tự do gắn liền với nhau. Hiện nay thuật ngữ “phần mềm tự do nguồn mở”, viết tắt tiếng Anh là FOSS (Free Open Source Software) hay được dùng hơn.

Giấy phép phần mềm công cộng đã mở rộng cánh cửa để mọi người tiếp cận những sản phẩm trí tuệ của xã hội và đóng góp phần mình vào vốn kiến thức chung của nhân loại.

2) Thực hành với phần mềm khai thác trực tuyến miễn phí

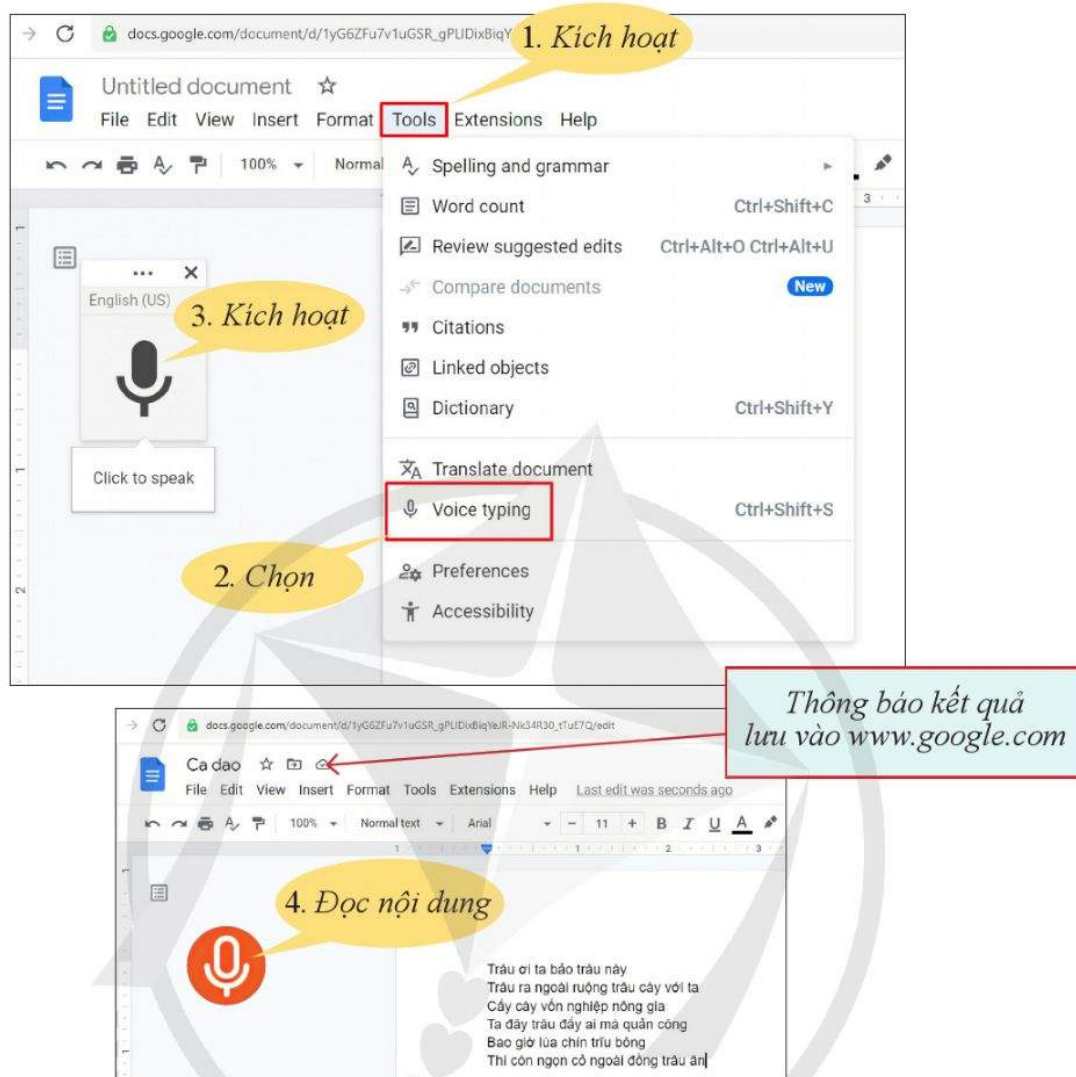
Yêu cầu:

Làm quen với soạn thảo văn bản, làm bài trình chiếu và sử dụng bảng tính qua các phần mềm trực tuyến của Google.

Hướng dẫn thực hiện:

Mở trang web www.google.com, mở bảng chọn các ứng dụng của Google, chọn ứng dụng muốn dùng thử.

a) Thử nghiệm nhập nội dung soạn thảo vào Google Docs bằng giọng nói theo các bước ở Hình 1:



Hình 1. Soạn thảo văn bản với Google Docs

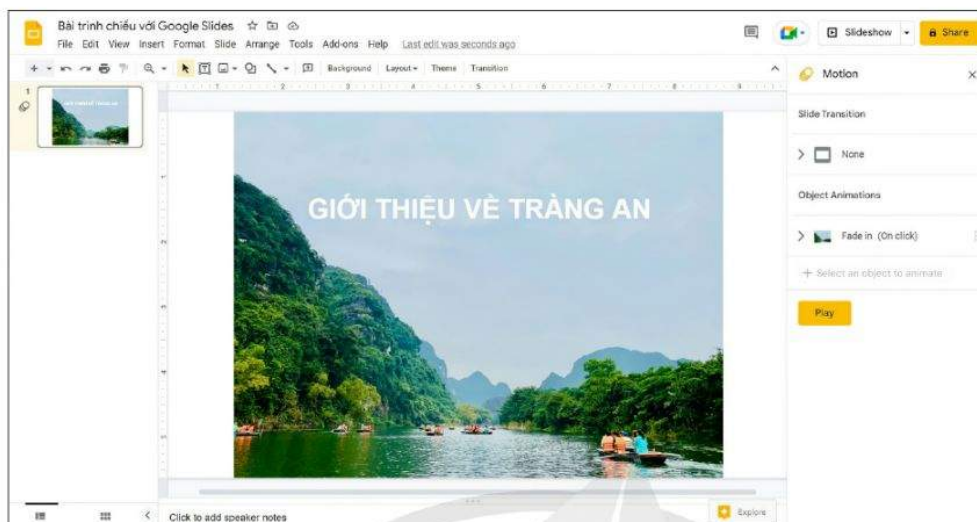
b) Thử mở Google Sheets và tạo một bảng tính ghi thông tin các bạn trong tổ theo mẫu như ở Hình 2.

Bảng tính với Google Sheets

	A	B	C	D	E	F	G	H	I
1	STT	Họ và tên	Ngày sinh	Điện thoại	Email				
2	1	Trần Tiến Đức	25/02/2007	0960617888	ductt@gmail.com				
3	2	Mai Tú Linh	13/10/2007	0912273508	linhmt@gmail.com				
4	3	Vũ Văn Minh	11/12/2007	0865454646	minhvv@gmail.com				
5									
6									

Hình 2. Một mẫu soạn thảo bảng tính với Google Sheets

c) Thử nghiệm mở Google Slides và thiết kế một trang trình chiếu theo mẫu như ở Hình 3.



Hình 3. Ví dụ mẫu soạn thảo slide với Google Slides



Câu 1. Hãy sử dụng công cụ dịch trực tuyến, ví dụ như *translate.google.com.vn* để dịch một đoạn văn bản từ tiếng Việt sang tiếng Anh và ngược lại.

Câu 2. Hãy tìm trên mạng phần mềm chuyển định dạng, chạy trực tuyến miễn phí. Sử dụng phần mềm tìm được để chuyển một số file “.docx” sang tương ứng thành file “.pdf” và ghép nối các file kết quả thành một file “.pdf”.



Trong các câu sau, những câu nào đúng?

- Trong hệ thống luôn cần có các chương trình ứng dụng.
- Phần mềm nguồn mở được cung cấp dưới dạng các mô đun phần mềm viết bằng ngôn ngữ máy.
- Phần mềm là một sản phẩm thương mại nên luôn phải có bản quyền mới sử dụng được.
- Không phải tất cả các phần mềm nguồn mở đều miễn phí.

Tóm tắt bài học

- ✓ Tùy thuộc vào nhu cầu sử dụng, người dùng cài đặt thêm vào hệ thống của mình các phần mềm ứng dụng khác nhau.
- ✓ Có các loại phần mềm: nguồn đóng và nguồn mở, phần mềm khai thác trực tuyến và phần mềm cài trên máy tính cá nhân, phần mềm thương mại và phần mềm miễn phí.
- ✓ Không phải mọi phần mềm nguồn mở đều miễn phí.
- ✓ Việc sử dụng phần mềm nguồn mở sẽ giúp người dùng dễ dàng cải tiến, nâng cấp khi giải quyết vấn đề của mình.

BÀI 1**LƯU TRỮ TRỰC TUYẾN**

Học xong bài này, em sẽ:

- ✓ Biết được ưu, nhược điểm cơ bản của việc lưu trữ trực tuyến.
- ✓ Sử dụng được một số công cụ trực tuyến như: Google Driver, Dropbox,... để lưu trữ và chia sẻ tệp tin.



Em đã từng dùng USB để sao lưu các tệp dữ liệu, chuyển dữ liệu từ máy tính này sang máy tính khác chưa? Em có biết cách nào khác để thực hiện việc đó không?

1 Dịch vụ lưu trữ trực tuyến

Việc lưu trữ dữ liệu số được xử lý bởi máy tính bằng các thiết bị có dung lượng lớn như ổ đĩa cứng (HDD), ổ đĩa cứng thể rắn (SSD) và ổ đĩa USB flash đã rất phổ biến. Cách lưu trữ này gọi là lưu trữ tại chỗ hay lưu trữ vật lý. Ngày nay, việc lưu trữ dữ liệu cần đáp ứng nhu cầu tính toán như trong các dự án dữ liệu lớn (Big Data), trí tuệ nhân tạo (AI), máy học và Internet vạn vật (IoT). Thêm nữa, các yêu cầu đặt ra là bảo vệ chống mất mát dữ liệu do thảm họa, lỗi hoặc gian lận. Đặc biệt, để dữ liệu lưu trữ được chia sẻ cho nhiều người ở các vị trí địa lý khác nhau và đảm bảo dữ liệu lưu trữ được đồng bộ hoá trên nhiều thiết bị, cách lưu trữ trực tuyến đang tỏ ra rất hiệu quả.

Lưu trữ trực tuyến là một cách tiếp cận lưu trữ cho phép người dùng sử dụng Internet lưu trữ, quản lý, sao lưu và chia sẻ dữ liệu. Việc lưu trữ trực tuyến thường liên quan đến hợp đồng với dịch vụ của một công ty gọi là bên thứ ba và chấp nhận dữ liệu được định tuyến thông qua giao thức Internet (IP). Hầu hết các nhà cung cấp dịch vụ lưu trữ trực tuyến đều miễn phí một lượng nhỏ dung lượng lưu trữ và trả phí (theo tháng hoặc năm) với dung lượng lưu trữ lớn hơn. Một số dịch vụ lưu trữ trực tuyến rất phổ biến như: Google Drive (www.drive.google.com) là dịch vụ lưu trữ dữ liệu dựa trên đám mây của Google, OneDrive (www.onedrive.live.com) là dịch vụ đám mây của Microsoft, Fshare (www.fshare.vn) là một dịch vụ lưu trữ trực tuyến lớn của Việt Nam hiện nay.

2 Thực hành lưu trữ và chia sẻ dữ liệu trên Google Drive

Nhiệm vụ 1. Tải và chia sẻ dữ liệu trên Google Drive

Yêu cầu:

Lớp được chia thành các nhóm, mỗi nhóm thực hiện những công việc sau:

a) Tìm hiểu một trong những dịch vụ lưu trữ trực tuyến: Dropbox, OneDrive, Mega, Box, Mediafire. Tóm tắt các nội dung tìm hiểu được bằng một tệp văn bản và một tệp trình chiếu để thuyết trình trong khoảng 5 phút.

b) Tạo một thư mục trên Google Drive có tên là tên của nhóm em (ví dụ: Nhóm 1, Nhóm 2,...). Tải các tệp ở câu a) lên thư mục, chia sẻ cho giáo viên và các bạn trong lớp với các bình luận (nếu có) được ghi kèm vào các tệp dữ liệu.

Hướng dẫn thực hiện:

a) Tìm kiếm và tạo các tệp tóm tắt nội dung tìm kiếm

Bước 1. Sử dụng máy tìm kiếm (chẳng hạn Google) để tìm thông tin về một trong các dịch vụ Dropbox, OneDrive, Mega, Box, Mediafire.

Bước 2. Tạo tệp văn bản và nhập nội dung về dịch vụ lưu trữ trực tuyến đã tìm ở Bước 1 gồm: giới thiệu, cách lưu trữ và chia sẻ dữ liệu, ưu và nhược điểm.

Bước 3. Tạo tệp trình chiếu giới thiệu tóm tắt nội dung có được ở Bước 2.

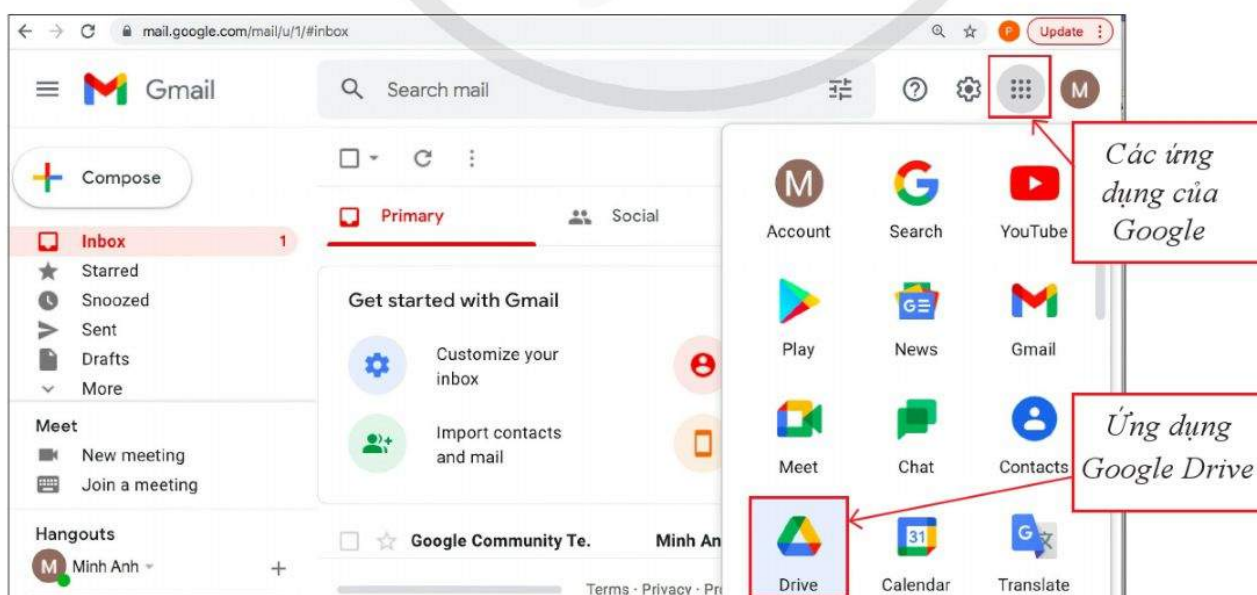
b) Lưu và chia sẻ dữ liệu trên Google Drive

Để sử dụng được dịch vụ Google Drive, người dùng phải có tài khoản Gmail (nếu chưa có thì đăng kí tài khoản Gmail tại www.gmail.com).

Bước 1. Đăng nhập Google Drive.

Cách 1: Truy cập địa chỉ www.drive.google.com, đăng nhập tài khoản Gmail.

Cách 2: Đăng nhập tài khoản Gmail, chọn **Google apps** (các ứng dụng của Google) để mở cửa sổ các ứng dụng của Google và chọn **Drive** (Hình 1).

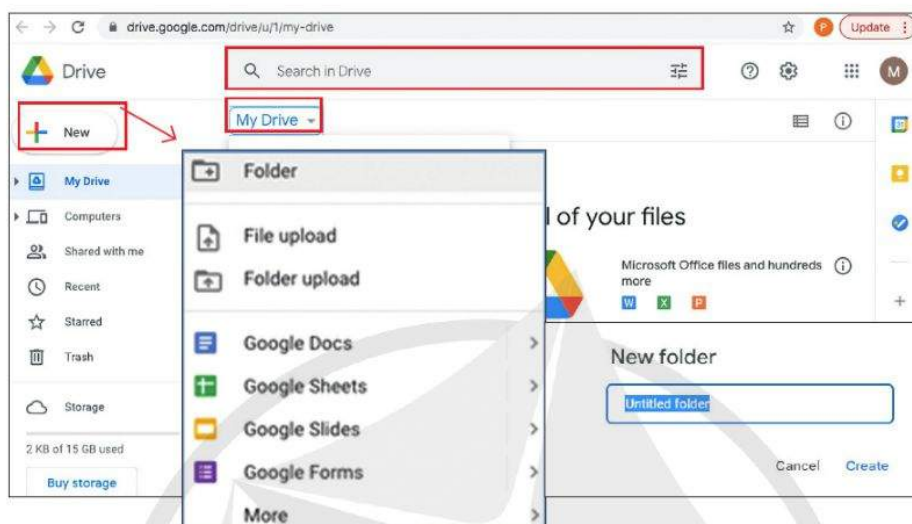


Hình 1. Cửa sổ các ứng dụng của Google

Bước 2. Tải dữ liệu lên Google Drive.

Người dùng có thể tải thư mục hoặc tệp từ máy tính lên Google Drive. Để dễ quản lý, người dùng nên tạo các thư mục để lưu các tệp theo cấu trúc cây như trên Windows.

– Tạo thư mục mới: Chọn mục **New** trên cửa sổ Google Drive, chọn **Folder** (Hình 2), đặt tên thư mục tại ô **New folder** và chọn **Create**.



Hình 2. Cửa sổ giao diện Google Drive

– Tải dữ liệu: Mở thư mục muốn tải dữ liệu lên, chọn **New** (hoặc nhấp chuột phải), chọn **File upload** (tải tệp) hoặc **Folder upload** (tải thư mục), chọn tệp hoặc thư mục trong máy tính muốn tải, chọn **Open**.

Bước 3. Chia sẻ dữ liệu.

Google Drive cho phép chia sẻ các tệp hoặc thư mục cho nhiều người dùng qua email hoặc đường liên kết.

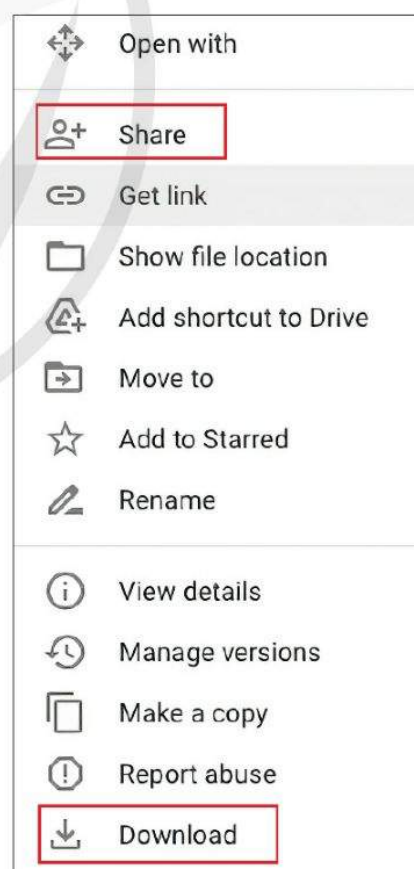
– Chọn tệp hoặc thư mục muốn chia sẻ, nhấp chuột phải, xuất hiện cửa sổ như ở Hình 3.

– Chọn **Share**, xuất hiện cửa sổ chia sẻ dữ liệu như ở Hình 5.

– Nhập địa chỉ email của người hoặc nhóm người nhận dữ liệu được chia sẻ, chọn **Send**.

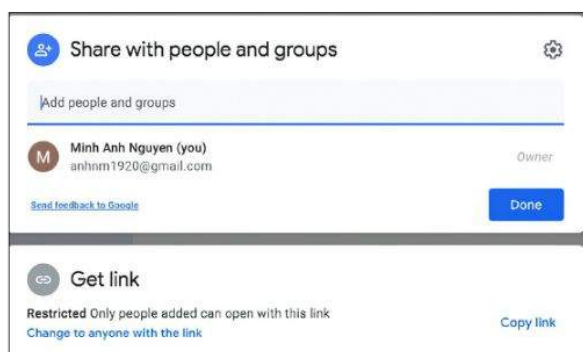
– Cũng có thể gửi đường liên kết (link) của tệp hoặc thư mục dữ liệu muốn chia sẻ: chọn **Copy link** (Hình 4) và gửi link này cho người nhận.

– Chọn quyền chia sẻ (Hình 5): **Viewer** (chỉ xem dữ liệu), **Commenter** (được bình luận), **Editor** (được sửa dữ liệu).

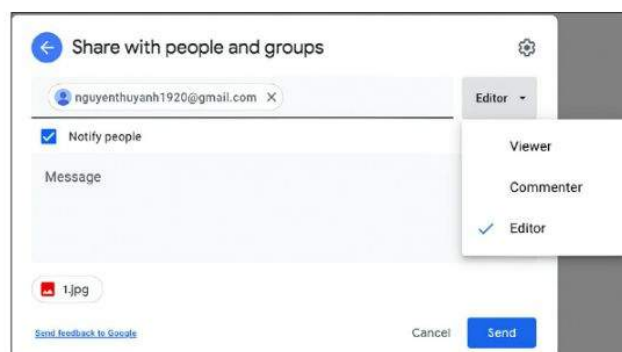


Hình 3. Các chức năng quản lý tệp trên Google Drive

Người được chia sẻ tệp hoặc thư mục có thể mở trực tiếp tệp hoặc thư mục trên Google Drive để xem và sửa dữ liệu (nếu được cấp quyền) mà không cần phải tải về.



Hình 4. Cửa sổ sao chép liên kết



Hình 5. Cửa sổ quyền chia sẻ dữ liệu

Bước 4. Tải xuống các tệp hoặc thư mục từ Google Drive.

Chọn tệp hoặc thư mục cần tải xuống, nhấp chuột phải, chọn **Download** (Hình 3).

Nhiệm vụ 2. Tìm kiếm, truy cập và cập nhật dữ liệu

Yêu cầu:

Các nhóm hãy chỉnh sửa nội dung tệp theo bình luận của giáo viên (được lưu ở thư mục chung có tên *Bail*) và di chuyển tệp vào thư mục của nhóm mình.

Hướng dẫn thực hiện:

Bước 1. Tìm kiếm và truy cập dữ liệu trên Google Drive.

– Truy cập Google Drive, chọn **Search in Drive** (Hình 2), xuất hiện danh sách các loại tệp ngay bên dưới bao gồm: Documents, SpreadSheets, Presentations, Forms,...

– Chọn loại tệp Documents, xuất hiện danh sách các tệp tài liệu (nếu muốn tìm chi tiết thì chọn **More search tools**), chọn và mở tệp cần truy cập.

Bước 2. Cập nhật dữ liệu.

Cách 1: Mở tệp trực tiếp trên Google Drive và sửa nội dung tệp.

Cách 2: Cập nhật và lưu cả phiên bản cũ và mới của tệp dữ liệu trên Google Drive.

– Cập nhật nội dung tệp và lưu trên máy tính.

– Mở Google Drive, chọn tệp cần cập nhật, nhấp chuột phải và chọn **Manage versions**.

– Chọn **Upload new version**, chọn tệp cập nhật từ máy tính, chọn **Close** sau khi tệp mới được tải xong.

Bước 3. Di chuyển tệp.

Chọn tệp cần di chuyển, nhấp chuột phải, chọn **Move to** (Hình 3), chọn thư mục cần di chuyển đến, chọn **Move here**.

Xoá tệp hoặc thư mục: Chọn tệp hoặc thư mục cần xoá, nhấp chuột phải, chọn **Remove**.

Lợi ích và lưu ý khi sử dụng lưu trữ trực tuyến

Lưu trữ trực tuyến mang lại rất nhiều lợi ích:

- Truy cập được dữ liệu từ mọi nơi và mọi lúc, truyền dữ liệu và đồng bộ hoá dữ liệu giữa các thiết bị. Điều này làm tăng tính sẵn có của dữ liệu.

- Chia sẻ tập tin với nhiều người dùng ở các vị trí địa lý khác nhau.

- Sao lưu và khôi phục tệp dữ liệu sau thảm hoạ vì nó nằm ngoài địa điểm của các thiết bị vật lý.

- Tránh được các sự cố như mất điện, thảm hoạ,... và khả năng sao lưu tự động để đảm bảo rằng dữ liệu không bị mất. Cần phải có tài khoản mới truy cập được dữ liệu.

Tuy nhiên, lưu trữ trực tuyến có một số mặt trái tiềm ẩn như:

- Các dịch vụ lưu trữ đám mây có thể có những lỗ hổng bảo mật.

- Một số nhà cung cấp đôi khi gặp tình trạng ngừng hoạt động, dẫn đến lo ngại về độ tin cậy.

- Sử dụng bộ nhớ tại chỗ nhanh hơn so với sử dụng bộ nhớ Internet, vì thời gian đợi tệp tải lên hoặc tải xuống phụ thuộc vào tốc độ đường truyền.

Nhiều cá nhân và tổ chức sử dụng kết hợp lưu trữ tại chỗ và trực tuyến. Chẳng hạn, sử dụng bộ nhớ cục bộ cho các tệp được sử dụng thường xuyên và bộ nhớ trực tuyến để sao lưu hoặc lưu trữ dữ liệu. Hoặc có thể sử dụng bộ nhớ cục bộ cho dữ liệu cá nhân và bộ nhớ trực tuyến cho các tệp muốn chia sẻ với người khác.



Em hãy mở tệp của một nhóm đã được chia sẻ trên Google Drive ở bài thực hành, bình luận hoặc sửa nội dung tệp và chia sẻ lại tệp này.



Theo em, những trường hợp nào sau đây nên sử dụng lưu trữ trực tuyến?

a) Chia sẻ tệp dữ liệu cho nhiều người ở nhiều nơi và sao lưu dữ liệu tự động.

b) Giải phóng bộ nhớ cho máy tính.

c) Thiết bị lưu trữ dữ liệu không có kết nối Internet.

d) Truy cập tệp dữ liệu từ nhiều thiết bị.

Tóm tắt bài học

- ✓ Lợi ích chính của lưu trữ trực tuyến là khả năng bảo mật dữ liệu, sao lưu tự động, tăng tính an toàn cho dữ liệu, truyền và chia sẻ dữ liệu cho nhiều người dùng một cách thuận lợi.
- ✓ Phần lớn các dịch vụ lưu trữ trực tuyến đều cung cấp tính năng đồng bộ hoá dữ liệu trên các thiết bị, bảo vệ dữ liệu bằng mật khẩu, tìm kiếm, truy cập tệp dữ liệu và chỉnh sửa tệp dữ liệu trực tuyến.

Học xong bài này, em sẽ:

- ✓ Xác định được các lựa chọn theo tiêu chí tìm kiếm để nâng cao hiệu quả tìm kiếm thông tin.
- ✓ Thực hành tìm kiếm thông tin bằng cách nhập từ khoá hoặc giọng nói với Google.

1 Sử dụng biểu thức tìm kiếm

Khi tìm kiếm thông tin trên Internet bằng máy tìm kiếm, có thể sử dụng một số toán tử và các kí hiệu đặc biệt, kết hợp với các từ khoá tìm kiếm. Điều này nhằm tăng hiệu quả tìm kiếm như: tìm kiếm nhanh hơn, trả về kết quả phù hợp với mong đợi hơn.

Nhiệm vụ 1. Kết hợp các từ khoá tìm kiếm thành biểu thức tìm kiếm

Yêu cầu:

Em hãy sử dụng máy tìm kiếm Google để thực hiện tìm kiếm với các biểu thức sau và so sánh kết quả nhận được về: thời gian tìm kiếm, số lượng trang web trả về, nội dung một số trang web kết quả.

- a) Cá heo xanh.
- b) “Cá heo xanh” + “cửa hàng”.
- c) Cửa hàng cá heo xanh.

Hướng dẫn thực hiện:

Truy cập trang web www.google.com, tại ô tìm kiếm nhập lần lượt các biểu thức tìm kiếm ở trên, quan sát và nhận xét các kết quả nhận được.

Google hỗ trợ các kí hiệu đặc biệt và toán tử nhằm tăng hiệu quả tìm kiếm, một số kí hiệu đó như sau (kí hiệu A, B là các từ khoá tìm kiếm):

- “A”: Tìm trang chứa chính xác từ khoá A.
- A-B: Tìm trang chứa từ khoá A nhưng không chứa từ khoá B.
- A+B: Tìm trang kết quả chứa cả từ khoá A và B nhưng không cần theo thứ tự.
- A*: Tìm trang chứa từ khoá A và một số từ khác mà Google xem là có liên quan. Ví dụ: Từ khoá “*tin học**” tìm các trang có chứa từ “*tin học ứng dụng*”, “*tin học văn phòng*”,...
- A AND B: Tìm trang chứa cả từ khoá A và B.
- A OR B (hoặc A | B): Tìm trang chứa từ khoá A hoặc B. Toán tử này hữu ích khi tìm từ đồng nghĩa hoặc một từ có nhiều cách viết.
- A + filetype (loại tệp): Tìm thông tin chính xác theo loại tệp như “.txt”, “.doc”, “.pdf”,... Sử dụng từ khoá này thuận lợi trong tìm kiếm tài liệu, sách điện tử.

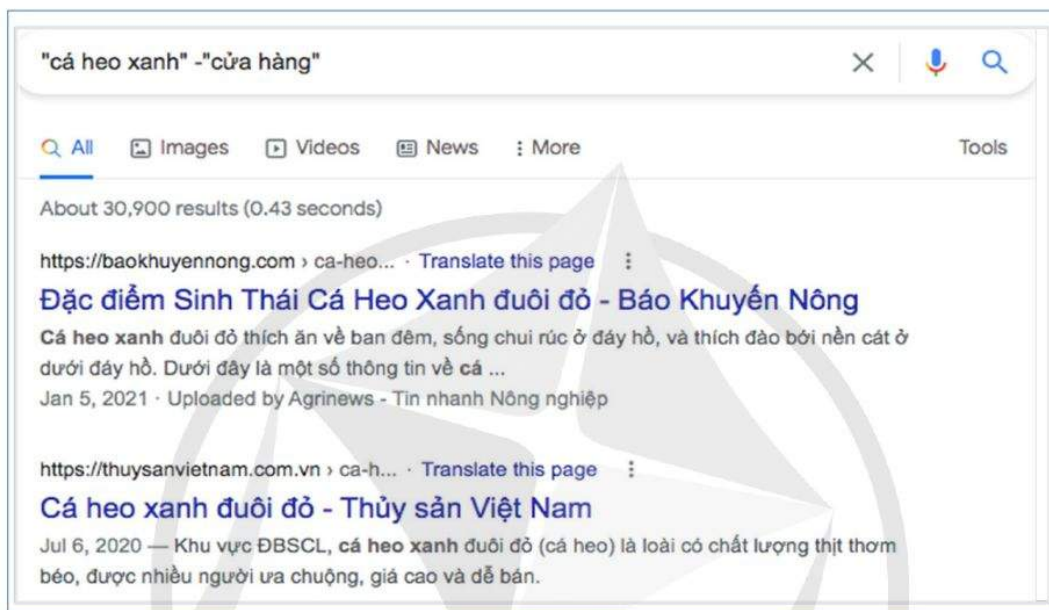
Nhiệm vụ 2. Điều chỉnh biểu thức tìm kiếm

Yêu cầu:

Dựa trên kết quả Bài 1, em hãy điều chỉnh biểu thức tìm kiếm để nhận được kết quả phù hợp với mong đợi hơn. Ví dụ “Đặc điểm sinh thái của cá heo xanh”.

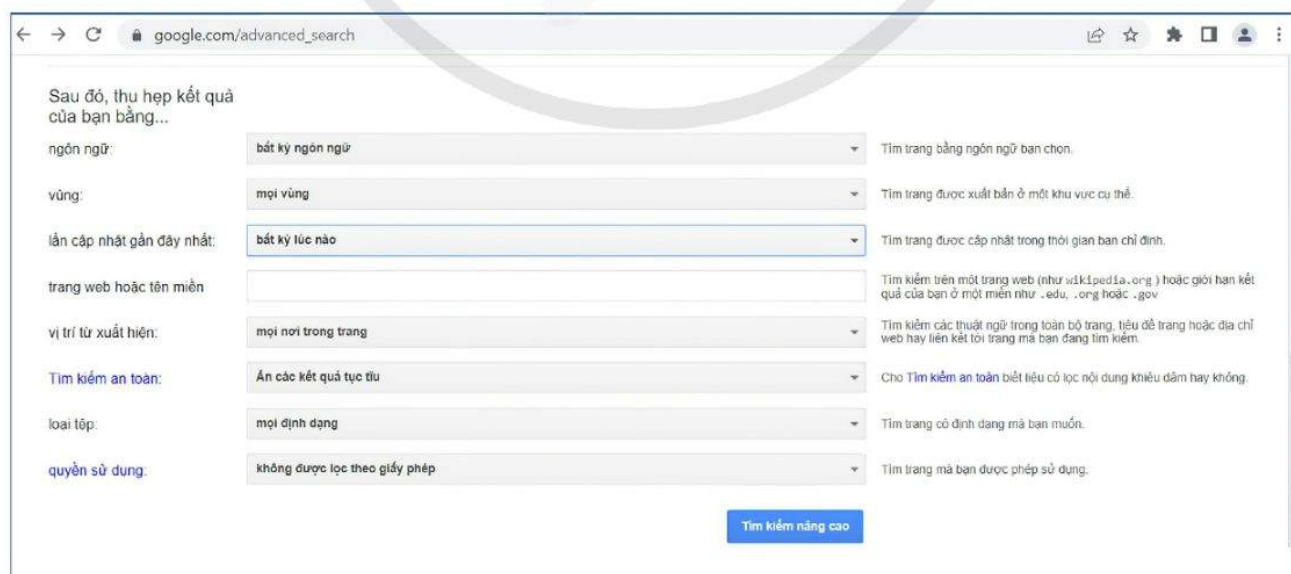
Hướng dẫn thực hiện:

Dùng toán tử trừ (-) để loại các trang web về các cửa hàng có tên cá heo xanh: Nhập vào ô tìm kiếm từ khoá “cá heo xanh” - “cửa hàng” (Hình 1).



Hình 1. Kết quả tìm kiếm theo từ khoá “cá heo xanh” - “cửa hàng”

Một cách khác để thu hẹp kết quả tìm kiếm là sử dụng bộ lọc trên một hoặc nhiều thuộc tính dữ liệu như ở Hình 2 bằng cách truy cập trang tìm kiếm nâng cao www.google.com/advanced_search.



Hình 2. Cửa sổ giao diện tìm kiếm nâng cao của Google

2 Tìm kiếm thông tin bằng giọng nói

Nhiệm vụ 3. Tìm kiếm thông tin bằng giọng nói trên Google

Yêu cầu:

Em hãy tìm những trường học ở quận/huyện nơi em ở bằng giọng nói trên máy tìm kiếm Google.

Hướng dẫn thực hiện:

Chẳng hạn, nếu muốn tìm trường học ở quận Cầu Giấy thì em thực hiện như sau:

Bước 1. Truy cập trang web *www.google.com* và chọn ngôn ngữ tiếng Việt (Hình 3).



Hình 3. Cửa sổ giao diện tìm kiếm của Google

Bước 2. Chọn biểu tượng  (Tìm kiếm bằng giọng nói), xuất hiện cửa sổ như ở Hình 4, bật micro của máy tính và nói “trường học ở quận Cầu Giấy”. Kết quả tìm kiếm là các trang web của các trường học ở Cầu Giấy.



Hình 4. Cửa sổ giao diện tìm kiếm bằng giọng nói

Tìm kiếm bằng giọng nói rất thuận lợi khi sử dụng tìm kiếm trên các thiết bị di động, thiết bị điều khiển trên ô tô.

Ngoài ra, Google còn cung cấp tính năng tìm kiếm bằng hình ảnh. Kết quả tìm kiếm là các trang web có hình ảnh cần tìm kiếm, thông tin về vật thể trong hình và các hình ảnh tương tự khác.



Em hãy sử dụng một máy tìm kiếm để tìm thông tin về lĩnh vực ngành nghề mà mình quan tâm. Trong đó, có sử dụng tìm kiếm theo từ khóa được nhập vào ô tìm kiếm, tìm một vài địa điểm của đơn vị hoạt động về lĩnh vực ngành nghề này bằng giọng nói, tìm kiếm dựa vào hình ảnh nhân vật hoặc sự kiện nổi bật trong lĩnh vực đó.

BÀI 3

THỰC HÀNH MỘT SỐ TÍNH NĂNG NÂNG CAO CỦA MẠNG XÃ HỘI

Học xong bài này, em sẽ:

- ✓ Sử dụng được một số chức năng nâng cao của dịch vụ mạng xã hội.
- ✓ Thực hành các chức năng nâng cao trên mạng xã hội Facebook.

1 Cài đặt bảo mật và quyền riêng tư trên mạng xã hội

Các mạng xã hội cung cấp các cơ chế bảo vệ tài khoản bằng cách sử dụng mật khẩu mạnh và thay đổi mật khẩu thường xuyên hoặc thiết lập bảo mật hai lớp. Bảo mật hai lớp là thêm một bước xác thực, ví dụ: Người dùng phải nhập một dãy số được gửi đến số điện thoại đã đăng ký kèm theo tài khoản.

Nhiệm vụ 1. Bảo mật hai lớp tài khoản Facebook

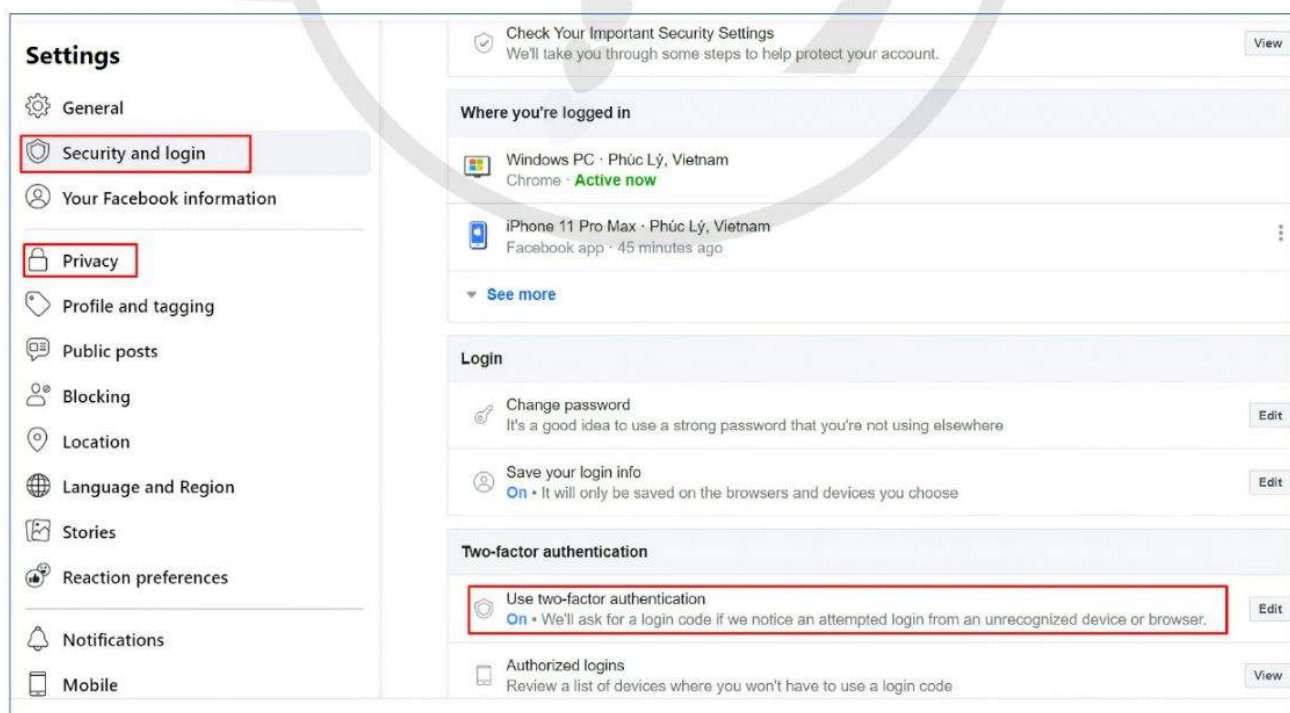
Yêu cầu:

Em hãy kích hoạt bảo mật hai lớp cho tài khoản Facebook của mình.

Hướng dẫn thực hiện:

Bước 1. Đăng nhập tài khoản Facebook, chọn **Your Profile**, chọn **Settings & privacy**, chọn **Settings** tại cửa sổ Settings & privacy.

Bước 2. Chọn **Security and login**, chọn **Use two-factor authentication** trong mục Two-factor authentication và chọn **Edit** (Hình 1).



Hình 1. Cửa sổ Security and login

Bước 3. Lựa chọn các phương án:

(1) **Use Authentication App**: Sử dụng một ứng dụng như Google Authenticator hoặc Duo Mobile để tạo mã xác minh nhằm bảo vệ tốt hơn.

(2) **Use Text Message (SMS)**: Sử dụng tin nhắn văn bản để nhận mã xác minh.

(3) **Use security key**: Sử dụng khoá bảo mật đăng nhập qua USB hoặc NFC.

Chẳng hạn, lựa chọn (2), nhập số điện thoại và chọn **Continue**, nhập mã xác nhận 6 số được gửi vào điện thoại, chọn **Done**.

Nhiệm vụ 2. Cài đặt quyền riêng tư với các thông tin chia sẻ trên Facebook

Yêu cầu:

Em đăng một bức ảnh chụp với bạn trên Facebook và chỉ chia sẻ bức ảnh với người bạn này.

Hướng dẫn thực hiện:

Bước 1. Đăng nhập tài khoản Facebook trên máy tính.

Bước 2. Tạo bài đăng, tại cửa sổ **Create Post** chọn **Photo/Video**, chọn **Add photos/videos**, chọn ảnh cần đăng.

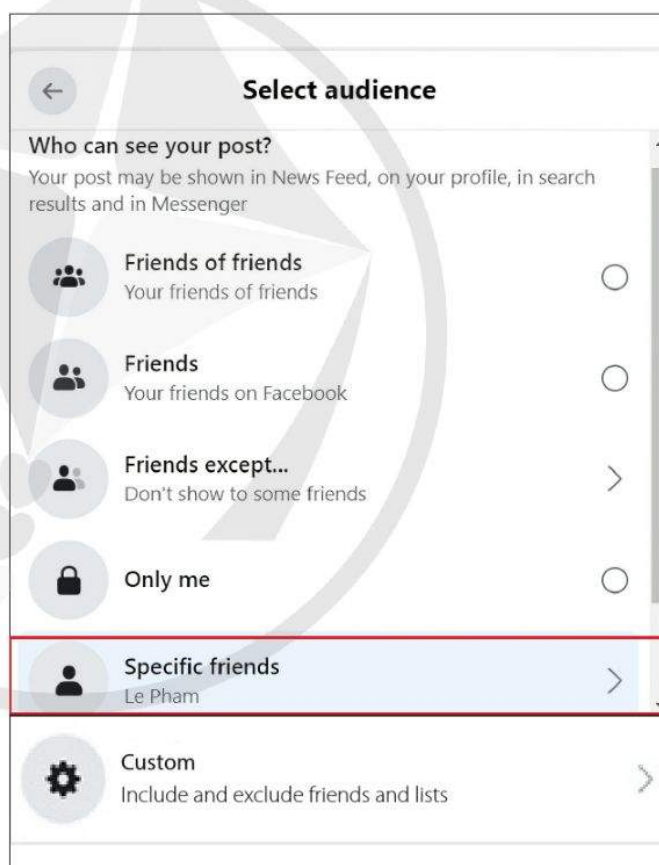
Bước 3. Chọn **Friends**, chọn **Specific friends** (Hình 2), nhập tài khoản Facebook của người mà em muốn chia sẻ ảnh, chọn **Save changes**, chọn **Post**.

Trong bảng chọn Hình 2, có thể lựa chọn đối tượng khác để chia sẻ bài đăng.

Facebook cũng cung cấp tính năng thiết lập quyền chia sẻ các hoạt động của người dùng (cho các bài đăng trong tương lai, những bài đăng mà người dùng được gắn thẻ,...) trừ khi người dùng muốn thay đổi nó, thực hiện như sau:

– Tại cửa sổ **Settings** (cột bên trái ở Hình 1) chọn **Privacy**.

– Chọn *Who can see your future posts?* trong mục **Your activity** tại cửa sổ **Privacy Settings and Tools**. Chọn các đối tượng để chia sẻ bài đăng như ở Hình 2.



Hình 2. Cửa sổ chọn người chia sẻ

2 Trao đổi và chia sẻ thông tin

Nhiệm vụ 3. Tạo phòng họp nhóm để trao đổi thông tin trên Facebook

Yêu cầu:

Em hãy tạo phòng họp trên Facebook để trao đổi về làm bài tập nhóm.

Hướng dẫn thực hiện:

Bước 1. Đăng nhập tài khoản Facebook, chọn  hoặc chọn  (**Menu**), rồi chọn **Room** trong cửa sổ **Create** (Hình 3).

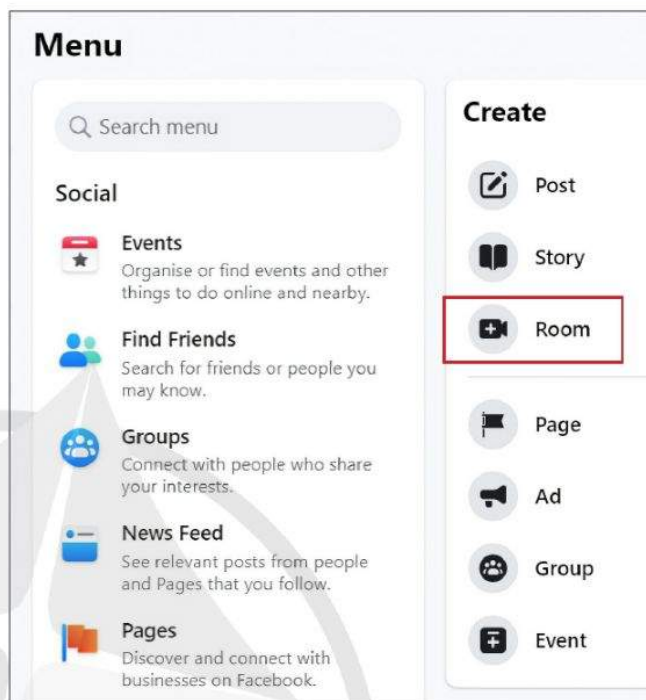
Bước 2. Đặt tên phòng họp và thời gian cuộc họp:

– Tại cửa sổ **Create your room** (Hình 4): chọn **Room name**, chọn **New** và nhập tên phòng họp, chọn thời điểm bắt đầu cuộc họp tại **Start time**, sau đó chọn **Save**.

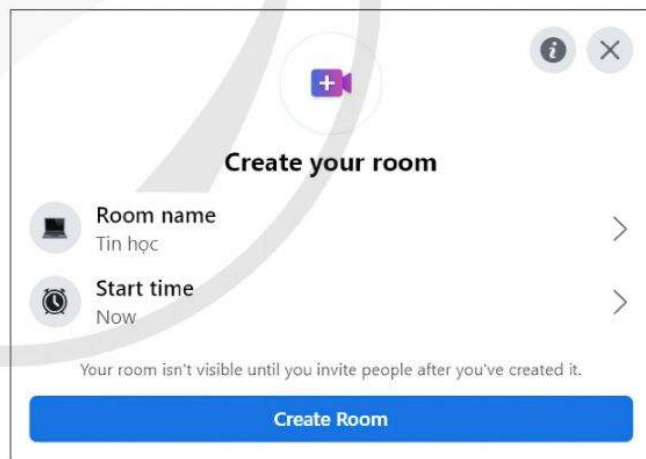
– Chọn **Create Room**.

Bước 3. Gửi lời mời tham gia phòng họp.

Chọn nút **Send** bên cạnh tài khoản muốn mời, chọn **Join Room** để tham gia và bắt đầu cuộc họp.



Hình 3. Cửa sổ Create



Hình 4. Đặt tên và thời gian cho phòng họp



Em hãy tạo phòng họp nhóm và hẹn mời các bạn cùng nhóm học tập thảo luận để lựa chọn thiết lập quyền riêng tư khi chia sẻ thông tin trên Facebook sao cho thuận tiện. Đồng thời, em tạo sự kiện thông báo với các bạn trong nhóm về buổi thảo luận này.

Gợi ý: Để tạo sự kiện, chọn **Event** trong cửa sổ Create.

BÀI 4

THỰC HÀNH MỘT SỐ TÍNH NĂNG HỮU ÍCH CỦA DỊCH VỤ THƯ ĐIỆN TỬ

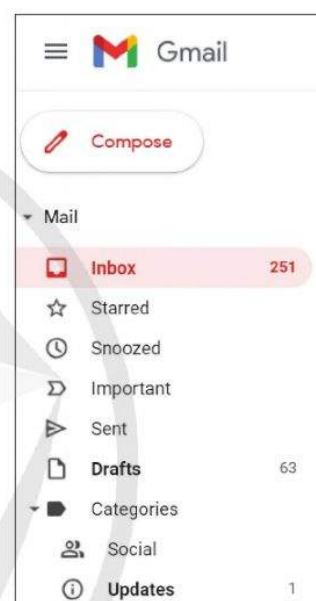
Học xong bài này, em sẽ:

- ✓ Biết cách phân loại và đánh dấu thư điện tử.
- ✓ Thực hành phân loại và đánh dấu thư điện tử Gmail.

1 Phân loại thư điện tử

Ở lớp 6, em đã biết sử dụng các chức năng cơ bản của dịch vụ thư điện tử (như: tạo tài khoản, đăng nhập tài khoản, soạn thư, gửi thư và đăng xuất khỏi hộp thư) để trao đổi thông tin với thầy cô, bạn bè, người thân. Sau một thời gian sử dụng, hộp thư đến sẽ chứa rất nhiều email và khi cần tìm kiếm có thể phải thực hiện mất nhiều thời gian. Để thuận tiện, các dịch vụ thư điện tử đều hỗ trợ tính năng quản lý bằng cách phân loại và gắn nhãn các thư đến.

Việc phân loại cho phép người dùng dễ dàng gắn thẻ, gắn nhãn vào nhóm các thư cũng như sự kiện lịch trong các dịch vụ thư điện tử. Gmail có sẵn các nhóm được gắn nhãn như: **Inbox** (các thư được gửi đến), **Sent** (các thư đã được gửi đi), **Drafts** (các thư nháp),... (Hình 1).



Hình 1. Các nhãn và phân loại mặc định trong Gmail

Nhiệm vụ 1. Phân loại và đánh dấu email trong Gmail

Yêu cầu:

Em hãy tạo nhãn “Học tập” cho các email về học tập, gồm các email được gửi từ giáo viên và các bạn trong lớp. Đánh dấu ★ cho những email được gửi đến từ giáo viên.

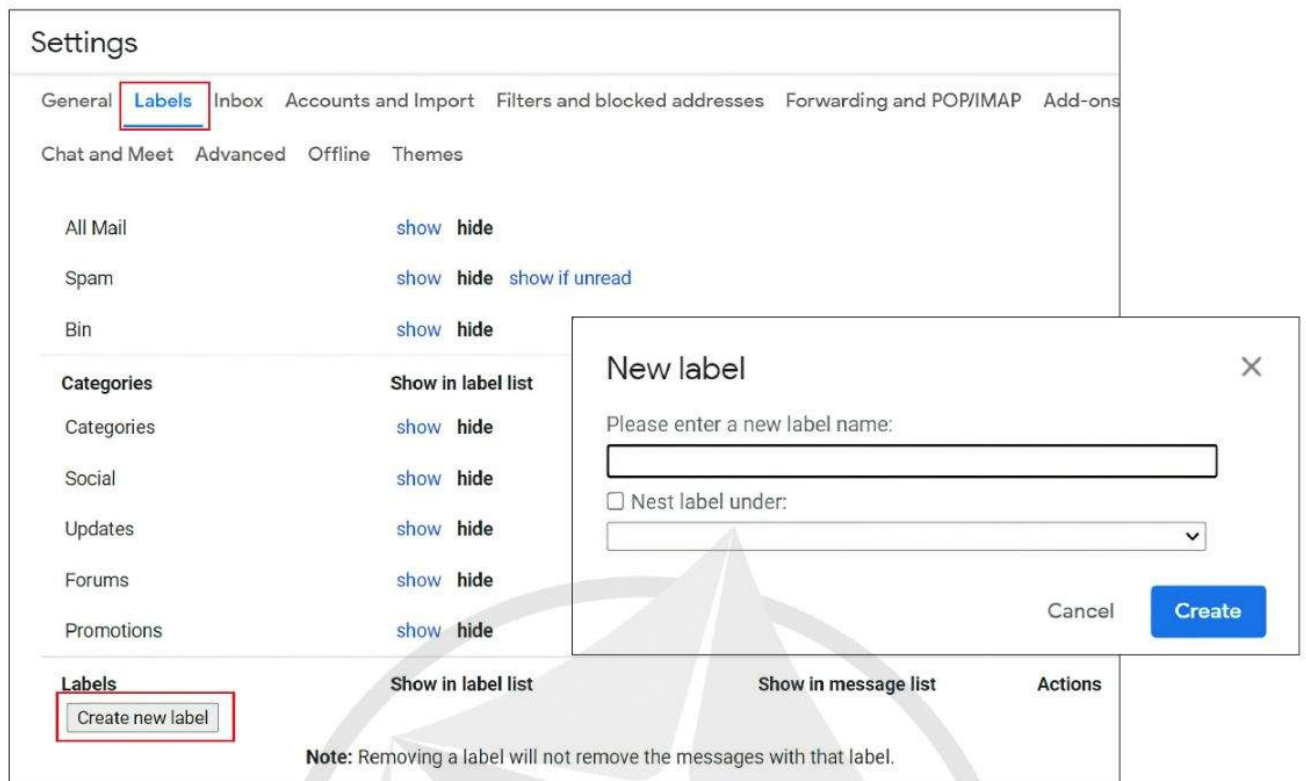
Hướng dẫn thực hiện:

Bước 1. Đăng nhập tài khoản Gmail của em.

Bước 2. Chọn ⚙️ (**Settings**) ở góc phải trên màn hình, xuất hiện cửa sổ Settings như ở Hình 2, chọn mục **Labels**, chọn **Create new label**.

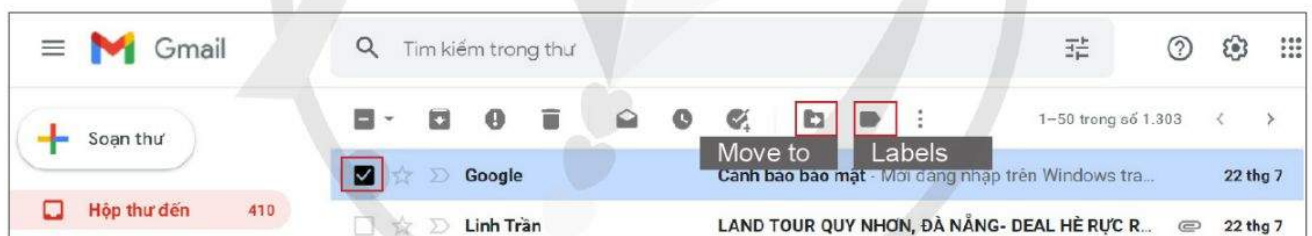
Bước 3. Tại cửa sổ New label nhập tên nhãn “Học tập” và chọn **Create**.

Lưu ý: Có thể tạo nhãn con của một nhãn khác giống như tạo thư mục con, bằng cách chọn **Nest label under** và chọn tên nhãn cha (Cửa sổ New label ở Hình 2).



Hình 2. Cửa sổ tạo nhãn mới trong Gmail

Nếu muốn gán thêm nhãn khác cho email: chọn email cần gán thêm nhãn, chọn **Labels** (Hình 3), chọn nhãn muốn gán thêm.



Hình 3. Cửa sổ chọn nhãn/di chuyển sang nhãn khác cho email được chọn

Nếu muốn di chuyển email sang nhãn khác: chọn email muốn di chuyển, chọn **Move to** (Hình 3), chọn nhãn muốn email di chuyển đến.

Bước 4. Chọn dấu ★ ở cạnh email cần đánh dấu (email do giáo viên gửi).

Lưu ý: Em nên đánh dấu những email quan trọng để dễ tìm kiếm.

2 Quản lý email bằng bộ lọc và tìm kiếm email

Nhiệm vụ 2. Tìm kiếm và tạo bộ lọc email trong Gmail

Yêu cầu:

Em hãy tìm kiếm các email do giáo viên dạy môn Tin học gửi đến và có tệp đính kèm.

Hướng dẫn thực hiện:

Bước 1. Đăng nhập tài khoản Gmail của em.

Bước 2. Tại ô **Search in mail** chọn , xuất hiện cửa sổ tìm kiếm như ở **Hình 4**.

Bước 3. Chọn tiêu chí tìm kiếm: chọn mục **From** và nhập địa chỉ email của giáo viên môn Tin học, tích chọn **Has attachment** (chứa tệp đính kèm), chọn **Search**.

Khi muốn tìm các email đã được gắn sao, chọn **Starred** ở cột bên trái cửa sổ màn hình tài khoản email (**Hình 1**).

Lưu ý: Có thể chọn nhiều tiêu chí tìm kiếm trong cửa sổ tìm kiếm (**Hình 4**) để thu hẹp kết quả tìm kiếm.



Hình 4. Cửa sổ tìm kiếm email

Để thuận lợi cho việc tìm kiếm và quản lý email, có thể tạo bộ lọc email. Muốn tạo bộ lọc email, chọn **Create filter** , chọn các tiêu chí cho bộ lọc, chọn **Create filter**.



Em hãy tạo nhãn “Tin học” là nhãn con của nhãn “Học tập” (đã được tạo trong bài thực hành). Nhãn “Tin học” dành cho các email được gửi từ giáo viên môn Tin học và các bạn trong nhóm làm bài tập môn Tin học cùng với em. Sau đó, hãy tìm kiếm và tạo bộ lọc những email mà em cho là quan trọng.

ỨNG XỬ VĂN HOÁ VÀ AN TOÀN TRÊN MẠNG
PHÒNG TRÁNH LỪA ĐẢO VÀ ỨNG XỬ VĂN HOÁ TRÊN MẠNG

Học xong bài này, em sẽ:

- ✓ Nêu được một số dạng lừa đảo phổ biến trên mạng và những biện pháp phòng tránh.
- ✓ Giao tiếp được trên mạng qua email, chat, mạng xã hội,... và trong môi trường số một cách văn minh, phù hợp với văn hoá ứng xử.



Theo em, lừa đảo trên không gian mạng dễ gặp hay hiếm thấy? Dễ tránh hay khó tránh? Vì sao?

1 Lừa đảo qua mạng**a) Một số dạng lừa đảo****1**

Em hãy sử dụng máy tìm kiếm tìm cụm từ “dạng lừa đảo phổ biến trên mạng” và cho biết:

- 1) Số kết quả trả về là nhiều hay ít?
- 2) Có thể tính được có bao nhiêu dạng lừa đảo hay không?

Không thể tin tưởng mọi điều nhìn thấy, nghe thấy trên mạng. Biết cách phát hiện nội dung giả mạo, lừa đảo là một kỹ năng quan trọng khi sử dụng Internet. Bọn lừa đảo trên mạng dùng nhiều thủ đoạn tinh vi, nhằm những mục đích khác nhau. Có nhiều ví dụ về lừa đảo, từ lừa “nhấn chuột là được tiền”, lừa nạp thẻ điện thoại, đến lừa tiền đặt cọc, tiền chuyển hàng hay đánh cắp thông tin cá nhân qua trang web giả mạo. Dưới đây nêu ví dụ một số thủ đoạn lừa đảo qua mạng.

Lừa đảo trúng thưởng, tặng quà để lấy tiền phí vận chuyển. Gửi email hay tin nhắn qua mạng xã hội và thông báo, ví dụ: “Bạn thật may mắn đã trúng thưởng...” hoặc “Nhân ngày lễ lớn, ngày kỉ niệm của công ty, để tri ân khách hàng, công ty xin tặng món quà... Cần trả phí vận chuyển (từ 100 đến 200 nghìn đồng) để nhận quà”. Kèm theo thông báo là ảnh món quà rất bắt mắt, ghi giá bán tới vài triệu đồng. Nạn nhân mất tiền phí vận chuyển, không nhận được gì hoặc món quà chỉ đáng giá 10 đến 20 nghìn đồng.

Lừa đảo chiếm tiền đặt cọc hoặc bán hàng giả. Lập tài khoản mạo danh các gian hàng trực tuyến uy tín để lừa khách hàng đặt mua, sau đó yêu cầu chuyển tiền đặt cọc để chiếm đoạt hay yêu cầu thanh toán và trả hàng giả, chất lượng khác xa với hình ảnh quảng cáo.

Lừa đảo để lấy cắp thông tin cá nhân. Cũng dùng thủ đoạn giống như hai trường hợp ở trên, nhưng thay vì lừa lấy tiền, bọn lừa đảo yêu cầu nhấn vào link để xác nhận sớm, nếu chậm sẽ mất cơ hội. Đường link gửi kèm sẽ dẫn tới một trang web (giả mạo) yêu cầu cung cấp thông tin cá nhân (ví dụ số tài khoản, mật khẩu,...) để có thể thực hiện giao dịch và chúng sẽ lấy cắp những thông tin ấy.

Các đối tượng lừa đảo có thể mạo danh các cơ quan, doanh nghiệp, người có uy tín, bạn bè hoặc người quen, thậm chí là đối tác nước ngoài, mời chào hợp tác kinh doanh, mua hàng giá rẻ,... kèm link lừa đảo.

b) Dấu hiệu lừa đảo và lời khuyên phòng ngừa

Trong tin học, việc lừa đảo để lấy cắp thông tin cá nhân bằng các trang web giả gọi là *phishing*. Cần nhận biết các dấu hiệu lừa đảo và luôn có ý thức đề phòng để tự bảo vệ, tránh bị lừa.

– Các cơ quan, tổ chức, doanh nghiệp uy tín phải đảm bảo giao tiếp quan hệ công chúng với chất lượng cao, có tính chuyên nghiệp. Nếu email, trang web có lỗi chính tả, lỗi hành văn thì đó có thể là lừa đảo. Những lỗi này có thể là do sự thiếu chuyên nghiệp của kẻ lừa đảo, do cố gắng tránh các bộ lọc thông minh, phát hiện kiểu lừa đảo đã biết, do được dịch từ một ngoại ngữ, từ kẻ lừa đảo xuyên biên giới, nhằm đến nạn nhân thích mới lạ.

Tên miền gồm vài phần cách nhau dấu chấm. Phần đầu viết tắt tên cơ quan, tổ chức, doanh nghiệp dễ nhớ nhưng các phần đuôi như: “com”, “net”, “org”,... ít được chú ý hơn. Các đuôi tên miền khác với tên miền chính thức mà cơ quan, tổ chức, doanh nghiệp vẫn dùng là dấu hiệu lừa đảo.

Cần chú ý nhận biết những cách viết sai chính tả trong tên miền để đánh lừa người đọc. Ví dụ, thay chữ “o” bằng số 0; thay “m” bằng “r” và “n”. Đây là những thủ đoạn phổ biến.

– Trỏ chuột vào một liên kết nhưng không nháy chuột, ta sẽ nhìn thấy địa chỉ đích thực sự mà liên kết sẽ mở ra. Nếu nó không khớp với địa chỉ hiển thị mời nháy chuột thì đó là dấu hiệu lừa đảo.

– Cảnh giác với email, tin nhắn từ người lạ, với cách xưng hô chung chung hoặc đột xuất bất ngờ từ người quen cũ lâu nay ít liên hệ. Tạo ra tình huống khẩn cấp là một thủ đoạn phổ biến của kẻ lừa đảo. Nạn nhân sẽ không kịp suy nghĩ về hậu quả.

Hãy tìm cách kiểm tra lại thông tin bằng con đường khác, chẳng hạn như gọi điện thoại trực tiếp, truy cập địa chỉ trang web in trên các tài liệu chính thức.

Khi nghi ngờ email, tin nhắn là lừa đảo, đừng mở bất kì liên kết hoặc tệp đính kèm nào mà hãy kiểm tra địa chỉ đích thực sự để phát hiện liên kết lừa đảo.

c) Nguyên tắc để hạn chế thiệt hại

Nếu nghi ngờ rằng mình đã có thể vô tình bị lừa qua mạng, hãy làm ngay một vài việc sau:

- Lập tức thay đổi mật khẩu cho những tài khoản giao tiếp qua mạng bị ảnh hưởng. Cần thiết lập xác minh hai bước cho những tài khoản quan trọng.
- Nếu tài khoản bị ảnh hưởng có liên quan đến nhà trường hay một cơ quan, tổ chức, cần thông báo ngay cho người có trách nhiệm.
- Nếu đã lỡ chia sẻ thông tin về thẻ tín dụng, tài khoản cá nhân, hãy báo ngay cho ngân hàng biết.
- Nếu đã bị thiệt hại, hãy báo ngay cho cơ quan chức năng.

2 Văn hoá ứng xử trên mạng



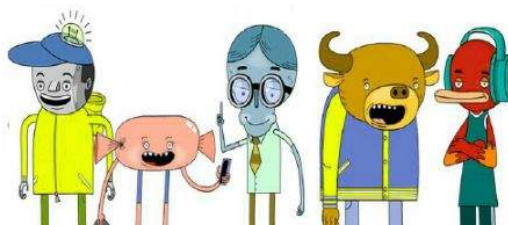
- 1) Theo em cụm từ “anh hùng bàn phím” có hàm ý gì? Hãy nêu vài ví dụ cụ thể về “anh hùng bàn phím”.
- 2) Nếu em là người có nhiều fan hâm mộ trên mạng xã hội, em nên làm gì và tránh những gì?

a) Quy tắc nền tảng: Thế giới ảo, cuộc sống thực

Trên không gian mạng, các tiêu chuẩn về hành xử có đạo đức, có văn hoá, tuân thủ pháp luật cũng như trong cuộc sống thực. Hãy ý thức rằng khi lên mạng là đang ở giữa cộng đồng (Hình 1).

Trong cuộc sống thực, hầu hết mọi người đều tuân thủ luật pháp, hành xử lịch sự, có văn hoá. Một số người hành xử trên mạng theo cách khác hẳn với khi đối mặt trực tiếp vì họ cho rằng, trên không gian mạng thì yêu cầu thấp hơn về đạo đức, văn hoá trong hành xử.

✓ common sense
digital compass™



Hình 1. Ý thức cộng đồng là kim chỉ nam cho mọi hành vi trên mạng

b) Một số nguyên tắc về ứng xử trên mạng

Hãy đặt mình vào vị trí người khác. Cha mẹ, thầy cô vẫn dạy: “Ta đối xử với người khác thế nào thì họ đối xử với ta như thế”. Trên không gian mạng, lời khuyên này được cụ thể hoá là: “Hãy nhớ ở đầu kia của mạng là những người khác, cũng có cảm xúc giống ta!”. Vì không nhìn thấy, ta dễ dàng quên rằng họ đang có mặt; không thấy các phản ứng tức thì, người ta dễ hiểu lầm nhau và khi biết thì đã muộn.

Rộng lượng với người khác, không gây chiến trên mạng. Khi ai đó mắc lỗi với bạn, hãy rộng lượng. Cần phản ứng lịch sự và tốt nhất là theo cách riêng tư hơn là to tiếng công khai.

Có người thích “thể hiện”, dù sự việc không liên quan trực tiếp đến mình cũng phản ứng theo cách cực đoan. Phán xử người khác bằng ngôn từ bất lịch sự, hành vi thiếu văn hoá chỉ dẫn đến có thêm kẻ thù mà thôi. Những gì bạn nói ra, viết ra trên mạng có thể được lưu trữ ở một nơi nào đó, được chuyển tiếp đi bất cứ đâu mà bạn không còn quyền kiểm soát nữa.

Tôn trọng “văn hoá nhóm”. Khi tham gia một nhóm mạng mới, hãy tìm hiểu xem “văn hoá nhóm” có phù hợp với bạn. Có những điều chấp nhận được ở một nơi này lại thành thô lỗ, bất lịch sự ở nơi khác. Một số câu chuyện cười kể trong nhóm nhỏ là bình thường nhưng không nên mang kể trong nhóm chat của cả lớp.

Bạn không phải là trung tâm của không gian mạng. Đừng cố lấn át, nói hết phần người khác khi tham gia một nhóm mạng. Đừng mong đợi tất cả bài đăng, câu hỏi của bạn được phản hồi ngay; đừng cho rằng tất cả người đọc sẽ đồng ý hoặc quan tâm đến những bài viết đầy tâm huyết của bạn.

Tôn trọng thời gian và công sức của người khác. Thật dễ nhấn nút “đăng” hay “gửi” bản sao cho nhiều người. Nhưng đăng bài nhiều lần, đăng tin rác, gửi thư rác,... sẽ làm phiền người khác và chiếm dụng đường truyền, chiếm dụng dung lượng lưu trữ trên máy chủ. Người có trách nhiệm luôn ý thức rằng “Không lãng phí thời gian và công sức của người khác” (Hình 2).



Hình 2. Thư rác quấy nhiễu mọi người

Tôn trọng quyền riêng tư của người khác. Không tìm cách đọc email, tin nhắn của người khác. Không chuyển tiếp email, tin nhắn riêng tư mà mình được chia sẻ cho người tiếp theo nếu không chắc đó là việc nên làm. Một số người thích thu thập thông tin về những người nổi tiếng hay bất kì ai có các vụ bê bối rồi chia sẻ cho nhau. Đó là xâm phạm quyền riêng tư, có thể gây ra áp lực lớn cho người “được quan tâm”; nạn nhân có thể bị stress thậm chí tự tử.

Không lợi dụng vị thế của mình để làm việc xấu. Một số người trong không gian mạng có nhiều ảnh hưởng hơn, dễ điều khiển luồng dư luận theo hướng ủng hộ ý kiến của họ. Đó là các KOL (*Key Opinion Leader*) trong một lĩnh vực, các quản trị viên hệ thống, các quản trị diễn đàn, nhóm tin trên mạng. Đạo đức trên mạng không cho phép bạn, với tư cách là một KOL, lợi dụng vị thế của mình vì mục đích xấu.



Câu 1. Hãy cho biết những dấu hiệu để phát hiện lừa đảo qua mạng.

Câu 2. Hãy cho biết quy tắc nền tảng về văn hoá, đạo đức trên mạng.



Câu 1. Em sẽ làm gì khi nhận được email báo được thưởng một phần quà vì là khách hàng trung thành và phải gửi ngay một khoản tiền để nhận thưởng?

Câu 2. Ngày 17/6/2021, Bộ Thông tin và Truyền thông ban hành Bộ Quy tắc ứng xử trên mạng xã hội, trong đó có quy định chung về Quy tắc Lành mạnh: hành vi, ứng xử trên mạng xã hội phù hợp với các giá trị đạo đức, văn hoá, truyền thống tốt đẹp của dân tộc Việt Nam. Em hãy trích ra một số quy tắc ứng xử cho cá nhân về điều này.



Câu 1. Cần làm gì trước khi nhấp vào một liên kết trong email từ người gửi chưa chắc chắn đáng tin?

Câu 2. Quy tắc ứng xử văn minh và có đạo đức trên mạng có gì khác với trong cuộc sống thực?

Tóm tắt bài học

- ✓ Cảnh giác để phòng lừa đảo trên mạng: thận trọng với email, tin nhắn khác thường; không vội vã, cần kiểm tra để tránh trang web giả mạo trước khi nhập thông tin cá nhân hay chuyển tiền; hành động ngay để hạn chế thiệt hại khi cần thiết.
- ✓ Ứng xử văn hoá và có đạo đức trên mạng: rộng lượng với người khác, không gây chiến trên mạng; tôn trọng “văn hoá nhóm”; tôn trọng quyền riêng tư, thời gian và công sức của người khác; không lợi dụng vị thế của mình để làm việc xấu.

GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

GIỚI THIỆU CÁC HỆ CƠ SỞ DỮ LIỆU

BÀI 1

BÀI TOÁN QUẢN LÝ VÀ CƠ SỞ DỮ LIỆU

Học xong bài này, em sẽ:

- ✓ Nhận biết được nhu cầu lưu trữ dữ liệu và khai thác thông tin cho bài toán quản lý.
- ✓ Diễn đạt được khái niệm hệ cơ sở dữ liệu, nêu được ví dụ minh họa.



Có một số cụm từ mà em đã từng nghe và có thể em đã từng dùng, ví dụ: “Quản lý học sinh”, “Quản lý nhân sự”, “Quản lý chi tiêu cá nhân”,... Theo em, việc quản lý có liên quan đến việc lưu trữ và xử lý dữ liệu không? Hãy nêu một việc em đã làm để quản lý một hoạt động nào đó của mình.

1 Bài toán quản lý

Có rất nhiều bài toán quản lý cho các tổ chức lớn, nhỏ khác nhau với mức độ phức tạp khác nhau và ngay cả mỗi cá nhân cũng có những nhu cầu quản lý của riêng mình. Quản lý là công việc rất phổ biến. Xã hội càng phát triển, càng văn minh thì nhu cầu và chất lượng quản lý các hoạt động càng cao.

Việc quản lý một tổ chức gắn liền với những dữ liệu phản ánh thông tin về hoạt động của tổ chức đó. Ví dụ: Dựa trên kết quả học tập của lớp mà giáo viên có thể đề xuất với nhà trường danh sách những em tham gia bồi dưỡng học sinh giỏi môn Tin học; để khách sạn quyết định có nhận cho khách thuê phòng hay không tùy thuộc vào thông tin về số phòng còn trống chưa ai thuê trong thời gian cụ thể đó. Trong hai ví dụ trên đây, dễ thấy rằng nếu thông tin không chính xác sẽ dẫn đến những hậu quả đáng tiếc.

Thông tin dùng trong bài toán quản lý phải chính xác, kết quả xử lý thông tin phải đáng tin cậy để giúp có được quyết định đúng đắn, hợp lý.

2 Xử lý thông tin trong bài toán quản lý

Các bài toán quản lý đều có chung đặc điểm là lưu trữ và xử lý dữ liệu về hoạt động của một tổ chức. Thông thường, từ “hồ sơ” được dùng để chỉ một tập hợp dữ liệu được tổ chức và thể hiện theo những khuôn mẫu nào đó. Xử lý thông tin trong bài toán quản lý bao gồm: *tạo lập hồ sơ, cập nhật và khai thác thông tin*.

a) Tạo lập hồ sơ

Ví dụ, để quản lí việc học tập của một lớp, hồ sơ của lớp thường có cấu trúc dạng bảng để dễ theo dõi như ở *Bảng 1*.

Bảng 1. Ví dụ về hồ sơ học sinh của một lớp

STT	Mã định danh	Họ và tên	Ngày sinh	Giới tính	Đoàn viên	Địa chỉ	Toán	Ngữ văn	...	Tin học
1	13109413	Phan Thuỳ Anh	29/10/2007	Nữ	x	39 Hùng Vương	7.3	7.4	...	8.5
2	13109735	Lê Minh Đức	05/9/2007	Nam	x	15 Văn Cao	6.4	7.2	...	7.0
3	13124595	Hoàng Giang	21/12/2007	Nam		27 Lò Sũ	7.7	7.6	...	9.3
4	31387552	Nguyễn Thị Hà	12/6/2007	Nữ	x	29 Hàng Bún	7.1	6.5	...	8.4
5	13169292	Trần Minh Tú	14/11/2007	Nữ	x	18 Quán Thánh	7.8	6.5	...	7.7
...	...	...	...	...	...	...	...	...	...	...

Để phản ánh đúng thực tế, dữ liệu trong bảng phải đầy đủ và chính xác.

– Dữ liệu phải đầy đủ so với yêu cầu quản lí. Ví dụ, muốn quản lí thông tin mỗi học sinh đã là đoàn viên hay chưa, bảng hồ sơ của lớp cần có thêm cột ghi nhận thông tin này, nếu sĩ số lớp là 45 thì bảng phải có 45 hàng dữ liệu.

– Dữ liệu phải chính xác. Ví dụ, không thể có hai hàng trong bảng hoàn toàn giống nhau ở họ tên, ngày sinh và địa chỉ, vì hoặc đó là dư thừa dữ liệu hoặc không phân biệt được chính xác điểm của mỗi bạn trong hai bạn trùng tên đó.

Khi tạo lập hồ sơ cho mỗi bài toán quản lí, phải xác định đầy đủ những dữ liệu cần được lưu trữ, đồng thời dữ liệu nhập vào phải đúng đắn.

b) Cập nhật dữ liệu

Dữ liệu được lưu trữ cần được cập nhật để phản ánh kịp thời những thay đổi diễn ra trên thực tế. Ví dụ, trong quản lí học tập của một lớp như ở *Bảng 1*, những việc làm sau đây là cập nhật dữ liệu:

– Học sinh Hoàng Giang vừa chuyển nhà về địa chỉ “20 Chùa Bộc”, cần sửa đổi dữ liệu tương ứng, dữ liệu “27 Lò Sũ” không còn đúng nữa.

– Cần bổ sung một hàng mới ghi dữ liệu cho học sinh Trần Anh Tuấn mới chuyển đến lớp.

– Cần xoá dữ liệu của học sinh Nguyễn Thị Hà vì học sinh này đã chuyển trường do bố mẹ chuyển công tác về tỉnh khác.

Cập nhật dữ liệu gồm các thao tác: thêm, sửa, xoá dữ liệu. Toàn bộ dữ liệu sau mỗi lần cập nhật cũng phải thoả mãn tính đầy đủ và đúng đắn.

c) Khai thác thông tin

Mục đích của việc lưu trữ và cập nhật dữ liệu là để khai thác thông tin, phục vụ cho việc điều hành công việc và ra quyết định của người quản lí. Một số việc khai thác thông tin thường gặp là: *tìm kiếm dữ liệu, thống kê, lập báo cáo*.

– *Tìm kiếm dữ liệu* là việc rút ra được các dữ liệu thoả mãn một số điều kiện nào đó từ dữ liệu đã lưu trữ. Ví dụ: tìm họ và tên học sinh có điểm môn Tin học cao nhất.

– *Thống kê* là khai thác hồ sơ dựa trên tính toán để đưa ra các thông tin không có sẵn trong hồ sơ. Ví dụ: xác định điểm cao nhất và điểm thấp nhất của môn Tin học; xác định số học sinh là đoàn viên.

– *Lập báo cáo* là sử dụng các kết quả tìm kiếm, thống kê, sắp xếp dữ liệu được rút ra để tạo lập một bộ hồ sơ mới có nội dung và cấu trúc theo một số yêu cầu cụ thể trong quản lí. Ví dụ: Hết mỗi học kì, giáo viên chủ nhiệm cần có một danh sách học sinh đề nghị nhà trường khen thưởng, cuối năm học cần báo cáo phân loại học tập để lên kế hoạch ôn tập hè cho lớp và trao đổi với phụ huynh về hướng nghiệp cho các em.

Khai thác thông tin là để phục vụ kịp thời cho công tác quản lí. Do vậy, việc xử lí dữ liệu trong hồ sơ phải nhanh chóng, chính xác và thông tin kết xuất ra phải ở dạng dễ hiểu cho người quản lí.

3 Cơ sở dữ liệu và phần mềm hệ quản trị cơ sở dữ liệu



Theo em, có nên dùng phần mềm soạn thảo văn bản hay phần mềm bảng tính để tạo lập hồ sơ, cập nhật và khai thác thông tin trong hồ sơ phục vụ công tác quản lí của một tổ chức hay không? Vì sao?

Ngày nay, với khả năng lưu trữ dữ liệu không lồ, tốc độ truy xuất và xử lí dữ liệu vô cùng nhanh, máy tính là công cụ hỗ trợ đắc lực cho con người trong mọi hoạt động thông tin. Tập hợp hồ sơ dữ liệu làm cơ sở cho việc quản lí các hoạt động của một tổ chức, được số hoá để máy tính truy cập, cập nhật và xử lí, được gọi là một *cơ sở dữ liệu* (CSDL).



Cơ sở dữ liệu: tập hợp dữ liệu được tổ chức sao cho máy tính có thể lưu trữ, truy cập, cập nhật và xử lí để phục vụ cho hoạt động của một đơn vị nào đó.

Để giúp tạo lập, cập nhật CSDL và khai thác thông tin trong CSDL có loại phần mềm được gọi là *hệ quản trị CSDL* (Database Management System – DBMS).

Hệ quản trị CSDL là một hệ thống chương trình giúp người dùng tương tác với CSDL qua các giao diện dễ hiểu, dễ dùng (như hệ thống bảng chọn, hộp thoại, các biểu mẫu, báo cáo,...). Với CSDL, hệ quản trị CSDL là hệ thống chương trình truy cập được dữ liệu nhưng tuân theo những ràng buộc để đảm bảo tính đúng đắn cho mỗi thao tác cập nhật dữ liệu và khai thác dữ liệu.

Mỗi đơn vị, mỗi tổ chức có những yêu cầu riêng và cụ thể trong khai thác CSDL thể hiện qua các mẫu (giao diện) cập nhật dữ liệu, các mẫu tìm kiếm dữ liệu và báo cáo thường dùng. *Hệ cơ sở dữ liệu* của một đơn vị là cách gọi chung một tập hợp gồm: CSDL của đơn vị, hệ quản trị CSDL và các phần mềm ứng dụng có các giao diện tương tác với CSDL đáp ứng được nhu cầu quản lý của đơn vị đó.

Các phần mềm ứng dụng khác muốn sử dụng dữ liệu trong CSDL đều phải thông qua hệ quản trị CSDL giống như mọi chương trình máy tính đều phải chạy dưới sự kiểm soát, điều phối của hệ điều hành.



Hệ quản trị CSDL: phần mềm cung cấp môi trường thuận lợi và hiệu quả để tạo lập, lưu trữ và khai thác dữ liệu của CSDL.

Thực hành tìm hiểu các yêu cầu của một bài toán quản lý và CSDL phục vụ bài toán đó

Em hãy hình dung việc quản lý thư viện của một trường học, thảo luận với bạn và thực hiện các yêu cầu sau đây.

a) Mô tả hoạt động của thư viện

Gợi ý: Cho mượn sách hoặc trả sách như thế nào? Căn cứ vào đâu để biết ai đã mượn, trả sách gì? Căn cứ vào đâu để biết một quyển sách cụ thể đã được cho mượn và chưa được trả lại?...

b) Liệt kê những dữ liệu cần có trong CSDL

Gợi ý: Những đối tượng cần quản lý là người đọc, sách cho mượn,...

– Với người đọc, cần quản lý thông tin gì? (Thông tin trên thẻ thư viện gồm: Số thẻ TV, Họ và tên,...).

– Với sách cho mượn, cần quản lý thông tin gì? (Thông tin về quyển sách gồm: Mã sách, Tên sách, Tác giả,...).

c) Nêu ví dụ

Nêu thêm ít nhất hai ví dụ cho mỗi công việc sau đây:

– Cập nhật dữ liệu (cho CSDL):

Ví dụ 1. Khi có thêm một học sinh làm thẻ thư viện, cần bổ sung một số thông tin của học sinh này vào CSDL.

– Tìm kiếm dữ liệu:

Ví dụ 2. Tìm xem trong thư viện có quyển “Tôi tài giỏi, Bạn cũng thế” không?

– Thống kê và báo cáo:

Ví dụ 3. Xác định trong thư viện có bao nhiêu quyển sách về Tin học (giả sử sách về Tin học sẽ có hai chữ cái đầu trong mã sách là “TH”).



Giả sử dùng một bảng để chứa dữ liệu thể hiện thông tin về những người được mượn sách ở thư viện (những người có thẻ thư viện), em hãy chỉ ra một vài điều kiện cho dữ liệu trong bảng đó nhằm đảm bảo tính chính xác của thông tin. Theo em, nếu dùng một phần mềm bảng tính để tạo lập, lưu trữ bảng dữ liệu đó thì phần mềm bảng tính có tự động kiểm soát các cập nhật dữ liệu để đảm bảo được các điều kiện đã đặt ra hay không?



Câu 1. Trong các câu sau, những câu nào đúng?

- a) CSDL là tập hợp dữ liệu được lưu trữ trên thiết bị nhớ phục vụ cho hoạt động của một cơ quan, đơn vị nào đó.
- b) Hệ CSDL của một đơn vị là phần mềm quản trị CSDL của đơn vị đó.
- c) Các giá trị dữ liệu được lưu trữ trong CSDL phải thoả mãn một số ràng buộc để góp phần đảm bảo được tính đúng đắn của thông tin.
- d) Hệ quản trị CSDL là chương trình kiểm soát được các cập nhật dữ liệu.

Câu 2. Theo em, những ứng dụng nào dưới đây cần có CSDL?

- a) Quản lí bán vé máy bay.
- b) Quản lí chi tiêu cá nhân.
- c) Quản lí cước phí điện thoại.
- d) Quản lí một mạng xã hội.

Tóm tắt bài học

- ✓ Các tổ chức hoạt động trong xã hội đều có nhu cầu lưu trữ dữ liệu và khai thác thông tin cho bài toán quản lí.
- ✓ Muốn máy tính hỗ trợ đắc lực được cho công tác quản lí, dữ liệu của một đơn vị phải được tổ chức trong một CSDL với tính đầy đủ và đúng đắn.
- ✓ Phần mềm quản trị CSDL là loại phần mềm tạo ra môi trường thuận lợi để tạo lập CSDL, cập nhật cho CSDL theo cách đúng đắn, đồng thời kiểm soát được các truy cập đến dữ liệu, đảm bảo tính chính xác và sự an toàn của dữ liệu.

BẢNG VÀ KHOÁ CHÍNH TRONG CƠ SỞ DỮ LIỆU QUAN HỆ

Học xong bài này, em sẽ:

- ✓ Diễn đạt được khái niệm quan hệ (bảng) và khoá của một quan hệ. Giải thích được các khái niệm đó qua ví dụ minh hoạ.
- ✓ Giải thích được ràng buộc khoá là gì.
- ✓ Biết được các phần mềm quản trị CSDL có cơ chế kiểm soát các cập nhật dữ liệu để đảm bảo ràng buộc khoá.



Hồ sơ học sinh một lớp được tổ chức theo dạng bảng: mỗi hàng chứa dữ liệu về một học sinh, mỗi cột chứa dữ liệu về một thuộc tính của học sinh như: Họ và tên, Ngày sinh,... Theo em, cách tổ chức như vậy có ưu điểm gì trong việc quản lí thông tin học sinh của lớp?

1 Tổ chức dữ liệu trong CSDL quan hệ và các thao tác trên dữ liệu

a) Cơ sở dữ liệu quan hệ

Cơ sở dữ liệu quan hệ là một tập hợp các bảng dữ liệu có liên quan với nhau. *Hình 1* cho thấy một phần bảng HỌC SINH 11 trong một CSDL quan hệ của một trường học.

Mã định danh	Họ và tên	Ngày sinh	Giới tính	Đoàn viên	Địa chỉ
13109413	Phan Thuỳ Anh	29/10/2007	Nữ	☒	39 Hùng Vương
13109735	Lê Minh Đức	05/09/2007	Nam	☒	15 Văn Cao
13124595	Hoàng Giang	21/12/2007	Nam	☐	27 Lò Sũ
31387552	Nguyễn Thị Hà	12/06/2007	Nữ	☒	29 Hàng Bún
13169292	Trần Minh Tú	14/11/2007	Nữ	☒	18 Quán Thánh

Hình 1. Một phần bảng HỌC SINH 11 của một cơ sở dữ liệu quan hệ

Tên của mỗi cột trong bảng cho biết ý nghĩa dữ liệu ở các ô thuộc cột đó. Tên bảng cùng với tên cột giúp hiểu nghĩa của mỗi hàng trong bảng. Mỗi hàng trong bảng chứa một bộ các giá trị, ví dụ, trong *Hình 1*, bộ gồm 6 giá trị: “13109413”, “Phan Thuỳ Anh”, “29/10/2007”, “Nữ”, “☒”, “39 Hùng Vương” cho thông tin về một học sinh.

Mỗi một hàng trong bảng của CSDL quan hệ còn được gọi là một *bản ghi*. Mỗi cột của bảng còn được gọi là một *trường*, trong *Hình 1* bảng HỌC SINH 11 có 6 trường phản ánh 6 thuộc tính của mỗi học sinh.

b) Cập nhật dữ liệu trong CSDL quan hệ

Cập nhật dữ liệu của một bảng bao gồm các thao tác thêm, sửa và xoá dữ liệu của bảng. Cấu trúc của một bảng bao gồm các mô tả cho các cột của bảng; người thiết kế CSDL sẽ định nghĩa cấu trúc của các bảng dựa vào các yêu cầu quản lý của đơn vị chủ quản. Cập nhật dữ liệu của một bảng không làm thay đổi cấu trúc của bảng.

c) Truy vấn trong CSDL quan hệ

Dữ liệu được tổ chức, lưu trữ trong CSDL là để người sử dụng có thể khai thác dữ liệu, rút ra thông tin phục vụ các hoạt động hoặc giúp đưa ra các quyết định phù hợp, kịp thời. Bản chất việc khai thác một CSDL là tìm kiếm dữ liệu và kết xuất ra thông tin cần tìm, công việc này còn được gọi là *truy vấn CSDL*.

d) Các ràng buộc dữ liệu trong CSDL quan hệ



Theo em, mỗi học sinh cần phải có riêng một *Mã định danh* để đưa vào hồ sơ quản lý hay không? Vì sao?

Dữ liệu trong CSDL quan hệ phải thỏa mãn một số ràng buộc gọi là *ràng buộc toàn vẹn về dữ liệu* để đảm bảo tính xác định và đúng đắn của dữ liệu. Ví dụ một số ràng buộc dữ liệu:

- Trong một bảng không có hai bản ghi giống nhau hoàn toàn.
- Trong cùng một bảng, mỗi trường có một tên phân biệt với tất cả các trường khác.
- Mỗi bảng có một tên phân biệt với các bảng khác trong cùng CSDL.
- Mỗi ô của bảng chỉ chứa một giá trị.

Ngoài ra, tùy theo yêu cầu của bài toán quản lý cụ thể mà người thiết kế CSDL đặt thêm một số ràng buộc khác cho dữ liệu. Ví dụ: Người thiết kế CSDL cho trường học có thể yêu cầu *Mã định danh* của mỗi học sinh phải là một dãy số không quá 12 kí tự, tất cả các kí tự đều là số. Với ràng buộc như thế, việc nhập “0011234567899” vào cột *Mã định danh* là không hợp lệ, đó là vi phạm ràng buộc miền giá trị.

2 Khoá của một bảng

Trong một bảng, mỗi bản ghi thể hiện thông tin về một đối tượng (một cá thể hoặc một sự kiện) nên không thể có hai bản ghi giống nhau hoàn toàn. Trong bảng chứa dữ liệu học sinh, ví dụ như bảng HỌC SINH 11 (*Hình 1*), hai học sinh khác nhau sẽ có hai *Mã định danh* khác nhau. Điều này giống như trường hợp số căn cước công dân của mỗi người xác định người đó là duy nhất, không nhầm lẫn với bất cứ ai.

Trong một bảng, có những tập hợp gồm một trường hay một số trường mà giá trị của chúng ở các bản ghi khác nhau là khác nhau. Ví dụ ở *Hình 2*: một giá trị của trường *STT* chỉ xuất hiện ở một bản ghi; một bộ giá trị của hai trường *CCCD* và *BHYT*, chẳng hạn (“001160017719”, “HT3010101040124”) chỉ xuất hiện ở một bản ghi. Nói cách khác, tập hợp gồm một trường *STT* và tập hợp gồm hai trường *CCCD* và *BHYT* đều có tính chất: Dùng giá trị của nó xác định được duy nhất một bản ghi trong bảng. Với ví dụ bảng trong *Hình 2*, có thể kể ra thêm một số tập hợp trường có tính chất như vậy:

- Tập chỉ gồm một trường *CCCD*.
- Tập gồm hai trường: *STT*, *Họ và tên*.
- Tập gồm tất cả sáu trường.
- ...

Khoá của một bảng là tập hợp một số trường có tính chất: mỗi bộ giá trị của các trường đó xác định duy nhất một bản ghi trong bảng và không thể bỏ bớt bất cứ trường nào mà tập hợp gồm các trường còn lại vẫn còn tính chất đó.

Ví dụ với bảng ở *Hình 2*:

- Tập hợp chỉ có một trường *CCCD* là một khoá.
- Tập hợp gồm hai trường *STT*, *Họ và tên* không phải là khoá vì nếu bỏ trường *Họ và tên* ra khỏi tập hợp này thì chỉ riêng *STT* cũng có tính chất xác định duy nhất một bản ghi trong bảng.

Nếu không thể có hai nhân viên trùng nhau hoàn toàn ở *Họ và tên*, *Ngày sinh* thì tập hợp gồm hai trường *Họ và tên*, *Ngày sinh* cũng tạo thành một khoá. Nhưng tập gồm ba thuộc tính *STT*, *Họ và tên*, *Ngày sinh* không phải là một khoá.

STT	Họ và tên	Ngày sinh	Giới tính	CCCD	BHYT
1	Nguyễn Thành An	27/3/1970	Nam	001160017719	HT3010101040124
2	Đỗ Thu Cúc	05/5/1973	Nữ	001250025170	HT3012101340125
3	Hoàng Thị Dung	12/4/1971	Nữ	001171123635	HT1013101240124
...	...	...	...	...	...

Hình 2. Bảng NHÂN VIÊN trong một CSDL của một công ty

Khi bảng có hơn một khoá, người ta thường chọn (chỉ định) một khoá làm *khoá chính* (Primary Key), ưu tiên chọn khoá gồm ít trường nhất, tốt nhất nếu chọn được khoá chỉ là một trường. Bởi vậy, với bảng ở *Hình 2*, thay vì chọn khoá chính là tập hợp gồm hai trường

Khoá của một bảng:

tập hợp các trường (có thể chỉ là một trường) mà mỗi bộ giá trị của nó xác định duy nhất một bản ghi ở trong bảng và ta không thể bỏ đi trường nào mà tập hợp các trường còn lại vẫn còn có tính chất xác định duy nhất một bản ghi trong bảng.

Họ và tên và *Ngày sinh*, ta có thể chọn trường *STT* hay trường *CCCD* làm khoá chính của bảng bởi các hàng trong bảng phân biệt với nhau bởi số thứ tự (*STT*). Trên thực tế, người ta thường tạo thêm trường *MaNV* (Mã nhân viên) làm khoá chính cho bảng chứa thông tin nhân viên để phù hợp với cách tổ chức quản lý của đơn vị đó.

Việc cập nhật dữ liệu cho một bảng cũng phải thoả mãn yêu cầu không làm xuất hiện hai bản ghi có giá trị khoá giống nhau. Yêu cầu này còn được gọi là *ràng buộc khoá*.

3 ► Hệ quản trị CSDL đảm bảo ràng buộc khoá

Bất cứ hệ quản trị CSDL nào cũng có cơ chế kiểm soát, ngăn chặn những vi phạm ràng buộc khoá đối với việc cập nhật dữ liệu. Để thực hiện điều đó, phần mềm yêu cầu người tạo lập CSDL chỉ định trường làm khoá chính và mỗi khi xuất hiện thao tác cập nhật dữ liệu, phần mềm sẽ tự động kiểm tra xem cập nhật đó có vi phạm ràng buộc khoá hay không.

4 ► Thực hành với khoá của bảng trong CSDL

Yêu cầu:

Sử dụng phần mềm Microsoft Access 365 tạo bảng *SÁCH* có cấu trúc như ở *Hình 3*, chỉ định trường *Mã sách* làm khoá chính và nhập nhiều hơn 5 bản ghi cho bảng.

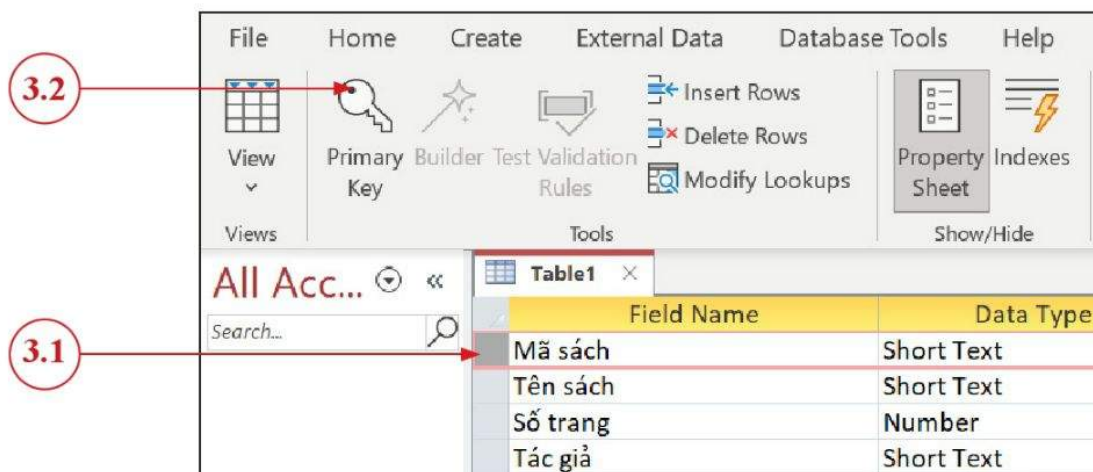
Hướng dẫn thực hiện:

Bước 1. Khởi chạy Microsoft Access 365 bằng cách nháy đúp chuột vào biểu tượng **Access** của phần mềm này.

Bước 2. Tạo một CSDL mới, trong CSDL mới này tạo cấu trúc cho bảng *SÁCH* bằng cách thực hiện tuần tự các thao tác sau:

- Chọn **Blank Desktop Database** rồi đặt tên cho CSDL mới (hoặc nháy đúp chuột vào biểu tượng của **Blank Desktop Database**, Access sẽ tự đặt tên cho CSDL mới tạo).

- Chọn **Create\Table Design** để xuất hiện cửa sổ khai báo cấu trúc bảng (*Hình 3*).



Hình 3. Chỉ định khoá chính

– Trên mỗi hàng nhập tên một trường (ở cột *Field Name*), chọn kiểu dữ liệu cho trường đó bằng cách đưa con trỏ chuột vào ô ở cột *Data Type* để làm xuất hiện danh sách cho chọn.

Bước 3. Chỉ định khoá chính cho bảng bằng cách chọn hàng có trường Mã sách (3.1), sau đó chọn **Primary Key** (3.2) (Hình 3).

Bước 4. Chọn **Save**  để lưu cấu trúc bảng và đặt tên cho bảng.

Bước 5. Chọn **View**  để xuất hiện cửa sổ cho nhập các bản ghi vào bảng.

Chú ý: Nên thử nhập hai bản ghi giống nhau để xem phần mềm báo lỗi vi phạm ràng buộc khoá ra sao.



Để tiếp tục xây dựng CSDL quản lý một thư viện, em hãy cho biết:

- Dự kiến của em về cấu trúc bảng NGƯỜI ĐỌC, biết rằng bảng này dùng để lưu trữ dữ liệu về những người có thẻ thư viện.
- Trong các trường của bảng NGƯỜI ĐỌC, nên chọn trường nào làm khoá chính? Giải thích vì sao?
- Hãy nêu ví dụ cụ thể về nhập dữ liệu cho bảng NGƯỜI ĐỌC nhưng vi phạm ràng buộc khoá.



Trong các câu sau, những câu nào đúng?

- Trong CSDL quan hệ, mỗi bảng chỉ có một khoá.
- Khoá của một bảng chỉ là một trường.
- Nếu hai bản ghi khác nhau thì giá trị khoá của chúng phải khác nhau.
- Các hệ quản trị CSDL quan hệ tự động kiểm tra ràng buộc khoá để đảm bảo tính đúng đắn của dữ liệu.

Tóm tắt bài học

- ✓ Một CSDL quan hệ là một tập hợp các bảng dữ liệu (quan hệ) có liên quan với nhau.
- ✓ Mỗi bảng trong CSDL đều phải có khoá, đó là tập hợp gồm một hay một số trường cho phép xác định duy nhất một bản ghi trong bảng.
- ✓ Dữ liệu trong một bảng phải thoả mãn ràng buộc khoá: Không có hai bản ghi giống nhau ở giá trị khoá. Mọi hệ quản trị CSDL quan hệ đều có cơ chế kiểm soát việc cập nhật dữ liệu để không xảy ra vi phạm ràng buộc khoá đối với mỗi bảng.

BÀI 3**QUAN HỆ GIỮA CÁC BẢNG VÀ KHOÁ NGOÀI TRONG CƠ SỞ DỮ LIỆU QUAN HỆ****Học xong bài này, em sẽ:**

- ✓ Diễn đạt được khái niệm khoá ngoài của một bảng và mối liên kết giữa các bảng. Giải thích được các khái niệm đó qua ví dụ minh hoạ.
- ✓ Giải thích được ràng buộc khoá ngoài là gì.
- ✓ Biết được các phần mềm quản trị CSDL có cơ chế kiểm soát các cập nhật dữ liệu để đảm bảo ràng buộc khoá ngoài.



Để quản lí sách, người đọc và việc mượn/trả sách của một thư viện (TV) trường học, bạn Anh Thư dự định chỉ dùng một bảng như mẫu ở *Hình 1*. Theo em, trong trường hợp cụ thể này, việc đưa tất cả dữ liệu cần quản lí vào trong một bảng như Anh Thư thực hiện có ưu điểm và nhược điểm gì?

Số thẻ TV	Họ và tên	Ngày sinh	Lớp	Mã sách	Tên sách	Số trang	Tác giả	Ngày mượn	Ngày trả
...	...	...	...	...	...	...	...	...	...

Hình 1. Cấu trúc bảng SÁCH-NGƯỜI ĐỌC-MƯỢN-TRẢ

Gợi ý: Xét một số trường hợp sau:

- 1) Một học sinh mượn sách nhiều lần, mỗi lần mượn nhiều quyển sách.
- 2) Cần bổ sung dữ liệu về số sách mới mua của thư viện.

1 Tính dư thừa dữ liệu

a) Dư thừa dữ liệu có thể dẫn đến dữ liệu không nhất quán khi cập nhật

Có thể một số người nghĩ rằng nên đưa tất cả dữ liệu cần lưu trữ vào trong một bảng vì khi cần tìm thông tin nào đó thì chỉ phải tìm trong một bảng. Nhưng thực tế cho thấy đa số bài toán quản lí cần dùng nhiều hơn một bảng dữ liệu. Nếu chỉ dùng một bảng thì rất có thể dẫn đến tình trạng dư thừa dữ liệu. Ví dụ, trường hợp nêu ở phần khởi động, giả sử học sinh có số thẻ TV “HS-002” tên là “Lê Bình” sinh ngày “02/3/2007”, học lớp “11A1” đã có 68 lần mượn sách. Như vậy, bộ giá trị (“HS-002”, “Lê Bình”, “02/3/2007”, “11A1”) phải xuất hiện 68 lần (trên 68 bản ghi của bảng). Tình trạng dư thừa dữ liệu có thể dẫn đến sai nhầm, không nhất quán về dữ liệu. Việc gõ nhập 68 lần

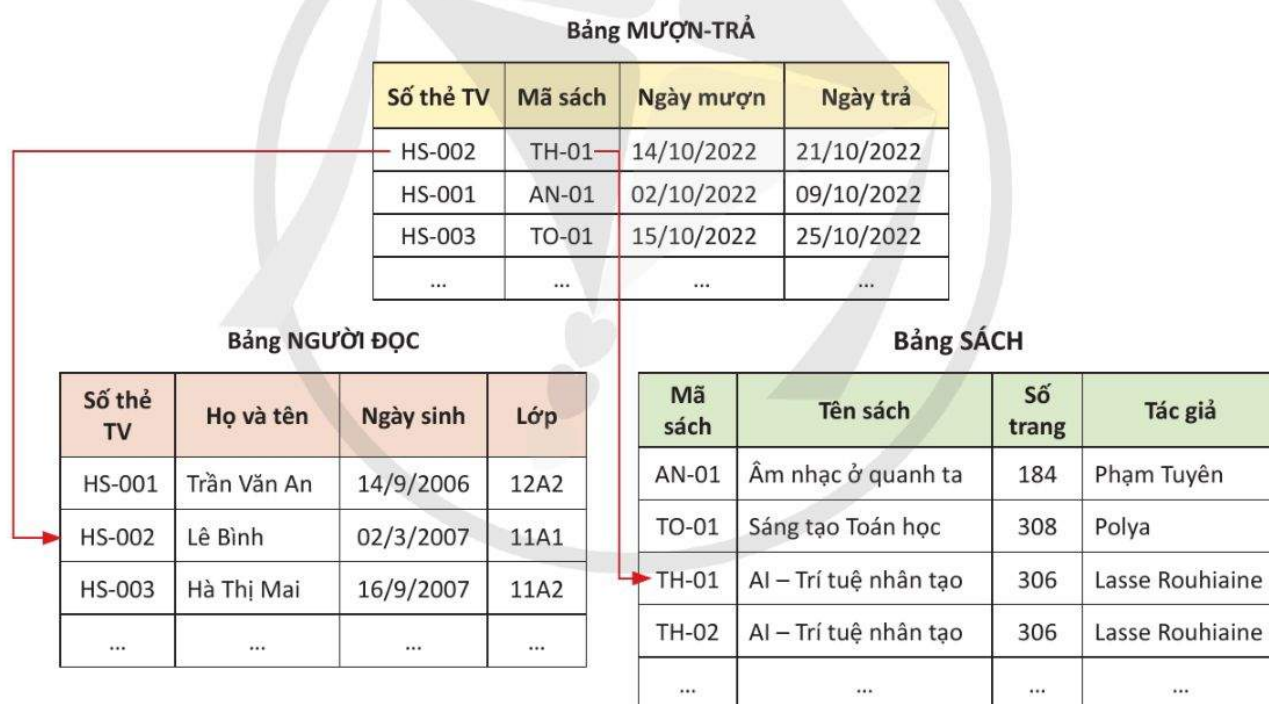
bộ dữ liệu về Lê Bình sẽ dễ xuất hiện sai nhầm hơn so với 68 lần chỉ gõ *Số thẻ TV* của Lê Bình vào bảng. Giải pháp tránh dư thừa là có thể dùng một bảng khác chỉ chứa dữ liệu về người đọc và có khoá chính là trường *Số thẻ TV*.

b) CSDL cần được thiết kế để tránh dư thừa dữ liệu

Dư thừa dữ liệu do trùng lặp dữ liệu có các nhược điểm là tốn nhiều vùng nhớ lưu trữ không cần thiết và dữ liệu có thể không nhất quán (dữ liệu bị mâu thuẫn) khi cập nhật dữ liệu.

Để tránh những nhược điểm do dư thừa dữ liệu gây ra, CSDL quan hệ thường được thiết kế gồm một số bảng, có bảng chứa dữ liệu về riêng một đối tượng (cá thể) cần quản lí, có bảng chứa dữ liệu về những sự kiện liên quan đến các đối tượng được quản lí. Ví dụ, ở một thư viện nhỏ, CSDL có thể gồm 3 bảng (Hình 2):

- Bảng SÁCH chứa dữ liệu về các quyển sách của thư viện.
- Bảng NGƯỜI ĐỌC chứa dữ liệu về những người đọc (có thẻ thư viện).
- Bảng MƯỢN-TRẢ chứa dữ liệu về sự việc một người mượn/trả một quyển sách, sự việc này liên quan đến hai đối tượng quản lí (một người đọc và một quyển sách).



Hình 2. Một CSDL gồm 3 bảng

Với cách tổ chức CSDL như trong ví dụ vừa nêu, mỗi bảng sẽ giảm được dữ liệu lặp lại, tránh thông tin dư thừa và việc cập nhật dữ liệu sẽ bớt được nhiều rủi ro sai nhầm.

2 Liên kết giữa các bảng và khoá ngoài

Để trích xuất thông tin từ CSDL quan hệ, ta có thể cần dữ liệu trong hơn một bảng và phải ghép nối đúng được dữ liệu giữa các bảng với nhau.

Ví dụ: Xét CSDL Thư viện gồm ba bảng như ở Hình 2 và yêu cầu “Cho biết *Họ và tên, Lớp* của những học sinh đã mượn quyển sách có mã TH-01”. Để trả lời yêu cầu này cần dữ liệu ở hai bảng (MUỖN-TRẢ và NGƯỜI ĐỌC). Chú ý rằng giá trị “HS-002” của *Số thẻ TV* trong bảng MUỖN-TRẢ đã “dẫn” ta đến (tham chiếu đến) một bản ghi trong bảng NGƯỜI ĐỌC chứa thông tin cần tìm. Thông qua thuộc tính *Số thẻ TV* mà hai bảng MUỖN-TRẢ và NGƯỜI ĐỌC có được mối liên kết với nhau: mỗi giá trị của *Số thẻ TV* xuất hiện trong MUỖN-TRẢ được giải thích chi tiết hơn trong NGƯỜI ĐỌC. Trong mỗi liên kết đó, bảng MUỖN-TRẢ được gọi là *bảng tham chiếu*, NGƯỜI ĐỌC là *bảng được tham chiếu* của mỗi liên kết. Tương tự, hai bảng MUỖN-TRẢ và SÁCH có mối liên kết với nhau qua thuộc tính *Mã sách*, bảng MUỖN-TRẢ là bảng tham chiếu và bảng SÁCH là bảng được tham chiếu.

Để tham chiếu xác định thì thuộc tính liên kết hai bảng phải là khoá của bảng được tham chiếu, trong ví dụ này *Số thẻ TV* phải là khoá chính của bảng NGƯỜI ĐỌC và còn được gọi là *khoá ngoài* của bảng MUỖN-TRẢ. Liên kết giữa hai bảng trong CSDL được thực hiện thông qua cặp khoá chính – khoá ngoài.



Khoá ngoài của một

bảng: một trường (hay một số trường) của bảng này và đồng thời là khoá của một bảng khác.

3 Hệ quản trị CSDL đảm bảo ràng buộc khoá ngoài



Hãy xét tình huống sau đây: CSDL Thư viện có bảng MUỖN-TRẢ liên kết với bảng NGƯỜI ĐỌC qua khoá ngoài *Số thẻ TV*. Hiện tại, bảng NGƯỜI ĐỌC có bốn bản ghi (ghi nhận dữ liệu về bốn học sinh đã làm thẻ thư viện). Người thủ thư đang muốn thêm một bản ghi cho bảng MUỖN-TRẢ (Hình 3). Theo em, cập nhật đó có hợp lí không? Giải thích vì sao?

Bảng MUỖN-TRẢ

Số thẻ TV	Mã sách	Ngày mượn	Ngày trả
HS-002	TH-01	14/10/2022	21/10/2022
HS-001	AN-01	02/10/2022	09/10/2022
HS-007	TO-01	16/9/2022	26/9/2022

Bảng NGƯỜI ĐỌC

Số thẻ TV	Họ và tên	Ngày sinh	Lớp
HS-001	Trần Văn An	14/9/2006	12A2
HS-002	Lê Bình	02/3/2007	11A1
HS-003	Hà Thị Mai	16/9/2007	11A2
HS-004	Nguyễn Lộc	05/10/2006	12A3

Hình 3. Dự kiến bổ sung một bản ghi vào bảng MUỖN-TRẢ

a) Ràng buộc khoá ngoài

Khi hai bảng trong một CSDL có liên kết với nhau, mỗi giá trị khoá ngoài ở bảng tham chiếu sẽ được giải thích chi tiết hơn ở bảng được tham chiếu. Ví dụ, “HS-001”

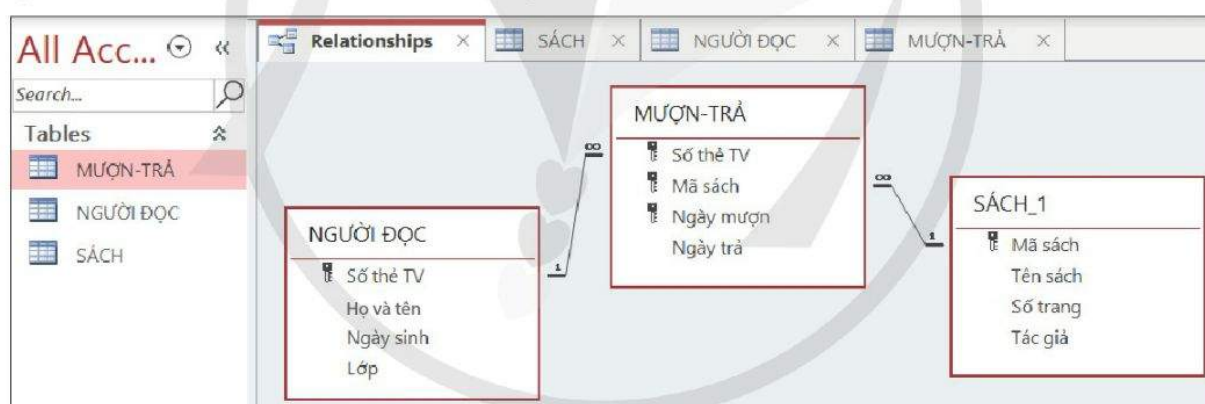
được giải thích bằng thông tin “Họ và tên: Trần Văn An; Ngày sinh: 14/9/2006; Lớp: 12A2”. Nếu có giá trị khoá ngoài nào không xuất hiện trong giá trị khoá ở bảng được tham chiếu thì xảy ra hiện tượng mất tham chiếu. Trong *Hình 3*, “HS-007” không xuất hiện trong *Số thẻ TV* của bảng NGƯỜI ĐỌC. Do vậy, việc bổ sung cho bảng MUỖN-TRẢ một bản ghi mới có giá trị khoá ngoài là “HS-007” sẽ làm cho dữ liệu trong CSDL không còn đúng đắn nữa, không giải thích được “HS-007” là số thẻ thư viện của ai. Muốn cập nhật đó hợp lệ, phải bổ sung bản ghi có giá trị khoá là “HS-007” vào bảng NGƯỜI ĐỌC trước.

Đảm bảo tính tham chiếu đầy đủ giữa các bảng có liên kết với nhau cũng là một phần của việc đảm bảo tính toàn vẹn của dữ liệu. Ràng buộc này áp dụng cho khoá ngoài nên được gọi là *ràng buộc khoá ngoài*. Nói một cách cụ thể hơn, *ràng buộc khoá ngoài* là yêu cầu mọi giá trị của khoá ngoài trong bảng tham chiếu phải xuất hiện trong giá trị khoá ở bảng được tham chiếu.

b) Khai báo liên kết giữa các bảng

Các hệ quản trị CSDL đều cho người tạo lập CSDL được khai báo liên kết giữa các bảng. Phần mềm quản trị CSDL sẽ căn cứ vào các liên kết đó để kiểm soát tất cả thao tác cập nhật, không để xảy ra những vi phạm ràng buộc khoá ngoài.

Hình 4 cho thấy kết quả trực quan của việc khai báo liên kết giữa 3 bảng khi dùng hệ quản trị CSDL Microsoft Access (phiên bản 365).



Hình 4. Kết quả của việc khai báo thành công hai liên kết giữa 3 bảng

4 Thực hành về bảng với khoá ngoài

Yêu cầu:

Khám phá cách khai báo liên kết giữa các bảng trong môi trường Access và nhận biết các cập nhật vi phạm ràng buộc khoá ngoài.

Hướng dẫn thực hiện:

Bước 1. Mở CSDL Thư viện đã có bảng SÁCH (kết quả mục thực hành ở Bài 2). Tạo cấu trúc như ở *Hình 2* cho bảng NGƯỜI ĐỌC và bảng MUỖN-TRẢ. Chọn *Số thẻ TV* làm khoá chính cho bảng NGƯỜI ĐỌC, chọn khoá chính của bảng MUỖN-TRẢ gồm ba thuộc tính: *Số thẻ TV*, *Mã sách* và *Ngày mượn*.

Bước 2. Khám phá cách khai báo liên kết giữa các bảng.

– Trong dải **Database Tools**, chọn **Relationships**.

– Dùng chuột kéo thả các bảng vào cửa sổ khai báo liên kết (vùng trống ở giữa).

– Dùng chuột kéo thả khoá ngoài của bảng tham chiếu thả vào khoá chính của bảng được tham chiếu, làm xuất hiện hộp thoại Edit Relationships.

– Đánh dấu hộp kiểm **Enforce Referential Integrity** và chọn **Create**.

Bước 3. Khám phá báo lỗi của phần mềm quản trị CSDL khi cập nhật vi phạm ràng buộc khoá ngoài.

– Thêm một vài bản ghi trong đó có bản ghi vi phạm lỗi ràng buộc khoá ngoài (tham khảo *Hình 3*), quan sát báo lỗi của phần mềm.

– Chọn xoá một bản ghi trong bảng NGƯỜI ĐỌC nếu giá trị *Số thẻ TV* trong bản ghi này xuất hiện trong bảng MUỐN-TRẢ, quan sát báo lỗi của phần mềm.



Trong việc tạo lập CSDL, sau khi tạo xong cấu trúc cho hai bảng mà ta dự kiến có liên kết với nhau bằng khoá ngoài, ta nên khai báo liên kết trước hay nên nhập dữ liệu cho hai bảng trước? Hãy giải thích vì sao.



Trong các câu sau, những câu nào đúng?

a) Một trường là khoá ngoài của một bảng nếu nó là khoá của bảng đó và đồng thời xuất hiện trong một bảng khác.

b) Khoá ngoài của một bảng là tập hợp một số trường của bảng đó và đồng thời là khoá của một bảng khác.

c) Khi hai bảng có liên kết với nhau qua khoá chính – khoá ngoài, chỉ khi bổ sung bản ghi vào các bảng mới cần thoả mãn ràng buộc khoá ngoài.

d) Các hệ quản trị CSDL quan hệ tự động kiểm tra và chỉ chấp nhận các cập nhật thoả mãn ràng buộc khoá ngoài.

Tóm tắt bài học

- ✓ CSDL quan hệ có thể gồm một số bảng, trong đó có những bảng có mối liên kết với nhau. Những liên kết này giúp tìm được những thông tin đúng đắn và đầy đủ.
- ✓ Nếu hai bảng có chung một trường và trường này là khoá của một trong hai bảng thì trường đó là khoá ngoài của bảng còn lại. Hai bảng có thể liên kết với nhau thông qua khoá ngoài.
- ✓ Dữ liệu trong hai bảng liên kết với nhau qua khoá ngoài cần phải thoả mãn ràng buộc khoá ngoài: Mọi giá trị khoá ngoài đều phải xuất hiện trong trường khoá ở bảng được tham chiếu. Mọi hệ quản trị CSDL quan hệ đều có cơ chế đảm bảo cập nhật dữ liệu không vi phạm ràng buộc khoá ngoài đối với các liên kết giữa các bảng.

BÀI 4

CÁC BIỂU MẪU CHO XEM VÀ CẬP NHẬP DỮ LIỆU

Học xong bài này, em sẽ:

- ✓ Diễn đạt được khái niệm biểu mẫu trong các CSDL và ứng dụng CSDL.
- ✓ Giải thích được những ưu điểm khi người dùng xem và cập nhật dữ liệu cho CSDL thông qua biểu mẫu.



Khi nhập dữ liệu vào một bảng của CSDL quan hệ, theo em có thể gặp những lỗi nào? Em hãy cho ví dụ.

1 Khái niệm và chức năng của biểu mẫu

a) Chức năng của biểu mẫu

Phục vụ cho hoạt động của một tổ chức nên một CSDL có thể được nhiều người cùng sử dụng. Thay vì cho phép người dùng tương tác trực tiếp với CSDL, các giao diện người dùng đã được thiết kế phù hợp với mỗi nhóm người làm việc với CSDL. Biểu mẫu là giao diện thuận tiện để người dùng tương tác với CSDL khi xem dữ liệu, hạn chế bớt lỗi, tránh vi phạm ràng buộc về dữ liệu khi cập nhật CSDL. Biểu mẫu được thiết kế nhằm các mục đích sau:

- Hiện thị dữ liệu trong bảng dưới dạng phù hợp để xem.
- Cung cấp một khuôn dạng thuận tiện để nhập và sửa dữ liệu.
- Cung cấp các nút lệnh để người dùng có thể sử dụng, thông qua đó thực hiện một số thao tác với dữ liệu.

Phổ biến nhất là các biểu mẫu hiển thị dữ liệu cho từng nhóm người dùng và biểu mẫu cho người nhập dữ liệu.

b) Tạo biểu mẫu

Các hệ quản trị CSDL thường cung cấp các công cụ để tạo được biểu mẫu cho người dùng CSDL. Muốn nhanh chóng có được biểu mẫu theo ý mình, ta có thể dùng công cụ thiết kế biểu mẫu tự động, sau đó điều chỉnh thêm để có một biểu mẫu thân thiện hơn, thuận tiện hơn trong sử dụng. Những ứng dụng CSDL đơn giản sử dụng các biểu mẫu được tạo ra theo cách này. Trong khi đó, ở những ứng dụng CSDL lớn và phức tạp, (thường là những phần mềm được xây dựng trên nền hệ quản trị CSDL), các biểu mẫu như một thành phần của phần mềm ứng dụng được tạo ra nhờ một ngôn ngữ lập trình.

2 Biểu mẫu cho xem dữ liệu

Các hệ quản trị CSDL quan hệ thường cung cấp công cụ tạo lập nhanh chóng những biểu mẫu cho xem dữ liệu. Những biểu mẫu loại này không cho người xem sửa đổi dữ liệu. Việc thiết kế những biểu mẫu như vậy là để hỗ trợ cho những nhóm người dùng tra cứu thông tin của CSDL trong phạm vi được phép:

- Biểu mẫu chỉ hiển thị dữ liệu người dùng cần hoặc phần dữ liệu được phép xem. Có thể thiết kế biểu mẫu hiển thị chỉ một phần của dữ liệu trong bảng.
- Biểu mẫu hiển thị các bản ghi theo thứ tự sắp xếp của một trường nào đó.
- Biểu mẫu cho xem dữ liệu được lọc theo một tiêu chí nào đó và có thể lọc dần nhiều bước.

ĐỊA CHỈ LIÊN LẠC CỦA HỌC SINH LỚP 11				
Mã định danh	Họ và tên	Giới tính	Địa chỉ	
13109413	Phan Thùy Anh	Nữ	39 Hùng Vương	
13109735	Lê Minh Đức	Nam	15 Văn Cao	
13124595	Hoàng Giang	Nam	27 Lò Sũ	
13126236	Đặng Phương	Nam	148 Hàng Gà	

Record: 1 of 16

Thanh trượt ngang

Thanh trượt dọc

Hình 1. Một biểu mẫu cho xem dữ liệu ở một số trường của một bảng

Các biểu mẫu minh họa trong bài đều được tạo ra bởi hệ quản trị CSDL Microsoft Access 365.

Biểu mẫu như ở Hình 1 chỉ hiển thị một số trường của bảng dữ liệu nguồn THÔNG TIN HỌC SINH LỚP 11 (không hiển thị điểm các môn học). Các thanh trượt dọc và ngang được dùng để xem những dữ liệu bị khuất trong cửa sổ biểu mẫu. Các nút ► ◀ được dùng để chuyển đến xem bản ghi đứng trước hoặc đứng sau bản ghi hiện thời. Có thể chỉ hiển thị danh sách các bản ghi thỏa mãn điều kiện nào đó (ví dụ xem danh sách học sinh là Đoàn viên) bằng cách sử dụng chức năng lọc bản ghi theo điều kiện. Người dùng biểu mẫu có thể thay đổi các điều kiện lọc, điều kiện sắp xếp ngay trên biểu mẫu để xem được dữ liệu tương ứng.

Biểu mẫu cũng có thể hiển thị các trường từ nhiều bảng khác nhau. Hình 2 cho thấy dữ liệu trong biểu mẫu lấy từ 3 bảng SÁCH, NGƯỜI ĐỌC và MUỐN-TRẢ của CSDL Thư viện trong ví dụ đã nêu.

THEO DÕI SÁCH MƯỢN						
Số thẻ TV	Họ và tên	Mã sách	Tên sách	Ngày mượn	Ngày trả	
HS-002	Lê Bình	TH-01	AI-Trí tuệ nhân tạo	14/10/2022	21/10/2022	
HS-001	Trần Văn An	AN-01	Âm nhạc ở quanh ta	02/10/2022	9/10/2022	
HS-003	Hà Thị Mai	AN-01	Âm nhạc ở quanh ta	14/10/2022		
HS-001	Trần Văn An	TH-02	AI-Trí tuệ nhân tạo	15/10/2022	27/10/2022	
HS-003	Hà Thị Mai	TO-01	Sáng tạo Toán học	15/10/2022	25/10/2022	

Hình 2. Một biểu mẫu của CSDL Thư viện

3 Biểu mẫu cho cập nhật dữ liệu



Theo em, có những bất lợi nào trong việc mở một bảng của CSDL quan hệ rồi trực tiếp cập nhật dữ liệu (thêm bản ghi, sửa các bản ghi trong đó)?

Các hệ quản trị CSDL quan hệ cũng thường cung cấp công cụ cho phép tạo lập nhanh chóng những biểu mẫu cập nhật dữ liệu. Những biểu mẫu loại này có các ô nhập dữ liệu còn để trống hoặc chứa dữ liệu đã có nhưng cho phép sửa đổi. Các ô và nhãn đi kèm được bố trí hợp lý cho việc xem và thực hiện thao tác cập nhật.

Việc thiết kế những biểu mẫu như vậy giúp việc cập nhật dữ liệu được tiện lợi hơn, hạn chế được những sai nhầm khi cập nhật:

– Tránh được các cập nhật vi phạm ràng buộc toàn vẹn như ràng buộc khoá, ràng buộc khoá ngoài.

– Tránh được các cập nhật vi phạm ràng buộc miền giá trị, tức là không đưa vào giá trị nằm ngoài tập giá trị được chấp nhận.

Ví dụ 1. Biểu mẫu ở Hình 3 dùng để nhập dữ liệu. Dữ liệu của các trường ở nửa bên trên biểu mẫu đó (*Mã định danh*,... *Giới tính*) được hiển thị và bị khoá lại không cho thay đổi.

NHẬP ĐIỂM HỌC SINH LỚP 11			
Mã định danh	13109413	Họ và tên	Phan Thuỷ Anh
Ngày sinh	29/10/2007	Giới tính	NỮ
Đoàn viên	☒		
Điểm Toán	7.3	Điểm Hoá học	8.8
Điểm Ngữ văn	7.4	Điểm Sinh học	7.5
Điểm Ngoại ngữ 1	8.5	Điểm Tin học	8.5
Điểm Lịch sử	9.5		

Hình 3. Một biểu mẫu nhập điểm cho học sinh

Có thể thiết kế biểu mẫu dùng để cập nhật dữ liệu cho bảng MUON-TRẢ và tránh được vi phạm ràng buộc khoá ngoài. Hình 4a cho thấy biểu mẫu cập nhật được thiết kế để Số thẻ TV của người mượn (hay người trả) không thể gõ nhập vào mà chỉ lựa chọn trong một danh sách thả xuống. Hình 4b cho thấy biểu mẫu cho phép nhập dữ liệu Ngày mượn, Ngày trả theo cách mở lịch và chọn ngày trên đó.

Hình 4a. Biểu mẫu cho chọn Số thẻ thư viện trong danh sách

Hình 4b. Biểu mẫu cho chọn dữ liệu kiểu ngày tháng trong lịch

Ví dụ 2. Một số CSDL trực tuyến cũng có các biểu mẫu cho sẵn phục vụ người dùng, như biểu mẫu khai báo y tế mà người dân có thể điền thông tin trên điện thoại di động.

4 Thực hành tạo biểu mẫu và cập nhật dữ liệu

Nhiệm vụ 1. Thầy, cô giáo đã dựng sẵn 3 bảng: SÁCH, NGƯỜI ĐỌC, MUON-TRẢ cùng một vài biểu mẫu trong CSDL Thư viện (tạo bằng Access). Em hãy sử dụng biểu mẫu NHẬP DỮ LIỆU MUON-TRẢ SÁCH đã có để nhập 3 bản ghi mới cho bảng MUON-TRẢ.

Hướng dẫn thực hiện:

Bước 1. Kích hoạt Microsoft Access.

Bước 2. Mở CSDL Thư viện, chọn biểu mẫu NHẬP DỮ LIỆU MUON-TRẢ SÁCH.

Bước 3. Trên biểu mẫu vừa mở, hãy nhập ít nhất 3 bản ghi.

Bước 4. Tìm và mở biểu mẫu XEM THÔNG TIN MUON-TRẢ SÁCH để kiểm tra xem những bản ghi nhập vào ở Bước 3 đã xuất hiện trong bảng MUON-TRẢ chưa.

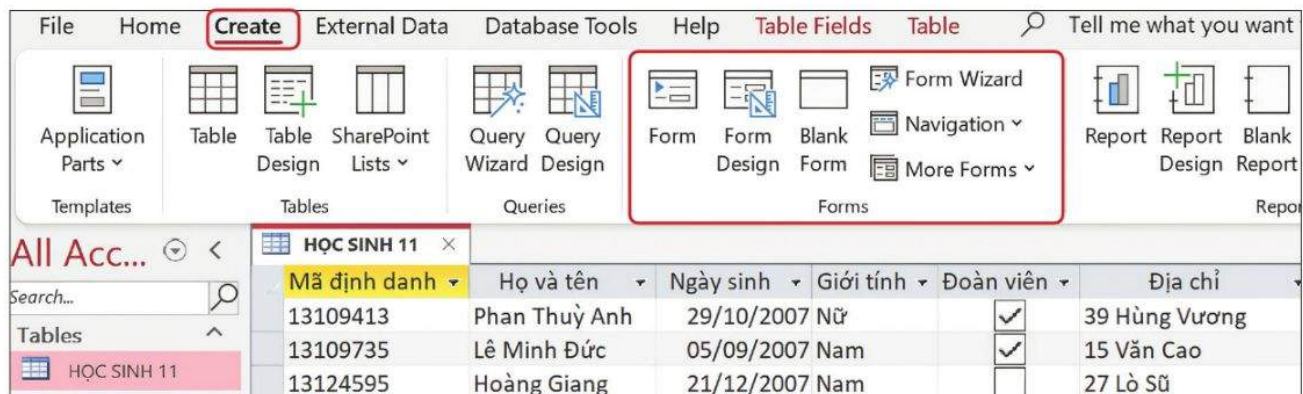
Bước 5. Kết thúc phiên làm việc với CSDL Thư viện, trong bảng chọn **File** chọn nút lệnh **Close** để đóng CSDL này.

Nhiệm vụ 2. Khám phá cách dùng công cụ tạo biểu mẫu trong Access.

Hướng dẫn thực hiện:

Bước 1. Chọn mở CSDL HỌC SINH 11. Mở bảng HỌC SINH 11.

Bước 2. Nháy chuột vào **Create** để xuất hiện các công cụ tạo lập, trong đó có các công cụ tạo lập biểu mẫu (Hình 5).



Hình 5. Các công cụ tạo lập biểu mẫu trong Access

Bước 3. Chọn và khám phá công cụ **Form Wizard**: Chọn các trường cho biểu mẫu, kiểu cho biểu mẫu, đặt tên biểu mẫu, chọn **Finish**.

Bước 4. Đóng CSDL HỌC SINH 11 để kết thúc phiên làm việc với CSDL này.



Nếu là người xây dựng một CSDL quản lý học sinh khối 11 của trường mình, em sẽ xây dựng những biểu mẫu nào? Mỗi biểu mẫu em định thiết kế sẽ có chức năng nào và đem lại thuận lợi gì, cho ai?



Trong các câu sau, những câu nào đúng?

- Mỗi biểu mẫu đều được dùng chung cho tất cả mọi người sử dụng CSDL.
- Mỗi biểu mẫu là một cửa sổ cho người dùng xem toàn bộ thông tin trong một bảng của CSDL.
- Khi cập nhật dữ liệu, cần sử dụng biểu mẫu vì có thể đảm bảo được ràng buộc khoá và khoá ngoài, tránh được nhiều sai nhầm về dữ liệu.
- Biểu mẫu là một giao diện được thiết kế để kiểm soát các truy cập của người dùng đến dữ liệu trong CSDL.

Tóm tắt bài học

- ✓ Biểu mẫu là một loại giao diện cho người dùng CSDL tương tác với dữ liệu nguồn trong việc xem và cập nhật dữ liệu.
- ✓ Biểu mẫu đem lại sự thuận tiện cho các nhóm người dùng làm việc với CSDL và giúp hạn chế những vi phạm trong cập nhật nhằm tăng cường sự đảm bảo tính đúng đắn của dữ liệu.

Học xong bài này, em sẽ:

- ✓ Diễn đạt được khái niệm truy vấn CSDL.
- ✓ Giải thích được cấu trúc cơ bản **SELECT...FROM...WHERE...** của câu lệnh SQL.
- ✓ Nêu được một vài ví dụ minh họa việc dùng truy vấn để tổng hợp, tìm kiếm dữ liệu trên một bảng.



Em hãy nêu một vài ví dụ cụ thể về khai thác thông tin trong một CSDL mà em biết.

1) Khái niệm truy vấn CSDL

Truy vấn CSDL (Query) là một phát biểu thể hiện yêu cầu của người dùng đối với CSDL. Đó có thể là yêu cầu thao tác trên dữ liệu như: thêm, sửa, xóa bản ghi,... Đó cũng có thể là yêu cầu khai thác CSDL. Dù đơn giản hay phức tạp thì bản chất việc khai thác một CSDL là tìm kiếm dữ liệu đã lưu giữ trong đó và hiển thị kết quả theo khuôn dạng thuận lợi cho người khai thác.

Để máy tính có thể hiểu và thực thi được yêu cầu của người dùng, truy vấn phải được viết theo một số quy tắc của hệ quản trị CSDL. Nói cách khác, mỗi hệ quản trị CSDL có ngôn ngữ truy vấn của nó. Đối với các hệ quản trị CSDL quan hệ, ngôn ngữ truy vấn phổ biến nhất và nổi tiếng nhất cho đến nay là SQL (Structured Query Language). Hầu hết các hệ quản trị CSDL quan hệ, ngay cả những hệ thống có ngôn ngữ của riêng chúng, đều hỗ trợ một số phiên bản SQL. Bài học này tập trung vào việc dùng SQL thể hiện yêu cầu tìm và trích rút dữ liệu trong CSDL.

Chẳng hạn, giáo viên chủ nhiệm cần danh sách những học sinh của lớp có điểm tổng kết môn Tin học từ 8,0 trở lên. Tùy theo hệ quản trị CSDL, có thể có những truy vấn tóm tắt dữ liệu và thực hiện một số phép tính trên dữ liệu để đưa ra kết quả. Với những truy vấn như vậy, kết quả trả ra có thể là hình ảnh, đồ thị, ví dụ như kết quả của yêu cầu phân tích xu hướng mua/bán một mặt hàng trong 6 tháng đầu năm của một công ty thương mại.

2 Khai thác CSDL bằng câu truy vấn SQL đơn giản



Em hãy quan sát mẫu câu truy vấn ở *Hình 1a* dùng để tìm dữ liệu trong CSDL và một ví dụ truy vấn ở *Hình 1b*. Muốn tìm Họ và tên, Ngày sinh, điểm môn Toán và điểm môn Ngữ văn của những học sinh có điểm môn Toán trên 7.0 thì em sẽ dùng câu truy vấn SQL như thế nào?

Cấu trúc cơ bản của một câu truy vấn viết bằng ngôn ngữ SQL như ở *Hình 1a*:

SELECT <  >	Tên các trường dữ liệu cần đưa ra kết quả
FROM <  >	Tên bảng trong CSDL được truy cập để lấy dữ liệu
WHERE <  >	Biểu thức logic chọn các bản ghi đưa ra kết quả

SELECT [Mã định danh], [Họ và tên], [Toán], [Ngữ văn]
FROM [HỌC SINH 11]
WHERE [Ngữ văn] >= 7.0

Hình 1a. Mẫu câu truy vấn thông dụng của SQL

Hình 1b. Một câu truy vấn SQL

Để có kết quả của câu truy vấn, hệ quản trị CSDL sẽ truy cập vào các bảng dữ liệu có tên được chỉ ra sau **FROM**. Các bản ghi thỏa mãn điều kiện tìm kiếm đứng sau **WHERE** sẽ được lựa chọn. Kết quả câu truy vấn là những bản ghi đã được lựa chọn và chỉ giá trị của những trường có tên đứng sau **SELECT** mới được hiển thị.

Chú ý: Khi thực hiện các câu truy vấn, hệ quản trị CSDL sẽ coi tên trường là biến trong chương trình xử lý, do vậy, nếu tên trường có chứa dấu cách thì cần phải dùng các dấu [] để đánh dấu bắt đầu và kết thúc tên trường.

Để dễ theo dõi các ví dụ về câu truy vấn trong mục này, CSDL nói đến ở các ví dụ có bảng HỌC SINH 11 với dữ liệu như ở *Hình 2*.

Mã định danh	Họ và tên	Ngày sinh	Giới tính	Đoàn viên	Địa chỉ	Toán	Ngữ văn	...	Tin học
13109413	Phan Thuỳ Anh	29/10/2007	Nữ	x	39 Hùng Vương	7.3	7.4	...	8.5
13109735	Lê Minh Đức	05/9/2007	Nam	x	15 Văn Cao	6.4	7.2	...	7.0
13124595	Hoàng Giang	21/12/2007	Nam		27 Lò Sũ	7.7	7.6	...	9.3
13126236	Đặng Phương	21/01/2007	Nam		148 Hàng Gà	8.5	6.8	...	9.0
13146782	Nguyễn Minh Trí	03/12/2007	Nam	x	37 Chu Văn An	9.0	7.0	...	7.5
13169292	Trần Minh Tú	14/11/2007	Nữ	x	18 Quán Thánh	7.8	6.5	...	7.7

Hình 2. Bảng HỌC SINH 11

Ví dụ 1. Để tìm Mã định danh, Họ và tên, điểm môn Toán và điểm môn Ngữ văn của những học sinh có điểm môn Ngữ văn từ 7.0 trở lên thì cần dùng truy vấn SQL như trong *Hình 1b*. Kết quả nhận được từ truy vấn đó sẽ như trong *Hình 3*.

Mã định danh	Họ và tên	Toán	Ngữ văn
13109413	Phan Thuỳ Anh	7.3	7.4
13109735	Lê Minh Đức	6.4	7.2
13124595	Hoàng Giang	7.7	7.6
13146782	Nguyễn Minh Trí	9.0	7.0

Hình 3. Kết quả của câu truy vấn ở Hình 1b với bảng HỌC SINH 11

3 Ngôn ngữ truy vấn QBE

Có những hệ quản trị CSDL cho phép truy vấn bằng cách điền vào chỗ trống trong một bảng, như thể hiện một ví dụ về kết quả cần nhận được (nên ngôn ngữ truy vấn này là Query By Example – QBE). Access là một hệ quản trị CSDL cho truy vấn bằng cả SQL và QBE.

Ví dụ 2. Tương ứng với câu truy vấn SQL ở *Hình 1b*, ta có thể điền vào bảng thiết kế QBE của Access như ở *Hình 4* dưới đây:

Những bản ghi có giá trị trường Ngữ văn ≥ 7.0 mới được đưa vào kết quả của truy vấn

Field:	[Mã định danh]	[Họ và tên]	[Ngày sinh]	[Toán]	[Ngữ Văn]
Table:	HỌC SINH 11	HỌC SINH 11	HỌC SINH 11	HỌC SINH 11	HỌC SINH 11
Sort:					
Show:	☒	☒	☐	☒	☒
Criteria:					≥ 7.0
or:					

Cho biết trường Mã định danh xuất hiện trong kết quả truy vấn

Cho biết trường Ngày sinh không xuất hiện trong kết quả truy vấn

Hình 4. Một truy vấn trên bảng QBE



Câu 1. Hãy viết câu truy vấn SQL để tìm điểm môn Ngữ văn của những học sinh là Đoàn viên trong bảng HỌC SINH 11 (*Hình 2*). Kết quả của câu truy vấn là gì?

Câu 2. Hình bên là một câu truy vấn SQL được viết để tìm dữ liệu trong CSDL Thư viện (*Hình 2 Bài 3*). Theo em, người viết truy vấn đó muốn tìm biết gì?

```
SELECT [Mã sách], [Tên sách], [Số trang]
FROM SÁCH
WHERE [Tác giả] = "Nguyễn Nhật Ánh"
```




Hãy nêu một yêu cầu tìm thông tin trong bảng HỌC SINH 11 (Hình 2) và viết câu truy vấn SQL để có được thông tin cần tìm.



Trong các câu sau, những câu nào đúng?

- a) Truy vấn CSDL là một biểu mẫu.
- b) Có thể dùng các câu truy vấn để tìm kiếm dữ liệu trong CSDL.
- c) SQL là ngôn ngữ truy vấn thường được dùng trong các hệ CSDL quan hệ.
- d) Trong câu truy vấn SQL, sau từ khoá **FROM** là tên của bảng dữ liệu nguồn cho các trích xuất dữ liệu.

Tóm tắt bài học

- ✓ Đối với các hệ CSDL quan hệ, có hai loại truy vấn dữ liệu: truy vấn cập nhật dữ liệu và truy vấn khai thác dữ liệu.
- ✓ Ngôn ngữ truy vấn phổ biến nhất trong các hệ quản trị CSDL quan hệ là SQL. Câu truy vấn khai thác dữ liệu của SQL có cấu trúc cơ bản là **SELECT...FROM...WHERE...**
- ✓ Mệnh đề **SELECT** xác định thông tin ta muốn hiển thị; mệnh đề **FROM** xác định dữ liệu được lấy từ đâu; mệnh đề **WHERE** xác định điều kiện lọc dữ liệu.
- ✓ Trong một số hệ quản trị CSDL, truy vấn còn có thể được thể hiện bằng ngôn ngữ QBE.

BÀI ĐỌC THÊM

VÀI NÉT VỀ CSDL NoSQL

Vào cuối những năm 2000 xuất hiện các CSDL NoSQL. Một số người quen dùng truy vấn SQL trong các hệ CSDL quan hệ có thể suy đoán rằng CSDL NoSQL là CSDL không dùng truy vấn SQL. Thực ra “NoSQL” trong tên gọi đó nên được hiểu là “Không chỉ SQL” (Not Only SQL). Các hệ CSDL NoSQL ra đời là để lưu trữ và xử lý lượng dữ liệu tăng rất nhanh trong nhiều ứng dụng, đặc biệt là các ứng dụng web. Khác với các CSDL quan hệ truyền thống, CSDL NoSQL hỗ trợ nhiều kiểu lưu trữ dữ liệu khác nhau tùy thuộc vào ứng dụng cụ thể thay vì sử dụng cấu trúc chặt chẽ dạng bảng. Một số hệ CSDL NoSQL có thể sử dụng cú pháp giống SQL để làm việc với dữ liệu nhưng chỉ ở một mức độ hạn chế. CSDL NoSQL nổi lộng rằng buộc và tính nhất quán dữ liệu để đạt tốc độ nhanh trong phục vụ, tính linh hoạt cũng như khả năng mở rộng quy mô phục vụ. CSDL quan hệ và CSDL NoSQL, mỗi loại có cách lưu trữ và truy xuất dữ liệu khác nhau, được thiết kế để giải quyết các loại nhu cầu khác nhau do các ứng dụng CSDL đòi hỏi.

BÀI 6**TRUY VẤN TRONG CƠ SỞ DỮ LIỆU QUAN HỆ**

(tiếp theo)

Học xong bài này, em sẽ:

Đưa ra được một vài ví dụ minh họa cho việc dùng truy vấn để tổng hợp, tìm kiếm dữ liệu trên hơn một bảng.



Theo em, việc khai báo liên kết giữa một số bảng trong một CSDL quan hệ có ý nghĩa gì?

1 Câu lệnh truy vấn SQL với liên kết các bảng

**1**

Xét CSDL được mô tả như ở *Hình 1*. Nếu cần biết tên quyển sách mà người có thẻ thư viện HS-001 đã mượn vào ngày 02/10/2022, ta có thể dùng câu truy vấn trên một bảng được không? Nếu tìm thông tin này bằng cách tra cứu thủ công (không dùng máy tính) thì em sẽ làm như thế nào?

Bảng MƯỢN-TRẢ

ID	Số thẻ TV	Mã sách	Ngày mượn	Ngày trả
1	HS-001	AN-01	02/10/2022	09/10/2022
2	HS-002	TH-01	14/10/2022	21/10/2022
3	HS-003	AN-01	14/10/2022	
4	HS-003	TO-01	15/10/2022	25/10/2022
5	HS-001	TH-02	15/10/2022	27/10/2022

Bảng NGƯỜI ĐỌC

Số thẻ TV	Họ và tên	Ngày sinh	Lớp
HS-001	Trần Văn An	14/9/2006	12A2
HS-002	Lê Bình	02/3/2007	11A1
HS-003	Hà Thị Mai	16/9/2007	11A2
...	...	...	...

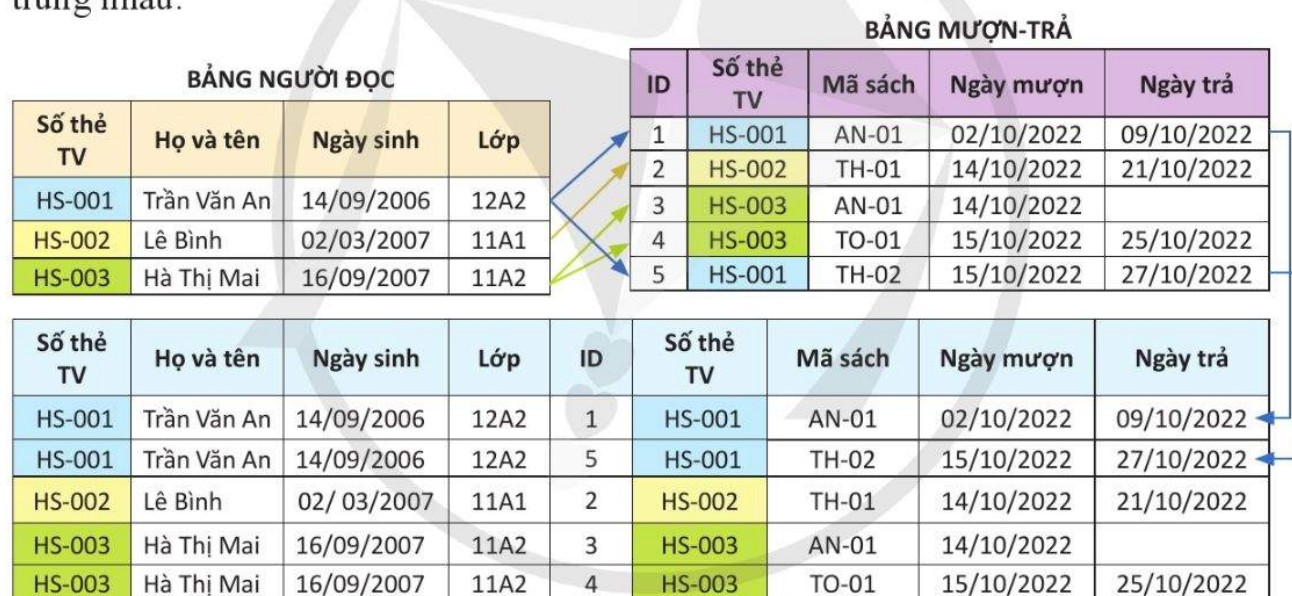
Bảng SÁCH

Mã sách	Tên sách	Số trang	Tác giả
AN-01	Âm nhạc ở quanh ta	184	Phạm Tuyên
TO-01	Sáng tạo Toán học	308	Polya
TH-01	AI – Trí tuệ nhân tạo	306	Lasse Rouhiaine
TH-02	AI – Trí tuệ nhân tạo	306	Lasse Rouhiaine
...	...	...	...

Hình 1. Mối quan hệ giữa ba bảng trong CSDL Thư viện

Khi khai thác CSDL quan hệ, nhiều tình huống cần phải kết hợp dữ liệu ở hai hoặc nhiều bảng để đưa ra được dữ liệu cần tìm. Kiểu kết hợp thường gặp là ghép nối một bản ghi của bảng này với một hay nhiều bản ghi của bảng khác tạo nên một hay nhiều bản ghi mới đầy đủ thông tin hơn. Kết quả của các ghép nối là các bản ghi mới được đưa vào một bảng tạm thời. Hệ quản trị CSDL sẽ chọn lựa trong bảng tạm thời này những dữ liệu thoả mãn điều kiện tìm để đưa ra kết quả. Chẳng hạn, để tìm *Mã sách* của những quyển sách mà học sinh Trần Văn An đã mượn, hệ quản trị CSDL cần kết hợp dữ liệu ở bảng NGƯỜI ĐỌC với dữ liệu ở bảng MUỐN-TRẢ. Mục đích kết hợp dữ liệu của hai bảng này là để có một bảng dữ liệu tạm thời mà mỗi bản ghi của nó cho ta dữ liệu đúng đắn và đầy đủ gồm *Họ và tên*, *Số thẻ TV*,... của một người đọc cùng với *Mã sách* của quyển sách họ mượn và ngày mượn,... (Hình 2).

Khi kết hợp dữ liệu, hai bản ghi thuộc hai bảng khác nhau trong CSDL chỉ được ghép lại nếu chúng thoả mãn một điều kiện mà ta gọi là *điều kiện kết nối*. Trong tình huống nêu trên, điều kiện kết nối là giá trị ở trường *Số thẻ TV* của hai bản ghi đó phải trùng nhau.



Hình 2. Kết nối bảng NGƯỜI ĐỌC và bảng MUỐN-TRẢ với điều kiện *Số thẻ TV* trùng nhau

Việc trích rút dữ liệu từ nhiều bảng khác nhau được thực hiện như những truy vấn trên một bảng dữ liệu, đó là bảng dữ liệu tạm thời chứa kết quả kết nối các bản ghi. Trong trường hợp nói đến ở Hình 2, hệ quản trị CSDL chỉ việc lựa chọn dữ liệu trong bảng kết quả kết nối đó để đưa ra “Trần Văn An” đã mượn quyển sách có mã sách “AN-01” và quyển sách có mã sách “TH-02”.

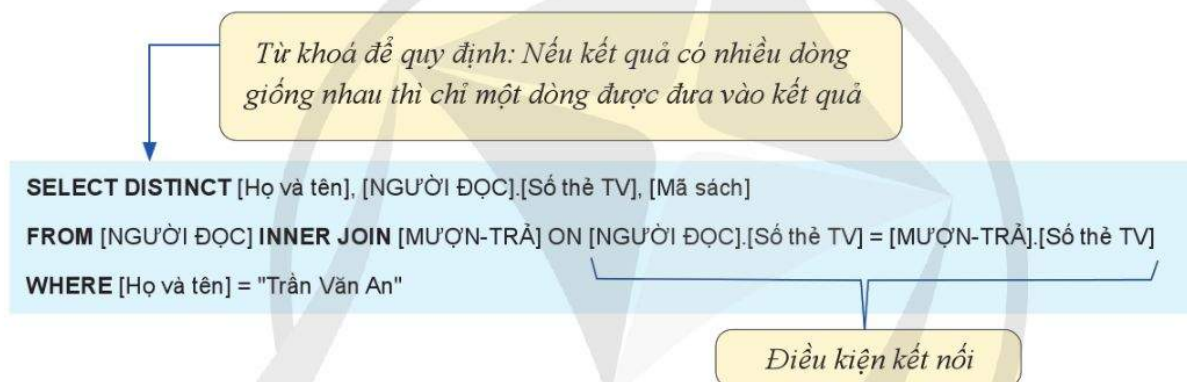
Để kết hợp dữ liệu từ các bảng có trường chung theo cách ghép nối các bản ghi thoả mãn một điều kiện nào đó, SQL sử dụng từ khoá **JOIN** trong mệnh đề **FROM**. Có một số kiểu **JOIN** khác nhau, trong đó **INNER JOIN** được dùng phổ biến nhất. Dưới đây là mẫu viết mệnh đề **FROM** (trong câu truy vấn) sử dụng **INNER JOIN**.

FROM *bảng1* **INNER JOIN** *bảng2* **ON** *bảng1.TrườngA* $\emptyset$ *bảng2.TrườngB*

Điều kiện kết nối 2 bản ghi

Trong mẫu nêu trên, kí hiệu $\emptyset$ để chỉ bất cứ toán tử so sánh nào: =, <, <=, <>, >, >= (trong đó kí hiệu <> thể hiện toán tử so sánh khác). Tuy nhiên, trên thực tế **INNER JOIN** được dùng phổ biến với điều kiện kết nối là sự trùng khớp giá trị trên một trường chung của hai bảng kết nối.

Ví dụ 1. Trong *Hình 3* là một câu truy vấn dùng kết nối hai bảng. Mệnh đề **FROM** yêu cầu kết nối hai bản ghi: một ở bảng NGƯỜI ĐỌC và một ở bảng MUỐN-TRẢ. Điều kiện để hai bản ghi được kết nối là giá trị trường *Số thẻ TV* của chúng bằng nhau. Câu truy vấn SQL này được dùng để tìm mã sách của các quyển sách mà học sinh “Trần Văn An” đã mượn. Thông tin đưa ra gồm có thông tin về Trần Văn An (gồm *Họ và tên*, *Số thẻ TV*) và *Mã sách* của các cuốn sách đã mượn.



Hình 3. Một câu truy vấn dùng kết nối hai bảng

Mỗi giá trị khoá (một *Số thẻ TV*) chỉ xuất hiện trong một bản ghi duy nhất ở bảng NGƯỜI ĐỌC nhưng có thể xuất hiện trong nhiều bản ghi ở bảng MUỐN-TRẢ. Do vậy, ta nói quan hệ giữa NGƯỜI ĐỌC và MUỐN-TRẢ là quan hệ *một – nhiều*, ý nói *một* bản ghi trong bảng thứ nhất tương ứng với *nhiều* bản ghi trong bảng thứ hai và một bản ghi trong bảng thứ hai chỉ tương ứng với một bản ghi trong bảng thứ nhất.

Chú ý: Từ khoá **INNER JOIN** nằm giữa tên hai bảng nguồn cho kết nối và từ khoá **ON** đứng ngay trước điều kiện kết nối.

2 Kết xuất thông tin bằng báo cáo



2

Em đã biết, có thể truy vấn CSDL Quản lí học tập 11 để có được thông tin về kết quả học tập của học sinh lớp 11 ở một số môn học. Theo em, với công cụ truy vấn ta có được dữ liệu trình bày như ở *Hình 4* hay không?

Báo cáo CSDL là một văn bản trình bày thông tin kết xuất từ CSDL, có thể xem trực tiếp trên màn hình hoặc in ra. Dữ liệu để đưa vào báo cáo được lấy từ một hay nhiều bảng và truy vấn. Báo cáo trình bày dữ liệu trực quan, làm nổi bật những mục quan trọng và thường theo mẫu quy định. Vì lẽ đó nên nhu cầu xem báo cáo trong công tác quản lý rất lớn.

Hình 4 là một báo cáo có được từ CSDL của một trường trung học phổ thông ở các ví dụ đã từng nêu. Để có kết quả học tập của học sinh ở một số môn học, ta có thể dùng truy vấn CSDL. Tuy nhiên, dùng hình thức báo cáo thì việc trình bày những thông tin kết xuất được sẽ đạt hiệu quả cao hơn, phù hợp hơn với người cần những thông tin này.

KẾT QUẢ MỘT SỐ MÔN HỌC		
Họ và tên	Môn học	Điểm
Hồ Quốc Trung	Toán	7.8
	Ngữ văn	6.5
	Ngoại ngữ	7.0
	Tin học	7.7
Hoàng Giang	Toán	7.7
	Ngữ văn	7.6
	Ngoại ngữ	8.8
	Tin học	9.3
Lê Minh Đức	Toán	6.4
	Ngữ văn	7.2
	Ngoại ngữ	7.0
	Tin học	7.0
Nguyễn Thị Hà	Toán	7.1
	Ngữ văn	6.5
	Ngoại ngữ	7.2
	Tin học	8.4

Hình 4. Một báo cáo kết quả học tập

Báo cáo-So sánh số lượng và tổng tiền bán được theo mặt hàng (Nửa đầu tháng 2 năm 2022)			
Mã mặt hàng	Tên mặt hàng	Tổng số lượng bán	Tổng số tiền
3	Vở Thanh Trúc	375	2,062,500.00
4	Vở Hoàng Hà	175	1,050,000.00
2	Bút bi Hồng Hải	250	750,000.00
11	Bút chì	220	660,000.00
5	Vở Hướng Dương	100	650,000.00
1	Bút bi Thiên Nga	120	540,000.00
9	Bút chì 2B	60	300,000.00
6	Thuốc kẻ	75	262,500.00
7	Tẩy	65	195,000.00
10	Bút chì 3B	20	140,000.00
8	Ghim	20	70,000.00
12	Bút xoá	50	50,000.00

Báo cáo lập ngày 16/02/2022

Page 1 of 1

Hình 5. Một báo cáo tổng hợp thông tin theo mặt hàng

Quan sát một báo cáo ở Hình 5, người xem dễ dàng so sánh thực tế mua bán giữa các mặt hàng vì dữ liệu về mỗi mặt hàng được tổng hợp (tính tổng) ở số lượng đã bán và tiền thu được. Rất hữu ích với người làm kinh doanh khi sắp xếp các mặt hàng trong báo cáo theo thứ tự giảm dần của tổng tiền thu được.

Bên cạnh các công cụ tạo biểu mẫu, tạo truy vấn, các hệ quản trị CSDL quan hệ đều cung cấp công cụ tạo báo cáo tự động. Người phát triển ứng dụng CSDL có thể sử dụng công cụ tạo báo cáo tự động rồi tiếp tục chỉnh sửa bố cục, định dạng dữ liệu của báo cáo. Với những ứng dụng CSDL, người phát triển ứng dụng có thể dùng ngôn ngữ lập trình để thiết kế các báo cáo phù hợp với nhu cầu người dùng.

3 Thực hành truy vấn trong CSDL quan hệ

Trong CSDL Thư viện được tạo bởi hệ quản trị CSDL Access, giáo viên đã chuẩn bị sẵn một số truy vấn.

a) Em hãy mở xem một truy vấn và chạy thử để biết kết quả.

b) Trong các truy vấn được thiết kế sẵn, em hãy cho biết câu truy vấn nào trả lời cho câu hỏi: Các quyển sách “AI-Trí tuệ nhân tạo” đã được những người nào mượn đọc? Truy vấn đó kết nối những bảng nào của cơ sở dữ liệu? Vì sao em biết điều đó?



Xét CSDL được mô tả như ở *Hình 1*. Nếu cần biết tên cuốn sách đã được mượn với ID = 1 trong bảng MUỐN-TRẢ, em sẽ viết câu truy vấn như thế nào?



Trong các câu sau, những câu nào đúng?

a) Chỉ có thể viết câu truy vấn SQL trên một bảng của CSDL.

b) Các từ khoá kết nối phải viết trong mệnh đề **FROM** của câu truy vấn SQL.

c) Chỉ có thể kết nối với điều kiện giá trị ở trường chung giữa hai bảng là bằng nhau.

d) Dữ liệu để đưa vào báo cáo được lấy từ một hay nhiều bảng và truy vấn.

Tóm tắt bài học

- ✓ Để trích rút dữ liệu trong một CSDL quan hệ, có những truy vấn phải thực hiện kết nối dữ liệu của các bảng.
- ✓ Mệnh đề **FROM** có thể chứa từ khoá chỉ định kiểu **JOIN** để thực hiện kết nối các bản ghi ở các bảng khác nhau. **INNER JOIN** là một kiểu kết nối phổ biến.
- ✓ Các hệ quản trị CSDL đều cung cấp công cụ tạo báo cáo tự động và người dùng cũng có thể điều chỉnh bố cục, định dạng báo cáo để nâng cao chất lượng trình bày thông tin.

BÀI 7

CÁC LOẠI KIẾN TRÚC CỦA HỆ CƠ SỞ DỮ LIỆU

Học xong bài này, em sẽ:

- ✓ Phân biệt được CSDL tập trung và CSDL phân tán.
- ✓ Biết được một số kiến trúc thường gặp của hai loại hệ CSDL tập trung và hệ CSDL phân tán.

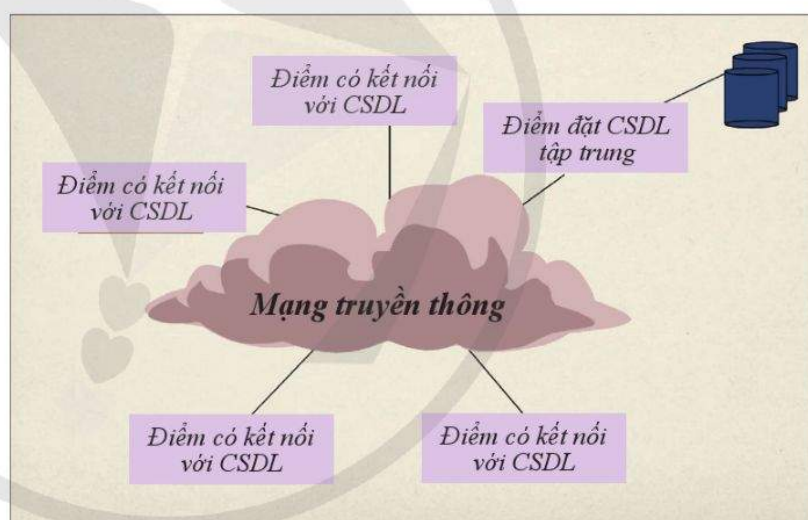


Theo em, CSDL của trường em được đặt trong một máy tính hay trong tất cả các máy tính có sử dụng CSDL đó? CSDL của một ngân hàng được đặt trong một máy tính hay nhiều máy tính?

1 Cơ sở dữ liệu tập trung và cơ sở dữ liệu phân tán

a) Cơ sở dữ liệu tập trung

Một CSDL tập trung được lưu trữ trên một máy tính (Hình 1). Việc quản lý, cập nhật được thực hiện tại chính vị trí này. Tùy trường hợp cụ thể, người dùng có thể truy cập và khai thác thông tin bằng chính máy tính chứa CSDL hay thông qua kết nối mạng (Internet, LAN, WAN,...).



Hình 1. Hệ cơ sở dữ liệu tập trung

Vì tất cả dữ liệu được lưu trữ tại một máy tính duy nhất nên việc truy cập và điều phối dữ liệu dễ dàng hơn, đây là một ưu điểm lớn. Bởi vậy, phần lớn các cơ quan, doanh nghiệp, tổ chức dùng hệ CSDL tập trung. Hệ thống quản lý học sinh của trường em là một hệ CSDL tập trung cỡ nhỏ. Hệ thống bán vé tàu hỏa của Tổng công ty Đường sắt Việt Nam cũng là một ví dụ về hệ CSDL tập trung.

Tuy nhiên, hệ CSDL tập trung cũng có những hạn chế. Trong quá trình khai thác, nếu CSDL tập trung gặp sự cố thì các chương trình ứng dụng CSDL không thể chạy được.

b) Cơ sở dữ liệu phân tán



Theo em, các hệ thống thư điện tử trên Internet có thể sử dụng hệ CSDL tập trung không? Vì sao?

Một CSDL phân tán (Hình 2) là một tập hợp dữ liệu được lưu trữ phân tán trên các máy tính khác nhau của một mạng máy tính (mỗi máy tính như vậy được gọi là một *site* hay một *trạm* của mạng) cùng với những đặc điểm sau đây:

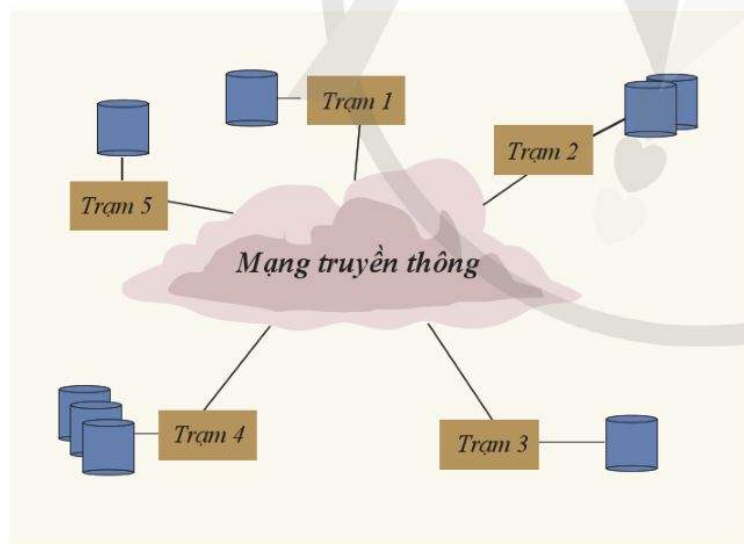
- Mỗi trạm có một CSDL được gọi là CSDL cục bộ của trạm này. Mỗi trạm thực hiện ít nhất một ứng dụng cục bộ, tức là chỉ sử dụng CSDL cục bộ để cho ra kết quả. Khả năng thực hiện ứng dụng cục bộ được gọi là xử lý độc lập.

- Mỗi trạm phải tham gia thực hiện ít nhất một ứng dụng toàn cục. Ứng dụng toàn cục là ứng dụng chạy tại một trạm và phải sử dụng CSDL của ít nhất hai trạm.



Cơ sở dữ liệu phân tán:

tập hợp dữ liệu được phân tán trên các máy tính khác nhau của một mạng máy tính. Mỗi nơi (site) của mạng máy tính có khả năng xử lý độc lập và thực hiện các ứng dụng cục bộ. Mỗi nơi cũng tham gia thực hiện ít nhất một ứng dụng toàn cục, yêu cầu truy xuất dữ liệu tại nhiều nơi bằng cách dùng hệ thống truyền thông con.



Hình 2. Hệ cơ sở dữ liệu phân tán

Ví dụ 1. Một ngân hàng có nhiều chi nhánh, ở mỗi thành phố có một chi nhánh, ở mỗi chi nhánh có dữ liệu quản lý các tài khoản của dân cư và đơn vị đăng kí kinh doanh tại thành phố đó. Thông qua mạng truyền thông, tập hợp dữ liệu của ngân hàng này tại các chi nhánh tạo thành một hệ CSDL phân tán. Người chủ của một tài khoản có thể thực hiện các giao dịch (chẳng hạn rút một khoản tiền trong tài khoản) ở

một chi nhánh nào đó (ví dụ ở Hà Nội), nhưng cũng có thể thực hiện giao dịch ở một chi nhánh đặt tại một thành phố khác (ví dụ ở Đà Nẵng).

Ví dụ 2. Hệ thống tìm kiếm Google có hệ CSDL phân tán. Mỗi yêu cầu có thể được thực hiện bởi hàng trăm máy tính thu thập dữ liệu web và trả về các kết quả có liên quan.

Đối với người dùng, Google dường như là một hệ thống, nhưng nó thực sự là nhiều máy tính làm việc cùng nhau và truy xuất dữ liệu tại nhiều trạm để hoàn thành một nhiệm vụ duy nhất (trả lại kết quả cho truy vấn tìm kiếm).

So với hệ CSDL tập trung, hệ CSDL phân tán có một số ưu điểm chính:

- Sự phân tán dữ liệu về mặt vật lý phù hợp với các tổ chức, doanh nghiệp lớn hoạt động trải rộng về mặt địa lý, phù hợp với các dịch vụ phủ rộng trên toàn cầu, ví dụ như: các hệ thống dịch vụ dựa trên web, hệ thống thương mại điện tử,...

- Tính sẵn sàng và tính tin cậy của dữ liệu cao hơn. Tính sẵn sàng phục vụ cao là do những dữ liệu được đơn vị nào sử dụng nhiều nhất sẽ được lưu trữ và quản lý tại đơn vị đó, thêm nữa khi có sự cố không truy cập được dữ liệu tại một trạm thì vẫn có thể khai thác bản sao của dữ liệu đặt tại một trạm khác. Cũng như vậy, về tính tin cậy, khi một trạm gặp sự cố, có thể khôi phục được dữ liệu tại đây do có bản sao của nó được lưu trữ và vận hành tại một hay vài trạm khác nữa.

- Mở rộng các tổ chức một cách linh hoạt. Có thể thêm trạm mới vào mạng máy tính mà không ảnh hưởng đến hoạt động của các trạm sẵn có.

Tuy nhiên, hệ CSDL phân tán có một số hạn chế so với hệ CSDL tập trung:

- Chi phí cao hơn do hệ thống phức tạp hơn, hệ thống phải làm ẩn đi sự phân tán dữ liệu đối với người dùng.

- Khó khăn hơn trong đảm bảo tính nhất quán dữ liệu và tính an ninh, đồng thời rất khó cung cấp một cái nhìn thống nhất cho người dùng vì dữ liệu đặt tại nhiều địa điểm khác nhau.

Các loại kiến trúc của các hệ cơ sở dữ liệu

Mỗi hệ CSDL bao gồm 3 lớp:

- Lớp CSDL.
- Lớp hệ quản trị CSDL.
- Lớp các ứng dụng CSDL.

Nói về kiến trúc của một hệ CSDL là muốn nhìn hệ thống đó dưới cách phân chia nó thành các thành phần chức năng để có thể hiểu và chỉnh sửa, thay thế mỗi thành phần đó một cách khá độc lập. Dưới đây giới thiệu sơ lược một số kiến trúc phổ biến của hai loại hệ CSDL tập trung và hệ CSDL phân tán.

a) Kiến trúc phổ biến của hệ CSDL tập trung

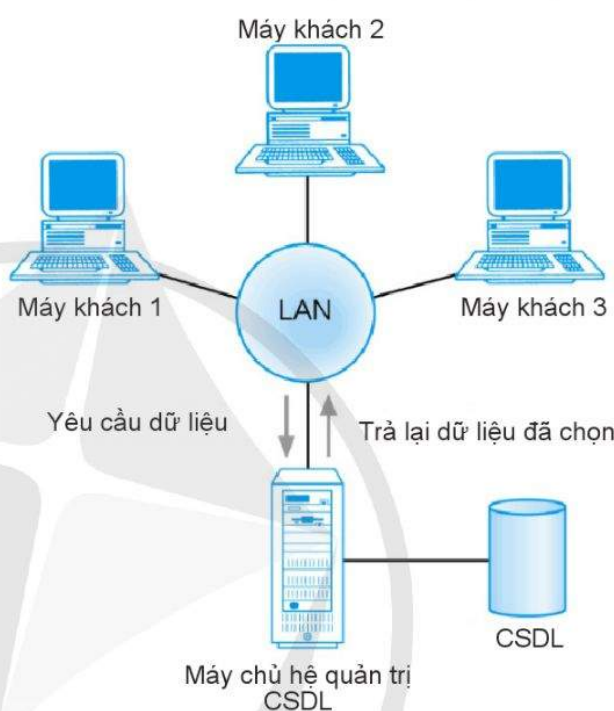
Nhìn chung các hệ CSDL tập trung theo kiến trúc khách – chủ (Client – Server), các thành phần của hệ quản trị CSDL gồm thành phần yêu cầu tài nguyên (dữ liệu) và

thành phần cung cấp tài nguyên (dữ liệu) không nhất thiết phải cài đặt trên cùng một máy tính.

Thành phần cung cấp tài nguyên thường đặt tại một máy chủ (server). Thành phần yêu cầu tài nguyên có thể cài đặt tại nhiều máy khác trên mạng, ta gọi là máy khách (client).

Kiến trúc 1 tầng (1-Tier Architecture) là kiến trúc đơn giản nhất, toàn bộ CSDL được lưu trữ tại một máy tính và cũng chỉ được khai thác tại máy tính này. Máy tính như vậy vừa là máy chủ CSDL vừa là máy khách duy nhất khai thác CSDL. Tuy nhiên, kiến trúc đơn giản này không phù hợp cho các ứng dụng phức tạp.

Kiến trúc 2 tầng (2-Tier Architecture) là kiến trúc có CSDL được lưu trữ ở một máy chủ trên mạng (được xem là tầng 2), thành phần trình bày dữ liệu cho người khai thác được cài đặt trên máy khách kết nối được với mạng (được xem là tầng 1). Máy khách có thể là PC, máy tính bảng hay điện thoại di động,... Tuy nhiên, hiệu suất hoạt động của hệ thống này sẽ kém trong trường hợp có nhiều máy khách cùng khai thác CSDL.



Hình 3. Mô hình Khách – Chủ 2 tầng của CSDL tập trung

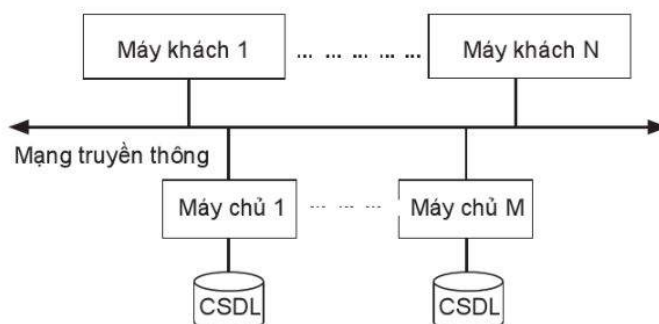
Kiến trúc 3 tầng (3-Tier Architecture) là kiến trúc mở rộng của kiến trúc 2 tầng. Tầng 1 vẫn là thành phần trình bày dữ liệu. Tầng 3 là máy chủ chứa CSDL. Tầng 2 nằm giữa gọi là tầng ứng dụng, hoạt động như một phương tiện để trao đổi dữ liệu đã được xử lý một phần giữa máy chủ và máy khách. Tầng trung gian này chứa các chương trình ứng dụng thường xử lý các vấn đề nghiệp vụ trước khi chuyển dữ liệu qua lại giữa tầng 1 và tầng 3. Loại kiến trúc này thường được sử dụng trong trường hợp các ứng dụng web lớn.

b) Các kiến trúc phổ biến của hệ CSDL phân tán

Hệ CSDL phân tán có một số mô hình kiến trúc phổ biến: mô hình ngang hàng (peer to peer), mô hình khách – chủ cho hệ CSDL phân tán.

Kiến trúc ngang hàng cho hệ CSDL phân tán có mỗi máy tính hoạt động như một máy khách và máy chủ để truyền tải các dịch vụ CSDL. Các máy tính ngang hàng với nhau trong khả năng chia sẻ nguồn tài nguyên dữ liệu của nó với các máy khác và cùng ngang hàng trong khả năng điều phối các hoạt động.

Kiến trúc khách – chủ cho hệ CSDL cũng là kiến trúc khách – chủ như đã biết, nhưng khác với ở hệ CSDL tập trung, hệ CSDL phân tán có nhiều máy chủ CSDL (Hình 4).



Hình 4. Mô hình khách – chủ 2 tầng của CSDL phân tán với nhiều máy chủ và nhiều máy khách



Hãy nêu đặc điểm quan trọng nhất để phân biệt một hệ CSDL tập trung với một hệ CSDL phân tán.



Dựa vào quy mô và đặc điểm tổ chức của mình mà các doanh nghiệp lựa chọn xây dựng cho mình loại hệ CSDL (tập trung hay phân tán) và mô hình kiến trúc phù hợp. Em hãy giải thích và lấy vài ví dụ để minh họa.



Trong các câu sau đây, những câu nào đúng?

- CSDL luôn chỉ được lưu trữ và khai thác tại một máy tính.
- Trong hệ CSDL tập trung, việc quản lý và cập nhật dữ liệu dễ dàng hơn so với hệ CSDL phân tán.
- Trong tất cả các hệ CSDL, hễ có sự cố không truy cập được một máy chủ CSDL thì toàn bộ hệ thống CSDL đó ngừng hoạt động.
- Một hệ CSDL phân tán đắt hơn so với một hệ CSDL tập trung vì nó phức tạp hơn nhiều.

Tóm tắt bài học

- ✓ Điểm khác biệt quan trọng giữa CSDL tập trung và CSDL phân tán là: CSDL tập trung có toàn bộ dữ liệu được lưu trữ trên một máy tính, trong khi đó CSDL phân tán có dữ liệu phân tán trên các máy tính khác nhau của một mạng máy tính và mỗi máy tính khai thác CSDL đều tham gia ít nhất một ứng dụng toàn cục.
- ✓ Kiến trúc khách – chủ là kiến trúc phổ biến của các hệ CSDL tập trung, tùy theo ứng dụng mà có kiến trúc theo mô hình 1 tầng, 2 tầng hay nhiều tầng hơn.
- ✓ Có vài loại mô hình kiến trúc phổ biến của các hệ CSDL phân tán: khách – chủ (cho CSDL phân tán), ngang hàng,...

BÀI 8

BẢO VỆ SỰ AN TOÀN CỦA HỆ CSDL VÀ BẢO MẬT THÔNG TIN TRONG CSDL

Học xong bài này, em sẽ:

- ✓ Nêu được tầm quan trọng của an toàn và bảo mật hệ CSDL.
- ✓ Nêu được một số biện pháp bảo vệ sự an toàn và bảo mật hệ CSDL.



Hệ CSDL của một tổ chức thường có nhiều người dùng truy cập, do vậy có những nguy cơ đe dọa sự an toàn của hệ CSDL. Em hãy cho một vài ví dụ về những nguy cơ đó và hậu quả có thể xảy ra.

1) Tầm quan trọng của việc bảo vệ sự an toàn của hệ CSDL và bảo mật thông tin trong CSDL

Bảo vệ sự an toàn của hệ CSDL và tầm quan trọng của an toàn hệ CSDL

Bảo vệ sự an toàn hệ CSDL là bảo vệ hệ CSDL khỏi các mối đe dọa cố ý hoặc vô tình. Nguy cơ phá vỡ sự an toàn của hệ CSDL có thể đến từ những sự cố, tai họa ngẫu nhiên. Ví dụ, do thao tác vô tình hoặc do lỗi bất chợt ở phần cứng làm hỏng các ổ đĩa lưu trữ dữ liệu hay sự cố cháy nổ,... Tất nhiên sự cố tình phá hoại hoạt động của hệ CSDL, sử dụng hệ CSDL một cách bất hợp pháp hay đánh cắp dữ liệu cũng là nguy cơ làm mất đi sự an toàn của hệ CSDL.

Bảo vệ sự an toàn của hệ CSDL là rất quan trọng đối với bất cứ tổ chức nào vì bất kì một hỏng hóc hay mất mát nào cũng sẽ ảnh hưởng đến hoạt động hằng ngày của tổ chức và hiệu suất làm việc của mọi người.

Bảo mật thông tin trong CSDL và tầm quan trọng của bảo mật thông tin

Một CSDL có thể có những dữ liệu cần được bảo mật. Điều này có nghĩa là cần kiểm soát được việc xem dữ liệu, mỗi cá nhân chỉ được phép xem dữ liệu mà họ được quyền xem. Bảo mật được thông tin trong CSDL là bảo vệ được tính bí mật của những thông tin có tính riêng tư của cá nhân hay tổ chức. Ví dụ, để lộ những thông tin cá nhân bí mật như hồ sơ sức khỏe, số tài khoản cũng như mật khẩu thẻ tín dụng là vi phạm tính bảo mật thông tin. Những thông tin như bí mật thương mại, phân tích cạnh tranh, kế hoạch tiếp thị và bán hàng cũng là những ví dụ về thông tin cần bảo mật của một công ty thương mại, không phải ai trong công ty đó cũng được quyền biết.

Bảo mật thông tin trong CSDL cũng rất quan trọng. Các tổ chức không thực hiện được bảo mật thông tin sẽ phải gánh chịu nhiều hậu quả khó giải quyết hoặc tổn thất. Ví dụ, nếu thông tin bị đánh cắp là tài sản trí tuệ của một công ty thì họ có thể đánh mất

lợi thế cạnh tranh trên thị trường. Một ví dụ khác, nếu thông tin bí mật của khách hàng bị lộ, đơn vị kinh doanh chủ quản hệ CSDL lưu trữ thông tin đó có thể phải đối mặt với pháp luật, bồi thường cho khách hàng, mất uy tín trong kinh doanh.

Bảo vệ tính an toàn của hệ CSDL và bảo mật thông tin trong CSDL là vô cùng cần thiết. Đó không chỉ là bảo vệ dữ liệu bên trong CSDL, bảo vệ tính bí mật của thông tin, mà còn gồm cả bảo vệ hệ quản trị CSDL và tất cả các ứng dụng CSDL sao cho không có truy cập sử dụng dữ liệu sai mục đích và làm hư hỏng dữ liệu.

2) Một số biện pháp bảo vệ sự an toàn của hệ CSDL và bảo mật thông tin trong CSDL



Theo em, sự an toàn của hệ CSDL và bảo mật thông tin trong CSDL có liên quan đến nhau không? Em hãy giải thích ý kiến của mình về điều đó.

a) Bảo vệ sự an toàn của hệ CSDL

Có nhiều biện pháp và cách thức khác nhau mà các tổ chức, doanh nghiệp thực hiện để hệ CSDL của họ được an toàn. Dưới đây là một số biện pháp thường được sử dụng.

Xác thực người truy cập: Hai loại xác thực thường được thực hiện đồng thời là xác thực bằng *thẻ vào cửa* và xác thực bằng *kiểm tra quyền truy cập tài khoản*. Hệ thống bảo vệ (người bảo vệ và camera an ninh) được các tổ chức thiết lập nhằm ngăn chặn người xâm nhập trái phép các thành phần vật lý của hệ thống như: khu vực, toà nhà, phòng chứa máy chủ. Đồng thời với điều đó, các hình thức *thẻ vào cửa* (thẻ nhân viên và mã truy cập vào cửa,...) là một biện pháp không thể bỏ qua. Để *kiểm tra quyền truy cập tài khoản*, xác thực qua mật khẩu là biện pháp phổ biến. Nhiều hệ thống sử dụng thêm các hình thức xác thực khác nữa như: chữ ký điện tử, nhận dạng vân tay, nhận dạng giọng nói, nhận dạng khuôn mặt,... Cơ chế xác thực mạnh sẽ bảo vệ quyền truy cập hiệu quả hơn.

Sử dụng tường lửa: Sử dụng một kỹ thuật được cài vào hệ thống mạng để thiết lập một rào cản giữa một mạng nội bộ đáng tin cậy và mạng bên ngoài không tin cậy.

Sao lưu dự phòng và duy trì biên bản hệ thống: Tạo các bản sao lưu của CSDL và các tệp biên bản (nhật kí) theo định kì, đồng thời đảm bảo rằng các bản sao ở một vị trí an toàn. Trong trường hợp xảy ra lỗi khiến CSDL không thể sử dụng được, bản sao lưu và các chi tiết được ghi lại trong tệp nhật kí được sử dụng để khôi phục CSDL về trạng thái nhất quán mới nhất có thể.

b) Bảo mật thông tin trong CSDL

Nhiều trường hợp cố tình truy cập trái phép, tấn công vào hệ CSDL là để nhằm lấy cắp dữ liệu, đặc biệt là dữ liệu bí mật. Bởi vậy, tất cả các biện pháp nhằm bảo vệ sự an toàn của hệ thống CSDL cũng có vai trò thiết yếu để tăng cường bảo mật thông tin trong CSDL.

Mã hoá dữ liệu là biện pháp bảo mật dữ liệu trong CSDL, là lớp bảo vệ trong trường hợp các biện pháp kiểm soát truy cập đã bị vượt qua. Mã hoá dữ liệu là quá trình chuyển đổi dữ liệu sang một định dạng khác gọi là bản mã. Chỉ những người dùng được uỷ quyền có khoá giải mã mới có thể truy cập được thông tin đó (Hình 1). Mục tiêu của mã hoá dữ liệu là để bảo vệ tính bí mật của dữ liệu kỹ thuật số trong quá trình lưu trữ hoặc trong quá trình truyền trên mạng.



Hình 1. Mã hoá dữ liệu và giải mã

Nén dữ liệu cũng góp phần tăng cường tính bảo mật dữ liệu ngoài mục đích giảm dung lượng lưu trữ. Khi có dữ liệu dạng nén, cần biết quy tắc nén, giải nén mới có dữ liệu gốc được. Việc áp dụng các biện pháp an toàn và bảo mật hệ CSDL có ý nghĩa rất quan trọng nhằm bảo vệ hệ CSDL.



Em hãy nêu một trường hợp cụ thể về hệ CSDL không được an toàn hoặc lộ bí mật thông tin. Với trường hợp đó, cần áp dụng biện pháp nào để tăng cường khả năng bảo vệ sự an toàn của của hệ CSDL và bảo mật thông tin trong CSDL.



Em hãy tìm hiểu các giải pháp đảm bảo an toàn cho hệ CSDL của trường em và đề xuất bổ sung giải pháp cụ thể để nâng cao tính an toàn cho hệ thống đó.



Câu 1. Vì sao cần đảm bảo sự an toàn của hệ CSDL và bảo mật thông tin trong CSDL?

Câu 2. Hãy nêu một vài biện pháp thông dụng bảo vệ sự an toàn cho hệ CSDL và giải thích mục đích của việc mã hoá dữ liệu.

Tóm tắt bài học

- ✓ Cần thiết phải bảo vệ hệ CSDL và bảo mật thông tin trong CSDL khỏi những mối đe dọa: phá hoại hoạt động của hệ thống, thay đổi dữ liệu, lấy cắp dữ liệu, làm lộ bí mật thông tin.
- ✓ Một số biện pháp bảo vệ sự an toàn của hệ CSDL được dùng rất phổ biến là: xác thực người truy cập kiểm soát các truy cập, sử dụng tường lửa, sao lưu dự phòng và duy trì bản hệ thống.
- ✓ Mã hoá và nén dữ liệu là những biện pháp thường dùng để bảo mật thông tin trong CSDL, ngoài ra các biện pháp bảo vệ sự an toàn của hệ CSDL cũng giúp ngăn chặn nguy cơ xâm nhập lấy cắp thông tin bí mật.

NGHỀ QUẢN TRỊ CƠ SỞ DỮ LIỆU

Học xong bài này, em sẽ:

- ✓ Biết được một số thông tin cơ bản về nghề quản trị CSDL: các công việc chính; yêu cầu về kiến thức, kĩ năng; các ngành học có liên quan ở các bậc học tiếp theo; nhu cầu nhân lực hiện tại và tương lai.
- ✓ Tự tìm kiếm và khai thác được thông tin hướng nghiệp (qua các chương trình đào tạo, thông báo tuyển dụng nhân lực,...) về một vài ngành nghề liên quan khác trong lĩnh vực tin học.
- ✓ Giao lưu được với bạn bè qua các kênh truyền thông số để tham khảo và trao đổi ý kiến về những thông tin trên.



Em đã được giới thiệu về hệ quản trị CSDL, đã thực hành tạo lập CSDL và khai thác thông tin trong CSDL cho một bài toán quản lí nhỏ. Hãy thảo luận nhóm và trả lời các câu hỏi sau:

- 1) Quản trị CSDL là gì và nhằm mục đích gì?
- 2) Em có muốn trở thành nhà quản trị CSDL hay không?

1 Công việc chính của nhà quản trị CSDL

Quản trị CSDL là để đảm bảo việc khai thác thông tin trong CSDL phục vụ mọi hoạt động thường ngày của tổ chức, doanh nghiệp và chuẩn bị để ứng phó tốt nhất với các sự cố có thể xảy ra đối với CSDL. Người làm việc quản trị CSDL (*Database Administrator*) gọi là *nhà quản trị CSDL* và có các nhiệm vụ chính sau đây:

a) Đảm bảo an toàn dữ liệu và xác thực quyền truy cập

Nhà quản trị CSDL cần kiểm soát và đảm bảo tính toàn vẹn an toàn cho dữ liệu. Cụ thể, nhà quản trị CSDL thực hiện cấp quyền và kiểm soát truy cập CSDL cho các đối tượng người dùng, đồng thời phát triển các biện pháp bảo mật CSDL và đảm bảo dữ liệu đến từ các nguồn đáng tin cậy. Đặc biệt, vấn đề bảo mật dữ liệu càng cần được coi trọng đối với các hệ thống trực tuyến, doanh nghiệp thương mại điện tử, các công ty và tổ chức có lưu giữ thông tin cá nhân và tài chính của khách hàng.

b) Giám sát hiệu suất và điều chỉnh CSDL

Giám sát hiệu suất CSDL là một phần của quá trình bảo trì hệ thống do nhà quản trị CSDL thực hiện. Nhà quản trị CSDL cần xác định nguyên nhân làm giảm hiệu suất xử lý của hệ thống để khắc phục như: thay đổi các thông số thiết lập trong phần mềm, thay phần cứng có cấu hình mạnh hơn hoặc điều chỉnh các tham số CSDL. Các tham số CSDL ví dụ: số lượng dữ liệu tối đa, số lượng khoá tối đa.

c) Lập kế hoạch phát triển CSDL

Nhà quản trị cần cập nhật định kỳ nhu cầu mới về khai thác dữ liệu của CSDL để đề xuất mở rộng, nâng cấp các khả năng đáp ứng của CSDL. Nhà quản trị CSDL cần thường xuyên cập nhật xu thế phát triển CSDL để có những dự báo tương lai về: không gian lưu trữ của CSDL, công suất sử dụng CSDL. Từ đó, nhà quản trị CSDL sẽ phải chuẩn bị tăng khả năng xử lý khối lượng công việc khi cần (Hình 1).



Hình 1. Minh họa công việc của nhà quản trị CSDL

(Nguồn: <https://www.techrepublic.com>)

d) Sao lưu, phục hồi và khắc phục sự cố

Nhà quản trị CSDL cần có khả năng phán đoán sự cố, nhanh chóng khắc phục các sự cố, khôi phục dữ liệu để giảm thiểu thiệt hại cho tổ chức, đưa các hoạt động với CSDL sớm trở lại bình thường. Nhà quản trị CSDL có trách nhiệm thực hiện sao lưu hệ thống thường xuyên để không có dữ liệu nào bị mất khi ngắt điện đột ngột hoặc các loại thảm họa khác.

e) Cài đặt và bảo trì phần mềm CSDL

Nhà quản trị CSDL cài đặt phần mềm CSDL, thực hiện bảo trì, cập nhật và vá lỗi. Điều này góp phần quan trọng đảm bảo tính toàn vẹn dữ liệu của tổ chức.

2 Yêu cầu của nghề quản trị CSDL

Để trở thành nhà quản trị CSDL, em có thể học các chuyên ngành về quản trị CSDL, khoa học máy tính, hệ thống thông tin quản lý hoặc một chuyên ngành về công nghệ thông tin. Ngoài ra, tùy thuộc vào quy mô của tổ chức và mức độ phức tạp của công việc, nhà quản trị CSDL có thể cần thêm các chứng chỉ của nhà cung cấp phần mềm hệ quản trị CSDL và kinh nghiệm làm việc.

Các kỹ năng cụ thể để quản trị CSDL thường khác nhau tùy theo tổ chức, vị trí công việc và dự án. Tuy nhiên, nhà quản trị CSDL cần có những kiến thức và kỹ năng cơ bản sau:

- Kiến thức vững chắc về CSDL, ngôn ngữ truy vấn CSDL. Nhà quản trị CSDL nên nắm được một số ngôn ngữ truy vấn CSDL phổ biến như: SQL, Oracle SQL và DB2 của IBM.

– Kiến thức về hệ điều hành (các hệ điều hành thông dụng như: Unix, Linux, Windows), phần cứng và mạng.

– Hiểu biết về các ứng dụng liên quan đến CSDL mà mình quản trị.

– Kỹ năng phân tích dữ liệu: Nhà quản trị CSDL phân tích các tập dữ liệu, trích xuất ra thông tin hữu ích cho tổ chức và khách hàng, đồng thời cung cấp thông tin chi tiết về các cải tiến hệ thống, ra quyết định.

– Kỹ năng giao tiếp: Nhà quản trị CSDL thường phải giám sát và làm việc nhóm với các nhân viên công nghệ thông tin. Ngoài ra họ cũng giao tiếp với người quản lý điều hành, nhà cung cấp và các chuyên gia công nghệ tại các tổ chức khác.

– Kỹ năng giải quyết vấn đề: Năng lực xác định, kiểm tra và phát hiện các vấn đề tiềm ẩn, nguyên nhân và giải pháp khắc phục các sự cố là rất có giá trị đối với nhà quản trị CSDL. Nhà quản trị CSDL cũng cần sáng tạo trong việc đưa ra giải pháp cho các vấn đề mới.

– Kỹ năng tổ chức: Tổ chức dữ liệu để đưa ra các quyết định về CSDL. Nhà quản trị CSDL cũng tổ chức các nhiệm vụ cho nhân viên của bộ phận công nghệ thông tin.

– Cân trọng và tỉ mỉ: Vì phải làm việc với khối lượng dữ liệu lớn, nên sai sót nhỏ cũng có thể dẫn đến những lỗi nghiêm trọng. Điều đó đòi hỏi nhà quản trị CSDL phải làm việc cẩn trọng, tỉ mỉ.

Nhu cầu nhân lực và triển vọng phát triển

Hiện nay, ở Việt Nam có rất nhiều trường đại học đào tạo cử nhân hoặc kỹ sư công nghệ thông tin. Các trường đại học có uy tín về đào tạo công nghệ thông tin như: Trường Đại học Bách Khoa Hà Nội, Trường Đại học Bách Khoa – Đại học Quốc gia Thành phố Hồ Chí Minh, Trường Đại học Công nghệ – Đại học Quốc gia Hà Nội, Trường Đại học Khoa học tự nhiên – Đại học Quốc gia Hà Nội, Trường Đại học Công nghệ thông tin – Đại học Quốc gia Thành phố Hồ Chí Minh,... Các chương trình đào tạo công nghệ thông tin trong các trường đại học ở Việt Nam đều trang bị cho sinh viên cơ sở lý thuyết về hệ CSDL, thực hành với hệ quản trị cơ sở dữ liệu, an toàn hệ thống thông tin, lập trình web và phần mềm ứng dụng. Ngoài ra, sinh viên có thể học thêm các khóa học về các hệ CSDL hoặc về các phần mềm cụ thể của Microsoft, IBM, Oracle, Altibase,...

Cơ hội việc làm và mức lương cũng khác nhau đáng kể theo vị trí công việc, quy mô và địa điểm của tổ chức. Dữ liệu của Cục Thống kê Lao động Hoa Kỳ (BLS) năm 2018 cho thấy rằng các nhà quản trị CSDL có mức lương trung bình hàng năm là 90 070 USD. BLS dự đoán rằng các công việc quản trị CSDL sẽ tăng 9% từ năm 2018 đến năm 2028, các vị trí phân tích thông tin y tế dự kiến sẽ tăng 13% vào năm 2026 (www.bls.gov/ooh/healthcare). Các vị trí nhà quản trị CSDL, chuyên gia phân tích thông tin sức khỏe được tuyển dụng trong nhiều tổ chức như: bệnh viện, các cơ sở y tế, các tổ

chức chăm sóc sức khỏe, các công ty bảo hiểm, công ty dược phẩm, công ty sản xuất thiết bị y tế,... Các lĩnh vực như: giáo dục, viễn thông, bảo hiểm, ngân hàng, xuất bản phần mềm cũng sử dụng số lượng lớn nhà quản trị CSDL. Ngoài ra, nhiều công việc khác cần đến kỹ năng quản trị CSDL bao gồm: tư vấn công nghệ thông tin, quản lý dự án công nghệ thông tin, tư vấn ứng dụng và quản trị mạng.

Ở Việt Nam, theo đánh giá của các chuyên gia tư vấn nhân sự, nhu cầu nhân lực ngành công nghệ thông tin sẽ tiếp tục tăng cao trong những năm tới. Đặc biệt, chúng ta đang triển khai rộng rãi về thương mại điện tử, thực hiện chính phủ điện tử, nền kinh tế số, xã hội số. Do đó, vị trí nhà quản trị CSDL càng trở nên quan trọng, cần thiết trong các tổ chức chính quyền và các cơ quan doanh nghiệp.

Nhà quản trị CSDL có thể chọn tiếp tục con đường học vấn bằng cách học lên thạc sĩ về công nghệ thông tin hay theo đuổi các chứng chỉ chuyên môn thuộc lĩnh vực CSDL hoặc mở rộng sang các lĩnh vực liên quan, chẳng hạn như an ninh mạng.

4 Thực hành tìm hiểu một số ngành nghề liên quan

Yêu cầu:

Học sinh được chia thành các nhóm, mỗi nhóm lựa chọn tìm hiểu một trong các ngành nghề: nhà phân tích CSDL, kiến trúc sư CSDL, nhà quản trị dữ liệu. Mỗi nhóm mô tả nghề tìm hiểu bằng một tệp văn bản và giới thiệu nghề bằng một tệp trình chiếu.

Hướng dẫn thực hiện:

Bước 1. Phân công nhiệm vụ cho các thành viên trong nhóm.

Bước 2. Tìm kiếm thông tin về ngành nghề lựa chọn (qua Internet, phỏng vấn và giao lưu với khách mời qua mạng xã hội, email,...), tổng hợp thông tin.

Bước 3. Trao đổi, thảo luận trong nhóm về các nội dung tìm hiểu được.

Bước 4. Soạn nội dung báo cáo (tệp văn bản và trình chiếu).

– Gợi ý chuẩn bị tệp văn bản:

+ Về nội dung, nên gồm những phần chính sau:

- | |
|---|
| 1. Giới thiệu về nghề |
| 2. Kiến thức, kỹ năng cần thiết cho nghề |
| 3. Đặc điểm lao động và yêu cầu của nghề (sản phẩm chính là gì?) |
| 4. Đào tạo và tuyển sinh |
| 5. Tình hình tuyển dụng, môi trường làm việc, thu nhập và phúc lợi xã hội |

+ Về hình thức, cần định dạng văn bản và trình bày khoa học.

- Gợi ý chuẩn bị tệp trình chiếu:
- + Về nội dung: Tùy vào thời lượng trình bày (do giáo viên quy định) để chọn lọc các nội dung (trong tệp văn bản) đưa vào bài trình chiếu. Có thể đưa thêm các hình ảnh, video minh họa cho các phần nội dung để bài trình bày sinh động hơn.
- + Về hình thức: Thiết kế, định dạng bài trình bày và lựa chọn hiệu ứng phù hợp.

Bước 5. Trình bày báo cáo.



Bài 1. Nếu thầy, cô giáo môn Tin học ở trường em được giao quản lí điểm của học sinh trong trường. Theo em, có thể gọi thầy, cô giáo môn Tin học này là nhà quản trị CSDL được không? Vì sao?

Bài 2. Nếu muốn trở thành nhà quản trị CSDL thì em sẽ chuẩn bị những gì?



Trong các câu sau, những câu nào đúng?

- Công việc của nhà quản trị CSDL là đảm bảo CSDL luôn sẵn sàng trong trạng thái tốt nhất và được bảo mật.
- Nhà quản trị CSDL chỉ cần cho các doanh nghiệp lớn.
- Nhà quản trị CSDL nên có hiểu biết về các ứng dụng liên quan đến CSDL mà mình quản trị.
- Nhà quản trị CSDL cần thành thạo các ngôn ngữ lập trình thông dụng hiện nay.
- Nhà quản trị CSDL có hiểu biết sâu về tất cả các ứng dụng liên quan đến CSDL.
- Nhà quản trị CSDL cần sử dụng thành thạo ngôn ngữ truy vấn CSDL như SQL.
- Nhà quản trị CSDL cần có kĩ năng phân tích và giải quyết vấn đề.

Tóm tắt bài học

Nhà quản trị CSDL có vai trò rất quan trọng trong các tổ chức, doanh nghiệp vì họ chịu trách nhiệm duy trì và quản lí CSDL của tổ chức, đảm bảo sự vận hành thông suốt của hệ thống. Nhu cầu nhân lực cho ngành nghề này rất lớn, đặc biệt là trong các tổ chức sử dụng lượng dữ liệu lớn như: bảo hiểm, tài chính, y tế, thương mại điện tử, giáo dục.

BÀI 1**Kiểu mảng và cấu trúc mảng**

Học xong bài này, em sẽ:

- ✓ Trình bày được cấu trúc dữ liệu mảng một chiều.
- ✓ Biết và sử dụng được một số hàm có sẵn trong Python để thao tác với biến kiểu mảng.



Khi lập trình Python, nếu cần xử lý một dãy số thì em dùng kiểu dữ liệu gì?

1) Biến mảng và cấu trúc mảng

Xét bài toán phân tích kết quả học tập cuối năm của một lớp, ví dụ: Lớp 11A, có 45 học sinh. Đầu vào là bảng điểm tổng kết của tất cả học sinh trong lớp, có các cột *Họ và tên, Điểm Toán, Điểm Ngữ văn, Điểm Tin học,...* Cần viết chương trình máy tính cho biết các kết quả như: điểm trung bình mỗi môn học, điểm cao nhất từng môn học, họ và tên học sinh đạt được điểm cao nhất đó,...

Mỗi cột điểm môn học là một dãy số thực gồm 45 số. Hầu hết các ngôn ngữ lập trình bậc cao đều có sẵn kiểu dữ liệu mảng (**array**) nên phù hợp để chứa một dãy số nguyên hay dãy số thực có độ dài định trước.

a) Khai báo mảng một chiều

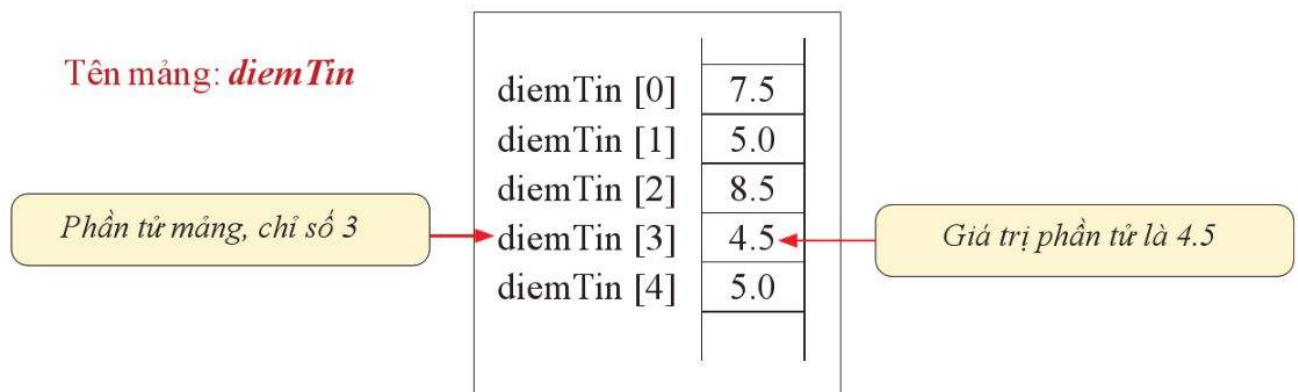
Khai báo tức là cung cấp đủ các thông tin: *tên biến mảng, kiểu dữ liệu, kích thước*. Ví dụ:

- Tên biến mảng: *diemTin*.
- Kiểu dữ liệu của mảng: Số thực.
- Kích thước: 45.

b) Tổ chức mảng một chiều

Trong bộ nhớ, mảng một chiều được lưu trữ thành một khối các ô nhớ liên kế liên tục, có dung lượng bằng tích *kích thước $\times$ độ dài kiểu dữ liệu* (Hình 1). Ví dụ, nếu để lưu trữ một số thực (**float**) cần dùng 32 bit (4 byte) thì mảng *A* gồm 10 phần tử trong bộ nhớ sẽ chiếm 40 byte.

Mảng có kích thước n thì các phần tử mảng được đánh chỉ số tuần tự từ 0 đến $n - 1$.



Hình 1. Phần đầu của mảng *diemTin* (5 phần tử) trong bộ nhớ

Có thể hình dung bộ nhớ RAM là một dãy bit rất dài, chia thành nhiều ô nhớ liên nhau. Mỗi ô nhớ được đánh số gọi là địa chỉ truy cập. Tùy theo cách tổ chức bộ nhớ cho chương trình máy tính, một ô nhớ có thể dài 1 byte, 2 byte hay 4 byte.

Một số nguyên (integer) có thể chiếm 1 byte hoặc 2 byte. Một số thực (float hay double) có thể chiếm 4 byte hoặc 8 byte. Để dễ minh họa với mảng số thực, ta coi một ô nhớ chứa vừa đúng một số thực.

c) Truy cập ngẫu nhiên

Các thông tin có trong khai báo mảng sẽ được máy tính dùng để xác định độ lớn phần bộ nhớ dành cho một biến mảng. Nó cũng cho phép tìm được vị trí chính xác của từng phần tử trong mảng khi biết chỉ số tương ứng.

Ví dụ, nếu mảng *diemTin* được lưu trữ trong bộ nhớ bắt đầu từ địa chỉ A thì phần tử mảng có chỉ số i là *diemTin*[i] sẽ được lưu trữ tại ô nhớ cách vị trí bắt đầu của mảng đúng i ô nhớ. Địa chỉ ô nhớ chứa *diemTin*[3] sẽ là $A + 3$. Máy tính tìm ngay được địa chỉ này. Nếu mảng rất dài, ví dụ gồm 10 000 phần tử thì máy tính cũng tìm ngay được địa chỉ phần tử chỉ số i bất kì theo cách tính trên.

Mảng được sử dụng nhiều vì thời gian truy cập đọc giá trị hay gán giá trị mới cho một phần tử bất kì (đã cho biết chỉ số) là hằng số.

 **Mảng:** một cấu trúc dữ liệu gồm các phần tử có cùng kiểu dữ liệu, được lưu trữ thành một khối nhiều ô nhớ liên tiếp trong bộ nhớ.

 **Thời gian thực hiện hằng số:** một câu lệnh máy, một thuật toán, một chương trình được coi là có thời gian thực hiện hằng số nếu thời gian thực hiện đều không vượt quá một hằng số cho trước, không phụ thuộc kích thước dữ liệu đầu vào.

2 Mảng một chiều trong Python

Cú pháp khai báo mảng một chiều trong Python như sau:

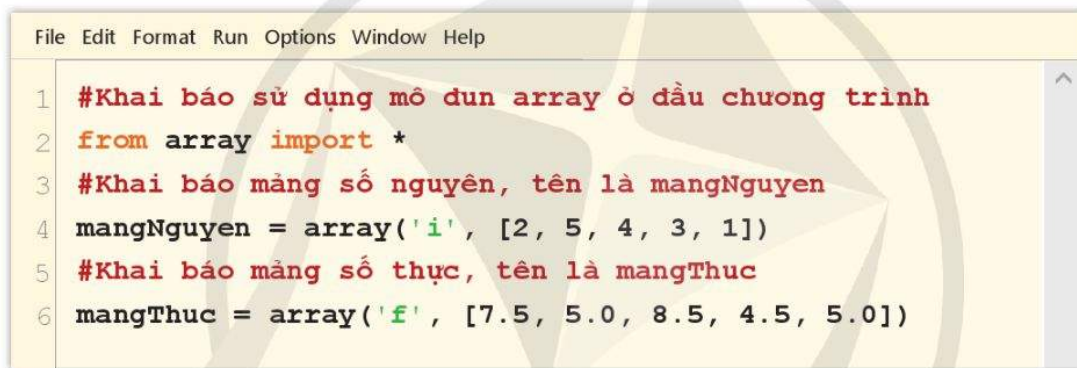
- Khai báo sử dụng mô đun `array` ở đầu chương trình.
- Khai báo biến kiểu mảng theo mẫu dưới đây:

```
mang_1 = array('i', [...])  
mang_2 = array('f', [...])
```

Trong đó:

- Ký tự `'i'` là viết tắt của `integer`; ký tự `'f'` là viết tắt của `float`.
- Thay cho dấu `"..."` ở dòng thứ nhất là một danh sách các số nguyên trong `mang_1`.
- Thay cho dấu `"..."` ở dòng thứ hai là một danh sách các số thực trong `mang_2`.

Hình 2 là một ví dụ khai báo mảng trong Python kèm giải thích câu lệnh.



```
File Edit Format Run Options Window Help  
1 #Khai báo sử dụng mô đun array ở đầu chương trình  
2 from array import *  
3 #Khai báo mảng số nguyên, tên là mangNguyen  
4 mangNguyen = array('i', [2, 5, 4, 3, 1])  
5 #Khai báo mảng số thực, tên là mangThuc  
6 mangThuc = array('f', [7.5, 5.0, 8.5, 4.5, 5.0])
```

Hình 2. Ví dụ khai báo mảng trong Python

Dùng danh sách Python làm mảng

Có thể dùng kiểu danh sách của Python làm mảng một chiều. Không những thế, kiểu danh sách linh hoạt hơn nhiều và có thêm một số hàm (chính xác hơn là các phương thức) mà kiểu mảng không áp dụng được.

Danh sách dùng làm mảng được khai báo và sử dụng như một danh sách Python thông thường.



Em hãy khám phá các phép toán cơ sở với mảng trong Python, sao chép lại và chạy thử các câu lệnh ở Hình 3 và Hình 4; thêm dần từng dòng lệnh, sau đó thực hiện các công việc sau:

- 1) Đoán trước kết quả và chạy chương trình để kiểm tra.
- 2) Xem kết quả và cho biết có sự tương tự giữa mảng với danh sách hay không.


```
File Edit Format Run Options Window Help
1 from array import *
2 mangNguyen = array('i', [2, 5, 4, 3, 1])
3 mangThuc = array('f', [7.5, 5.0, 8.5, 4.5, 5.0])
4 dsNguyen = [2, 5, 4, 3, 1]
5 dsThuc = [7.5, 5.0, 8.5, 4.5, 5.0]
6 #Đọc giá trị; gán giá trị cho phần tử mảng và danh sách tương tự nhau
7 print(dsNguyen[1])
8 print(mangNguyen[2])
9 mangThuc[3] = 7.0
10 print(mangThuc [3])
11 print(dsThuc)
12 mangNguyen[3] = 3.5 #Báo lỗi sai kiểu phần tử
13 dsNguyen[3] = 3.5
14 mangThuc[3] = 8
15 print(mangThuc[3]) #In ra 8.0
```

Hình 3. Ví dụ xử lý mảng một chiều

```
File Edit Format Run Options Window Help
1 #Hàm sorted() để sắp xếp, mặc định theo thứ tự tăng dần
2 #Dãy số kết quả là bản sao khác; dãy số ban đầu giữ nguyên
3 print (sorted(mangThuc))
4 print (mangThuc)
5 #Phương thức sort() không áp dụng được cho mảng, sẽ báo lỗi
6 mangThuc.sort()
7 #Hàm tolist() chuyển mảng thành danh sách, sau đó mới áp dụng sort
8 ds = mangThuc.tolist()
9 ds.sort()
10 print(ds)
```

Hình 4. Minh họa về hàm không áp dụng được cho mảng

③ Một số hàm gộp và hàm phân tích thống kê

Các hàm gộp `max`, `min`, `sum` đã có sẵn và có thể sử dụng ngay cho kiểu mảng cũng như kiểu danh sách. Python có một số hàm phân tích thống kê áp dụng cho kiểu mảng và kiểu danh sách các số. Để sử dụng các hàm trong mô đun này cần khai báo mô đun `statistics` ở đầu chương trình theo cú pháp như sau:

```
from statistics import *
```

**2**

Em hãy khám phá các hàm gộp và hàm phân tích thống kê (tham khảo Hình 5). Thử áp dụng từng hàm với mảng và với danh sách rồi quan sát kết quả.

#Các hàm phân tích thống kê

```
print(mean(mangThuc))
```

```
print(median(dsThuc))
```

```
print(mode(mangThuc))
```

```
...
```

Tên hàm	Chức năng
mean	Trả về trung bình cộng các phần tử
median	Trả về trung vị
mode	Trả về giá trị xuất hiện nhiều lần nhất trong dãy số

Hình 5. Một số hàm phân tích thống kê



Câu 1. Hãy kể tên một số hàm Python áp dụng được cho cả danh sách và mảng.

Câu 2. Hãy kể tên một hàm Python áp dụng cho danh sách nhưng không áp dụng cho mảng.



Cho *diemTin* là dãy điểm tổng kết môn Tin học của lớp 11A. Lập trình đưa ra màn hình các phân tích thống kê: điểm cao nhất, điểm thấp nhất, điểm trung bình, trung vị,... và dãy điểm sắp xếp theo thứ tự từ cao xuống thấp.

Gợi ý: Hàm *sorted* có lựa chọn sắp thứ tự giảm dần.

Cú pháp:

```
sorted(biến_mảng, reverse=True)
```



Câu 1. Khai báo mảng là cung cấp những thông tin gì?

Câu 2. Nói “Thời gian thực hiện câu lệnh là hằng số” có nghĩa là gì?

Tóm tắt bài học

- ✓ Mảng là một tập hợp các phần tử có cùng kiểu dữ liệu, được lưu trữ thành một khối nhiều ô nhớ liên kề trong bộ nhớ. Các phần tử mảng được đánh chỉ số tuần tự và có thể truy cập ngẫu nhiên với thời gian hằng số.
- ✓ Trong Python, mảng và danh sách có nhiều điểm tương tự; có thể dùng danh sách thay cho mảng.

Học xong bài này, em sẽ:

- ✓ Trình bày được cấu trúc dữ liệu mảng hai chiều.
- ✓ Sử dụng được danh sách để thể hiện mảng hai chiều trong Python.



Em có biết nếu xếp nối tiếp nhau n mảng số thực cùng độ dài m phần tử trong bộ nhớ thì đó gọi là gì không?

1 Mảng hai chiều

Mảng hai chiều dùng để lưu trữ một bảng số liệu hình chữ nhật, ví dụ như hình chữ nhật khung đỏ ở Hình 1a và Hình 1b. Mảng hai chiều cũng gọi là *ma trận*.



Ma trận: (*matrix*) bảng hình chữ nhật gồm các phần tử được sắp xếp thành hàng và cột.

Họ và tên	Điểm Toán	Điểm Văn	Điểm Tin	...
Nguyễn Văn Anh	7.5	6.5	5.0	...
Hoàng Đình Bảo	5.0	9.0	4.5	...
Trần Kim Cúc	8.5	8.0	8.0	...
Phạm Cao Dương	4.5	5.5	7.0	...
...	...	...	...	...

Hình 1a. Bảng điểm tổng kết các môn học

bangDiem	Cột [][0]	Cột [][1]	Cột [][2]	...
bangDiem[0]	7.5	6.5	5.0	...
bangDiem[1]	5.0	9.0	4.5	...
bangDiem[2]	8.5	8.0	8.0	...
bangDiem[3]	4.5	5.5	7.0	...
...	...	...	...	...

Hình 1b. Mảng hai chiều bangDiem

Mảng hai chiều là mảng một chiều mà mỗi phần tử là một mảng một chiều. Hầu hết các ngôn ngữ lập trình bậc cao đều có sẵn kiểu mảng hai chiều. Thậm chí, nếu cần thiết, có thể tạo ra cả *mảng nhiều chiều* (có hơn hai chiều).

Khai báo mảng hai chiều

Khai báo mảng hai chiều cần cung cấp đủ các thông tin: *tên biến mảng*; *kiểu dữ liệu*; *kích thước*. Kích thước gồm hai số nguyên dương, mỗi số xác định kích thước một chiều của hình chữ nhật.

Cấu trúc mảng hai chiều

Trong bộ nhớ, mảng hai chiều cũng được tổ chức tương tự như mảng một chiều, tức là lưu trữ thành một khối các ô nhớ liên tục, có độ lớn bằng: $\text{số hàng} \times \text{số cột} \times \text{độ dài kiểu dữ liệu}$.

Truy cập ngẫu nhiên

Các thông tin có trong khai báo mảng hai chiều giúp máy tính xác định dung lượng phần bộ nhớ dành cho một biến mảng hai chiều.

Để truy cập phần tử mảng hai chiều cần biết hai chỉ số: chỉ số hàng và chỉ số cột. Ví dụ, `bangDiem[3][1]` là phần tử ở hàng thứ tư, cột thứ hai của mảng `bangDiem` (Hình 1b). Các thông tin về mảng và hai chỉ số kèm theo cho phép tìm được vị trí chính xác của từng phần tử.

Thời gian thực hiện việc đọc giá trị hay gán giá trị mới cho một phần tử mảng hai chiều cũng là *hằng số*, không phụ thuộc vào kích thước mảng.

2 Sử dụng danh sách làm mảng hai chiều trong Python

Kiểu danh sách (*list*) có sẵn trong Python, rất linh hoạt, hoàn toàn đáp ứng các nhu cầu xử lý dãy số (mảng một chiều) và bảng chữ nhật các số (mảng hai chiều).

Danh sách dùng làm mảng được khai báo và sử dụng như một danh sách Python thông thường. Cú pháp cụ thể như sau:

(1) Khai báo danh sách dùng làm mảng (một chiều hoặc hai chiều) với các phần tử hay các danh sách con sẽ được thêm dần vào sau đó.

```
Tên_danh_sach = []
```

(2) Khai báo danh sách với cặp dấu “`[]`” chứa danh sách các danh sách con cùng độ dài cho kết quả là một danh sách dùng như mảng hai chiều.

```
Tên_danh_sach = [[...], [...], ..., [...]]
```

Ví dụ: `matranThuc = [[7.5, 6.5, 5.0, 5.0, 9.0], [6.5, 8.5, 8.0, 8.0, 4.5]]` cho kết quả là một danh sách dùng như mảng hai chiều, gồm 2 hàng, 5 cột.

Trong bộ nhớ máy tính, mảng hai chiều n hàng và m cột được lưu trữ thành dãy $n \times m$ số bằng cách xếp các hàng tiếp nối nhau, bắt đầu là hàng 0, tiếp theo là hàng 1, hàng 2,... cho đến hết.



Em hãy khai báo một danh sách để làm mảng hai chiều khi lập trình giải bài toán thực tế với dữ liệu đầu vào là bảng điểm tổng kết các môn học như mô tả ở trên. Để tiết kiệm thời gian, ta tạm thời minh họa với mảng 4×3 bằng bảng trích từ Hình 1a như sau:

7.5	6.5	5.0
5.0	9.0	4.5
...		

điểm các môn của học sinh thứ nhất
điểm các môn của học sinh thứ hai
...

Hình 2. Bảng điểm trích từ Hình 1a

Thời gian thực hiện các phép toán của mảng

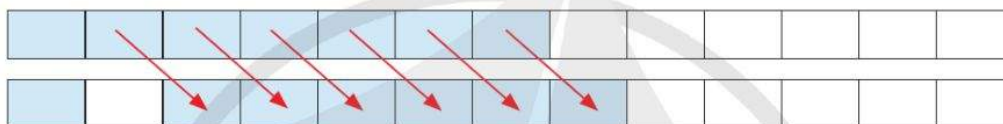
Phép chèn thêm hay xóa phần tử ở cuối mảng có thời gian thực hiện không vượt quá một hằng số nếu còn chỗ trống, bất kể mảng có độ dài bao nhiêu. Trường hợp hết chỗ trống phải di chuyển sang vùng nhớ mới thì thời gian thực hiện là tương đương với độ dài danh sách vào lúc đó.

Phép chèn thêm hay gỡ bỏ ở vị trí bất kì trong mảng sẽ cần dịch chuyển tạo chỗ trống hoặc lấp chỗ trống (Hình 3). Thời gian thực hiện các phép toán sẽ phụ thuộc độ dài n của mảng, không còn là hằng số. Tính trung bình sẽ cần khoảng $\frac{n}{2}$ thao tác dịch chuyển.



Thời gian thực hiện (là) tuyến tính:

Một câu lệnh máy, một thuật toán, một chương trình được coi là có thời gian thực hiện tuyến tính nếu số thao tác để hoàn thành nó tỉ lệ thuận (nhân thêm hằng số) với kích thước dữ liệu đầu vào.



Hình 3. Dịch chuyển tạo chỗ trống khi chèn thêm vào giữa mảng



Câu 1. Vì sao có thể nói mảng hai chiều là mảng các mảng một chiều?

Câu 2. Hãy cho ví dụ một bài toán thực tế cần tính toán trên một bảng số hình chữ nhật.



Hoạt động khám phá trong bài đã minh họa cấu trúc mảng hai chiều, đồng thời cũng chuẩn bị sẵn dữ liệu đầu vào là các dãy điểm số môn học. Hãy viết tiếp các câu lệnh thực hiện phân tích kết quả học tập:

- Cho chỉ số i ứng với một học sinh nào đó trong danh sách: in ra tên học sinh kèm điểm cao nhất; điểm thấp nhất; điểm trung bình các môn.
- Cho chỉ số k ứng với một môn học nào đó trong danh sách: in ra điểm cao nhất; điểm thấp nhất; điểm trung bình môn học.



Câu 1. Trong Python, danh sách dùng làm mảng một chiều và danh sách dùng làm mảng hai chiều có gì khác nhau?

Câu 2. Nói “Thời gian thực hiện (là) tuyến tính” nghĩa là gì?

Tóm tắt bài học

- ✓ Mảng hai chiều là bảng hình chữ nhật các phần tử có cùng kiểu dữ liệu, gồm n hàng và m cột.
- ✓ Có thể truy cập các phần tử bằng hai chỉ số: chỉ số hàng và chỉ số cột.

BÀI 3**THỰC HÀNH VỀ TỆP, MẢNG VÀ DANH SÁCH**

Học xong bài này, em sẽ:

- ✓ Sử dụng được lát cắt để xử lý mảng, danh sách theo ý muốn.
- ✓ Biết và sử dụng được một số hàm xử lý tệp dữ liệu đầu vào, đầu ra.

Nhiệm vụ 1. Lát cắt

Yêu cầu:

a) Cho a là mảng (danh sách) các số. Hãy dùng lát cắt tạo danh sách b và dùng vòng lặp `for` in kết quả ra màn hình (xem mẫu ở Hình 1) để kiểm tra kết quả trong mỗi trường hợp sau:

```

File Edit Format Run Options Window Help
1  b = a[...]          #Thay dấu "..." bằng một danh sách các số
2  for x in b: print(x) #hoặc print(b)
  
```

Hình 1. Mẫu xử lý dùng vòng lặp `for`

- b là nửa cuối của a .
- b là một phần tử kể từ đầu trái của a .
- b là các phần tử chỉ số lẻ của a .

b) Cho a là ma trận (bảng số) hình vuông $n \times n$ các số thực. Hãy viết các câu lệnh (dùng lát cắt khi có thể) để in kết quả ra màn hình và kiểm tra kết quả trong mỗi trường hợp sau:

- Các hàng chỉ số chẵn của a .
- Hai phần tử đầu tiên của hàng đầu tiên của a .
- Hai cột đầu tiên của a .
- Các cột chỉ số lẻ của a .

Gợi ý:

Toán tử lát cắt (Hình 2) trích ra *đoạn con* liên mạch hay *dãy con* (có bước nhảy `step` cách quãng) từ một dãy tuần tự nhiều mục dữ liệu, ví dụ như một biến kiểu danh sách.

Lát cắt	Kết quả
<code>a[lo:hi]</code>	Cắt đoạn con, từ chỉ số <code>lo</code> đến chỉ số <code>hi-1</code> .
<code>a[:hi]</code>	Cắt đoạn đầu danh sách (mặc định <code>lo = 0</code>) đến chỉ số <code>hi-1</code> .
<code>a[lo:]</code>	Cắt đoạn cuối danh sách từ vị trí <code>lo</code> đến cuối danh sách.
<code>a[lo:hi:step]</code>	Cắt danh sách con, từ chỉ số <code>lo</code> đến chỉ số <code>hi-1</code> , với bước nhảy <code>step</code> .

Hình 2. Các lát cắt

Lưu ý: Nếu bước nhảy **step** nhận giá trị âm thì toán tử lát cắt sẽ đảo chiều, đi từ cuối danh sách lên đầu danh sách, từ phải sang trái, kết quả nhận được giống như dùng phương thức **reverse()** (xem ví dụ *Hình 3*).

```
File Edit Format Run Options Window Help
1 ds = [1, 2 , 3, 4, 5]
2 dsLui = ds[::-1] #dsLui = [5, 4, 3, 2, 1]
```

Hình 3. Ví dụ về lát cắt

Nhiệm vụ 2. Vòng lặp

Yêu cầu: Cho a là mảng hai chiều hình vuông gồm n hàng và n cột các số thực. Hãy tính:

- Tổng các phần tử chỉ số chẵn ở hàng i của a .
- Tổng các phần tử âm; tổng các phần tử không âm ở hàng i của a .
- In ra chỉ số các phần tử bằng số x cho trước.

Gợi ý: Vòng lặp **for** hoặc **while** duyệt qua các phần tử trong danh sách a và thân vòng lặp có thể xử lý lần lượt tất cả các phần tử hoặc chọn một số phần tử thoả mãn điều kiện nào đó: theo chỉ số i hoặc theo giá trị $a[i]$.

Nhiệm vụ 3. Đọc dữ liệu từ tệp đầu vào và viết ra tệp

Yêu cầu:

Cho tệp “bangDiem.txt” gồm nhiều dòng; các mục dữ liệu cách nhau khoảng trống:

- Dòng thứ nhất: Hai số nguyên dương n và m ; với n là số học sinh, m là số môn học.
- Dòng thứ hai: *TênHS Toán Văn Tin Li...* gồm $(m + 1)$ từ.
- n dòng tiếp theo, mỗi dòng có tên học sinh và điểm các môn học của học sinh đó.

Hãy viết một hàm **nhapTuTep()** để đọc tệp dữ liệu đầu vào “bangDiem.txt” và khởi tạo dữ liệu sẵn sàng để tính toán phân tích kết quả học tập:

- Một mảng hai chiều $n \times m$ các số thực.
- Hai danh sách: danh sách tên học sinh và danh sách tên môn học.

Hướng dẫn thực hiện:

Có thể tạo tệp “bangDiem.txt” bằng cách chỉnh sửa và bổ sung bảng trong *Hình 1a* ở Bài 2; từ Word hay Excel, thao tác **Copy\Paste** vào cửa sổ của Notepad hay cửa sổ soạn thảo của Python; ghi lưu thành tệp có định dạng text.

- Đọc từng dòng của tệp đầu vào.
- Chuyển đổi mỗi mục của danh sách sang kiểu dữ liệu cần thiết và nối thêm vào danh sách tương ứng trong chương trình (tham khảo chương trình ở *Hình 4*).

```
File Edit Format Run Options Window Help
1  fi = open("bangDiem.txt", encoding = "utf-8")
2  dong = fi.readline().split()
3  ...
4  for i in range(n):
5      ...
6      hang = [float(d) for d in dong]
7      a.append(hang) #Thêm một hàng vào mảng a[][]
```

Hình 4. Ví dụ câu lệnh thêm một hàng vào mảng hai chiều

Trong Python, nếu một dòng gồm nhiều mục khác kiểu dữ liệu, xen kẽ nhau, thì phải truy cập từng phần tử của danh sách và chuyển từ chuỗi ký tự thành kiểu dữ liệu đúng mô tả.

Các thao tác với tệp dữ liệu

Đầu vào là tệp thuần văn bản chữ và số (đuôi tên tệp “txt”) gồm nhiều dòng; mỗi dòng gồm nhiều từ, mỗi từ là một mục dữ liệu, phân cách bằng khoảng trống.

Các bước thực hiện như sau:

Bước 1. Mở tệp để đọc hay viết, sử dụng hàm `open()` như ví dụ ở Hình 5.

```
File Edit Format Run Options Window Help
1  f = open(name, mode) #name = đường dẫn và tên tệp
2  #mode = 'r' là mở để đọc (đây là trường hợp mặc định);
3  #mode = 'w' là mở để viết vào; mode = 'a' là mở để viết nối thêm
```

Hình 5. Cách mở tệp để đọc, viết và viết nối thêm

Bước 2. Đọc từ tệp, có thể dùng các phương thức `read()`, `readline()`, `readlines()` kết hợp với `split()`:

`read().split()`: Đọc từng từ và nối liền toàn bộ các dòng thành một danh sách các từ. Sử dụng khi tệp ngắn và cần xử lý toàn bộ nội dung tệp.

`readline().split()`: Đọc một dòng, trả về danh sách các từ, thường dùng nhất.

`readlines()`: Đọc toàn bộ tệp, trả về danh sách các dòng, mỗi dòng là một chuỗi ký tự, kết thúc bằng ‘\n’ (dấu xuống dòng).

Bước 3. Xuất ra tệp thuần văn bản: có thể dùng hàm `print`, sau khi đã chuyển đầu ra chuẩn từ màn hình sang tệp đã mở để viết vào như sau:

```
sys.stdout = f #f: tên biến tệp đã mở để viết vào
```

Bước 4. Đóng tệp, dùng phương thức `close()`.

Lưu ý: Nếu giữa các từ được phân cách nhau bằng dấu phẩy thì ta có tệp kiểu “csv” (*comma separated value*) và cần dùng `split(',')` thay vì dạng mặc định `split()`. Cần chuyển thành kiểu danh sách (hay mảng) nên sẽ kết hợp xử lý bằng `split(',')`.

Đọc và viết tệp văn bản có chữ tiếng Việt

Một số kí tự trong tiếng Việt không có trong bảng mã ASCII. Để đọc, viết tệp văn bản có chữ tiếng Việt hay các văn bản nói chung, có dùng các kí tự không phải ASCII thì cần thêm tham số tùy chọn `encoding = 'utf-8'` vào các hàm xử lí tệp.



Tạo dãy số thực ngẫu nhiên; sử dụng các hàm `mean`, `median`, `mode` trong mô đun `statistics` để:

- Tìm `mean` của `a` và đếm số phần tử bé hơn, bằng, lớn hơn `mean`.
- Tìm `median` của `a` và cho biết đó là phần tử nào hay nó ở giữa hai phần tử nào.
- Tìm `mode` của `a` và cho biết số lần xuất hiện và dãy các chỉ số tương ứng.
- Áp dụng để phân tích dãy điểm từng môn học của lớp 11A.

BÀI TÌM HIỂU THÊM

LIỆT KÊ MỘT SỐ PHƯƠNG THỨC TRONG PYTHON

Python có sẵn nhiều phương thức áp dụng cho kiểu danh sách. *Bảng 1* dưới đây chọn liệt kê một số phương thức chưa được thực hành nhiều. Kí hiệu `ds` là tên của một biến kiểu danh sách.

Bảng 1. Một số phương thức xử lí danh sách thường dùng

Phương thức	Kết quả
<code>ds.clear()</code>	Xoá rỗng <code>ds</code>
<code>ds.copy()</code>	Tạo bản sao của <code>ds</code>
<code>ds.reverse()</code>	Đảo ngược vị trí các phần tử trong <code>ds</code>
<code>ds.sort()</code>	Sắp xếp <code>ds</code> theo thứ tự tăng dần
<code>ds.append(x)</code>	Thêm một phần tử <code>x</code> vào cuối <code>ds</code>
<code>ds.extend(ds_2)</code>	Nối thêm danh sách <code>ds_2</code> vào cuối <code>ds</code>
<code>ds.index(x)</code>	Tìm vị trí xuất hiện của phần tử <code>x</code> trong <code>ds</code>
<code>ds.count(x)</code>	Đếm số lần xuất hiện phần tử <code>x</code> trong <code>ds</code>
<code>ds.insert(i, x)</code>	Chèn thêm <code>x</code> thành phần tử chỉ số <code>i</code> trong <code>ds</code>
<code>ds.pop(i)</code>	Gỡ bỏ phần tử chỉ số <code>i</code>
<code>ds.clear(i)</code>	Xoá các phần tử bắt đầu từ vị trí <code>i</code> đến hết <code>ds</code>
<code>ds.remove(x)</code>	Xoá phần tử <code>x</code> gặp đầu tiên trong <code>ds</code>

BÀI 4**LÀM MỊN DẦN TỪNG BƯỚC TỪ THUẬT TOÁN
ĐẾN CHƯƠNG TRÌNH MÁY TÍNH**

Học xong bài này, em sẽ:

- ✓ Giải thích được sơ bộ phương pháp làm mịn dần trong lập trình.
- ✓ Biết được mã giả là gì và sử dụng được mã giả làm mịn dần một số thuật toán đơn giản.



Khi lập trình giải bài toán theo một thuật toán đã cho, em sẽ bắt đầu như thế nào? Theo em cách làm như thế có đúng phương pháp không?

1 Mã giả và mô tả thuật toán bằng mã giả

Mã giả thường được sử dụng trong sách giáo khoa, giáo trình hay các bài nghiên cứu để mô tả thuật toán. Mã giả có thể mô tả thuật toán theo cách ngắn gọn mà vẫn làm rõ ý tưởng chính của thuật toán và đảm bảo sự chính xác. Mã giả độc lập với ngôn ngữ lập trình, môi trường lập trình thực hiện thuật toán. Mã giả là cách mô tả thuật toán rất gần với văn bản mã lệnh chương trình. Một mô tả thuật toán bằng mã giả thậm chí có thể coi như chương trình khung.

Không có một quy ước thống nhất chính thức nào về các “từ khoá”, về các kí hiệu được sử dụng trong mã giả. Mã giả phỏng theo các mẫu câu lệnh rẽ nhánh, câu lệnh lặp của ngôn ngữ lập trình bậc cao; sử dụng các kí hiệu toán học, các dấu phép toán quen thuộc hay các kí hiệu gọi tả khác, dễ hiểu với nhiều người. Các kí hiệu được chọn và quy ước rõ để loại bỏ sự nhầm lẫn do khác biệt cách dùng giữa các ngôn ngữ lập trình. Có thể dùng thêm một số từ ngắn gọn sau khi đã định nghĩa rõ ràng.

Quy ước cụ thể khi viết mã giả

Ta sẽ ưu tiên dùng một số yếu tố của ngôn ngữ lập trình Python trong các bài học khi mô tả thuật toán bằng mã giả. Dưới đây là một số quy ước:

- Lời chú thích bắt đầu bằng dấu “#” cho đến hết dòng.
- Cấu trúc rẽ nhánh (phép lựa chọn) dùng mẫu câu lệnh **if...else**.
- Cấu trúc lặp (phép lặp):

+ Số lần lặp biết trước: Phỏng theo mẫu lệnh **for** của Python nhưng mô tả danh sách giá trị theo kiểu toán học. Ví dụ: **for biến in { i | i chẵn, $j + 1 \leq i \leq n - 1$ } : ...**

+ Số lần lặp chưa biết trước: Phỏng theo mẫu lệnh **while** của Python. Ví dụ: **while điều kiện : ...**

– Sử dụng các mức thụt lùi đầu dòng để đánh dấu kết thúc dãy lệnh tuần tự trong mỗi nhánh rẽ của phép lựa chọn hay trong thân vòng lặp của phép lặp.

– Các phép toán gồm:

+ Phép toán số học, phép so sánh.

Ví dụ: $+$, $-$, $*$, $/$, $>$, $<$, $=$, $\neq$, $\geq$, $\leq$, ...

+ Phép gán dùng dấu mũi tên trái.

Ví dụ: $x \leftarrow 5$ nghĩa là gán x nhận giá trị bằng 5. Không viết “ $x = 5$ ” vì nó có nghĩa là phép so sánh x có bằng 5 hay không, cho kết quả là “đúng” (**True**) hoặc “sai” (**False**).

– Một số thành phần khác:

Các lời gọi hàm thư viện hay hàm do người lập trình định nghĩa có thể mô tả ngắn gọn bằng cách viết toán học. Ví dụ: $\min \{a_i \mid j + 1 \leq i \leq n - 1\}$.

Có thể định nghĩa thêm các kí hiệu phép toán để chỉ một việc cụ thể nào đó. Ví dụ: Khi mô tả các thuật toán sắp xếp, người ta thường viết phép đổi chỗ hai phần tử x, y trong dãy số một cách ngắn gọn là **swap** (x, y).

2 ➤ Làm mịn dần các bước mô tả thuật toán

Mô tả thuật toán bằng liệt kê các bước còn chứa nhiều cụm từ của ngôn ngữ tự nhiên, mỗi cụm từ nêu một việc phải làm. Để lập trình thực hiện thuật toán, cần làm chi tiết dần từng bước, tiến gần hơn đến các câu lệnh của ngôn ngữ lập trình. Ở đây lựa chọn sử dụng mã giả để trình bày, vì nó ngắn gọn, dễ hiểu và không phụ thuộc vào ngôn ngữ lập trình.

Cách thức chung: Chuyển các cụm từ mô tả một “việc cần làm” thành các đoạn mã giả, tiến gần hơn một bước đến các câu lệnh của chương trình chi tiết. Sau đây là các ví dụ minh họa.

Ví dụ 1. Thuật toán kiểm tra một số n là số nguyên tố.

– Đầu vào: một số nguyên dương n .

– Đầu ra: Nếu n là số nguyên tố trả về **True**, ngược lại trả về **False**.

Thuật toán khởi đầu đơn giản nhất là làm theo định nghĩa số nguyên tố.

Bước 1. Nếu $n = 1$ thì n không là số nguyên tố;

Bước 2. Nếu $n = 2$ thì n là số nguyên tố;

Bước 3. Nếu $n > 2$ thì kiểm tra tính nguyên tố của n ; trả kết quả kiểm tra **True/False**;

Bước 4. Kết thúc.

Nhận thấy Bước 1 và Bước 2 đã chi tiết và đơn giản nên có thể chuyển thành câu lệnh Python dễ dàng. Riêng Bước 3 “Kiểm tra tính nguyên tố của n với $n > 2$ ” cần được chi tiết và “làm mịn dần” lần lượt theo từng nhận xét sau:

Nhận xét 1: Nếu $n > 2$ thì với k ($2 \leq k < n$) là số nguyên dương bất kì mà n chia hết cho k thì n không là số nguyên tố. Một cách dễ chi tiết cho Bước 3 như sau:

– Với k nào đó thoả mãn $2 \leq k < n$:

Nếu n chia hết cho k thì n không là số nguyên tố.

– Các số k ($2 \leq k < n$) được biểu diễn qua hàm `range` bằng câu lệnh: `range(2, n)`.

Hình 1 minh hoạ mã giả và các câu lệnh Python kiểm tra số nguyên tố sau khi chi tiết Bước 3 theo Nhận xét 1.

Mã giả		Chương trình
if $n = 1$:	1	if ($n == 1$):
Trả về <i>False</i>	2	return False
if $n = 2$:	3	if ($n == 2$):
Trả về <i>True</i>	4	return True
for k in $\{k \mid 2 \leq k \leq n - 1\}$:	5	for k in <code>range(2, n)</code> :
if n chia hết cho k :	6	if ($n \% k == 0$):
Trả về <i>False</i>	7	return False
Trả về <i>True</i>	8	return True

Hình 1. Mã giả và các câu lệnh Python sau khi chi tiết Bước 3 theo Nhận xét 1

Nhận xét 2: n chia hết cho k nghĩa là $n = k.m$. Như vậy, hoặc $k \leq \sqrt{n}$ hoặc $m \leq \sqrt{n}$.

Khi đó, ta chỉ cần kiểm tra với k không lớn hơn $\sqrt{n}$. Một cách khác để chi tiết cho Bước 3 như sau:

– Với k nào đó thoả mãn $2 \leq k \leq \sqrt{n}$:

Nếu n chia hết cho k thì n không là số nguyên tố.

– Các số k ($2 \leq k \leq \sqrt{n}$) được biểu diễn qua hàm `range` bằng câu lệnh: `range(2, int(math.sqrt(n)) + 1)`.

Hình 2 minh hoạ mã giả và các câu lệnh Python kiểm tra số nguyên tố ở Bước 3 sau khi chi tiết theo Nhận xét 2.

Mã giả		Chương trình
for k in $\{k \mid 2 \leq k \leq \sqrt{n}\}$:	5	for k in <code>range(2, int (math.sqrt(n)) + 1)</code> :
if n chia hết cho k :	6	if ($n \% k == 0$):
Trả về <i>False</i>	7	return False

Hình 2. Mã giả và các câu lệnh Python sau khi chi tiết Bước 3 theo Nhận xét 2

Nhận xét 3: Số chẵn lớn hơn 2 không là số nguyên tố. Như vậy chỉ cần kiểm tra với k lẻ và không lớn hơn $\sqrt{n}$. Một cách khác để chi tiết hơn cho Bước 3 như sau:

– Nếu n chẵn và $n > 2$ thì n không là số nguyên tố.

- Trái lại, kiểm tra n chia hết cho k với k lẻ và $3 \leq k \leq \sqrt{n}$.
- Các số k lẻ ($3 \leq k \leq \sqrt{n}$) được biểu diễn qua hàm **range** bằng câu lệnh: `range(3, int(math.sqrt(n))+1, 2)`.

Hình 3 minh hoạ mã giả và các câu lệnh Python kiểm tra số nguyên tố ở Bước 3 sau khi chi tiết theo Nhận xét 3.

Mã giả		Chương trình
<i>if</i> $n > 2$ và n chẵn:	5	<code>if n>2 and n%2 == 0:</code>
Trả về False	6	<code> return False</code>
<i>else</i> :	7	<code>else:</code>
<i>for</i> k <i>in</i> $\{k \mid k \text{ lẻ}, 3 \leq k \leq \sqrt{n}\}$:	8	<code> for k in range(3, int (math.sqrt(n))+1, 2) :</code>
<i>if</i> n chia hết cho k :	9	<code> if (n%k == 0) :</code>
Trả về False	10	<code> return False</code>

Hình 3. Mã giả và các câu lệnh Python sau khi chi tiết Bước 3 theo Nhận xét 3

Ví dụ 2. Bài toán sàng số nguyên tố.

Cho trước số tự nhiên n , hãy sàng lọc chỉ giữ lại những số là số nguyên tố trong dãy $\{0, 1, 2, \dots, n\}$.

- Đầu vào: một số nguyên dương n .
- Đầu ra: in ra danh sách tương ứng đánh dấu **True** (là số nguyên tố) hay **False** (là hợp số).

Thuật toán thô: Đục bỏ dần các số $m > 2$ là bội số của 2, 3, 4, 5, ... cho đến khi hết các bội số thì còn lại các số nguyên tố.

Bước 1. Tạo danh sách *prime* gồm $n + 1$ giá trị logic **True**;

Bước 2. Với $m > 2$, kiểm tra nếu m là một bội số của k ($k < m$) thì gán $prime[m] = \text{False}$;

Bước 3. Gán $prime[0] = \text{False}$; $prime[1] = \text{False}$;

Một cách chi tiết Bước 2 như sau:

- Bắt đầu với $m = 3$;

- **Lặp khi** $m \leq n$:

Nếu với k nào đó ($2 \leq k \leq m - 1$) mà m chia hết cho k thì m không là số nguyên tố;

$m \leftarrow m + 1$

Hết lặp

Mã giả và các câu lệnh Python tương ứng cho Bước 2 có trong Hình 4.

Mã giả		Chương trình
$m \leftarrow 3$	1	<code>m = 3</code>
while $m \leq n$:	2	while (<code>m <= n</code>) :
for k in $\{k \mid 2 \leq k \leq m-1\}$:	3	for <code>k in range(2, m)</code> :
if $m \% k = 0$:	4	if <code>m % k == 0</code> :
$prime[m] \leftarrow False$	5	<code>prime[m] = False</code>
$m \leftarrow m+1$	6	<code>m += 1</code>

Hình 4. Mã giả và các câu lệnh Python chi tiết Bước 2

Thuật toán sàng Eratosthenes: Sàng Eratosthenes là một thuật toán cổ để tìm tất cả các số nguyên tố nhỏ hơn hay bằng n . Thuật toán được cho là tìm ra bởi nhà toán học Hy Lạp trước Công nguyên tên là Eratosthenes. Thuật toán kiểm tra số m là hợp số theo cách hiệu quả hơn.

Ý tưởng: Đục bỏ dần các số không nguyên tố bằng cách đánh dấu “là hợp số” (không phải số nguyên tố) mỗi khi biết số đó là bội số của một số nguyên tố.



Đọc mã lệnh của thuật toán Eratosthenes cho ở Hình 5 và mô tả liệt kê các bước của thuật toán và bằng mã giả.

```

File Edit Format Run Options Window Help
1  def SieveOfEratosthenes(n):
2      #Tạo mảng biến Boolean "prime[0..n]"; gán trị ban đầu tất cả là True
3      #Kết cục prime[i] sẽ là False nếu i không là số nguyên tố
4      #Còn lại là số nguyên tố
5      prime = [True for i in range(n + 1)]
6      p = 2
7      while (p * p <= n):
8          #Nếu prime[p] không bị sửa, p là nguyên tố
9          if prime[p]:
10             #Đục bỏ các bội số của p
11             for i in range(p * p, n + 1, p):
12                 prime[i] = False
13             p += 1
14         prime[0] = False
15         prime[1] = False
16         return prime

```

Hình 5. Mã lệnh của thuật toán Eratosthenes



Em hãy viết chương trình thực hiện sàng số nguyên tố sử dụng thuật toán thô và sử dụng thuật toán Eratosthenes. Sau đó chạy thử và so sánh kết quả.



Câu 1. Hãy nêu một điều kiện sàng khác cho bài toán sàng số: In ra danh sách các số nguyên dương nhỏ hơn n và thỏa mãn điều kiện sàng mới.

Gợi ý: Ví dụ “không là số chính phương”.

Câu 2. Viết mô tả mã giả cho thuật toán tương ứng với Câu 1.



Câu 1. Hãy cho biết cách viết các dấu phép toán số học, phép so sánh bằng mã giả.

Câu 2. Hãy cho biết cách viết phép gán bằng mã giả; dấu bằng “=” có ý nghĩa gì trong mã giả.

Câu 3. Cho câu lệnh lặp bằng mã giả như ở hình bên. Hãy diễn giải ý nghĩa và cho biết kết quả là gì nếu bắt đầu ta có j nhận giá trị 5 và n nhận giá trị 15.

```
for i in { i | i chẵn,  $j + 1 \leq i \leq n - 1$  } :  
    In ra i
```

Tóm tắt bài học

- ✓ Mã giả là một cách mô tả thuật toán độc lập với ngôn ngữ lập trình và tạo thuận lợi cho việc chuyển thuật toán thành chương trình máy tính.
- ✓ Từ mô tả thuật toán bằng liệt kê các bước, chuyển dần những cụm từ mô tả một công việc thành mã giả bằng cách làm chi tiết từng bước cách thực hiện công việc đó.
- ✓ Chuyển câu lệnh mã giả thành mã lệnh của ngôn ngữ lập trình để có văn bản chương trình.

BÀI TÌM HIỂU THÊM

KIỂM TRA TÍNH NGUYÊN TỐ THEO CÁCH NGẪU NHIÊN

Bài toán kiểm tra tính nguyên tố của một số nguyên dương rất lớn trở nên đặc biệt quan trọng khi các hệ mật mã khoá công khai ra đời.

Bên cạnh các phương pháp cho kết quả chính xác, có các phương pháp kiểm tra theo xác suất, dùng các thuật toán ngẫu nhiên. Sau một loạt lần kiểm tra, nếu không tìm được bằng chứng chứng tỏ n là hợp số thì ta kết luận n là số nguyên tố. Xác suất kết luận sai càng nhỏ khi số lần kiểm tra được tăng thêm càng nhiều.

Ví dụ, các phép kiểm tra tính nguyên tố theo cách ngẫu nhiên:

- Phép kiểm tra Fermat (ít được sử dụng).
- Phép Kiểm tra Miller-Rabin.
- Phép kiểm tra Solovay-Strassen.

Học xong bài này, em sẽ:

- ✓ Trình bày được sơ lược khái niệm độ phức tạp thời gian của thuật toán. Nêu được ví dụ minh họa.
- ✓ Biết được kí pháp O lớn và các bậc độ phức tạp thời gian.



Theo em, một thuật toán như thế nào thì được xem là chạy nhanh/chạy chậm?

➤ 1 Các khái niệm cơ bản

Một thuật toán ngắn gọn, dễ hiểu, dễ lập trình cũng là những điểm tốt, nhưng trong Tin học, các thuật toán được đánh giá và so sánh dựa trên một tiêu chuẩn quan trọng hơn – Đó là tính hiệu quả. Thuật toán được coi là hiệu quả hơn nếu thời gian thực hiện chương trình và lượng bộ nhớ mà máy tính cần dùng là ít hơn.

Ước lượng thời gian thực thi chương trình và hiệu quả thời gian của thuật toán

Python có lệnh `time()` cho phép bấm giờ tính thời gian chạy thực thi chương trình. Nhiều ngôn ngữ lập trình khác cũng có lệnh tương tự. Nhưng cách tính giờ chạy thực thi chương trình cụ thể không áp dụng được khi muốn so sánh hiệu quả để lựa chọn thuật toán vì nó dẫn đến các vấn đề sau đây:

- Phải lập trình và chạy thử chương trình của tất cả các thuật toán cần so sánh.
- Thời gian đo được phụ thuộc vào nhiều yếu tố không liên quan tới thuật toán bởi chương trình chạy nhanh hay chậm phụ thuộc vào phần cứng máy tính, ngôn ngữ lập trình, chương trình dịch và bản thân kĩ năng lập trình của người viết.
- Không khả thi nếu muốn chọn cách lập nhiều chương trình khác nhau (lập trình viên, ngôn ngữ lập trình) rồi tính thời gian thực thi trung bình.

Kích thước đầu vào

Thời gian chạy chương trình phụ thuộc kích thước dữ liệu đầu vào. Ví dụ, chương trình tìm kiếm một số x trong dãy số sẽ kết thúc sớm hơn khi dãy chỉ gồm 10 số và sẽ lâu hơn khi dãy gồm 10 000 số. Thông thường, kích thước dữ liệu đầu vào được đại diện bằng một số tự nhiên n . Ví dụ, trong trường hợp bài toán sắp xếp dãy số hay tìm kiếm một số x trong dãy, thì kích thước đầu vào n là độ dài dãy số.

2 Độ phức tạp thời gian của thuật toán

Thời gian chạy một chương trình với đầu vào có kích thước n sẽ là một hàm số $T(n)$ của biến số n . Độ phức tạp thời gian là một khái niệm trong khoa học máy tính, là kết quả ước lượng thời gian thực hiện các chương trình cài đặt thuật toán để xử lý một lượng dữ liệu đầu vào có độ lớn n . Ước lượng này thể hiện *số phép toán* cần thiết để thực hiện thuật toán khi đã biết dữ liệu đầu vào có kích thước n . Khó tính đếm chính xác con số này vì nhiều lý do:

- Bộ xử lý thực hiện các phép toán bit, khó có thể xác định tương ứng số các phép toán bit với mỗi phép toán mà chúng ta vẫn biết như các phép toán số học (cộng, trừ, nhân, chia), các phép so sánh,...

- Ngay cả khi tính đếm số phép toán theo nghĩa thông thường với con người thì thế nào là một phép toán cũng không dễ thống nhất. Ví dụ: Phép khai căn, phép lũy thừa,... là một hay nhiều phép toán số học.

Phép toán sơ cấp

Khi phân tích một thuật toán để tính đếm số lượng các phép toán cần thực hiện, người ta phân biệt các phép toán sơ cấp với phần còn lại được coi là không sơ cấp. Một phép toán sơ cấp là phép toán có thời gian thực hiện không lớn hơn một hằng số nào đó, không phụ thuộc n (n là kích thước dữ liệu đầu vào).

Ví dụ, những trường hợp dưới đây được coi là phép toán sơ cấp:

- Phép toán số học, phép so sánh,... với các toán hạng là giá trị cụ thể.
- Các hàm toán học với đầu vào là giá trị cụ thể không phụ thuộc n .

Khái niệm phép toán sơ cấp sẽ rõ dần qua một số ví dụ minh họa và thực hành.

Phép lặp (mô tả bằng cấu trúc lặp), *phép lựa chọn* (mô tả bằng cấu trúc rẽ nhánh) không phải là phép toán sơ cấp.

3 Ví dụ về độ phức tạp thời gian hằng số và độ phức tạp thời gian tuyến tính



Cho bài toán tính tổng dãy số: $S = 1 + 2 + \dots + n$. Hãy cho biết cách giải nào tốt hơn trong hai cách giải sau đây:

Cách thứ nhất: Tính cộng dồn dần từng số.

Cách thứ hai: Vì dãy số là cấp số cộng nên có thể dùng công thức tính tổng cấp số cộng $S = \frac{n(n+1)}{2}$.

Độ phức tạp thời gian hằng số

Thuật toán có *độ phức tạp thời gian hằng số* khi mà số phép toán cần thực hiện không phụ thuộc kích thước n của dữ liệu đầu vào.

Cách giải thứ hai cho bài toán trong Hoạt động trên là một thuật toán có độ phức tạp thời gian hằng số vì chỉ cần 3 phép toán để tính tổng S , $T(n) = 3$. Chuyện kể rằng, đây là cách giải mà khi còn là học sinh phổ thông, nhà toán học thiên tài người Đức – Friedrich Gauss đã nêu ra khi làm bài tập trong giờ học môn Toán.

Độ phức tạp thời gian tuyến tính

Thuật toán có *độ phức tạp thời gian tuyến tính* nếu số phép toán cần thực hiện là hàm tuyến tính của n (n là kích thước dữ liệu đầu vào). Cách giải thứ nhất trong hoạt động trên là một thuật toán có độ phức tạp thời gian tuyến tính vì $T(n) = n - 1$.

4 Kí pháp và các bậc độ phức tạp thời gian

Cách ước lượng làm già thêm

Số phép toán cần thiết để thực hiện thuật toán không chỉ phụ thuộc kích thước n của dữ liệu đầu vào mà còn phụ thuộc vào việc ta may mắn gặp trường hợp dễ, ít việc phải làm hay không may gặp trường hợp khó, nhiều việc phải làm hơn.

Xét thuật toán tìm số lớn nhất trong dãy số:

- Đầu tiên, tạm gán $max = a_0$; đọc giá trị tiếp theo, so sánh với max và gán lại nếu cần.
- Khi phần tử đầu dãy a_0 có giá trị lớn nhất thì số lần phải gán lại giá trị $max = a_i$ là bằng 0. Số phép toán là ít nhất.
- Khi dãy số ban đầu là dãy tăng chặt, mọi số đều khác nhau, thì số lần phải gán $max = a_i$ là bằng n . Số phép toán là nhiều nhất.

Một cách tổng quát, có thể xét ba trường hợp: trường hợp thuận lợi nhất (số phép toán cần thực hiện ít nhất); trường hợp bất lợi nhất (số phép toán cần thực hiện nhiều nhất) và trường hợp ngẫu nhiên (số phép toán cần thực hiện ở mức trung bình).

Nói chung, ta muốn có một *ước lượng trung bình* cho tất cả các trường hợp ngẫu nhiên xảy ra. Tuy nhiên, không dễ tìm được ước lượng trung bình này. Người ta chọn cách dễ làm hơn, đó là *ước lượng làm già thêm*. Cách ước lượng đảm bảo rằng trong thực tế sẽ không có trường hợp nào vượt quá ước lượng đã đưa ra.

Kí pháp O lớn

Theo định nghĩa, nếu số phép toán sơ cấp cần thực hiện không vượt quá một hằng số C , không phụ thuộc n thì thuật toán có độ phức tạp thời gian là hằng số. Kí hiệu $T(n) = O(1)$.

Nếu số phép toán sơ cấp cần thực hiện không vượt quá một hàm tuyến tính của n , $T(n) \leq C_1 n + C_2$ (với C_1, C_2 là hằng số) thì độ phức tạp thời gian của thuật toán là tuyến tính. Viết ngắn gọn $T(n) = O(n)$ nghĩa là độ phức tạp thời gian của thuật toán là tuyến tính. *Bảng 1* dưới đây là một số kí hiệu O lớn về thời gian thực hiện thuật toán thường gặp:

Bảng 1. Một số kí hiệu O lớn về thời gian thực hiện thuật toán

Kí hiệu O lớn	Tên gọi độ phức tạp thời gian thuật toán
$O(1)$	Hằng số
$O(\log_2 n)$	Logarit
$O(n)$	Tuyến tính (linear)
$O(n^2)$	Bậc hai (quadratic)
$O(C^n)$	Hàm mũ (exponential) ($C > 1$)
$O(n!)$	Giai thừa

Một số công thức liên quan:

Công thức 1:

Nếu $f_1(n) = O(g_1(n))$ và $f_2(n) = O(g_2(n))$ thì $f_1(n) + f_2(n) = O(\max(g_1(n), g_2(n)))$.
Công thức áp dụng cho hai cấu trúc điều khiển được thực hiện tuần tự.

Công thức 2:

Nếu $f_1(n) = O(g_1(n))$ và $f_2(n) = O(g_2(n))$ thì $f_1(n) \times f_2(n) = O(g_1(n) \times g_2(n))$.
Công thức áp dụng cho hai cấu trúc điều khiển lồng nhau.

5 Các quy tắc khi ước lượng thời gian thực hiện thuật toán

Quy tắc chung

Khi tính đếm số phép toán cần thực hiện, các quy tắc ước lượng cho phép bỏ bớt những phần có bậc lớn thấp hơn, chỉ giữ lại những phần có bậc lớn cao nhất và các hằng số nhân C đều coi là 1.

Mô tả thuật toán chỉ sử dụng ba cấu trúc: cấu trúc tuần tự, cấu trúc rẽ nhánh, cấu trúc lặp. Cấu trúc tuần tự thực hiện *dãy phép toán* nối tiếp nhau. Dây phép toán của một cấu trúc tuần tự gồm các phép toán sơ cấp và có thể có các phép toán không sơ cấp như *phép lựa chọn* (thực hiện bằng cấu trúc rẽ nhánh) hay *phép lặp* (thực hiện bằng cấu trúc lặp).

Lòng bên trong các cấu trúc rẽ nhánh và cấu trúc lặp lại là các dãy phép toán tuần tự khác. Cần ước lượng số phép toán từ bên trong trở ra ngoài.

Lời gọi hàm

Hàm trong chương trình thực chất là một chương trình con, thực hiện một thuật toán cụ thể. Ước lượng độ phức tạp thời gian một lời gọi hàm chia làm hai trường hợp:

- Lời gọi các hàm toán học sơ cấp, các hàm thư viện,... với đầu vào là giá trị cụ thể không phụ thuộc n . Trường hợp này có độ phức tạp thời gian là $T(n) = O(1)$.
- Lời gọi hàm trong các trường hợp còn lại sẽ được ước lượng độ phức tạp như với một thuật toán.

Cấu trúc tuần tự và quy tắc lấy max

Cấu trúc tuần tự là một dãy gồm C phép toán; C là số xác định, không phụ thuộc n .

- Nếu tất cả C phép toán là sơ cấp, độ phức tạp thời gian là $T(n) = O(1)$.
- Trái lại, thời gian thực hiện bằng *ước lượng lớn nhất* trong số các ước lượng của các phép toán có trong dãy.

Cấu trúc rẽ nhánh và quy tắc lấy max

Máy tính thực thi một cấu trúc rẽ nhánh (hai nhánh hay nhiều nhánh) sẽ phải kiểm tra điều kiện và thực hiện một trong số các nhánh. Thường việc kiểm tra điều kiện là tính giá trị biểu thức logic gồm biểu thức số học và một phép so sánh, độ phức tạp thời gian là $T(n) = O(1)$.

Độ phức tạp thời gian của cấu trúc rẽ nhánh là độ phức tạp thời gian lớn nhất trong các độ phức tạp thời gian của các nhánh. Lúc này, độ phức tạp thời gian của việc kiểm tra điều kiện trong cấu trúc rẽ nhánh có thể sẽ không còn là $O(1)$.

Ví dụ:

Câu lệnh mã giả	Độ phức tạp thời gian
if dãy là cấp số cộng:	$O(n)$
Tính tổng theo công thức Gauss	$O(1)$
else:	
Tính cộng dồn	$O(n)$

Hình 1. Mã giả và độ phức tạp thời gian để tính tổng dãy số

Nếu như chưa biết trước, việc kiểm tra dãy có phải là cấp số cộng hay không sẽ cần thời gian $O(n)$. Độ phức tạp thời gian của cấu trúc rẽ nhánh trong Hình 1 là $O(n)$ trong mọi trường hợp.

Cấu trúc vòng lặp và quy tắc nhân

Máy tính thực hiện một cấu trúc vòng lặp sẽ phải kiểm tra điều kiện và thực hiện thân vòng lặp. Thân vòng lặp là một cấu trúc tuần tự các phép toán (có thể có phép lựa chọn hay phép lặp khác).

Thời gian thực hiện cấu trúc vòng lặp được tính bằng số lần lặp nhân với tổng thời gian kiểm tra điều kiện lặp và thời gian thực hiện thân vòng lặp.

Ví dụ: Trong Hình 2, độ phức tạp thời gian của thuật toán tìm giá trị cực tiểu một dãy số $a_1, a_2, a_3, \dots, a_n$ là $O(n)$.

Câu lệnh mã giả	Số phép toán sơ cấp
$i_min \leftarrow 1$ # Tạm gán chỉ số đạt min i_min	1
for k in $\{k \mid 2 \leq k \leq n\}$:	$n - 1$
if $a[i_min] > a[k]$:	1
$i_min \leftarrow k$ # Gán lại chỉ số đạt min	1
return i_min	

Hình 2. Mã giả và số phép toán sơ cấp để tìm cực tiểu của một dãy số



Em hãy cho ví dụ một lời gọi hàm được tính là phép toán sơ cấp và một lời gọi hàm không được tính là phép toán sơ cấp.



Câu 1. Xét bài toán sắp xếp dãy số. Hãy cho biết khi nào ta có trường hợp thuận lợi nhất, số phép toán cần làm là ít nhất?

Câu 2. Ước lượng số phép toán sơ cấp cần thực hiện để tìm số lớn nhất trong dãy số:

a) Đầu vào là dãy ngẫu nhiên.

b) Đầu vào là dãy giảm dần.



Câu 1. Tại sao không thể đánh giá thuật toán qua chương trình cài đặt thuật toán?

Câu 2. Khi nào thì áp dụng quy tắc lấy max?

Câu 3. Quy tắc nhân áp dụng cho cấu trúc vòng lặp là gì?

Tóm tắt bài học

- ✓ Độ phức tạp thời gian của thuật toán thể hiện tổng số phép toán sơ cấp cần thực hiện để hoàn thành thuật toán và được ước lượng xấp xỉ bằng một hàm số phụ thuộc n (n là kích thước dữ liệu đầu vào).
- ✓ Để ước lượng độ phức tạp thời gian của thuật toán, phải xác định đúng phép toán sơ cấp và tuân thủ các quy tắc tính số phép toán cho cấu trúc tuần tự, cấu trúc rẽ nhánh, cấu trúc lặp.

BÀI 6

KIỂM THỬ VÀ SỬA LỖI CHƯƠNG TRÌNH

Học xong bài này, em sẽ:

- ✓ Biết được việc kiểm thử giúp lập trình viên phát hiện lỗi, làm tăng độ tin cậy của chương trình nhưng chưa chứng minh được chương trình đã hết lỗi.
- ✓ Biết được một số kinh nghiệm gỡ lỗi và các thói quen lập trình tốt để dễ gỡ lỗi.



Theo em, tại sao rất ít khi viết chương trình xong có thể chạy được ngay?

1 Nguyên nhân gây lỗi và truy vết lỗi

a) Các loại lỗi và nguyên nhân

Lỗi cú pháp là lỗi hay xảy ra trong quá trình soạn thảo chương trình. Người lập trình chỉ cần hiểu rõ ngôn ngữ lập trình mình sử dụng là có thể dễ dàng sửa lỗi cú pháp. Hiện nay các môi trường tích hợp phát triển phần mềm (IDE – Integrated Development Environment) có công cụ soạn thảo chương trình nhằm hạn chế những sai sót có thể sinh ra lỗi cú pháp.

Chương trình đã biên dịch, chạy thử thành công một vài lần vẫn có thể đột ngột dừng giữa chừng hoặc chạy mãi không dừng. Đó là lỗi thời gian chạy (*runtime errors*). Nguyên nhân thường do có giá trị không hợp lệ khi thực hiện một tính toán nào đó. Ví dụ: quên gán trị khởi tạo một biến, bỏ qua việc kiểm tra mẫu số khác 0 trước khi chia; chỉ số phần tử của danh sách ngoài phạm vi cho phép; quên tăng giá trị biến đếm để kiểm tra vòng lặp,...

b) Truy vết lỗi và thông báo lỗi

Vùng soạn thảo các câu lệnh trong môi trường lập trình IDE thường có hiển thị số thứ tự các dòng lệnh, đánh số tăng dần từ 1. Khi phát sinh một lỗi, chức năng gỡ lỗi sẽ truy ngược lùi về phía trên, tìm đến tận gốc, tới dòng lệnh có câu lệnh gây lỗi. Thông báo lỗi in ra danh sách các dòng lệnh truy vết được, ghi kèm số thứ tự dòng lệnh trong văn bản chương trình. Người lập trình dễ dàng tìm ra chuỗi dòng lệnh gây lỗi.

2 Chạy thử chương trình

Chạy thử là để phát hiện lỗi trong mã nguồn của chương trình. Gỡ lỗi là xác định vị trí có lỗi, nguyên nhân gây lỗi và sửa lỗi. Phát hiện lỗi và sửa lỗi là hai việc đan xen trong một quá trình. Mục đích cuối cùng là đảm bảo rằng chương trình hoạt động đúng, đáp ứng yêu cầu bài toán đặt ra.

Thuật toán sai thì chương trình thực hiện đúng thuật toán đó sẽ cho kết quả sai. Việc phát hiện chương trình còn lỗi và sửa lỗi sẽ không phân biệt đó là lỗi chương trình thực hiện thuật toán hay lỗi của bản thân thuật toán.

Tập hợp toàn bộ các trường hợp đầu vào có thể xảy ra của một chương trình thường là vô hạn. Đầu vào cho chương trình thực hiện một thuật toán sắp xếp là bất cứ dãy số nào. Không thể chạy thử chương trình với tất cả các đầu vào có thể có.

Chạy thử cho phép người lập trình dễ phát hiện lỗi hơn, qua đó kịp thời đưa ra các biện pháp xử lý lỗi. Mặc dù điều này không đảm bảo tuyệt đối rằng chương trình không còn lỗi nhưng nó cũng hạn chế được rất nhiều rủi ro phát sinh lỗi trong quá trình vận hành.

Một số kinh nghiệm thực hành gỡ lỗi chương trình

Các ca kiểm thử để phát hiện lỗi chương trình

Một *ca kiểm thử* là một trường hợp đã cho các đầu vào cụ thể và dự đoán trước kết quả đầu ra đúng yêu cầu của bài toán.

Các ca kiểm thử nhằm phát hiện các lỗi tiềm ẩn. Dưới đây là một số gợi ý các ca kiểm thử:

- Kiểm tra các lệnh rẽ nhánh với đầu vào tương ứng cho đủ các trường hợp.
- Kiểm tra các lệnh lặp với đầu vào khiến số lần lặp là 0 lần, 1 lần, nhiều lần.
- Kiểm tra với các giá trị ở các đầu mút trái, phải của một biểu thức điều kiện. Ví dụ, với điều kiện $a \leq x \leq b$, hãy thử với các giá trị a và b . Các hạn chế “không là số dương”, “không là số âm”,... hãy thử với đầu vào bằng 0.
- Cần thận trọng với điều kiện “bằng nhau” khi so sánh hai biến kiểu số thực vì kết quả tính toán có thể bị làm tròn. Ví dụ, sau khi tính tỉ lệ phần trăm, cộng các tỉ lệ phần trăm không chắc sẽ đúng bằng 100.
- Kiểm tra với các đầu vào “không mong đợi” nếu muốn biết chương trình sẽ hoạt động như thế nào khi người khác chạy “khám phá”. Các giá trị không mong đợi có thể là giá trị rất lớn hoặc rất gần số không, giá trị không hợp lệ,...



Hãy cho một số ví dụ ca kiểm thử:

- 1) Chương trình giải phương trình bậc hai.
- 2) Chương trình tính đếm (tính tổng, tính trung bình cộng,...) các số dương trong một mảng số thực.

Chia để trị

Kiểm thử và sửa lỗi một đoạn mã lệnh ngắn dễ hơn nhiều so với cả một văn bản chương trình dài. Hãy kiểm thử và sửa lỗi từng đoạn mã lệnh, từng hàm riêng biệt, chắc chắn rằng nó làm đúng việc cần làm trước khi chuyển sang phần khác. Văn bản chương trình có thể được gỡ lỗi, chỉnh sửa hoàn thiện dần từng phần nếu ta biết cách tổ chức tách biệt các phần công việc của chương trình để dễ sửa lỗi. Việc tổ chức tách biệt các phần công việc của chương trình cũng là một khía cạnh của *phương pháp lập trình theo mô đun* (sẽ được trình bày chi tiết hơn ở các bài học sau).

Hãy in ra

Có những lỗi logic rất khó phát hiện. Một kinh nghiệm là kiểm soát các giá trị biến, biểu thức trong quá trình chạy kiểm thử chương trình. Điều này có thể thực hiện bằng cách in ra các giá trị biến, biểu thức; hoặc theo dõi các giá trị biến, biểu thức bằng trình gỡ rối nếu nó được trang bị sẵn trong IDE.

4 Tập thói quen tốt khi lập trình để dễ gỡ lỗi

Các kĩ năng lập trình và gỡ lỗi chỉ có được qua kinh nghiệm thực hành. Hãy học từ những sai lầm của bản thân, ghi nhớ những lỗi đã mắc, cải tiến phong cách lập trình. Nên tập một số thói quen tốt sau đây để chương trình ít lỗi và việc gỡ lỗi dễ dàng hơn:

- Không lập tức bắt đầu viết các câu lệnh ngay sau khi đọc xong bài toán lập trình và nảy ra ý tưởng (thuật toán) giải bài toán. Nên bắt đầu với việc tách biệt các phần công việc cần làm và thiết kế tổng thể chương trình.
- Mô tả thuật toán bằng liệt kê các bước, chọn những phần việc chuyển thành chương trình con (hàm tự định nghĩa), xác định rõ đầu vào đầu ra của mỗi hàm.
- Chọn đặt *tên gọi nhớ* cho các hàm và các biến quan trọng, sao cho dễ nhận biết nó làm gì, đó là cái gì. Điều này giúp tránh nhầm lẫn khi viết câu lệnh và dễ phát hiện lỗi hơn.
- Viết chú thích đầy đủ, ngay trước hay ngay sau các khai báo tên hàm, tên biến quan trọng, các đoạn mã lệnh cần chú ý. Sau này, chính người lập trình có thể không còn nhớ mình có ý tưởng gì lúc viết các dòng lệnh đó.

5 Tổ chức tách biệt các phần của một chương trình

Định nghĩa hàm để thực hiện thuật toán

Người lập trình tự định nghĩa một (hay một số) hàm: chọn tên hàm, tên các biến đầu vào và cách trả về kết quả. Trong mô tả thuật toán đã có sẵn những thông tin này. Phần thân hàm là kết quả chuyển từ mô tả thuật toán thành câu lệnh của ngôn ngữ lập trình đã chọn.

Các lệnh để chạy thử phát hiện lỗi

Trong chương trình cần có thêm các lệnh làm những việc sau:

- Gán dữ liệu đầu vào: Một số câu lệnh gán giá trị cho các biến đầu vào. Dữ liệu đầu vào cũng có thể đọc từ tệp cho trước.
- Xuất kết quả đầu ra: Một số lệnh in ra màn hình. Để tiện kiểm tra, đỡ nhầm lẫn, nên in kèm lời mô tả đầu ra là gì; có thể in kèm cả dữ liệu đầu vào tương ứng.

Lợi ích của việc tổ chức tách biệt các phần công việc

- Dễ chạy thử: Các lệnh để chạy thử kiểm tra ở các chỗ cần theo dõi giá trị của các biến, việc thực hiện các đoạn chương trình. Dùng dấu chú thích “#” có thể liệt kê một danh sách các ca kiểm thử khác nhau và chạy thử từng ca.
- Dễ sửa lỗi: Bộ cục chương trình có logic rõ ràng, dễ thấy lỗi xảy ra ở việc nào.



Câu 1. Có các loại lỗi chương trình nào? Nguyên nhân gây ra loại lỗi đó có thể là gì?

Câu 2. Hãy nêu một vài thói quen lập trình tốt để chương trình ít lỗi và dễ gỡ lỗi.



Em hãy liệt kê một số ca kiểm thử cho chương trình:

- a) Tìm số x trong một dãy số (đã cho cụ thể).
- b) Sắp xếp một dãy số.



Câu 1. Tại sao nói kiểm thử chương trình làm tăng độ tin cậy của chương trình nhưng chưa chứng minh được chương trình đã hết lỗi?

Câu 2. Nên làm gì mỗi khi nghi ngờ một chức năng nào đó của chương trình chưa chắc đúng như ta mong muốn?

Tóm tắt bài học

- ✓ Chương trình đã chạy ra kết quả, có thể vẫn còn lỗi tiềm ẩn; kiểm thử để phát hiện lỗi và sửa lỗi nhằm đảm bảo rằng chương trình đáp ứng yêu cầu bài toán đặt ra.
- ✓ Cần kiểm thử: đủ các trường hợp của cấu trúc rẽ nhánh, các trường hợp ở đầu mút của một biểu thức điều kiện và các trường hợp của cấu trúc lặp có số lần lặp là 0 lần, 1 lần, nhiều lần.
- ✓ Cần tập các thói quen lập trình tốt để chương trình ít lỗi và dễ gỡ lỗi.

Học xong bài này, em sẽ:

- ✓ Phát biểu được bài toán tìm kiếm.
- ✓ Viết được chương trình cho một số thuật toán tìm kiếm.
- ✓ Vận dụng được quy tắc thực hành xác định được độ phức tạp của một vài thuật toán tìm kiếm đơn giản.



Khi tạo mới một tài khoản người dùng, em được yêu cầu nhập tên người dùng “user name”. Có trường hợp em phải nhập lại tên khác vì tên vừa nhập đã có người sử dụng rồi. Theo em, máy tính làm gì ngay sau khi nhận được yêu cầu tạo mới một tài khoản? Hãy phát biểu thành một bài toán.

1 Bài toán tìm kiếm

a) Khái niệm bài toán tìm kiếm

Các ví dụ thực tế dẫn đến bài toán tìm kiếm:

- Cho mã cuốn sách, hãy tìm cuốn sách trong kho sách của thư viện.
- Tìm một tên người, tên hàng hoá,... trong danh sách liệt kê.
- Tìm bản ghi có khoá là k trong bảng T của một cơ sở dữ liệu.

Theo nghĩa chung nhất, bài toán tìm kiếm là: Cho một *yêu cầu tìm kiếm* và một tập hợp dữ liệu là *phạm vi tìm kiếm*. Hãy tìm mục (các mục) dữ liệu đáp ứng yêu cầu tìm kiếm đã cho hoặc khẳng định không có mục dữ liệu nào đáp ứng yêu cầu đó.

Tuỳ theo yêu cầu và phạm vi tìm kiếm mà bài toán tìm kiếm là dễ hay khó. Một nhiệm vụ của máy tìm kiếm trên Internet nhằm giải quyết bài toán tìm kiếm có phạm vi tìm kiếm là dữ liệu văn bản (hình ảnh hoặc giọng nói) trên Internet.

b) Tìm kiếm tuần tự bằng hàm của Python

Python có phương thức `index` thực hiện tìm kiếm phần tử x trong một dãy tuần tự (xâu kí tự, mảng hoặc danh sách) và trả về:

- Nếu xuất hiện nhiều lần thì đưa ra chỉ số của lần xuất hiện đầu tiên.

- Báo lỗi “**ValueError**” nếu không tìm thấy.
- Phương thức **index** có hai tham số tùy chọn: **lo**, **hi** để hạn chế thực hiện tìm kiếm chỉ trong đoạn con của dãy số, bắt đầu từ chỉ số **lo** (**lowest**) và kết thúc ở **hi** (**highest**). Cú pháp:

```
dãy_số.index(giá_trị, lo, hi)
```

Ví dụ: Với mảng $a = [1, 2, 3, 4, 5, 6]$ như hình bên, câu lệnh `print(a.index(3, 1, 4))` sẽ in ra màn hình kết quả là 2, cho biết vị trí của phần tử 3 trong đoạn $[1, 4]$ ở mảng a .

```
File Edit Format Run Options Window Help
1 a = [1, 2, 3, 4, 5, 6]
2 print(a.index(3, 1, 4))
```

2 Thuật toán tìm kiếm tuần tự

Chi tiết dần từng bước thuật toán tìm kiếm tuần tự

Hình 1 mô tả liệt kê các bước của thuật toán tìm kiếm tuần tự một số x :

Số đang xét là số ở đầu dãy
Lặp khi Chưa hết dãy số:
 Nếu Số đang xét $\neq x$:
 Chuyển đến xét số tiếp theo trong dãy
 Trái lại:
 Thông báo vị trí tìm thấy x và kết thúc
Hết nhánh
Hết lặp
 Thông báo không tìm thấy x và kết thúc

Hình 1. Thuật toán tìm kiếm tuần tự

Trong mô tả trên có một số cụm từ như: “Số đang xét là số ở đầu dãy”, “Chưa hết dãy số”, “Chuyển đến xét số tiếp theo trong dãy”, “Thông báo vị trí tìm thấy x và kết thúc”, “Thông báo không tìm thấy x và kết thúc”. Hình 2 là kết quả thay thế những cụm từ trên bằng mã giả.

	Mã giả	Số phép toán sơ cấp
	$i \leftarrow 0$ # Số đang xét là a_0 ở đầu dãy	?
	while $(i < n)$: # $(i < n)$ tức là chưa hết dãy số	?
	if $a_i \neq x$:	
	$i \leftarrow i + 1$ # Chuyển đến xét số tiếp theo	?
	else:	
	return i # Đã tìm thấy	?
	return không tìm thấy	

Hình 2. Mã giả của thuật toán tìm kiếm tuần tự



1

Em hãy tính số phép toán sơ cấp, điền vào cột bên phải của bảng ở Hình 2 và ước lượng độ phức tạp thời gian của thuật toán tìm kiếm tuần tự.

3 Thuật toán tìm kiếm nhị phân

Nếu dãy số đã sắp thứ tự thì có thể áp dụng thuật toán tìm kiếm nhị phân như mô tả trong Hình 3: Chia đôi dần dãy số và loại bỏ nửa dãy chắc chắn không chứa x . Phạm vi tìm kiếm giảm đi một nửa sau mỗi bước. Kết thúc khi tìm thấy hoặc phạm vi tìm kiếm đã hết mà không thấy (Hình 3).

Xuất phát: Phạm vi tìm kiếm là dãy ban đầu

Lặp khi Vẫn còn Phạm vi tìm kiếm :

Xác định phần tử a_m ở giữa Phạm vi tìm kiếm

Nếu $x = a_m$:

Thông báo tìm thấy x ở vị trí m và kết thúc

Trái lại :

Loại bỏ nửa dãy chắc chắn không chứa x

Phạm vi tìm kiếm là nửa dãy còn lại

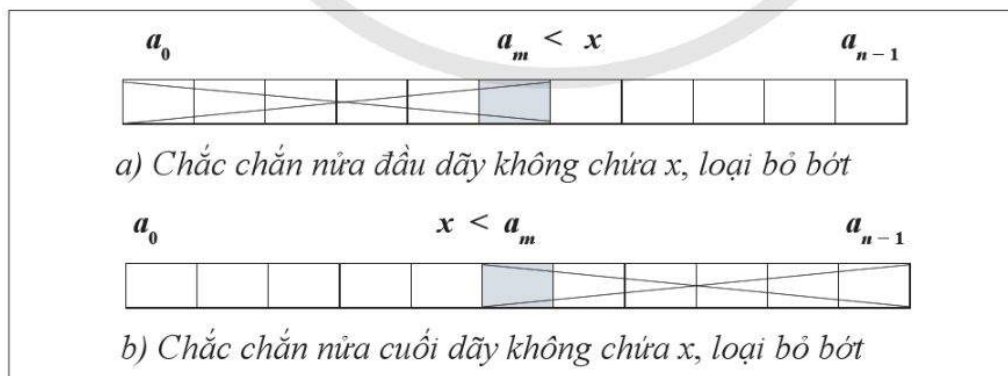
Hết nhánh

Hết lặp

Thông báo không tìm thấy x và kết thúc

Hình 3. Liệt kê các bước của thuật toán tìm kiếm nhị phân

Hình 4 minh họa cho dãy sắp thứ tự tăng dần (không giảm):



Hình 4. Sơ đồ một bước thuật toán tìm kiếm nhị phân

Hướng dẫn viết mã giả của thuật toán tìm kiếm nhị phân

– Xác định các cụm từ cần làm chi tiết hơn bằng mã giả: “Phạm vi tìm kiếm là dãy ban đầu”; “Vẫn còn phạm vi tìm kiếm”; “Xác định phần tử a_m ở giữa phạm vi tìm kiếm”,

“Loại bỏ nửa dãy chắc chắn không chứa x ”, “Phạm vi tìm kiếm là nửa dãy còn lại”, “Thông báo không tìm thấy x và kết thúc”.

– Bổ sung thêm **lo** là chỉ số phần tử ở đầu trái đoạn con và **hi** là chỉ số phần tử ở đầu phải đoạn con.

– Công thức tính chỉ số m của phần tử ở “giữa” đoạn con là $(lo + hi)/2$, kết quả đảm bảo là số nguyên.



Em hãy thực hiện các yêu cầu sau:

1) Viết mã giả cho thuật toán tìm kiếm nhị phân.

2) Ước lượng số lần thực hiện vòng lặp trong thuật toán tìm kiếm nhị phân.

Gợi ý: Sau lần chia đôi đầu tiên, phạm vi tìm kiếm còn lại $\frac{n}{2}$ số; sau khi chia đôi lần thứ hai, dãy còn lại $\frac{n}{4}$ số; sau khi chia đôi lần thứ ba, dãy còn lại $\frac{n}{8}$ số, ... sau khi chia đôi lần k dãy còn lại $\frac{n}{2^k}$ số. Kết thúc khi $2^k \sim n$.

3) Ước lượng độ phức tạp thời gian của thuật toán tìm kiếm nhị phân.

Thực hành lập trình giải bài toán tìm kiếm

Nhiệm vụ 1. a) Viết chương trình Python thực hiện tìm kiếm tuần tự.

b) Viết phiên bản tìm kiếm tuần tự thứ hai, dùng vòng lặp **for** thay cho vòng lặp **while** (hoặc ngược lại).

c) Viết phiên bản tìm kiếm tuần tự có thêm hai tham số đầu vào **lo** và **hi** tương tự như của hàm **index**. So sánh kết quả với phương thức **index** của Python.

Nhiệm vụ 2. Viết hàm thực hiện tìm kiếm nhị phân nhận hai tham số đầu vào: dãy số a và giá trị x cần tìm.



Viết chương trình tìm kiếm vị trí tên của một người trong mỗi danh sách sau đây:

a) Danh sách học sinh của lớp em.

b) Danh sách tên các chủ tài khoản ngân hàng (kí tự không dấu) và đã sắp thứ tự theo bảng chữ cái.



Câu 1. Em hãy nêu ra một vài ví dụ về bài toán tìm kiếm trong thực tế.

Câu 2. Theo em, với dãy đã sắp thứ tự và cho một số x cụ thể:

a) Trường hợp nào tìm kiếm tuần tự nhanh hơn tìm kiếm nhị phân?

b) Về trung bình thuật toán tìm kiếm tuần tự hay thuật toán tìm kiếm nhị phân tốt hơn?

Tóm tắt bài học

- ✓ Thực hiện tìm kiếm tuần tự bằng phép lặp duyệt từ đầu dãy số với điều kiện dừng khi “tìm thấy” hoặc “đã xét hết dãy số”.
- ✓ Phép lặp thực hiện tìm kiếm nhị phân chia đôi dãy số tại điểm “giữa” có chỉ số $(lo + hi) // 2$, bỏ bớt nửa dãy cho đến khi “tìm thấy” hoặc hết dãy.

BÀI TÌM HIỂU THÊM

CÁC THAM SỐ TỰY CHỌN CHO MỘT HÀM

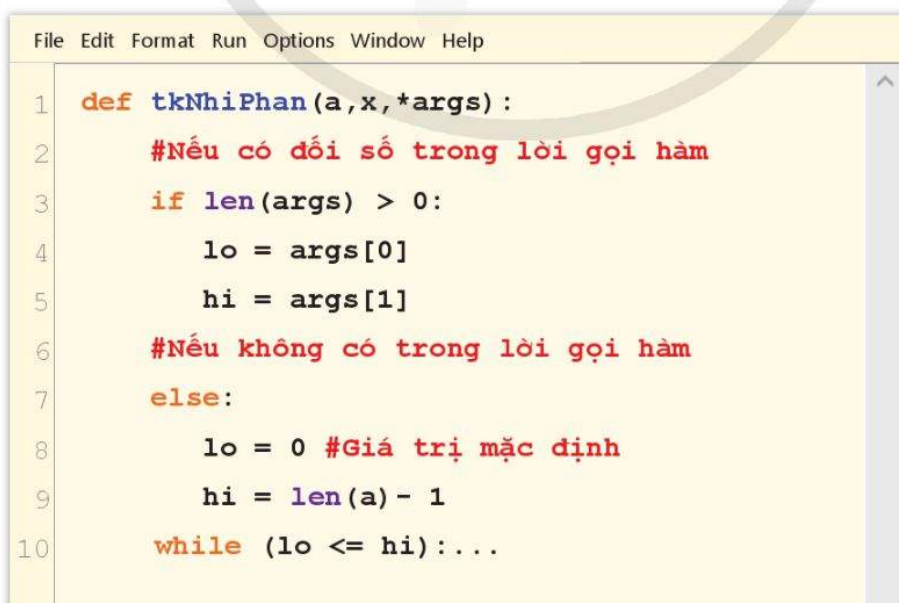
Các hàm thực hiện tìm kiếm của Python đều có các tham số tựy chọn `lo` và `hi` để xác định một phạm vi tìm kiếm là đoạn con của dãy số đầu vào. Ta đã viết các hàm thực hiện tìm kiếm có thêm hai tham số `lo` và `hi` có vai trò tương tự. Tuy nhiên, đây không phải là những tham số tựy chọn vì chúng bắt buộc phải có trong lời gọi sử dụng các hàm này. Làm thế nào để chúng trở thành *tựy chọn*?

Một giải pháp là dùng *bộ (tuple) các tham số *args* trong định nghĩa hàm.

1) Dấu sao (*) là kí hiệu toán tử, sau đó là một tên bộ, gộp nhiều tham số. Có thể tựy ý đặt tên bộ; `args` chỉ là thông lệ thường thấy.

2) Quy định `*args` phải đứng sau tất cả các tham số bắt buộc của hàm.

Theo mẫu trong *Hình 5*, ta có thể định nghĩa các hàm có tham số tựy chọn.



```
File Edit Format Run Options Window Help
1 def tkNhiPhan(a,x,*args):
2     #Nếu có đối số trong lời gọi hàm
3     if len(args) > 0:
4         lo = args[0]
5         hi = args[1]
6     #Nếu không có trong lời gọi hàm
7     else:
8         lo = 0 #Giá trị mặc định
9         hi = len(a) - 1
10    while (lo <= hi):...
```

Hình 5. Một hàm tự định nghĩa có tham số tựy chọn

BÀI 8

LẬP TRÌNH MỘT SỐ THUẬT TOÁN SẮP XẾP

Học xong bài này, em sẽ:

- ✓ Phát biểu được bài toán sắp xếp.
- ✓ Viết được chương trình cho một vài thuật toán sắp xếp.



Trình quản lý tệp của hệ điều hành cho phép lựa chọn hiển thị nội dung của thư mục được sắp xếp thứ tự theo vài cách khác nhau. Em hãy cho biết một trong số các lựa chọn này và giải thích rõ thêm tiêu chí (yêu cầu) sắp xếp tương ứng.

1 Bài toán sắp xếp

Một số bài toán sắp xếp như sau:

- Cho dãy các số, yêu cầu sắp xếp “theo thứ tự tăng dần (giảm dần)”.
- Cho dãy các xâu kí tự, yêu cầu sắp xếp “theo thứ tự bảng chữ cái”, “theo độ dài tăng dần”,...
- Sắp xếp các hàng trong một bảng gồm nhiều cột (hay bản ghi trong bảng cơ sở dữ liệu) theo một cột nào đó. Ví dụ, có bảng kết quả học tập gồm các cột *Họ và tên*, *Điểm Toán*, *Điểm Ngữ văn*, *Điểm Tin học*,... yêu cầu sắp xếp theo điểm môn Tin học giảm dần. Các hàng trong bảng có dạng như sau:

	File	Edit	Format	Run	Options	Window	Help
1	['Nguyễn Văn An' , 7.5 , 8.0 , 6.5 , ...]						
2	['Trần Thị Bình' , 6.5 , 9.0 , 4.5 , ...]						
3	['Phạm Minh Chí' , 3.5 , 9.5 , 8.5 , ...]						

Trong tin học, thuật ngữ sắp xếp đề cập đến việc tổ chức lại một tập hợp dữ liệu theo một tiêu chí sắp xếp, tức là đáp ứng một yêu cầu cụ thể về trình tự. Kết quả sắp xếp là một *danh sách* theo đúng thứ tự yêu cầu. Sắp xếp giúp tìm kiếm nhanh hơn. Yêu cầu sắp xếp cần chỉ rõ *cách so sánh* hai mục dữ liệu để quyết định *thứ tự*.

Dưới đây trình bày bài toán sắp xếp đơn giản và minh họa bằng sắp xếp dãy số.

- Đầu vào: Dãy n số $a_0, a_1, \dots, a_{n-1}$.
- Đầu ra: Dãy được sắp theo thứ tự tăng dần (không giảm).

Sắp xếp tại chỗ và không tại chỗ

Một thuật toán được gọi là sắp xếp tại chỗ khi không phải dùng thêm một dãy khác ở bên ngoài dãy ban đầu để thực hiện sắp xếp. Nó chỉ đổi chỗ các phần tử trong dãy ban đầu. Yêu cầu này rất quan trọng khi dãy cần sắp xếp rất dài.

Nếu thuật toán sử dụng một dãy khác ở bên ngoài dãy ban đầu để chứa kết quả thì gọi là sắp xếp không tại chỗ.

Các thuật toán được trình bày trong bài học đều có yêu cầu sắp xếp tại chỗ và thực hiện với cấu trúc mảng, phải dịch chuyển để lấy chỗ trống khi thao tác chèn để thay đổi vị trí.

Nghịch thế

Nếu $i < j$ mà $a_i > a_j$ thì cặp hai phần tử (a_i, a_j) gọi là một nghịch thế. Dãy số chưa sắp đúng thứ tự khi còn ít nhất một nghịch thế. Một số thuật toán sắp xếp dựa trên ý tưởng giảm dần và tiến đến triệt tiêu hết các nghịch thế trong dãy. Dãy được sắp xếp xong khi không còn nghịch thế nào.

2) Thuật toán sắp xếp nổi bọt (Bubble Sort)

Ý tưởng: Xét các cặp hai phần tử liên kề nhau (a_i, a_{i+1}) ; nếu là một nghịch thế thì đổi chỗ a_i cho a_{i+1} để loại bỏ nghịch thế này. Lặp lại việc này để soát hết dãy số từ đầu đến cuối. Số lượng nghịch thế giảm bớt sau một vòng lặp. Thực hiện nhiều vòng lặp thì cuối cùng sẽ hết nghịch thế và dãy được sắp đúng thứ tự.

Hình 1 trình bày diễn biến từng vòng lặp khi thực hiện thuật toán. Kí hiệu $\circlearrowright$ thể hiện thao tác đổi chỗ khi có nghịch thế (cặp phần tử màu đỏ).

Dãy (a)	a_0	a_1	a_2	a_3	a_4	a_5	Giải thích diễn biến từng vòng lặp
Ban đầu	55	19	42	94	18	67	Vòng lặp 1) 55 > 19 $\circlearrowright$, 55 > 42 $\circlearrowright$, 55 < 94, 94 > 18 $\circlearrowright$, 94 > 67 $\circlearrowright$
Sau vòng lặp 1	19	42	55	18	67	94	Vòng lặp 2) 19 < 42, 42 < 55, 55 > 18 $\circlearrowright$, 55 < 67, 67 < 94
Sau vòng lặp 2	19	42	18	55	67	94	Vòng lặp 3) 19 < 42, 42 > 18 $\circlearrowright$, 42 < 55, 55 < 67, 67 < 94
Sau vòng lặp 3	19	18	42	55	67	94	Vòng lặp 4) 19 > 18 $\circlearrowright$, 19 < 42, 42 < 55, 55 < 67, 67 < 94
Sau vòng lặp 4	18	19	42	55	67	94	Vòng lặp 5) 18 > 19, 19 < 42, 42 < 55, 55 < 67, 67 < 94

Hình 1. Diễn biến từng vòng lặp của thuật toán

Ở vòng lặp 5, không tìm ra nghịch thế nào nên không xảy ra đổi chỗ, dãy đã được sắp xếp đúng thứ tự. Để biết khi nào hết nghịch thế, tức là đã sắp xếp xong, ta dùng một biến logic “có đổi chỗ” nhận giá trị **True** hay **False** tùy theo có xảy ra đổi chỗ hay không. Hình 2 là mã giả của thuật toán sắp xếp nổi bọt phiên bản thô nhất.

Nhận xét:

– Có thể chứng minh sau vòng lặp 1 thì số lớn nhất trong dãy sẽ được chuyển đến vị trí cuối dãy, tức là phần tử $a[n - 1]$ đã ở đúng vị trí. Như vậy vòng lặp 2 chỉ cần rà soát nghịch thế và đổi chỗ đến vị trí $n - 2$.

– Sau vòng lặp i thì phần tử $a[n - i - 1]$ đã ở đúng vị trí.

```
while (True):
    CoDoiCho ← False
    for i in { i | 0 ≤ i < n - 1 }:
        if a[i] > a[i + 1]:
            CoDoiCho ← True
            swap (a[i], a[i + 1])
    if (not CoDoiCho):
        return
```

Hình 2. Mã giả của thuật toán sắp xếp nổi bọt

③ Thuật toán sắp xếp chèn tuyến tính (Insertion Sort)

Ý tưởng sắp xếp chèn tuyến tính:

- Vì dãy con a_0 chỉ có một phần tử, nên dãy con này có thứ tự.
- Lặp lại việc chèn a_i với $1 \leq i < n$ như sau:

Xét dãy con $a_0, \dots, a_{i-1}$ đã có thứ tự, ta chèn a_i vào dãy con này sao cho dãy con sau khi chèn sẽ có thứ tự.

Mô tả thuật toán chèn tuyến tính:

Hình 3 minh họa diễn biến từng bước của thuật toán sắp xếp chèn tuyến tính.

Dãy (a)	a_0	a_1	a_2	a_3	a_4	a_5	Giải thích diễn biến từng bước
Ban đầu	55	19	42	94	18	67	1) Xét $a_1 = 19$. $55 > 19$. Lấy ra 19. Dịch chuyển 55 qua phải, chèn 19 vào trước 55.
Sau bước 1	19	55	42	94	18	67	2) Xét $a_2 = 42$. $55 > 42$; $19 < 42$. Lấy ra 42. Dịch chuyển 55 qua phải, chèn 42 vào trước 55.
Sau bước 2	19	42	55	94	18	67	3) Xét $a_3 = 94$. $55 < 94$. 94 đã nằm đúng chỗ.
Sau bước 3	19	42	55	94	18	67	4) Xét $a_4 = 18$. $94 > 18$; $55 < 18$; $42 > 18$; $19 > 18$. Lấy ra 18. Dịch chuyển dãy con 19,..., 94 qua phải, chèn 18 vào trước 19.
Sau bước 4	18	19	42	55	94	67	5) Xét $a_5 = 67$. $94 > 67$; $55 < 67$. Lấy ra 67. Dịch chuyển 94 qua phải, chèn 67 vào trước 94.
Sau bước 5	18	19	42	55	67	94	Sắp xếp xong.

Hình 3. Minh họa thuật toán sắp xếp chèn tuyến tính

Mô tả các bước của thuật toán như sau:

Bước 0. $i \leftarrow 1$;

Bước 1.

if $i \geq n$: kết thúc;

else: $val \leftarrow a_i$; $k \leftarrow i - 1$;

Bước 2.

if $k \geq 0$:

if $a_k > val$: $a_{k+1} \leftarrow a_k$; $k \leftarrow k - 1$; đến Bước 2;

Bước 3. $a_{k+1} \leftarrow val$; $i \leftarrow i + 1$; đến Bước 1;

Để “chèn a_i vào đúng chỗ của nó” trong dãy $a_0, a_1, \dots, a_{i-1}$ cần:

a) Tìm được chỗ đúng thứ tự của a_i : cho k đi từ $i - 1$ qua trái cho đến khi $a_k \leq a_i$ hoặc $k = -1$.

b) Lấy a_i ra khỏi dãy; dịch chuyển dãy $a_{k+1}, \dots, a_{i-1}$ sang phải một vị trí để có chỗ trống tại a_{k+1} ; đưa a_i vào a_{k+1} .

Mã giả thuật toán sắp xếp chèn tuyến tính

Thuật toán sắp xếp chèn tuyến tính kết hợp làm đồng thời hai việc a) và b) theo cách dịch chuyển dần từng bước sang trái, từ vị trí i tới vị trí $k+1$.

<pre> for i in $\text{range}(1, n)$: $val \leftarrow a_i$ $k \leftarrow i - 1$ while $k \geq 0$ and $a_k > val$: $a_{k+1} \leftarrow a_k$ $k \leftarrow k - 1$ $a_{k+1} \leftarrow val$ </pre>	<pre> # Chép a_i vào val # Xét dãy $a_0, \dots, a_{i-1}$ # Chép a_k vào a_{k+1} # k di chuyển qua trái # Chép val vào a_k </pre>
---	---

Hình 4. Mã giả của thuật toán sắp xếp chèn tuyến tính

Vòng lặp **for** bên ngoài kiểm soát việc thực hiện đúng $n - 1$ bước (xem Hình 4).

Vòng lặp **while** lồng bên trong thực hiện đồng thời cùng lúc hai việc a) và b) trong mỗi bước (xem Hình 4).



Em hãy thực hiện các công việc sau:

- 1) Tính số lần lặp của vòng lặp bên trong của thuật toán sắp xếp chèn tuyến tính.
- 2) Tính số lần lặp của vòng lặp ngoài của thuật toán sắp xếp chèn tuyến tính.
- 3) Ước lượng độ phức tạp thời gian của thuật toán sắp xếp chèn tuyến tính.

4 Thực hành

Nhiệm vụ 1. Em hãy viết chương trình Python thực hiện thuật toán sắp xếp nổi bọt.

Nhiệm vụ 2. Em hãy viết chương trình Python thực hiện thuật toán sắp xếp chèn tuyến tính dựa trên mã giả đã cho trong bài học.



Cho danh sách *bangDiem* là kết quả học tập gồm các cột *Họ và tên*, *điểm Toán*, *điểm Ngữ văn*, *điểm Tin học*, ... Hãy viết chương trình sắp xếp *bangDiem* theo điểm môn Tin học giảm dần.

Gợi ý: Mỗi phần tử của *bangDiem* là một danh sách con, ứng với một học sinh. So sánh theo thành phần thứ ba (cột điểm Tin học) của danh sách con để sắp xếp.



Theo em, thuật toán sắp xếp nổi bọt và thuật toán sắp xếp chèn, thuật toán nào đơn giản và dễ cài đặt hơn?

Tóm tắt bài học

- ✓ Thuật toán sắp xếp nổi bọt có hai vòng lặp lồng nhau: vòng lặp trong thực hiện đổi chỗ một lượt các cặp phần tử là nghịch thế, vòng lặp ngoài kiểm tra điều kiện “không xảy ra đổi chỗ”.
- ✓ Việc tìm vị trí chèn đúng chỗ trong thuật toán sắp xếp chèn có thể thực hiện bằng cách dịch dần từng bước.

BÀI TÌM HIỂU THÊM

CÁC HÀM PYTHON SẮP XẾP DÃY SỐ

Python có hai hàm để sắp xếp dãy số:

Hàm **sorted**: Sắp xếp dãy đầu vào theo thứ tự tăng dần. Hàm trả về một dãy mới gồm các phần tử trong dãy cũ nhưng đã sắp xếp lại theo yêu cầu. Dãy cũ vẫn còn đó.

Hàm **sort**: Thực hiện sắp xếp tại chỗ, tức là đổi chỗ các phần tử trong dãy. Hàm trả về **None** để tránh nhầm lẫn.

Chú ý: **sort** chỉ áp dụng cho danh sách; trái lại, **sorted** chấp nhận đầu vào tổng quát hơn.

Cú pháp:

```
sorted (dãy_số, key = ..., reverse = ...)
danh_sách.sort (key = ..., reverse = ...)
```

Trong đó:

reverse = True nghĩa là xếp theo thứ tự giảm dần, mặc định là **reverse = False**.
key dùng để xác định một hàm (**key function**) được áp dụng cho mỗi phần tử trong dãy trước khi làm phép so sánh.

Học xong bài này, em sẽ:

- ✓ Hiểu được ý tưởng của thuật toán sắp xếp nhanh.
- ✓ Viết được chương trình thực hiện sắp xếp nhanh một dãy số dựa trên các mã lệnh thuật toán phân đoạn cho trước.



Nếu cần chọn một trong hai việc sau đây, em sẽ chọn làm việc nào? Vì sao?

- 1) Từ mô tả thuật toán bằng liệt kê các bước, viết chương trình Python thực hiện thuật toán.
- 2) Từ chương trình Python thực hiện thuật toán, viết lại ngắn gọn ý tưởng chính của thuật toán.

1) **Lược đồ phân đoạn trong sắp xếp nhanh**

Thuật toán sắp xếp nhanh (Quick Sort)

Thuật toán theo chiến lược *chia để trị*, lặp lại nhiều lần việc *phân đoạn* dãy đầu vào thành hai đoạn con.

Sau một lần phân đoạn, chỉ cần sắp xếp trong nội bộ hai đoạn con. Bài toán sắp xếp ban đầu được chia thành hai bài toán con nhỏ hơn. Việc phân đoạn lặp lại nhiều lần. Hai đoạn con lại được tiếp tục phân đoạn thành 4 đoạn con, 8 đoạn con,... cho đến khi tất cả các đoạn con đều chỉ còn không quá một phần tử. Dãy ban đầu được sắp xếp xong.

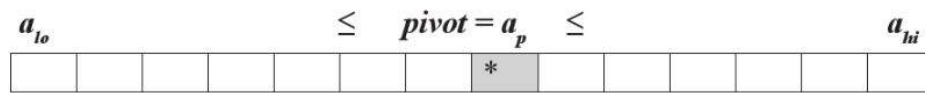
Lược đồ phân đoạn dãy số

– Lấy giá trị của một phần tử trong dãy làm pivot (giá trị chốt). Giá trị pivot có thể là bất cứ phần tử nào trong dãy. Phần tử này cũng được gọi là pivot cho ngắn gọn.

– Kết quả phân đoạn (*Hình 1*): đoạn con ở nửa dãy bên trái chỉ gồm các phần tử nhỏ hơn hay bằng pivot; đoạn con nửa dãy bên phải chỉ gồm các phần tử lớn hơn hay bằng pivot, phần tử làm pivot được chuyển đến đúng vị trí phân tách hai đoạn.

– Hàm thực hiện phân đoạn cần trả về vị trí phân tách dãy thành hai đoạn con vì sau đó sẽ sắp xếp chỉ trong nội bộ hai đoạn con.

Sau khi phân đoạn, chỉ cần sắp xếp trong nội bộ hai đoạn con thì hoàn thành việc sắp xếp cả dãy số. Bài toán sắp xếp ban đầu được chia thành hai bài toán con nhỏ hơn.



Hình 1. Kết quả phân đoạn: đoạn trái = $\{i \mid lo \leq i \leq p - 1\}$, đoạn phải = $\{j \mid p + 1 \leq j \leq hi\}$
 Có những lược đồ phân đoạn khác nhau để đạt mục đích trên.

2 Thuật toán sắp xếp nhanh áp dụng phân đoạn Lomuto

Ý tưởng thuật toán Lomuto: Chọn pivot là giá trị phần tử đứng cuối dãy số. Duyệt chỉ số i ở vị trí phân tách; duyệt dãy số bằng một chỉ số j khác và đảo giá trị các phần tử sao cho các phần tử ở vị trí từ $i - 1$ về đầu mút trái nhỏ hơn hay bằng pivot; các phần tử từ vị trí $i + 1$ đến j lớn hơn pivot, riêng phần tử ở vị trí i đúng bằng pivot.

Hình 2 là mã giả của thuật toán Lomuto phân đoạn dãy số a ; lo là chỉ số đầu mút trái; hi là chỉ số đầu mút phải.

```

pivot ← a[hi]           # pivot là phần tử cuối dãy số
i ← lo - 1              # i là vị trí phân tách
for j in {j | lo ≤ j ≤ hi}:
    if a[j] ≤ pivot:     # kiểm tra yêu cầu phân đoạn
        i ← i + 1       # tiến i một vị trí
        swap(a[i], a[j]) # đổi chỗ a[j] về vị trí phân tách
return i                # trả về vị trí phân tách
    
```

Hình 2. Mã giả thực hiện phân đoạn Lomuto

Mã lệnh Python thực hiện sắp xếp nhanh bằng phân đoạn Lomuto.

```

File Edit Format Run Options Window Help

1 def phandoanLomuto(a, lo, hi):
2     i = (lo-1)                #i là vị trí phân tách
3     pivot = a[hi]
4     for j in range(lo, hi):   #Duyệt dãy a[lo...hi]
5         if a[j] <= pivot:     #Phần tử a[j] <= pivot
6             i = i+1           #Tăng i lên,
7             a[i], a[j] = a[j], a[i] #Đảo giá trị a[i], a[j]
8     a[i+1], a[hi] = a[hi], a[i+1] #Đảo giá trị a[i+1], a[hi]
9     return (i+1)              #Hết vòng lặp, i là vị trí phân đoạn,
                                #a[i] = pivot
10
11 def quickSort(a, lo, hi):
12     if lo < hi:
13         p = phandoanLomuto(a, lo, hi) #p là vị trí phân tách
14                                         #a[p] đã ở đúng chỗ
15         quickSort(a, lo, p-1)          #Sắp xếp đoạn bên trái
16         quickSort(a, p+1, hi)          #Sắp xếp đoạn bên phải
    
```

Hình 3. Mã lệnh Python thuật toán sắp xếp nhanh sử dụng phân đoạn Lomuto



Dựa trên mã giả thuật toán phân đoạn Lomuto cho trong Hình 2, em hãy:

- 1) Mô tả diễn biến từng bước thực hiện phân đoạn Lomuto khi đầu vào là dãy 6 số nguyên tăng dần, ví dụ {1, 2, 3, 4, 5, 6}.
- 2) Tính số phép toán sơ cấp thực hiện phân đoạn Lomuto khi đầu vào là dãy tăng dần.

3) Thuật toán sắp xếp nhanh áp dụng phân đoạn Hoare

Lược đồ phân đoạn Hoare

Hoare là tác giả của thuật toán sắp xếp nhanh. Ý tưởng chính của thuật toán là đổi chỗ nhảy qua điểm phân tách (pivot), rà soát từ hai phía, trái và phải, cùng tiến dần từng bước vào giữa. Tạm dừng khi phát hiện phần tử vi phạm yêu cầu phân đoạn ở mỗi phía và đổi chỗ chúng cho nhau. Rà soát từ hai điểm tạm dừng đi vào giữa cho đến khi gặp nhau thì kết thúc việc phân đoạn. Điểm gặp nhau là vị trí phân tích dãy thành hai đoạn con.

Bước 1. $pivot \leftarrow a[lo]; i \leftarrow lo; j \leftarrow hi;$
 Bước 2.
 Tìm từ i qua phải đến khi $a[i] \geq pivot$;
 Tìm từ j qua phải đến khi $a[j] \leq pivot$;
 Bước 3.
 if $i < j$:
 Hoán đổi a_i với a_j ; cho i qua phải;
 cho j qua trái; đến Bước 2;
 else :
 Trả về j để tiếp tục phân đoạn 2 dãy con:
 $a[lo] \dots a[j]$ và $a[j+1] \dots a[hi]$
 Kết thúc;

Hình 4. Mã giả thực hiện phân đoạn Hoare

```
File Edit Format Run Options Window Help
1 def partitionHoare(a, lo, hi):
2     pivot = a[lo]
3     i, j = lo, hi
4     phanDoan = True
5     while phanDoan:                #Đang phân đoạn
6         #i qua phải đến khi a[i] >= pivot
7         while a[i] < pivot:
8             i = i + 1
9         #j qua trái đến khi a[j] <= pivot
10        while a[j] > pivot:
11            j = j - 1
12        if i < j:                    #i chưa gặp j
13            #Hoán đổi a[i] với a[j]
14            a[i], a[j] = a[j], a[i]
15            i = i + 1                #i qua phải
16            j = j - 1                #j qua trái
17        else:
18            phanDoan = False        #Kết thúc phân đoạn
19    return j
20 def quickSortHoare(a, lo, hi):
21     if lo < hi:
22         p = PartitionHoare(a, lo, hi)
23         quickSortHoare(a, lo, p)
24         quickSortHoare(a, p+1, hi)
```

Hình 5. Mã lệnh thực hiện phân đoạn Hoare

Hình 4 là mã giả của thuật toán sắp xếp nhanh áp dụng phân đoạn Hoare. Hình 5 là mã lệnh Python của thuật toán phân đoạn dãy số a , xuất phát với pivot là đầu dãy.

Thực hành

Nhiệm vụ 1. Viết chương trình thực hiện sắp xếp nhanh một dãy số và chạy thử kiểm tra.

- Dựa trên mã lệnh thuật toán cho trong Hình 3.
- Dựa trên mã lệnh thuật toán cho trong Hình 5.

Nhiệm vụ 2. Bổ sung thêm các câu lệnh in kết quả trung gian vào các chương trình nói trên để có thể quan sát diễn biến từng bước thực hiện sắp xếp nhanh một dãy số.



Em hãy thực hiện các công việc sau:

- Sửa lại thủ tục phân đoạn để có hàm `quickSort_down` sắp xếp theo thứ tự giảm dần.

Gợi ý: Sửa đổi phép so sánh trong câu lệnh `if a[j] <= pivot:` thành `if a[j] >= pivot:`

- Tiếp tục sửa lại để có hàm `quickSort_tuple_down` sắp xếp danh sách các cặp, ví dụ (*tên học sinh*, *điểm môn học*) theo điểm môn học giảm dần.

Gợi ý: Sửa đổi đầu vào thành danh sách các cặp (tên học sinh, điểm môn học) và thực hiện so sánh theo điểm môn học.



Câu 1. Em hãy giải thích tại sao lại nói thuật toán sắp xếp nhanh (QuickSort) theo chiến lược “chia để trị”.

Câu 2. Theo em thì diễn biến từng bước sắp xếp nhanh một dãy số cụ thể dùng phân đoạn Lomuto sẽ giống hay sẽ khác với dùng phân đoạn Hoare?

Tóm tắt bài học

- ✓ Thuật toán sắp xếp nhanh có thể áp dụng một trong hai lược đồ phân đoạn: theo Lomuto hoặc theo Hoare.
- ✓ Lược đồ Lomuto thực hiện phân đoạn bằng cách kiểm tra theo một chiều từ trái sang phải, đổi chỗ và dịch chuyển dần vị trí phân tách hai dãy con cho đến khi thoả mãn yêu cầu phân đoạn.
- ✓ Lược đồ Hoare thực hiện phân đoạn bằng cách kiểm tra theo hai chiều, từ hai đầu dãy số tiến dần vào giữa, đổi chỗ để thoả mãn yêu cầu phân đoạn; kết thúc khi gặp nhau.

THIẾT KẾ CHƯƠNG TRÌNH TỪ TRÊN XUỐNG VÀ PHƯƠNG PHÁP MÔ ĐUN HOÁ

Học xong bài này, em sẽ:

- ✓ Giải thích được phương pháp thiết kế chương trình thành các mô đun cho một bài toán cụ thể.
- ✓ Nhận biết được lợi ích của phương pháp nêu trên: hỗ trợ làm việc đồng thời, dễ dàng bảo trì, phát triển chương trình và tái sử dụng các mô đun.



Có người nói lập trình cũng giống như giải bài tập toán, sau khi có ý tưởng thuật toán, ta viết từng dòng lệnh lần lượt từ đầu đến cuối. Theo em, nói như thế có đúng không? Vì sao?

1 Thiết kế chương trình từ trên xuống theo phương pháp mô đun hoá

Mô đun trong lập trình

Một số loại thiết bị, đồ dùng được thiết kế theo mô đun để dễ tháo lắp, vận chuyển, sửa chữa, thay thế. Tên gọi mô đun hàm ý đó là một bộ phận, một phần của một thiết bị hay chương trình phần mềm, nhưng được thiết kế tách thành một khối riêng biệt, có tính độc lập nhất định với những phần còn lại. Tính độc lập của mỗi mô đun cho phép khi cần thiết có thể thay thế mô đun đó bằng cái tương đương, trong khi vẫn giữ nguyên các mô đun khác. Một chương trình lớn có thể gồm nhiều tệp mã nguồn. Một tệp mã nguồn là một mô đun phần mềm. Một hàm do người lập trình tự viết cũng có thể coi là một mô đun vì mỗi hàm trong chương trình tách thành một đoạn riêng, tương đối độc lập với những phần còn lại của chương trình.

Phương pháp mô đun hoá

Mô đun hoá là cách nói khái quát về một phương pháp thông dụng để làm phần mềm nói chung và lập trình nói riêng. Áp dụng phương pháp mô đun hoá, người lập trình làm theo các giai đoạn sau:

Giai đoạn 1. Liệt kê các việc lớn: sử dụng các gạch đầu dòng nhằm cho biết cần làm gì để nhận được kết quả mong muốn, tuần tự từ nhập dữ liệu đến kết quả cuối cùng.

Giai đoạn 2. Thiết kế các hàm: phân chia mỗi bước lớn thành một vài công việc độc lập và thiết kế các hàm thực hiện từng công việc đó, xác định rõ tên hàm, đầu vào, đầu ra.

Giai đoạn 3. Viết các hàm: lập trình từng hàm theo thiết kế; kiểm thử, gỡ lỗi từng hàm để chắc chắn nó làm đúng chức năng.

Giai đoạn 4. Viết chương trình chính: thực hiện các bước theo a) bằng các câu lệnh gọi sử dụng các hàm vừa hoàn thành; chạy thử, kiểm tra tổng thể.

Lập trình theo phương pháp mô đun hoá dẫn đến kết quả là chương trình có một số hàm do người lập trình định nghĩa. Chương trình chính sẽ ngắn gọn, gồm một số câu lệnh để nhập dữ liệu, gọi sử dụng các hàm do người lập trình viết, xuất kết quả cuối cùng ra màn hình hay ra tệp và kết thúc. Trái với lập trình theo phương pháp mô đun hoá gọi là lập trình kiểu nguyên khối. Nếu một chương trình khá dài mà không có hàm nào do người lập trình tự định nghĩa thì có thể coi đó không phải là kết quả lập trình theo mô đun.

2 Minh hoạ về lập trình theo phương pháp mô đun hoá



- 1) Trong Bài 9 có cho sẵn mã lệnh Python thực hiện thuật toán sắp xếp nhanh sử dụng phân đoạn Lomuto. Theo em, đây có phải là kết quả lập trình theo phương pháp mô đun hoá hay không? Vì sao?
- 2) Trong Bài 9 có yêu cầu thực hành dựa trên mã lệnh thực hiện phân đoạn Hoare cho sẵn, để viết chương trình thực hiện sắp xếp nhanh một dãy số. Em hãy cho biết đã làm như thế nào? Theo em như thế có phải là lập trình theo phương pháp mô đun hoá hay không?

Một dự án lập trình nhỏ

Về lí thuyết, ước lượng độ phức tạp thời gian của thuật toán tìm kiếm tuần tự là $O(n)$ còn độ phức tạp thời gian của thuật toán tìm kiếm nhị phân là $O(\log_2 n)$. Để minh hoạ trực quan kết luận trên ta lập một dự án lập trình nhỏ nhằm thực nghiệm bấm giờ thực tế chạy máy tính nhiều lần với các dãy số đầu vào ngẫu nhiên rồi lấy giá trị trung bình. Dưới đây là ví dụ minh hoạ việc áp dụng phương pháp mô đun hoá trong thiết kế chương trình và lập trình cho dự án này.

Giai đoạn 1. Liệt kê các việc lớn

- Sinh dãy ngẫu nhiên n số gọi là dãy A .
- Sắp xếp dãy A theo thứ tự tăng dần (không giảm) lưu thành dãy B .
- Tìm kiếm tuần tự một số có mặt trong dãy A ; tìm kiếm nhị phân số đó trong dãy B ; ghi lại khoảng thời gian từ lúc bắt đầu tìm kiếm cho đến khi tìm thấy trong cả hai trường hợp.
- Tìm kiếm một số x bất kì: Cho số x bất kì, tìm kiếm tuần tự số x trong dãy A , tìm kiếm nhị phân số x trong dãy B , bấm giờ tính khoảng thời gian từ lúc bắt đầu tìm kiếm cho đến khi kết thúc tìm kiếm trong cả hai trường hợp.
- Tính trung bình cộng thời gian thực hiện tìm kiếm tuần tự và tìm kiếm nhị phân; xuất kết quả ra cho cả hai trường hợp.

Giai đoạn 2. Thiết kế các hàm

Phân tích chi tiết hơn những việc lớn cần làm trong từng bước kể trên thành các việc cụ thể hơn. Nếu một việc không thể thực hiện bằng một vài câu lệnh ngắn gọn thì thiết kế một hàm: đặt tên gọi nhớ hàm sẽ làm gì, xác định rõ đầu vào là gì, đầu ra là gì. Ví dụ:

– Sinh dãy ngẫu nhiên n số với giá trị trong khoảng $(0, M)$: Tên hàm **dayngaunhien**; đầu vào: hai số nguyên n, M ; đầu ra: một dãy n số được sinh ngẫu nhiên với giá trị trong khoảng $(0, M)$.

– Sắp xếp dãy A theo thứ tự tăng dần (không giảm) lưu thành dãy B : Tên hàm **sapxep**; đầu vào: một dãy số; đầu ra: dãy số được sắp theo thứ tự tăng dần.

– Tìm kiếm tuần tự: tên hàm **tktuantu**;

– Tìm kiếm nhị phân: tên hàm **tknhiphan**;

– Chọn số có mặt trong dãy: Sinh ngẫu nhiên một số nguyên i trong khoảng $(0, n - 1)$ và trả về $x = a_i$ (phần tử chỉ số i trong dãy A). Chắc chắn x cũng có mặt trong dãy B . Không cần viết thành một hàm.

– Tìm số có mặt trong dãy: tên hàm **tkcomat**.

Đầu vào: $x = a_i$ nói trên; hai dãy số A, B .

Đầu ra: khoảng thời gian từ lúc bắt đầu tìm kiếm cho đến khi tìm thấy:

1) Bảng tìm kiếm tuần tự trong dãy A .

2) Bảng tìm kiếm nhị phân trong dãy B .

– Tìm số x bất kì: tên hàm **tkbatki**.

Đầu vào: một số x sinh ngẫu nhiên; hai dãy số A, B .

Đầu ra: khoảng thời gian từ lúc bắt đầu tìm kiếm cho đến khi kết thúc tìm kiếm:

1) Bảng tìm kiếm tuần tự trong dãy A .

2) Bảng tìm kiếm nhị phân trong dãy B .

– Ghi lại khoảng thời gian tìm kiếm: dùng hàm **time** hai lần, ngay trước và ngay sau câu lệnh gọi **tktuantu** (hay **tknhiphan**). Không cần viết thành hàm riêng.

– Tính trung bình cộng thời gian thực hiện tìm kiếm tuần tự và tìm kiếm nhị phân và xuất kết quả ra: không cần viết thành hàm riêng.

Người lập trình chỉ bắt đầu viết các câu lệnh sau khi đã thiết kế xong các hàm.

Với Giai đoạn 3 và Giai đoạn 4, học sinh thực hiện dựa vào các kiến thức đã học đồng thời làm theo gợi ý của thầy, cô giáo.

③ Các ưu điểm của lập trình theo phương pháp mô đun hoá

– *Phối hợp cùng lập trình*: Ví dụ, một nhóm học sinh làm dự án lập trình mà kết quả là một chương trình không nhỏ, cần viết nhiều hàm, sử dụng những thuật toán khác nhau,... thì cần tổ chức thành vài tệp mã nguồn riêng biệt. Hai người không thể viết cùng một lúc vào cùng một tệp để lập trình.

– *Chương trình dễ hiểu hơn*: Lập trình theo phương pháp mô đun hoá dẫn đến kết quả là chương trình chính thường ngắn gọn, gồm một số câu lệnh gọi sử dụng các hàm mà người lập trình viết. Mỗi hàm thường có tên gọi nhớ cho biết nó làm gì. Các câu lệnh trong chương trình chính thể hiện rõ các bước lớn đi từ dữ liệu đầu vào đến kết quả cuối cùng.

– *Dễ kiểm thử và sửa lỗi hơn*: Phương pháp mô đun hoá tách biệt các công việc nên dễ thấy lỗi xảy ra khi xử lý việc gì, ở phần chương trình thực hiện hàm nào. Một hàm là một đoạn chương trình ngắn, được thiết kế có đầu vào, đầu ra rõ ràng, chạy thử nhanh hơn và dễ tìm ra câu lệnh có lỗi hơn.

– *Khả năng tái sử dụng*: Những hàm do người lập trình tự định nghĩa có thể được dùng không chỉ trong chương trình vừa hoàn thành mà còn ở những chương trình khác sau này. Các môi trường lập trình đều hỗ trợ việc tạo lập thư viện các hàm do người lập trình tự định nghĩa.



Câu 1. Em hãy nêu ngắn gọn về lập trình mô đun hoá theo ý hiểu của mình.

Câu 2. Theo em, từ một chương trình có các hàm do người lập trình tự định nghĩa, có thể bỏ hết các hàm này để chuyển thành chương trình kiểu nguyên khối hay không? Việc này dễ hay khó?



Xét dự án nhỏ về lập trình để thực nghiệm so sánh thời gian thực tế chạy chương trình máy tính thực hiện một số thuật toán sắp xếp mà em đã biết theo cách bấm giờ chạy máy với các dãy số đầu vào ngẫu nhiên rồi lấy giá trị trung bình. Em hãy áp dụng phương pháp lập trình mô đun hoá:

a) Đưa ra thiết kế các hàm sẽ được sử dụng trong chương trình.

b) Viết các câu lệnh trong chương trình chính (không cần viết các hàm).



Câu 1. Hãy nêu các bước người lập trình cần thực hiện khi áp dụng phương pháp lập trình mô đun hoá.

Câu 2. Hãy nêu các ưu điểm của lập trình theo mô đun.

Tóm tắt bài học

- ✓ Lập trình theo phương pháp mô đun hoá là chia chương trình thành một số hàm (chương trình con) tách biệt để có thể viết mã lệnh, kiểm thử, gỡ lỗi từng hàm.
- ✓ Các hàm do người lập trình tự định nghĩa là kết quả của việc lập trình theo mô đun.

Học xong bài này, em sẽ:

- ✓ Vận dụng được phương pháp thiết kế chương trình thành các mô đun (hàm) cho một bài toán cụ thể.
- ✓ Viết được chương trình thực hiện một số hàm theo thiết kế.

1 **Nhiệm vụ, yêu cầu kết quả và tổ chức thực hiện**

Nhiệm vụ:

Thực hiện bài tập lớn về lập trình Python. Mô tả bài toán:

Dữ liệu đầu vào: Tập phần mềm bảng tính chứa dữ liệu là điểm tổng kết các môn học của lớp 11A, gồm các cột: *Họ và đệm*, *Tên*, *Điểm Toán*, *Điểm Ngữ văn*, *Điểm Tin học*, ... Để đơn giản, ta chưa xét cột *Họ và đệm* và giả thiết mỗi ô trong cột *Tên* là một từ, không có dấu cách; các tên cột bỏ bớt chữ “*Điểm*” và chỉ còn một từ cho ngắn gọn.

Kết quả (KQ) đầu ra:

KQ1 – Phân tích kết quả học tập của từng học sinh: điểm trung bình chung; điểm cao nhất; điểm thấp nhất, số lượng điểm thuộc các mức (Tốt, Khá, Đạt, Chưa đạt). Ghi lưu thành tệp văn bản “phantich_theoHS.txt”.

KQ2 – Phân tích kết quả học tập theo từng môn học; ghi lưu thành tệp văn bản “phantich_theoMon.txt”.

a) Danh sách sắp xếp điểm mỗi môn học theo thứ tự giảm dần, kèm tên học sinh.

b) Điểm cao nhất, điểm thấp nhất, trung bình cộng, tỉ lệ phần trăm điểm theo các mức: Tốt, Khá, Đạt, Chưa đạt.

KQ3 – Lập danh sách học sinh để xét khen thưởng; ghi lưu thành tệp văn bản “xetKhenThuong.txt” gồm hai cột *Tên*, *chamDiem*. Quy tắc chấm điểm:

a) Cứ mỗi điểm môn học đạt mức Tốt, *chamDiem* được cộng thêm 1 điểm.

b) Mỗi điểm môn học dưới mức Khá, *chamDiem* bị trừ 1 điểm.

Yêu cầu kết quả:

Với mục đích luyện kỹ năng lập trình, mỗi nhóm cần hoàn thành hai sản phẩm chương trình SP#1 và SP#2 với yêu cầu như sau:

– SP#1: tự viết các hàm (mô đun) chương trình, kế thừa những kết quả lập trình đã có được đến nay.

– SP#2: sử dụng tối đa các hàm đã có sẵn trong Python để hoàn thành nhiệm vụ.

Tổ chức thực hiện:

- Lập các nhóm dự án, mỗi nhóm khoảng 5 đến 6 học sinh; chọn nhóm trưởng.
- Cả nhóm cùng thực hiện Nhiệm vụ 1:
 - + Đọc hướng dẫn để biết cách phân tích, lựa chọn và thiết kế các hàm. Hướng dẫn chỉ là gợi ý, không bắt buộc phải theo.
 - + Thảo luận, đưa ra thiết kế cuối cùng sau các sửa đổi, điều chỉnh.
- Dựa trên danh sách các hàm cần thực hiện, xác định các nhiệm vụ cụ thể; phân công mỗi nhiệm vụ (các bài thực hành tiếp theo) cho 1 đến 2 học sinh đảm nhiệm.
- Nhóm trưởng phụ trách tích hợp các kết quả thành chương trình hoàn chỉnh với sự cộng tác của các thành viên khác, dưới sự hỗ trợ của thầy, cô giáo.

2 Các bài thực hành

Nhiệm vụ 1. Phân tích thiết kế chương trình theo mô đun

Yêu cầu:

Dựa trên mô tả kết quả đầu ra của bài tập lớn, hãy thiết kế một số hàm (mô đun chương trình) đáp ứng các yêu cầu sau:

- Với mỗi hàm, xác định rõ chức năng hàm sẽ làm gì, tên hàm, đầu vào, đầu ra và nêu cụ thể trả về chương trình chính cái gì (nếu có trả về).
- Nêu cách phối hợp các hàm này trong chương trình chính để nhận được tất cả các kết quả đầu ra đã mô tả trong *KQ1*, *KQ2* và *KQ3*.

Hướng dẫn thực hiện:

Giai đoạn 1. Liệt kê các việc lớn để nhận được các kết quả KQ1, KQ2 và KQ3

1. *Đọc dữ liệu từ tệp:* Tổ chức dữ liệu trong chương trình bằng các kiểu dữ liệu của Python sao cho thuận tiện để thực hiện các việc tiếp theo;

2. *Phân tích dãy điểm từng học sinh để có KQ1 và KQ3;* viết kết quả vào các tệp “*phantich_theoHS.txt*”, “*xetKhenThuong.txt*”;

3. *Với mỗi môn học, sắp xếp dãy điểm để có KQ2a;* viết kết quả vào tệp “*phantich_theoMon.txt*”;

4. *Với mỗi môn học, phân tích dãy điểm để có KQ2b;* viết kết quả vào tệp “*phantich_theoMon.txt*”.

Giai đoạn 2. Thiết kế các hàm

1. Đọc dữ liệu từ tệp

Dữ liệu đầu vào chứa trong một tệp, đọc vào từng dòng và xử lý không phức tạp. Có thể viết một hàm thực hiện việc này. Đặt tên hàm: ví dụ là **nhapTuTep**.

Đầu vào: tệp phần mềm bảng tính chứa dữ liệu như mô tả ở đầu bài học.

Đầu ra: dữ liệu trong chương trình được tổ chức như sau:

– Mảng hai chiều các điểm số: Mảng $n \times m$, mỗi hàng là dãy điểm của một học sinh, sẵn sàng để phân tích kết quả cho từng học sinh.

– Cột *Tên* trong bảng kết quả học tập tạo thành danh sách các tên học sinh để ghép với từng cột điểm số môn học, tách riêng được kết quả học tập theo từng môn.

– Hàng các tên môn học tạo thành danh sách tên môn học để dễ dàng lấy ra từng tên môn học theo chỉ số cột.

2. Phân tích điểm theo học sinh

Có thể tách thành các việc nhỏ, cụ thể hơn như sau:

2a) Phân tích dãy điểm số (là một hàng của mảng hai chiều) để có *KQ1*:

Thiết kế một hàm và đặt tên, ví dụ là **ptDiem**.

Đầu vào: một dãy điểm số.

Đầu ra: trả về **sum**, **max**, **min**, số lượng điểm thuộc các mức xếp hạng Tốt, Khá, Đạt, Chưa đạt.

2b) Xét khen thưởng:

Nếu *chamDiem* > 0 thì viết thêm (*tên*, *chamDiem*) thành một dòng vào tệp “xetKhenThuong.txt”; có thể thực hiện việc này bằng một vài câu lệnh ngắn gọn, không cần viết thành một hàm riêng.

Lặp lại các việc 2a) và 2b) cho mỗi hàng trong mảng hai chiều $n \times m$ sẽ hoàn thành phân tích điểm cho toàn bộ học sinh và lập xong danh sách học sinh được xét khen thưởng.

Có thể thiết kế thân vòng lặp thành một hàm và đặt tên, ví dụ là **ptHocSinh**.

Đầu vào: Một hàng trong mảng hai chiều $n \times m$ (một dãy điểm số).

Đầu ra:

– Thêm một dòng vào tệp “phantich_theoHS.txt” (gọi hàm **ptDiem**);

– Thêm (*tên*, *chamDiem*) vào tệp “xetKhenThuong.txt” nếu *chamDiem* > 0.

3. Phân tích điểm theo môn học

3a) Chuẩn bị đầu vào để sẵn sàng phân tích điểm theo môn học:

Dãy điểm số một môn học là một cột của mảng hai chiều $n \times m$ không sẵn có ngay như một danh sách Python. Cũng chưa có sẵn danh sách các cặp (*tên*, *điểm*) là kết quả của mỗi môn học (ở đây *tên* là tên học sinh).

Thiết kế một hàm, đặt tên ví dụ là **tachMon**:

- *Đầu vào*: dữ liệu trong chương trình (sau khi đọc từ tệp vào)
- *Đầu ra*: trả về tên danh sách dãy điểm số một môn học và tên danh sách các cặp (*tên, điểm*) cho môn học đó.

3b) Phân tích điểm một môn học:

Nhận thấy rằng yêu cầu kết quả đầu ra *KQ1* và *KQ2b* là tương tự như nhau. Hàm **ptDiem** sử dụng được cho cả hai việc, phân tích điểm từng học sinh và phân tích điểm từng môn học.

3c) Sắp xếp danh sách các cặp (*tên, điểm*) theo thứ tự *điểm* giảm dần để có *KQ2a*:

Ta đã viết một số chương trình thực hiện các thuật toán sắp xếp dãy số. Có thể cải biên để nhận được một hàm thực hiện sắp xếp danh sách các cặp (*tên, điểm*) theo thứ tự *điểm* giảm dần.

Lặp lại các việc 3b) và 3c) cho mỗi cột trong mảng hai chiều $n \times m$ sẽ hoàn thành phân tích điểm cho toàn bộ các môn học. Có thể thiết kế một hàm nhận kết quả từ **tachMon** và thực hiện 3b) và 3c) cho một môn học; đặt tên, ví dụ là **ptMonHoc**.

- *Đầu vào*: danh sách điểm một môn học và danh sách các cặp (*tên, điểm*).
- *Đầu ra*:
- + Thêm một dòng vào tệp “phantich_theoMon.txt” (gọi hàm **ptDiem**);
- + Thêm danh sách các cặp (*tên, điểm*) theo thứ tự *điểm* giảm dần vào tệp “phantich_theoMon.txt” (gọi hàm sắp xếp đã cải biên).

Nhiệm vụ 2. Đọc dữ liệu từ tệp và tổ chức dữ liệu trong chương trình

Yêu cầu:

Viết chương trình thực hiện hàm **nhapTuTep** và chạy thử kiểm tra.

Hướng dẫn thực hiện:

– Tạo tệp dữ liệu đầu vào: Một cách đơn giản là cắt dán cả khối ô cần thiết từ cửa sổ phần mềm bảng tính điện tử vào tệp đang soạn thảo trong IDE của Python. Lưu thành tệp có đuôi tên “txt”. Để tiện trình bày, ta đặt tên tệp đầu vào, ví dụ là “bangDiem.txt”. Bổ sung thêm vào dòng đầu tiên của tệp hai số nguyên dương n, m là *số học sinh* và *số môn học*.

- Mở tệp ở chế độ “đọc”;
- Viết các câu lệnh đọc dữ liệu từ tệp: kế thừa, sử dụng các câu lệnh đã viết trong các bài thực hành về cấu trúc mảng một và hai chiều. Kết quả đầu ra:

- + Danh sách *tenHS*: từ cột *Tên* của *bangDiem*.
- + Danh sách *tenMon*: từ hàng tên cột của *bangDiem*.
- + Mảng hai chiều $n \times m$, mỗi hàng là dãy điểm của một học sinh.
- Đóng tệp sau khi đọc xong.

Nhiệm vụ 3. Tách riêng kết quả học tập từng môn học

Yêu cầu:

Viết chương trình thực hiện hàm **tachMon** và chạy thử kiểm tra.

Hướng dẫn thực hiện:

Đầu vào của hàm này là kết quả thực hiện **nhapTuTep**, gồm có danh sách *tenHS* (các tên học sinh), danh sách *tenMon* (các tên môn học) và mảng hai chiều $n \times m$.

- Đọc từng cột của mảng hai chiều để có dãy số các điểm mỗi môn học.
- Ghép tương ứng mỗi tên học sinh từ danh sách *tenHS* với mỗi điểm môn học sẽ thành danh sách các cặp (*tên*, *điểm*) cho môn học đó.
- Trả về tên danh sách dãy điểm số môn học và tên danh sách các cặp (*tên*, *điểm*).

Nhiệm vụ 4. Sắp xếp kết quả một môn học theo thứ tự giảm dần

Yêu cầu:

Cải biên một hàm thực hiện thuật toán sắp xếp nào đó, ví dụ sắp xếp nhanh **quickSort** thành hàm **quickSort_tuple_down** để sắp xếp một danh sách các cặp (*tên*, *điểm*) theo thứ tự điểm giảm dần.

Hướng dẫn thực hiện:

- Cải biên hàm **phandoanLomuto** thành hàm **phandoanLomuto_tuple** để sắp các cặp (*Tên*, *điểm môn học*) theo thành phần điểm môn học.
- Trong hàm **phandoanLomuto_tuple** đảo chiều phép so sánh trong câu lệnh **if** từ “**<=**” thành “**>=**” để sắp thứ tự giảm dần; đặt tên hàm mới là **phandoanLomuto_tuple_down**.
- Dùng hàm **phandoanLomuto_tuple_down** để cải biên **quickSort** thành hàm **quickSort_tuple_down**.



Chạy chương trình thực hiện **quickSort_tuple_down** vừa hoàn thành với đầu vào là kết quả môn Tin học của lớp em và cho biết kết quả: điểm cao nhất; những bạn có điểm cao nhất; điểm thấp nhất.

Học xong bài này, em sẽ:

- ✓ Viết được chương trình thực hiện một số hàm theo thiết kế.
- ✓ Sử dụng các hàm đã viết để lập trình giải bài toán thực tế.

Nhiệm vụ 1. Viết hàm phân tích điểm

Yêu cầu:

Viết chương trình thực hiện hàm **ptDiem** và chạy thử kiểm tra.

Hướng dẫn thực hiện:

Tách thành các việc cụ thể:

- Đếm số điểm thuộc mỗi mức xếp hạng Tốt, Khá, Đạt, Chưa đạt.
- Tìm **sum**, **max**, **min**.

Có hai lựa chọn viết chi tiết các câu lệnh: 1-Duyệt dãy điểm số đầu vào nhiều lần, mỗi lần làm một việc hoặc 2-Duyệt dãy điểm số đầu vào chỉ một lần, làm đồng thời nhiều việc trong một lần duyệt.

- Trả về các giá trị: điểm trung bình, max, min, số điểm thuộc mỗi mức xếp hạng.

Nhiệm vụ 2. Thực hiện phân tích điểm một học sinh

Yêu cầu:

Viết chương trình thực hiện hàm **ptHocSinh** và chạy thử kiểm tra.

Hướng dẫn thực hiện:

- Gọi hàm **ptDiem**; viết kết quả vào tệp “phantich_theoHS.txt”.
- Theo kết quả đếm số điểm thuộc mỗi mức xếp hạng Tốt, Khá, Đạt, Chưa đạt, nếu *chamDiem* > 0 thì viết thêm tên học sinh vào tệp “xetKhenThuong.txt”;
- Định dạng in ra số thực bằng một trong các cách sau:
 - + Dùng hàm **round()** làm tròn số trước khi in ra để làm tròn, chỉ giữ lại *d* chữ số phần lẻ sau dấu chấm thập phân.

```
File Edit Format Run Options Window Help
1 round(số_thực, d) #Ví dụ, nếu số_thực = 345.678
2                     #thì round(x, 2) = 345.68
3 ...
```

- + Dùng lệnh in có giữ chỗ bằng “{ }” để định dạng bằng hàm **format**. Ví dụ, nếu muốn in ra số thực với 2 chữ số phần lẻ thì giữ chỗ bằng { : .2f }.

```
File Edit Format Run Options Window Help
1 kq = 'số dư tài khoản là {:.2f} VNĐ'.format(số_thực)
2 print(kq) #In ra 'số dư tài khoản là 345.68 VNĐ'
```

Nhiệm vụ 3. Thực hiện phân tích điểm một môn học

Yêu cầu:

Viết chương trình thực hiện hàm **ptMonHoc** và chạy thử kiểm tra.

Hướng dẫn thực hiện:

- Gọi hàm **ptDiem**; viết kết quả vào tệp “phantich_theoMon.txt”.
- Gọi hàm **quickSort_tuple_down**; viết kết quả vào tệp “phantich_theoMon.txt”.
- Chạy thử với đầu vào là hai danh sách: 1-Danh sách số thực; 2-Danh sách các cặp (tên người, số thực).

Nhiệm vụ 4. Phối hợp các hàm đã viết thành chương trình chính

Yêu cầu:

Viết chương trình chính và chạy thử kiểm tra.

Hướng dẫn thực hiện:

- Gọi hàm **nhapTuTep**.
- Mở tệp ở chế độ “viết” và gán làm đầu ra chuẩn (để có thể xuất kết quả ra bằng lệnh **print**).

```
File Edit Format Run Options Window Help
1 fo = open("tên_tệp.txt", "w", encoding = "utf-8")
2 #Đề viết và dùng kí tự tiếng Việt
3 sys.stdout = fo #Xuất ra tệp thay cho màn hình
4 #lệnh print sẽ viết nối thêm vào tệp
5 ...
```

- Lập theo i là chỉ số hàng của mảng (danh sách) 2 chiều $n \times m$ thực hiện **ptHocSinh**.
- Lập theo k là chỉ số cột ứng với điểm các môn học, thực hiện **tachMon** cho môn học k , thực hiện **ptMonHoc**.
- Đóng tệp.



Câu 1. Chạy chương trình thực hiện hàm **ptDiem** vừa hoàn thành với đầu vào là kết quả học tập của em; cho biết kết quả.

Câu 2. Chạy chương trình thực hiện hàm **ptMonHoc** vừa hoàn thành với đầu vào là kết quả học tập môn Tin học của lớp em; cho biết kết quả.

Học xong bài này, em sẽ:

Sử dụng được một số hàm thư viện có sẵn của Python để giải bài toán thực tế.

Nhiệm vụ và yêu cầu kết quả

Thực hiện phương án sử dụng tối đa các hàm đã có sẵn trong Python để tạo ra SP#2: Thay vì ưu tiên viết trực tiếp các câu lệnh chi tiết như trong Bài 11 và Bài 12, bài này yêu cầu thay thế bằng việc gọi sử dụng các hàm có sẵn trong Python mỗi khi có thể.

Nhiệm vụ 1. Viết phiên bản thứ hai cho hàm thực hiện phân tích dãy điểm***Yêu cầu:***

Viết chương trình thực hiện hàm **ptDiem** theo phương án sử dụng tối đa các hàm có sẵn trong Python và chạy thử kiểm tra.

Hướng dẫn thực hiện:

- Sử dụng hàm **sum** để tính tổng và điểm trung bình.
- Gọi hàm Python thực hiện sắp xếp thứ tự tăng dần (không giảm); sau khi sắp xếp thì tìm được ngay **max**, **min**.
- Dãy số đã sắp thứ tự tăng dần (không giảm) nên có thể dùng hàm **bisect_left** (trong mô đun **bisect**) tìm được các vị trí phân chia dãy điểm thành 4 đoạn điểm: Chưa đạt, Đạt, Khá và Tốt. Từ đó tính được số lượng điểm theo từng mức xếp hạng.



```

1  def ptDiem_bisect(ds):
2      ...
3      chuadat = bisect_left(ds,DAT) #Số điểm ở mức chưa đạt
4      dat = bisect_left (ds,KHA) - chuadat #Số điểm ở mức đạt
5      ...

```

Hình 1. Trích một số câu lệnh định nghĩa hàm **ptDiem_bisect**

Nhiệm vụ 2. Viết phiên bản thứ hai của hàm **ptMonHoc*****Yêu cầu:***

Trong hàm **ptMonHoc** thay vì sử dụng **quickSort_tuple_down** cải biên từ hàm **quicksort** hãy gọi sử dụng các hàm có sẵn trong Python.

Hướng dẫn thực hiện:

– Python có sẵn phương thức **sort**, hàm **sorted** với tham biến **key = lambda** để sắp xếp danh sách các cặp hay các bộ nhiều thành phần, theo giá trị của một thành phần bất kì trong bộ. Cách gọi sử dụng để sắp xếp danh sách các cặp (*tên, điểm*) theo thứ tự điểm giảm dần theo mẫu tham khảo *Hình 2*:

```
File Edit Format Run Options Window Help
1 dsGiam = sorted(ds, reverse = True, key = lambda bangGhi: bangGhi[1]))
```

Hình 2. Minh họa thực hiện sắp xếp bằng hàm **sorted**

- Trong thân hàm **ptMonHoc** sửa lại lời gọi hàm thực hiện việc sắp xếp, sử dụng hàm của Python;
- Để dùng hàm **bisect_left** cần sắp thứ tự tăng dần;
- Khi dãy đã sắp thứ tự tăng dần thì chỉ cần viết lại theo thứ tự đảo ngược để có dãy giảm dần.



Chạy chương trình thực hiện hàm **ptDiem_bisect** vừa hoàn thành với đầu vào là kết quả môn Tin học của lớp em và cho biết: Số lượng điểm của mức nào là nhiều nhất? Nên xếp hạng kết quả học tập môn Tin học chung của cả lớp ở mức nào?

BÀI TÌM HIỂU THÊM

ZEN OF PYTHON

Kỹ sư phần mềm Tim Peters đã tập hợp 19 “nguyên tắc hướng dẫn” để lập trình “tốt” và đăng lên danh sách gửi thư của nhóm phát triển Python vào năm 1999. Những nguyên tắc này đã ảnh hưởng đến thiết kế của ngôn ngữ lập trình Python. Danh sách của Peters đề nghị nguyên tắc thứ 20 “để Guido điền vào”, dành cho Guido van Rossum – tác giả ban đầu của ngôn ngữ Python.

“Zen of Python” của Peters đã được đưa vào mục số 20 trong “Bộ Đề nghị Cải tiến Python chính thức” và được phát hành công cộng. Trong cửa sổ trình thông dịch Python, có thể gõ nhập “import this” để hiển thị toàn văn. Dưới đây phỏng dịch một số điểm chính:

- Tường minh tốt hơn hiểu ngầm.
- Đơn giản tốt hơn phức hợp.
- Phức hợp tốt hơn phức tạp.
- Tuân tự tốt hơn lòng nhau.
- Viết câu lệnh rõ ràng tốt hơn viết rút ngắn cô đặc.
- Chương trình dễ đọc là quan trọng.
- Không “im lặng bỏ qua” tiềm năng lỗi.

Học xong bài này, em sẽ:

- ✓ Tạo được một thư viện nhỏ.
- ✓ Viết được chương trình sử dụng thư viện vừa tạo ra.

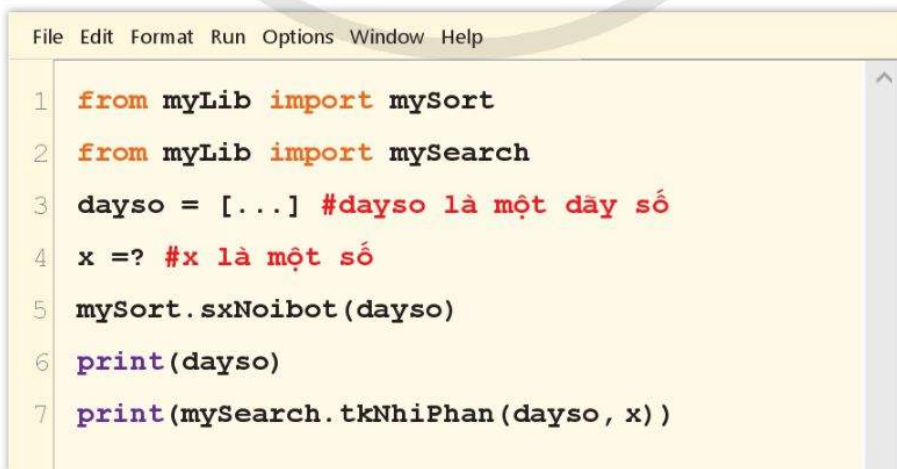
Nhiệm vụ 1. Tổ chức thư viện các hàm người lập trình tự viết

Yêu cầu:

Tạo được thư viện **myLib** gồm một số hàm thực hiện các thuật toán sắp xếp, tìm kiếm do ta tự viết trong các bài học trước.

Hướng dẫn thực hiện:

- Tạo thư mục dự án **myPrj** chứa thư mục con **myLib** là thư viện các hàm ta tự viết.
- Trong thư mục **myLib**, tạo một tệp rỗng, có tên “_init_.py”. Nếu có tệp này, Python biết đây sẽ là một gói chứa một số tệp mã nguồn.
- Trong thư mục **myLib**, tạo hai tệp “mySort.py” và “mySearch.py”. Sao chép mã lệnh của các hàm thực hiện sắp xếp, tìm kiếm mà ta đã viết thành công vào hai tệp tương ứng. Mỗi hàm bắt đầu từ câu lệnh **def** định nghĩa hàm đó cho đến hết toàn bộ cả hàm.
- Thử sử dụng **myLib** như một thư viện: Viết tệp chương trình “demoLib.py” bắt đầu với 2 dòng lệnh **import** khai báo sử dụng thư viện. Tham khảo mã lệnh trong *Hình 1*; Chú ý thay dấu “...” bằng một dãy số và dấu “?” bằng một số cụ thể.



```

File Edit Format Run Options Window Help

1 from myLib import mySort
2 from myLib import mySearch
3 dayso = [...] #dayso là một dãy số
4 x =? #x là một số
5 mySort.sxNoibot(dayso)
6 print(dayso)
7 print(mySearch.tkNhiPhan(dayso, x))
  
```

Hình 1. Minh họa cách sử dụng thư viện do người lập trình tạo ra

Nhiệm vụ 2. Sử dụng thư viện vừa tạo ra

Yêu cầu:

Sử dụng thư viện vừa tạo ra để viết phiên bản mới cho chương trình chính của Bài tập lớn.

Hướng dẫn thực hiện:

Mở văn bản chương trình sản phẩm SP#1; làm các việc sau:

- Bổ sung hai dòng khai báo sử dụng thư viện **myLib**.
- Rà soát từ đầu văn bản chương trình và cắt bỏ phần mã nguồn của các hàm đã có trong thư viện **myLib**.
- Nếu phát hiện còn hàm ta tự viết để thực hiện sắp xếp, tìm kiếm được sử dụng trong chương trình mà chưa có trong thư viện **myLib** thì cắt dán mã nguồn vào **myLib**.
- Chạy thử chương trình.



Chỉnh sửa các chi tiết để có thể áp dụng chương trình phân tích kết quả học tập theo các quy định của trường em.

BÀI TÌM HIỂU THÊM

KHAI THÁC THƯ VIỆN CỦA PYTHON

Python có nhiều mô đun. Bộ cài đặt Python cho nền tảng Windows chỉ tích hợp sẵn các mô đun tiêu chuẩn được viết bằng ngôn ngữ C. Ngoài các thư viện tiêu chuẩn, còn nhiều gói khác bên ngoài, các mô đun lẻ và các khung phát triển ứng dụng. Với người lập trình, cần phân biệt ba loại mô đun sau:

1) Các hàm tích hợp sẵn dùng ngay, không cần khai báo **import** ở đầu chương trình. Python 3 có gần 70 hàm có thể dùng ngay. Ta đã quen với một số trong đó như: **print()**, **input()**, **len()**, **sum()**, **max()**, **min()**, ... hay các hàm chuyển đổi kiểu dữ liệu số **int()**, **float()** chuyển đổi kiểu dữ liệu có cấu trúc **str()**, **list()**, **tuple()**, **set()**.

2) Các mô đun tiêu chuẩn đi kèm bản cài đặt Python, chỉ cần khai báo sử dụng bằng câu lệnh **import** là có thể dùng các hàm có trong thư viện. Hãy mở cửa sổ trình thông dịch (Python shell), thử chạy lệnh **help('modules')** và xem kết quả.

3) Các mô đun khác có thể gọi chung là các mô đun ngoài phải cài đặt gói bổ sung, sau đó mới có thể dùng và cách dùng giống như một mô đun tích hợp sẵn.

Một số thư viện ngoài chưa tích hợp sẵn là: **Matplotlib** để vẽ đồ thị hàm số, vẽ các biểu đồ; **PyGame** để làm việc với dữ liệu đồ họa, âm thanh trong trò chơi điện tử; **sqlite3** để làm việc với CSDL qua ngôn ngữ truy vấn SQL.

Học xong bài này, em sẽ:

Trình bày được cấu trúc dữ liệu danh sách liên kết và một số ứng dụng của nó.



Hãy cho biết danh sách mảng có nhược điểm gì?

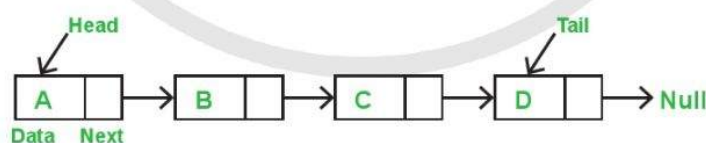
1 Cấu trúc danh sách liên kết

Danh sách liên kết (*linked list*) cũng gọi là danh sách móc nối, gồm các phần tử gọi là nút (*node*). Một nút có hai thành phần: phần *Data* chứa dữ liệu, phần liên kết gọi là *Next*. Phần *Next* chứa địa chỉ của nút liên kế và được thể hiện bằng mũi tên từ *Next* trở đến nút đứng kế sau nó (xem Hình 1). Trong hình minh họa, các chữ cái *A, B, C, D* thể hiện dữ liệu chứa trong nút.

Để tiện trình bày, ta gọi phần *Next* trong một nút là *con trỏ Next*. Kí hiệu mũi tên “→” trong hình minh họa thể hiện *Next* đang trỏ vào cái gì. Về bản chất, kí hiệu mũi tên “→” để thể hiện một kiểu dữ liệu kiểu đặc biệt, gọi là kiểu *con trỏ*. Con trỏ cho phép truy cập trực tiếp đến một địa chỉ ô nhớ cụ thể.

Đuôi danh sách là nút cuối cùng trong danh sách, không có nút nào đứng sau, được thể hiện bằng hình vẽ *Next* trỏ đến *Null* và được hiểu rằng “không trỏ đến đâu cả, không đi được tiếp nữa”. Con trỏ *Tail* trỏ đến nút đuôi danh sách.

Đầu danh sách được minh họa bằng mũi tên *Head* trỏ đến nút đầu tiên trong danh sách.



Hình 1. Minh họa một danh sách liên kết có 4 nút

Khi lập trình, cần phân biệt một nút với phần *Data* chứa dữ liệu trong nút đó, phân biệt phần *Data* với chính dữ liệu chứa trong nút đó, thể hiện bằng các chữ cái *A, B, C, D* trong hình minh họa. Nhưng để đơn giản trong trình bày các phép toán trong phần tiếp theo, ta gọi là nút *A*, nút *B*, nút *C*, nút *D*. Viết *A.Next* để nói về con trỏ từ nút *A* đến nút đứng sau nó và dễ nhận biết trên hình minh họa.

Sự khác nhau giữa danh sách liên kết và mảng

So với mảng, danh sách liên kết có những điểm khác biệt sau:

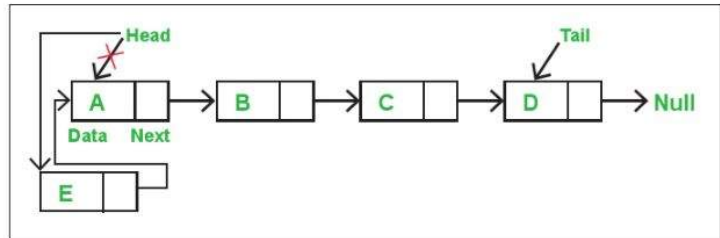
– Các nút danh sách liên kết không được lưu trữ thành một khối liên tục liền kề mà có thể nằm rải rác, tách rời nhau trong bộ nhớ.

– Các nút trong danh sách liên kết không có chỉ số. Phép lặp duyệt tuần tự từng nút của danh sách liên kết sử dụng một con trỏ *curr* (*current*) chỉ vào nút đang xét, thực hiện như sau:

+ $curr = Head$ bắt đầu từ *Head* để truy cập nút *A*;

+ $curr = A.Next$ để truy cập nút *B*; $curr = B.Next$ để truy cập nút *C*;...

+ Kết thúc khi gặp $curr = Null$ tức là ở tình huống $curr = D.Next$.



Hình 2a. Thêm nút vào đầu danh sách

Thêm nút và gỡ bỏ nút

• Thêm nút có ba trường hợp:

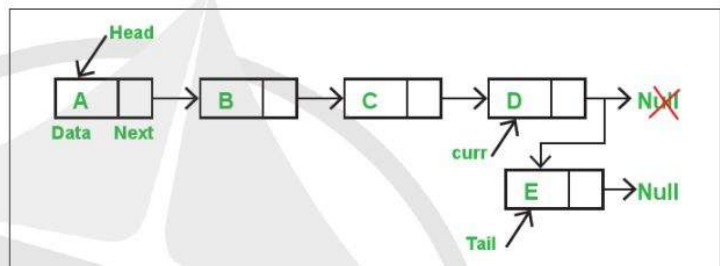
a) Thêm nút vào đầu danh sách

Nút mới thêm thành nút đầu tiên. Gọi nút mới là *E*. Thao tác theo hai bước sau (Hình 2a):

– Cho $E.Next$ trở đến nút *A*: gán $E.Next = Head$.

– Cho *Head* trở đến nút *E*: $Head \rightarrow E$.

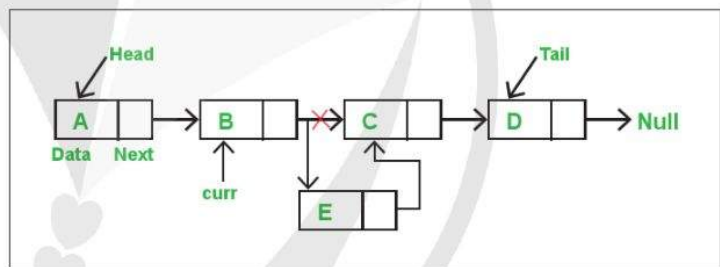
Thời gian thực hiện phép thêm nút vào đầu danh sách là $O(1)$, không phụ thuộc độ dài danh sách.



Hình 2b. Thêm nút vào cuối danh sách

b) Thêm nút vào cuối danh sách

Nối thêm nút mới vào cuối danh sách, nó trở thành nút cuối cùng. Con trỏ *Tail* trở đến nút cuối cùng của danh sách (Hình 1). Thao tác như minh họa trong Hình 2b. Chú ý rằng phải sửa lại, cho *Tail* trở vào *E*. Thời gian thực hiện phép thêm nút vào cuối danh sách là $O(1)$, không phụ thuộc độ dài danh sách.



Hình 2c. Thêm nút vào giữa danh sách

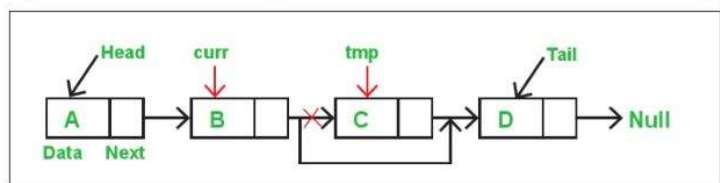
c) Thêm nút vào giữa danh sách

Tình huống minh họa: $curr \rightarrow B$. Thêm nút vào sau nút *B*. Thao tác như minh họa trong Hình 2c. Thời gian thực hiện phép thêm nút vào giữa danh sách là $O(1)$, không phụ thuộc độ dài danh sách.

• Gỡ bỏ nút:

Tình huống minh họa: $curr \rightarrow B$.

Gỡ bỏ nút sau nút *B* (Hình 3).



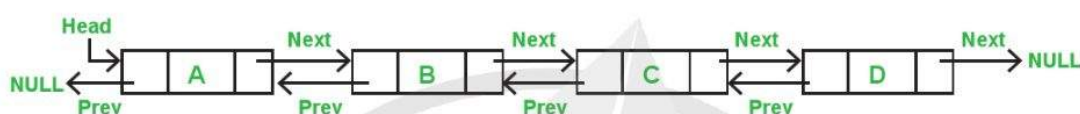
Hình 3. Gỡ bỏ nút sau nút B

- Ghi lưu con trỏ để truy cập nút C : $tmp = B.Next$, tức là $tmp \rightarrow C$.
- Cho $B.Next$ trỏ đến nút đứng sau C (là nút D): $B.Next = C.Next$.
- Sử dụng tmp để giải phóng phần bộ nhớ dành cho C .

Thao tác gỡ bỏ nút đầu danh sách hay cuối danh sách chỉ khác chút ít. Thời gian thực hiện gỡ bỏ là $O(1)$, không phụ thuộc vào độ dài danh sách.

Danh sách liên kết kép

Cấu trúc danh sách liên kết trình bày trên chỉ có một con trỏ $Next$ trỏ đến nút đứng kế ngay sau. Nếu mỗi nút có thêm một con trỏ nữa là $Prev$ trỏ đến nút đứng kế ngay trước thì ta sẽ có danh sách nối kép (Hình 4).



Hình 4. Mô hình một danh sách nối kép có 4 nút

Thời gian thực hiện các phép toán của danh sách liên kết

- **Phép tìm kiếm:** Tìm nút chứa dữ liệu X ($Data = X$) để xử lý. Phải thực hiện tìm kiếm tuần tự từ đầu danh sách. Độ phức tạp của phép tìm kiếm là $O(n)$ với n là số nút của danh sách.
- Các thao tác thêm nút, gỡ bỏ nút của danh sách liên kết dù ở bất cứ vị trí nào thì thời gian thực hiện đều là $O(1)$. Đây là điểm ưu việt hơn danh sách mảng.
- Danh sách liên kết tốn thêm chỗ để lưu trữ thành phần $Next$. Đây là nhược điểm so với danh sách mảng.

2 Ứng dụng của danh sách liên kết

Sử dụng cấu trúc móc nối để liên kết các nút thành một dãy tuần tự tạo ra kiểu danh sách rất linh hoạt. Danh sách liên kết phát huy ưu điểm trong những trường hợp thường xuyên phải:

- Thêm phần tử, gỡ bỏ phần tử ở bất cứ vị trí nào trong danh sách;
- Độ dài danh sách thay đổi nhanh và nhiều trong quá trình sử dụng.

Một số ví dụ ứng dụng danh sách liên kết

Danh sách nhóm đứng đầu *top N*: Ví dụ, dịch vụ cung cấp bài hát trực tuyến thường đưa ra một danh sách N bài hát đứng đầu một tuần, một tháng,... Sau đó, danh sách này được sử dụng theo nhiều cách khác nhau: phát lại từ bài $N - 1$ đến 0 hay ngược lại; phát lại theo trình tự ngẫu nhiên,...

Cập nhật danh sách *top N* cần các thao tác: gỡ bỏ một số phần tử tại các vị trí bất kì nào đó; chèn thêm phần tử vào vị trí bất kì nào đó. Độ dài N của danh sách cũng có các lựa chọn khác nhau, ví dụ là 5, 10, 20,...

Một số bài toán thực tế cần mô hình hoá một mạng lưới (điện, đường giao thông,...) hay một cấu trúc phân cấp hình cây (tổ chức hành chính, cây gia phả,...) thì không thể dùng một danh sách. Việc thể hiện mối liên kết giữa các nút (điểm giao cắt) phải cho phép cập nhật các thay đổi một cách linh hoạt. Danh sách liên kết sẽ được sử dụng trong các trường hợp này.



Dựa trên hình minh hoạ, mô tả các bước thực hiện các phép toán sau của danh sách liên kết để minh hoạ chúng đều có thời gian là $O(1)$.

- a) Thêm nút vào cuối danh sách, thêm nút vào giữa danh sách.
- b) Gỡ bỏ nút ở cuối danh sách, ở đầu danh sách.



Phân tích yêu cầu ứng dụng của một danh sách nhóm đứng đầu *top N* và cho biết, nếu dùng kiểu danh sách của Python để thực hiện thì:

- a) Những thao tác cần làm với danh sách *top N* sẽ thực hiện qua các phép toán danh sách Python như thế nào?
- b) Kể tên một vài phép toán danh sách của Python không cần dùng đến cho trường hợp này.



- Câu 1.** Hãy nêu các phép toán danh sách liên kết có thời gian thực hiện $O(1)$.
Câu 2. Hãy nêu các phép toán danh sách liên kết có thời gian thực hiện $O(n)$.
Câu 3. Nếu muốn truy cập nút chứa dữ liệu X thì phải làm gì? Ước lượng thời gian thực hiện.

Tóm tắt bài học

- ✓ Danh sách liên kết là một chuỗi nhiều nút, không đánh chỉ số tuần tự các nút, lưu trữ rải rác không liền kề trong bộ nhớ; các nút móc nối với nhau để có thể duyệt theo chiều tiến hoặc cả hai chiều tiến và lùi.
- ✓ Danh sách liên kết khắc phục nhược điểm của danh sách mảng: không bị giới hạn độ dài; thời gian thực hiện các phép thêm nút, gỡ bỏ nút là $O(1)$.
- ✓ Danh sách liên kết được dùng để mô hình hoá một mạng lưới (đồ thị) hay một cây phân cấp.

DANH SÁCH LIÊN KẾT TRONG PYTHON

Có thể lập trình Python thực hiện cấu trúc danh sách liên kết theo mô hình chung đã trình bày trên. Tuy nhiên, việc này đòi hỏi kiến thức cơ bản phương pháp hướng đối tượng và lập trình hướng đối tượng, một chủ đề rất quan trọng của Tin học nhưng chưa có trong nội dung chương trình bậc phổ thông.

Kiểu danh sách của Python rất tổng quát, có đủ các phép toán danh sách, phép thêm phần tử, gỡ bỏ phần tử có thể thực hiện tại bất cứ vị trí nào. Kiểu danh sách của Python đủ đáp ứng các yêu cầu ứng dụng nếu như không xem xét đến hiệu quả thời gian.

Python có một số kiểu dữ liệu làm sẵn có thể dùng để thực hiện những kiểu danh sách đặc thù, phù hợp hơn dùng kiểu danh sách tổng quát.

1) Thư viện **collections** có định nghĩa sẵn kiểu dữ liệu hàng đợi hai đầu **deque** (Doubly Ended Queue) là một kiểu danh sách được tối ưu hoá cho trường hợp thường xuyên phải lấy ra hay thêm vào một phần tử ở vị trí cuối hay ở đầu danh sách. Tên gọi phép thêm vào cuối là **append**, phép thêm vào đầu là **appendleft**, các phép lấy ra là **pop** và **popleft**. Các phép thêm vào và lấy ra đều có độ phức tạp thời gian $O(1)$ trong khi với kiểu **list** thì độ phức tạp là $O(n)$.

Để biểu diễn một hàng đợi thông thường (**queue**) trong thực tế, chỉ cần dùng **append** và **popleft**.

2) Trong một số ngôn ngữ lập trình khác, để biểu diễn dữ liệu thực tế không phải là dãy tuần tự, ví dụ như: cây gia phả, bản đồ giao thông,... người lập trình sẽ phải tự định nghĩa một kiểu dữ liệu dựa theo cấu trúc danh sách liên kết và sử dụng kiểu dữ liệu này trong văn bản chương trình.

Trong Python, có thể dễ dàng thể hiện một cây phân cấp hay đồ thị bằng cách sử dụng kiểu từ điển mà Python đã định nghĩa sẵn.

3) Có các gói thư viện bên ngoài cho Python đã làm sẵn danh sách liên kết, người lập trình chỉ cần tìm hiểu để sử dụng khi cần thiết.

MỤC LỤC

LỜI NÓI ĐẦU	3
BẢNG GIẢI THÍCH THUẬT NGỮ	4
CHỦ ĐỀ A MÁY TÍNH VÀ XÃ HỘI TRI THỨC THẾ GIỚI THIẾT BỊ SỐ – HỆ ĐIỀU HÀNH VÀ PHẦN MỀM ỨNG DỤNG	5
Bài 1. Bên trong máy tính	5
Bài 2. Khám phá thế giới thiết bị số thông minh	10
Bài 3. Khái quát về hệ điều hành	13
Bài 4. Thực hành với các thiết bị số	19
Bài 5. Phần mềm ứng dụng và dịch vụ phần mềm	24
CHỦ ĐỀ C TỔ CHỨC LƯU TRỮ, TÌM KIẾM VÀ TRAO ĐỔI THÔNG TIN TÌM KIẾM VÀ TRAO ĐỔI THÔNG TIN TRÊN MẠNG	28
Bài 1. Lưu trữ trực tuyến	28
Bài 2. Thực hành một số tính năng hữu ích của máy tìm kiếm	33
Bài 3. Thực hành một số tính năng nâng cao của mạng xã hội	36
Bài 4. Thực hành một số tính năng hữu ích của dịch vụ thư điện tử	39
CHỦ ĐỀ D ĐẠO ĐỨC, PHÁP LUẬT VÀ VĂN HOÁ TRONG MÔI TRƯỜNG SỐ ỨNG XỬ VĂN HOÁ VÀ AN TOÀN TRÊN MẠNG	42
Phòng tránh lừa đảo và ứng xử văn hoá trên mạng	42
CHỦ ĐỀ F GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH GIỚI THIỆU CÁC HỆ CƠ SỞ DỮ LIỆU	47
Bài 1. Bài toán quản lí và cơ sở dữ liệu	47
Bài 2. Bảng và khoá chính trong cơ sở dữ liệu quan hệ	52

Bài 3. Quan hệ giữa các bảng và khoá ngoài trong cơ sở dữ liệu quan hệ	57
Bài 4. Các biểu mẫu cho xem và cập nhập dữ liệu	62
Bài 5. Truy vấn trong cơ sở dữ liệu quan hệ	67
Bài 6. Truy vấn trong cơ sở dữ liệu quan hệ (tiếp theo)	71
Bài 7. Các loại kiến trúc của hệ cơ sở dữ liệu	76
Bài 8. An toàn và bảo mật hệ cơ sở dữ liệu	81
CHỦ ĐỀ G HƯỚNG NGHIỆP VỚI TIN HỌC GIỚI THIỆU NGHỀ QUẢN TRỊ CƠ SỞ DỮ LIỆU	84
Nghề quản trị cơ sở dữ liệu	84
CHỦ ĐỀ F^{CS} GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH KỸ THUẬT LẬP TRÌNH	89
Bài 1. Kiểu mảng và cấu trúc mảng	89
Bài 2. Mảng hai chiều	94
Bài 3. Thực hành về tệp, mảng và danh sách	97
Bài 4. Làm mịn dần từng bước từ thuật toán đến chương trình máy tính	101
Bài 5. Đánh giá thuật toán	107
Bài 6. Kiểm thử và sửa lỗi chương trình	113
Bài 7. Lập trình giải bài toán tìm kiếm	117
Bài 8. Lập trình một số thuật toán sắp xếp	122
Bài 9. Lập trình thuật toán sắp xếp nhanh	127
Bài 10. Thiết kế chương trình từ trên xuống và phương pháp mô đun hoá	131
Bài 11. Thực hành thiết kế và lập trình theo mô đun	135
Bài 12. Thực hành thiết kế và lập trình theo mô đun (tiếp theo)	140
Bài 13. Thực hành thiết kế và lập trình theo mô đun (tiếp theo)	142
Bài 14. Thực hành về thư viện các hàm tự định nghĩa	144
Bài 15. Cấu trúc dữ liệu danh sách liên kết và ứng dụng	146

Chịu trách nhiệm tổ chức bản thảo và bản quyền nội dung:
CÔNG TY CỔ PHẦN ĐẦU TƯ XUẤT BẢN – THIẾT BỊ GIÁO DỤC VIỆT NAM
Chủ tịch Hội đồng Quản trị: NGUYỄN NGÔ TRẦN ÁI
Tổng Giám đốc: VŨ BÁ KHÁNH

Biên tập:
TRẦN THỊ HIÊN

Thiết kế sách và minh họa:
TRẦN CƯỜNG

Trình bày bìa:
NGUYỄN MẠNH HÙNG

Sửa bản in:
TRẦN THỊ HIÊN – TRẦN THỊ THANH VÂN

Trong sách có sử dụng một số hình ảnh trên Internet. Trân trọng cảm ơn các tác giả!

Tin học 11 – Khoa học máy tính

Mã số:.....

ISBN:.....

In, khổ 19 x 26,5cm tại.....

Địa chỉ:

Số xác nhận đăng ký xuất bản.....

Quyết định xuất bản số:

In xong và nộp lưu chiểu tháng ... năm 20...