

TÀI LIỆU THAM KHẢO NÂNG CAO

TIN HỌC 10 CHO GÀ CHỌI

Dành cho học sinh ôn thi học sinh giỏi Tin học cấp tỉnh và cấp quốc gia

Advanced Computer Science for Grade 10 – a bilingual guide

Thông tin, dữ liệu, hệ nhị phân · Âm thanh, hình ảnh số

Mạng, an toàn và bản quyền · Thiết kế đồ họa với Inkscape

Lập trình Python: rẽ nhánh, vòng lặp, danh sách, chuỗi, hàm, kiểm thử

Robot giáo dục · Chương trình chạy thật, đề tự luyện ba mức có hướng dẫn chấm

Song ngữ Việt – Anh · Vietnamese – English

Phiên bản 1.0 – Tháng 10 năm 2026

<https://tailieuso.vn>

M U C L U C

C O N T E N T S

Chương 1. Thông tin, dữ liệu và hệ nhị phân – Chapter 1. Information, data and the binary system	5
1.1. Thông tin và dữ liệu khác nhau thế nào – <i>Information and data: what is the difference</i>	5
1.2. Cách đổi đơn vị lưu trữ dữ liệu – <i>How to convert units of data storage</i>	8
1.3. Kiểu dữ liệu và cách máy tính lưu văn bản (ASCII, Unicode) – <i>Data types and how a computer stores text (ASCII, Unicode)</i>	12
1.4. Cách đổi số thập phân sang nhị phân và ngược lại – <i>How to convert between decimal and binary</i>	16
1.5. Cách cộng và nhân hai số nhị phân – <i>How to add and multiply binary numbers</i>	23
1.6. Dữ liệu logic, phép toán logic và cách lập bảng chân trị – <i>Logical data, logical operations and how to build a truth table</i>	27
1.7. Đề tự luyện – <i>Practice questions</i>	34
1.8. Đáp án và hướng dẫn chấm bài tự luyện – <i>Answers and marking guide</i>	35
 Chương 2. Dữ liệu âm thanh, hình ảnh và thiết bị số – Chapter 2. Sound and image data, digital devices.....	 42
2.1. Thiết bị thông minh và Internet vạn vật – <i>Smart devices and the Internet of Things</i>	42
2.2. Tin học với xã hội: ứng dụng và các mốc phát triển – <i>Informatics and society: applications and milestones</i>	45
2.3. Số hóa âm thanh: lấy mẫu và thang giá trị mẫu – <i>Digitising sound: sampling and the sample-value scale</i>	47
2.4. Cách tính dung lượng tệp âm thanh chưa nén – <i>How to calculate the size of an uncompressed sound file</i>	51
2.5. Số hóa hình ảnh và cách tính dung lượng ảnh chưa nén – <i>Digitising images and how to calculate the size of an uncompressed image</i>	55
2.6. Thiết bị số cá nhân và cách quản lý tệp trên điện thoại – <i>Personal digital devices and managing files on a phone</i>	59
2.7. Đề tự luyện – <i>Practice questions</i>	64
2.8. Đáp án và hướng dẫn chấm bài tự luyện – <i>Answers and marking guide</i>	65
 Chương 3. Mạng, an toàn và đạo đức trong môi trường số – Chapter 3. Networks, safety and ethics in the digital world	 70
3.1. Mạng máy tính: LAN, Internet, điện toán đám mây và Internet vạn vật – <i>Computer networks: LAN, the Internet, cloud computing and the Internet of Things</i>	70
3.2. Nguy cơ trên mạng và cách phòng vệ (tin giả, lừa đảo, lộ thông tin, bắt nạt) – <i>Risks online and how to defend (fake news, scams, leaked data, bullying)</i>	77
3.3. Phần mềm độc hại: virus, worm, trojan và cách phòng chống – <i>Malware: viruses, worms, trojans and how to defend</i>	80

3.4. Cách dùng phần mềm dịch và kho học liệu mở để học – <i>How to use translation software and open learning repositories for study</i>	82
3.5. Ứng xử trên mạng, văn bản pháp luật và quyền tác giả (bản quyền) – <i>Conduct online, legal texts and copyright</i>	84
3.6. Cách phân tích một tình huống an toàn mạng và bản quyền – <i>How to analyse an online-safety or copyright situation</i>	92
3.7. Đề tự luyện – <i>Practice questions</i>	98
3.8. Đáp án và hướng dẫn chấm bài tự luyện – <i>Answers and marking guide</i>	100
Chương 4. Thiết kế đồ họa và nghề nghiệp tin học – Chapter 4. Graphic design and careers in computing	105
4.1. Cách so sánh đồ họa vector và đồ họa điểm ảnh theo tiêu chí – <i>How to compare vector and bitmap graphics on criteria</i>	105
4.2. Làm quen Inkscape: giao diện, hình khối, màu tô, màu vẽ và lớp – <i>Getting to know Inkscape: interface, basic shapes, fill, stroke and layers</i>	111
4.3. Cách ghép đối tượng bằng nhóm lệnh Path – <i>How to combine objects with the Path commands</i>	116
4.4. Đối tượng đường và văn bản: điểm neo, đường cong, chữ chạy theo đường – <i>Path objects and text: nodes, curves, text along a path</i>	121
4.5. Cách lập quy trình thiết kế một sản phẩm vector và xuất ảnh – <i>How to plan the design of a vector product and export it</i>	124
4.6. Nghề thiết kế đồ họa và nghề phát triển phần mềm – <i>The careers of graphic design and software development</i>	129
4.7. Đề tự luyện – <i>Practice questions</i>	135
4.8. Đáp án và hướng dẫn chấm bài tự luyện – <i>Answers and marking guide</i>	137
Chương 5. Lập trình Python: biến, biểu thức, vào ra – Chapter 5. Python programming: variables, expressions, input and output	145
5.1. Ngôn ngữ lập trình bậc cao và Python – <i>High-level programming languages and Python</i>	145
5.2. Cách dùng môi trường Python: gõ lệnh trực tiếp và soạn chương trình – <i>Using the Python environment: direct commands and program editing</i>	148
5.3. Biến, lệnh gán, quy tắc đặt tên và từ khóa – <i>Variables, assignment, naming rules and keywords</i>	151
5.4. Cách truy vết một đoạn lệnh gán – <i>How to trace a sequence of assignments</i>	154
5.5. Cách tính giá trị biểu thức Python – <i>How to evaluate a Python expression</i>	158
5.6. Cách nhập số, in kết quả và chuyển kiểu dữ liệu – <i>How to read numbers, print results and convert data types</i>	162
5.7. Đề tự luyện – <i>Practice questions</i>	169
5.8. Đáp án và hướng dẫn chấm bài tự luyện – <i>Answers and marking guide</i>	171
Chương 6. Rẽ nhánh và vòng lặp – Chapter 6. Branching and loops	179
6.1. Cách viết điều kiện bằng biểu thức logic – <i>How to write a condition as a Boolean expression</i>	179
6.2. Cách viết lệnh rẽ nhánh if, if-else và if-elif-else – <i>How to write if, if-else and if-elif-else</i>	184

6.3. Cách viết vòng for và dùng range – <i>How to write a for loop and use range</i>	190
6.4. Cách viết vòng while và chọn for hay while – <i>How to write a while loop and choose between for and while</i>	194
6.5. Cách truy vết vòng lặp bằng bảng – <i>How to trace a loop with a table</i>	198
6.6. Cách viết các thuật toán cơ bản bằng vòng lặp – <i>How to write the basic algorithms with loops</i>	202
6.7. Đề tự luyện – <i>Practice questions</i>	209
6.8. Đáp án và hướng dẫn chấm bài tự luyện – <i>Answers and marking guide</i>	211
Chương 7. Danh sách và chuỗi ký tự – Chapter 7. Lists and strings	220
7.1. Cách đọc, sửa, xóa và ghép phần tử của danh sách – <i>How to read, change, delete and join list items</i>	220
7.2. Cách thêm, chèn, xóa phần tử và kiểm tra có trong danh sách – <i>How to add, insert and delete items and test membership</i>	224
7.3. Cách duyệt danh sách để tìm lớn nhất, đếm và lọc – <i>How to traverse a list to find the largest, count and filter</i>	228
7.4. Chuỗi ký tự: chỉ số, duyệt và vì sao không sửa được từng ký tự – <i>Strings: indexes, traversal and why single characters cannot be changed</i>	233
7.5. Cách tìm chuỗi con, tách từ và nối từ: in, find, split, join – <i>How to find a substring, split into words and join words: in, find, split, join</i>	238
7.6. Cách đọc đề, chọn công cụ và thiết kế bộ test – <i>How to read a question, choose a tool and design a test set</i>	244
7.7. Đề tự luyện – <i>Practice questions</i>	247
7.8. Đáp án và hướng dẫn chấm bài tự luyện – <i>Answers and marking guide</i>	248
Chương 8. Hàm, kiểm thử và gỡ lỗi – Chapter 8. Functions, testing and debugging .	261
8.1. Cách viết một hàm và dùng lệnh return – <i>How to write a function and use return</i> ..	261
8.2. Cách truyền đối số vào hàm và chia bài toán thành hàm – <i>How to pass arguments and split a problem into functions</i>	266
8.3. Cách xác định phạm vi của biến – <i>How to decide the scope of a variable</i>	272
8.4. Cách đọc thông báo lỗi và phân loại lỗi – <i>How to read an error message and classify errors</i>	276
8.5. Cách thiết kế bộ test và gỡ lỗi – <i>How to design a test set and debug</i>	282
8.6. Ôn tập: viết chương trình hoàn chỉnh bằng hàm và hệ thống kiến thức – <i>Review: writing a complete program with functions, and the knowledge map</i>	289
8.7. Đề tự luyện – <i>Practice questions</i>	293
8.8. Đáp án và hướng dẫn chấm bài tự luyện – <i>Answers and marking guide</i>	295
Chương 9. Robot giáo dục – Chapter 9. Educational robots	306
9.1. Cấu tạo chung của robot giáo dục – <i>The general structure of an educational robot</i>	306
9.2. Cảm biến, phụ kiện và lắp ráp robot – <i>Sensors, accessories and assembling a robot</i>	312
9.3. Phần mềm, chương trình điều khiển và mã giả – <i>Software, control programs and pseudocode</i>	318
9.4. Điều khiển chuyển động: động cơ DC và động cơ servo – <i>Controlling movement: DC motors and servo motors</i>	324

9.5. Nhận biết vật cản và dẫn đường tự động – <i>Detecting obstacles and automatic guidance</i>	331
9.6. Sa bàn, phụ kiện và dự án robot – <i>The mat, accessories and the robot project</i>	337
9.7. Đề tự luyện – <i>Practice questions</i>	345
9.8. Đáp án và hướng dẫn chấm bài tự luyện – <i>Answers and marking guide</i>	347
Phụ lục A. Ba đề luyện thi học sinh giỏi – <i>Appendix A. Three practice papers</i>	356
A.1. Đề 1 – Luyện cấp tỉnh – <i>Paper 1: provincial-level practice</i>	356
A.2. Đề 2 – Luyện cấp quốc gia – <i>Paper 2: national-level practice</i>	368
A.3. Đề 3 – Chuyên sâu: lập trình và dự án – <i>Paper 3: in-depth programming and projects</i>	381
Phụ lục B. Tra cứu khái niệm – <i>Appendix B. Index of concepts</i>	397
B.1. Bảng tra theo chương – <i>B.1. Index by chapter</i>	397
Phụ lục C. Thuật ngữ Việt – Anh – <i>Appendix C. Vietnamese – English glossary</i>	402
C.1. Bảng tra từ A đến Z – <i>C.1. A to Z index</i>	402
Phụ lục D. Tài liệu đọc thêm – <i>Appendix D. Further reading</i>	407
D.1. Sách giáo khoa và chương trình – <i>D.1. Textbooks and curriculum</i>	407

Chương 1. Thông tin, dữ liệu và hệ nhị phân – Chapter 1.

Information, data and the binary system

Mục tiêu chương – Chapter objectives

- Phân biệt được thông tin và dữ liệu, chỉ ra được khi nào dữ liệu thiếu làm thông tin sai.
Tell information from data, and show when missing data makes the information wrong.
- Đổi được các đơn vị lưu trữ (bit, byte, KB, MB, GB, TB) và tính được thời gian truyền một tệp.
Convert storage units (bit, byte, KB, MB, GB, TB) and work out how long a file takes to transfer.
- Đổi được số giữa hệ thập phân và hệ nhị phân, viết được số âm bằng mã thuận, mã bù 1, mã bù 2.
Convert numbers between decimal and binary, and write negative numbers in sign-magnitude, 1's complement and 2's complement.
- Cộng và nhân được hai số nhị phân, rồi kiểm tra bằng cách đổi ra thập phân.
Add and multiply two binary numbers, then check the result by converting back to decimal.
- Lập được bảng chân trị cho biểu thức logic có hai hoặc ba biến và dùng nó để so sánh hai biểu thức.
Build a truth table for a logical expression with two or three variables and use it to compare two expressions.

1.1. Thông tin và dữ liệu khác nhau thế nào – Information and data: what is the difference

Bạn chụp tấm ảnh bảng điểm rồi gửi cho bố mẹ. Trong điện thoại, tấm ảnh là một dãy dài các số 0 và 1. Bố mẹ mở ảnh ra, đọc điểm và biết bạn học thế nào. Dãy số 0 và 1 là dữ liệu. Điều bố mẹ hiểu ra là thông tin.

You photograph your report card and send it to your parents. Inside the phone the photo is one long string of 0s and 1s. Your parents open it, read the marks and learn how you are doing. The string of 0s and 1s is data. What your parents understand is information.

KHÁI NIỆM 1.1 (Thông tin) / CONCEPT 1.1 (Information) – **Thông tin** là tất cả những gì mang lại cho con người hiểu biết. Thông tin gắn với việc nhận thức, mà nhận thức là việc của con người. Máy tính không tự nhận thức được, nó chỉ là công cụ giúp con người làm việc đó.

Information is everything that gives people understanding. It belongs to the act of understanding, which is a human act. A computer cannot understand by itself; it is a tool that helps people do so.

KHÁI NIỆM 1.2 (Dữ liệu) / CONCEPT 1.2 (Data) – Trong tin học, **dữ liệu** là thông tin (chữ, số, âm thanh, hình ảnh...) đã được đưa vào máy tính để máy nhận biết và xử lý. Thông tin là ý nghĩa của dữ liệu. Dữ liệu là những yếu tố thể hiện và xác định thông tin.

*In computing, **data** is information (text, numbers, sound, images...) that has been put into a computer so that the machine can recognise and process it. Information is the meaning of data. Data is what expresses and fixes the information.*

Máy tính xử lý thông tin theo ba bước nối nhau. Đầu vào có thể là dữ liệu hoặc thông tin, đầu ra cũng vậy.

A computer processes information in three linked steps. The input can be data or information, and so can the output.

1. Tiếp nhận dữ liệu. Dữ liệu đến từ thiết bị (ví dụ ảnh từ máy quét) hoặc do người gõ vào từ bàn phím.

Receive data. It arrives from a device (for example an image from a scanner) or is typed in by a person.

2. Xử lý dữ liệu. Máy biến đổi dữ liệu trong bộ nhớ để tạo ra dữ liệu mới. Ví dụ: từ ba điểm kiểm tra, tính điểm trung bình.

Process the data. The machine transforms data in memory to produce new data. Example: from three test marks, work out the average.

3. Đưa ra kết quả. Máy hiển thị chữ, số, âm thanh, hình ảnh để người đọc hiểu (dữ liệu thành thông tin), hoặc lưu vào thẻ nhớ, hoặc trao cho một bước xử lý khác.

Output the result. The machine shows text, numbers, sound or images that a person can understand (data becomes information), or saves it to a memory card, or hands it to another processing step. ■

ĐẶC ĐIỂM 3 (Tính độc lập tương đối và tính toàn vẹn) / FEATURES 3 (Relative independence and integrity) – Thông tin và dữ liệu độc lập tương đối với nhau, nên có bốn điều cần nhớ.

(a) Một thông tin có thể được ghi bằng nhiều loại dữ liệu. Tin “mai nghỉ học” có thể là tin nhắn chữ, đoạn ghi âm hoặc tờ thông báo.

(b) Một dữ liệu có thể mang nhiều thông tin. Số 40 có thể là tuổi, điểm hoặc giá tiền, tùy nó nằm ở đâu.

(c) Thông tin có tính toàn vẹn: thiếu dữ liệu thì thông tin sai hoặc không xác định được.

(d) Cùng một bộ dữ liệu, cách xử lý khác nhau cho thông tin khác nhau.

Information and data are only relatively independent, so remember four points.

(a) One piece of information can be recorded as several kinds of data. “No school tomorrow” can be a text message, a recording or a notice.

(b) One piece of data can carry several pieces of information. The number 40 may be an age, a mark or a price, depending on where it sits.

(c) Information is integral: with data missing, it becomes wrong or cannot be determined.

(d) The same data set, processed in different ways, gives different information.

Cách phân biệt dữ liệu và thông tin khi gặp đề – How to tell data from information in a question

Đề thường đưa một con số hoặc một tệp rồi hỏi “đây là dữ liệu hay thông tin”, hoặc hỏi vì sao kết luận bị sai. Làm theo bốn bước sau.

A question often gives a number or a file and asks “is this data or information”, or why a conclusion is wrong. Follow these four steps.

1. Gạch chân phần máy lưu được: con số, dòng chữ, tệp. Phần gạch chân là dữ liệu.

Underline what a machine can store: numbers, lines of text, files. The underlined part is data.

2. Hỏi “người đọc hiểu ra điều gì”, rồi viết câu trả lời một dòng. Dòng đó là thông tin.
Ask “what does a reader come to know”, then write the answer in one line. That line is the information.
3. Kiểm tra dữ liệu có đủ không: có đơn vị, có nói về đối tượng nào, có mốc thời gian không. Thiếu gì thì ghi ra.
Check whether the data is complete: is there a unit, an object it describes, a point in time? Write down what is missing.
4. Thử một cách xử lý khác trên cùng dữ liệu và xem thông tin có đổi không.
Try a different way of processing the same data and see whether the information changes.

PHÂN TÍCH ĐỀ / READING THE QUESTION – Đề hỏi hai thông tin khác nhau từ cùng một dữ liệu, nghĩa là hỏi tính (b) của Đặc điểm 3. Đề cũng hỏi cách làm cho dữ liệu rõ nghĩa, nghĩa là hỏi tính (c). Mỗi ý cần một ví dụ cụ thể.

The question asks for two different pieces of information from one datum, which is property (b) of Feature 3. It also asks how to make the data clear, which is property (c). Each point needs a concrete example.

VÍ DỤ 1 (Một dữ liệu, hai thông tin) / EXAMPLE 1 (One datum, two pieces of information) – Màn hình của một robot hút bụi chỉ hiện số 20. Nêu hai thông tin khác nhau mà con số này có thể mang, và cho biết phải bổ sung gì để con số rõ nghĩa.

The screen of a robot vacuum cleaner shows only the number 20. Give two different pieces of information this number could carry, and say what must be added to make it clear.

Hướng dẫn làm bài. / How to answer.

1. Bước 1: dữ liệu là con số 20, thiếu đơn vị và nhãn.
Step 1: the data is the number 20, with no unit and no label.
2. Bước 2: có thể đọc là “pin còn 20 phần trăm” hoặc “đã dọn 20 phút”.
Step 2: it could mean “battery at 20 per cent” or “cleaned for 20 minutes”.
3. Bước 3: chỗ thiếu là nhãn (pin hay thời gian) và đơn vị (phần trăm hay phút).
Step 3: what is missing is the label (battery or time) and the unit (per cent or minutes).

BÀI VIẾT MẪU / MODEL WRITING – Con số 20 trên màn hình là dữ liệu. Nó chưa cho biết điều gì chắc chắn vì thiếu nhãn và đơn vị. Nếu máy ghi “Pin: 20%”, người dùng hiểu rằng sắp hết pin và cần cho robot về sạc. Nếu máy ghi “Thời gian: 20 phút”, người dùng hiểu rằng robot đã chạy được hai mươi phút. Cùng một dữ liệu mang hai thông tin khác nhau. Muốn con số rõ nghĩa phải đi kèm nhãn và đơn vị.

The number 20 on the screen is data. It tells us nothing for certain because the label and unit are missing. If the machine shows “Battery: 20%”, the user learns that the battery is nearly flat and the robot should go to charge. If it shows “Time: 20 min”, the user learns that the robot has been running for twenty minutes. One datum carries two different pieces of information. To make the number clear it must come with a label and a unit. ■

LỖI HAY GẶP / COMMON MISTAKE – Viết “thông tin là con số 20”. Con số 20 là dữ liệu, còn thông tin là điều người đọc hiểu ra từ nó. Nhầm hai thứ này làm mất điểm câu định

nghĩa.

Writing “the information is the number 20”. The number 20 is data; the information is what the reader understands from it. Mixing the two loses the mark on the definition question.

NHẬN XÉT / REMARK – Câu hỏi “vì sao kết luận sai” hầu như luôn dẫn về tính toàn vẹn: dữ liệu thiếu đơn vị, thiếu đối tượng hoặc thiếu mốc thời gian.

A “why is the conclusion wrong” question nearly always leads back to integrity: the data lacks a unit, an object or a point in time.

1.2. Cách đổi đơn vị lưu trữ dữ liệu – How to convert units of data storage

Một ảnh chụp bằng điện thoại nặng vài MB, một bộ phim nặng vài GB. Các con số ấy đo lường dữ liệu. Muốn so sánh hay cộng chúng, bạn phải đưa về cùng một đơn vị.

A phone photo weighs a few MB and a film a few GB. These numbers measure the amount of data. To compare or add them you must bring them to one unit.

KHÁI NIỆM 4 (Bit và byte) / CONCEPT 4 (Bit and byte) – Bit là một chữ số nhị phân, 0 hoặc 1. Byte gồm 8 bit và là đơn vị dữ liệu nhỏ nhất mà máy tính truy cập được. Máy tính lưu và lấy dữ liệu theo nhóm bit chứ không theo từng bit, nên byte được dùng làm đơn vị đo lường dữ liệu.

A bit is one binary digit, 0 or 1. A byte is 8 bits and is the smallest unit of data a computer can access. Computers store and fetch data in groups of bits, not one bit at a time, so the byte is used to measure the amount of data.

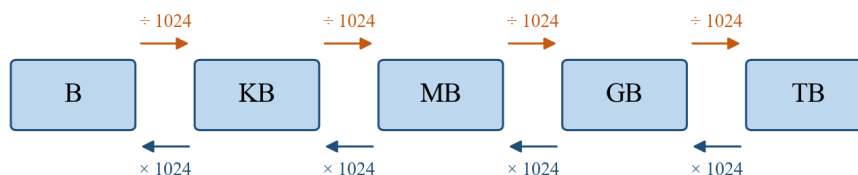
Các đơn vị lớn hơn byte cứ hơn kém nhau 1024 lần, vì $1024 = 2^{10}$. Bảng tính sẵn mỗi đơn vị bằng bao nhiêu byte, để bạn tra khi cần.

Units above the byte differ by a factor of 1024, because $1024 = 2^{10}$. The table gives each unit in bytes for quick reference.

Đơn vị <i>Unit</i>	Kí hiệu <i>Symbol</i>	Bằng bao nhiêu đơn vị nhỏ hơn liền kề <i>In the next smaller unit</i>	Bằng bao nhiêu byte <i>In bytes</i>
bit	b	1 bit	1 bit = 1/8 byte
Byte	B	8 bit	1
Kilobyte	KB	1024 B	1.024
Megabyte	MB	1024 KB	1.048.576 <i>1,048,576</i>
Gigabyte	GB	1024 MB	1.073.741.824 <i>1,073,741,824</i>
Terabyte	TB	1024 GB	1.099.511.627.776 <i>1,099,511,627,776</i>

Bảng cho thấy mỗi bước đi lên một đơn vị là chia 1024, mỗi bước đi xuống là nhân 1024. Các đơn vị lớn hơn (PB, EB, ZB, YB) cũng theo đúng quy luật đó.

The table shows that each step up one unit is a division by 1024 and each step down is a multiplication by 1024. The larger units (PB, EB, ZB, YB) follow the same rule.



Hình 1. Đổi sang đơn vị lớn hơn (sang phải) thì chia số cho 1024, đổi sang đơn vị nhỏ hơn (sang trái) thì nhân số với 1024

Figure 1. Converting to a larger unit (to the right) divides the number by 1024; converting to a smaller unit (to the left) multiplies it by 1024

Các bước đổi đơn vị – Steps

- Viết đơn vị đề cho và đơn vị cần đổi lên hai đầu một dòng, theo thứ tự B, KB, MB, GB, TB.
Write the given unit and the target unit at the two ends of a line, in the order B, KB, MB, GB, TB.
- Đếm số bước từ đơn vị đề cho đến đơn vị cần đổi. Đi sang đơn vị nhỏ hơn thì nhân 1024 cho mỗi bước, đi sang đơn vị lớn hơn thì chia 1024.
Count the steps from the given unit to the target. Each step towards a smaller unit multiplies by 1024; each step towards a larger unit divides by 1024.
- Gặp bit thì nhớ 1 byte = 8 bit: đổi từ byte ra bit thì nhân 8, từ bit ra byte thì chia 8.
When bits are involved, remember 1 byte = 8 bits: byte to bit multiplies by 8, bit to byte divides by 8.
- Viết phép tính ra giấy, tính, rồi đọc lại xem kết quả có hợp lí không: đổi sang đơn vị nhỏ hơn thì số phải lớn lên.
Write the calculation down, work it out, then check that it is sensible: converting to a smaller unit must make the number bigger. ■

PHẢI THUỘC / MEMORISE – Kí hiệu b nhỏ là bit, B hoa là byte. Tốc độ truyền thường ghi theo bit (ví dụ Mb/s), còn dung lượng tệp ghi theo byte (ví dụ MB). Muốn tính thời gian tải một tệp, đưa hai số về cùng loại đơn vị trước khi chia.

A small b stands for bit and a capital B for byte. Transfer speed is usually written in bits (for example Mb/s) while file size is in bytes (for example MB). To work out a download time, bring both numbers to the same kind of unit before dividing.

PHÂN TÍCH ĐỀ / READING THE QUESTION – Đề có hai việc. Việc thứ nhất cho dung lượng ở GB và hỏi KB, nên đi hai bước xuống: GB sang MB, MB sang KB. Số có phần thập phân (1,5) cứ nhân bình thường, không làm tròn giữa chừng. Việc thứ hai cho tốc độ ở Mb/s (chữ b nhỏ) và dung lượng ở MB (chữ B hoa): hai số khác loại đơn vị, nên đổi MB sang Mb bằng cách nhân 8 trước khi chia.

The question has two jobs. The first gives a size in GB and asks for KB, so go down two steps: GB to

MB, MB to KB. A number with a decimal part (1.5) is multiplied as usual, with no rounding in between. The second gives a speed in Mb/s (small b) and a size in MB (capital B): the two numbers use different kinds of unit, so convert MB to Mb by multiplying by 8 before dividing.

VÍ DỤ 2 (Đổi đơn vị và tính thời gian tải) / EXAMPLE 2 (Converting units and download time)

– Một tệp phim dài 1,5 GB. (a) Đổi ra MB và ra KB. (b) Một tệp khác 600 MB được tải qua đường truyền 40 Mb/s. Mất bao nhiêu giây nếu tốc độ giữ nguyên?

A film file is 1.5 GB. (a) Convert it to MB and to KB. (b) Another file of 600 MB is downloaded over a 40 Mb/s line. How many seconds does it take if the speed stays constant?

Hướng dẫn làm bài. / How to answer.

1. Bước 1: từ GB đi xuống KB là hai bước, mỗi bước nhân 1024.

Step 1: from GB down to KB is two steps, each multiplying by 1024.

2. Bước 2: $1,5 \text{ GB} = 1,5 \times 1024 \text{ MB} = 1.536 \text{ MB}$.

Step 2: $1.5 \text{ GB} = 1.5 \times 1024 \text{ MB} = 1,536 \text{ MB}$.

3. Bước 3: $1.536 \text{ MB} = 1.536 \times 1024 \text{ KB} = 1.572.864 \text{ KB}$.

Step 3: $1,536 \text{ MB} = 1,536 \times 1024 \text{ KB} = 1,572,864 \text{ KB}$.

4. Bước 4: sang việc (b), nhận ra b nhỏ (bit) và B hoa (byte). Đổi dung lượng sang bit: $600 \text{ MB} \times 8 = 4.800 \text{ Mb}$.

Step 4: for part (b), notice small b (bit) and capital B (byte). Convert the size to bits: $600 \text{ MB} \times 8 = 4,800 \text{ Mb}$.

5. Bước 5: chia cho tốc độ: $4.800 : 40 = 120$ giây.

Step 5: divide by the speed: $4,800 : 40 = 120 \text{ seconds}$.

6. Bước 6: kiểm tra ngược: 40 Mb/s là 5 MB mỗi giây, $5 \times 120 = 600 \text{ MB}$. Khớp.

Step 6: check backwards: 40 Mb/s is 5 MB per second, and $5 \times 120 = 600 \text{ MB}$. It matches.

Đáp số: (a) 1.536 MB và 1.572.864 KB; (b) 120 giây.

Answer: (a) 1,536 MB and 1,572,864 KB; (b) 120 seconds. ■

LỖI HAY GẶP / COMMON MISTAKE – Chia thẳng 600 cho 40 và ra 15 giây, vì không đổi MB sang Mb. Dấu hiệu nhận ra: một số có chữ B hoa, số kia có chữ b nhỏ.

Dividing 600 by 40 directly and getting 15 seconds because MB was not converted to Mb. The warning sign: one number has a capital B, the other a small b.

LỖI HAY GẶP / COMMON MISTAKE – Nhân 1000 thay vì 1024 giữa các bậc. Khi đề không nói gì khác, dùng 1024, đúng như quy ước của sách giáo khoa.

Multiplying by 1000 instead of 1024 between levels. Unless the question says otherwise, use 1024, as the textbook does.

Vì sao lưu, xử lý và truyền tin bằng thiết bị số lại có lợi – Why storing, processing and sending information on digital devices pays off

Hãy so một tấm bản đồ giấy với bản đồ trong điện thoại. Bản đồ giấy gấp lại mới bỏ vừa túi, còn điện thoại chứa cả nghìn tấm. Muốn tìm một con đường, bạn lật từng trang, còn điện thoại tìm trong giây lát.

Compare a paper map with a map on a phone. A paper map must be folded to fit a pocket, while a phone holds thousands. To find a road you turn pages one by one, while the phone finds it in moments.

Thiết bị số là thiết bị làm việc với thông tin số, tức thông tin được mã hóa trong hệ nhị phân: lưu trữ, truyền hoặc xử lý. Thẻ nhớ, bộ thu phát wifi và máy tính xách tay là thiết bị số.

*A **digital device** works with digital information, that is, information encoded in the binary system: it stores, transmits or processes it. A memory card, a wifi router and a laptop are digital devices.*

ĐẶC ĐIỂM 5 (Ưu điểm của thiết bị số) / FEATURES 5 (Advantages of digital devices) –

Thiết bị số có bốn ưu điểm.

Lưu trữ: chứa lượng dữ liệu lớn trong thiết bị gọn, giá thấp, tìm kiếm nhanh và dễ.

Xử lý: tính nhanh, chính xác, làm việc liên tục không mệt.

Truyền tin: tốc độ truyền lớn, độ tin cậy cao.

Tự động: thực hiện nhiều việc một cách chính xác, chi phí thấp và tiện lợi.

Digital devices have four advantages.

Storage: they hold large amounts of data in compact, cheap devices, with fast and easy searching.

Processing: they calculate quickly and accurately, and can work continuously without tiring.

Transmission: high transfer speed and high reliability.

Automation: they carry out many tasks accurately, at low cost and conveniently.

NHẬN XÉT / REMARK – Đề so sánh thiết bị số với thiết bị không số thường chia thành ba tiêu chí: lưu trữ, xử lý, truyền tin. Mỗi tiêu chí cần một ý cho thiết bị số và một ý cho thiết bị còn lại, không bỏ ô nào.

A question comparing digital and non-digital devices usually splits into three criteria: storage, processing, transmission. Each criterion needs one point for the digital device and one for the other, with no empty cell.

Bảng gom các ý để tra nhanh. Cột giữa là ý cho thiết bị số, cột phải là ý tương ứng cho tám bản đồ giấy, bạn có thể dùng làm mẫu câu trả lời.

The table collects the points for quick lookup. The middle column gives the point for the digital device and the right column the matching point for the paper map, which can serve as a model answer.

Tiêu chí Criterion	Thiết bị số Digital device	Đồ dùng không số (ví dụ bản đồ giấy) Non-digital item (for example a paper map)
Lưu trữ Storage	Một thẻ nhớ chứa cả thư viện bản đồ, tìm nhanh bằng ô tìm kiếm. <i>One memory card holds a whole library of maps, found with a search box.</i>	Mỗi khu vực một tờ giấy, tìm bằng cách lật. <i>One sheet per area, found by turning pages.</i>

Tiêu chí <i>Criterion</i>	Thiết bị số <i>Digital device</i>	Đồ dùng không số (ví dụ bản đồ giấy) <i>Non-digital item (for example a paper map)</i>
Xử lý <i>Processing</i>	Tự tính đường đi ngắn nhất, tính lại khi rẽ nhầm. <i>Works out the shortest route and recalculates after a wrong turn.</i>	Người đọc tự dò đường và tính nhẩm. <i>The reader traces the route and calculates in the head.</i>
Truyền tin <i>Transmission</i>	Gửi cả tấm bản đồ cho bạn qua mạng trong giây lát. <i>Sends the whole map to a friend over the network in moments.</i>	Phải photo hoặc gửi bản giấy qua bưu điện. <i>Must be photocopied or posted.</i>

1.3. Kiểu dữ liệu và cách máy tính lưu văn bản (ASCII, Unicode) – *Data types and how a computer stores text (ASCII, Unicode)*

Số điện thoại 0912 345 678 trông như một con số, nhưng bạn chẳng bao giờ cộng hai số điện thoại với nhau. Còn điểm 8,5 thì bạn cộng, so sánh được. Cách máy tính lưu và xử lý hai loại thông tin ấy khác nhau, vì vậy máy phân chúng thành hai kiểu dữ liệu.

The phone number 0912 345 678 looks like a number, yet you never add two phone numbers together. A mark of 8.5, on the other hand, can be added and compared. The computer stores and processes the two kinds of information differently, so it sorts them into two data types.

KHÁI NIỆM 6 (Kiểu dữ liệu) / CONCEPT 6 (Data type) – Thông tin đưa vào máy tính phải được mã hóa thành dữ liệu nhị phân. Tùy bản chất của thông tin mà dữ liệu có cách biểu diễn riêng, và từ đó có các **kiểu dữ liệu** thường gặp: văn bản, số, logic, hình ảnh và âm thanh. Việc chia kiểu giúp chọn đúng các phép xử lý: số thì tính và so sánh, văn bản thì tách, ghép và so sánh.

*Information put into a computer must be encoded as binary data. Depending on its nature, the data has its own representation, which gives the usual **data types**: text, number, logical, image and sound. Sorting data into types lets us choose the right operations: numbers are calculated and compared, text is split, joined and compared.*

Cách thử nhanh: hỏi “phép tính số học trên dữ liệu này có nghĩa không”. Cộng hai điểm trung bình có nghĩa, cộng hai số điện thoại thì không. Dữ liệu nào không tính được thì nên lưu dưới dạng văn bản. Số thực là số có phần thập phân, như điểm 8,5.

A quick test: ask “does arithmetic on this data make sense”. Adding two average marks does; adding two phone numbers does not. Data that cannot be calculated with should be stored as text. A real number has a fractional part, like the mark 8.5.

Bảng mã ASCII – *The ASCII character set*

Máy tính chỉ lưu được dãy bit, vậy chữ “A” nằm trong máy bằng cách nào? Người ta gán cho mỗi kí tự một số thứ tự, gọi là mã của kí tự, rồi lưu mã đó dưới dạng nhị phân. Danh sách kí tự và mã của chúng gọi là bảng mã.

A computer stores only strings of bits, so how does the letter “A” get into it? Each character is given a sequence number, called its code, and the code is stored in binary. The list of characters and their codes is called a character set.

ASCII (American Standard Code for Information Interchange, bảng mã chuẩn của Mỹ để trao đổi thông tin) ban đầu dùng mã 7 bit, tức $2^7 = 128$ mã, chỉ đủ cho tiếng Anh. Bảng mã được mở rộng thành 8 bit, tức $2^8 = 256$ mã, giữ nguyên 128 mã cũ. 128 vị trí thêm vào gọi là phần mở rộng, mỗi quốc gia có thể dùng cho kí tự riêng.

ASCII (American Standard Code for Information Interchange) originally used 7-bit codes, that is $2^7 = 128$ codes, enough only for English. The set was extended to 8 bits, that is $2^8 = 256$ codes, keeping the old 128. The 128 added positions are called the extension; each country may use them for its own characters.

Cách tìm mã nhị phân của một kí tự – *How to find the binary code of a character*

1. Mở bảng mã ASCII mở rộng ở Phụ lục (trang 165 của sách giáo khoa). Tìm kí tự trong bảng.

Open the extended ASCII table in the textbook appendix (page 165). Find the character in the table.

2. Nhìn sang đầu hàng bên trái, đọc 4 bit. Nhìn lên đầu cột phía trên, đọc 4 bit.

Look left to the row heading and read 4 bits. Look up to the column heading and read 4 bits.

3. Ghép 4 bit của hàng đứng trước, 4 bit của cột đứng sau, được mã nhị phân 8 bit của kí tự.

Join the 4 bits of the row first and the 4 bits of the column second to get the character's 8-bit binary code.

4. Đổi mã nhị phân sang thập phân (cách làm ở mục 1.4) để so với số thứ tự trong bảng, xem có khớp không.

Convert the binary code to decimal (the method is in section 1.4) and compare it with the sequence number in the table. ■

Khi không có bảng, nhớ ba mốc: “A” có mã 65, “a” có mã 97, chữ số “0” có mã 48. Các chữ cái liên tiếp có mã liên tiếp, nên “K”, đứng sau “A” mười vị trí, có mã 75.

Without the table, remember three landmarks: “A” has code 65, “a” has code 97, the digit “0” has code 48. Consecutive letters have consecutive codes, so “K”, ten places after “A”, has code 75.

Python là ngôn ngữ lập trình bạn sẽ học từ Chương 5. Lúc này chỉ cần đọc từng dòng lệnh và xem kết quả máy in ra.

Python is the programming language you will learn from Chapter 5. For now just read each line of code and look at what the machine prints.

```
chu = "K"
ma = ord(chu)
print(ma)
print(format(ma, "08b"))
```

Kết quả:

Output:

75
01001011

Hàm ord cho mã của một kí tự. Hàm format(ma, “08b”) viết số đó bằng nhị phân, đủ 8 chữ số, thêm số 0 ở đầu nếu thiếu. Máy in ra 75 và 01001011, khớp với cách tra bảng: hàng 0100, cột 1011.

The function ord gives the code of a character. The function format(ma, “08b”) writes that number in binary with 8 digits, padding with zeros on the left. The machine prints 75 and 01001011, which agrees with the table lookup: row 0100, column 1011.

Unicode và tiếng Việt – Unicode and Vietnamese

Tiếng Việt có 134 nguyên âm có dấu thanh và chữ đ (kể cả chữ in hoa) không nằm trong bảng ASCII gốc, trong khi phần mở rộng chỉ có 128 chỗ. Tiếng Trung, tiếng Nhật còn cần vài chục nghìn kí tự. Vì vậy đầu những năm 1980 người ta đề xuất một bảng mã chung cho mọi quốc gia, gọi là Unicode.

Vietnamese has 134 vowels carrying tone marks, plus the letter đ (capitals included), that are not in the original ASCII, while the extension has only 128 places. Chinese and Japanese need tens of thousands of characters. So in the early 1980s an international proposal was made for one set shared by all countries, called Unicode.

ĐẶC ĐIỂM 7 (Unicode và UTF-8) / FEATURES 7 (Unicode and UTF-8) – Unicode là bộ tiêu chuẩn biểu diễn kí tự dùng nhiều hơn 8 bit, mã hóa được hàng trăm nghìn kí tự của nhiều ngôn ngữ. Nước đã có mặt chữ nào thì nước khác dùng lại, không đặt mã mới, nên mã không bị lệch nhau. Việt Nam ban hành TCVN 6909:2001 (bộ mã kí tự tiếng Việt 16 bit, mỗi kí tự 2 byte, phù hợp Unicode) và từ năm 2017 bắt buộc dùng UTF-8. UTF-8 (8-bit Unicode Transformation Format) mã hóa mỗi kí tự với độ dài từ 1 đến 4 byte.

Unicode is a standard for representing characters with more than 8 bits and can encode hundreds of thousands of characters from many languages. A script already defined by one country is reused by the others without a new code, so codes do not clash. Viet Nam issued TCVN 6909:2001 (a 16-bit Vietnamese character set, 2 bytes per character, consistent with Unicode) and since 2017 has required UTF-8. UTF-8 (8-bit Unicode Transformation Format) encodes each character in 1 to 4 bytes.

Sách giáo khoa nêu quy ước dễ nhớ: kí tự La tinh không dấu dùng 1 byte, nguyên âm có dấu và chữ đ dùng 2 byte, kí tự đặc biệt dùng 3 byte. Khi bạn đếm bằng máy, kết quả cho thấy chữ đ, ă, â, ê, ô, ơ, u đúng là 2 byte, còn nguyên âm mang dấu thanh như ệ, ẹ, ế, ạ chiếm 3 byte. Vì vậy đề trắc nghiệm hỏi “mỗi kí tự tiếng Việt chiếm bao nhiêu byte” có đáp án đúng là từ 1 đến 3 byte. Để tính số byte của một câu cụ thể thì đếm từng chữ bằng máy, đừng chỉ dựa vào quy ước.

The textbook gives an easy rule: unaccented Latin letters take 1 byte, vowels with marks and the letter đ take 2 bytes, special characters take 3 bytes. When we count with a machine, đ, ă, â, ê, ô, ơ, u are indeed 2 bytes, while vowels carrying a tone mark such as ệ, ẹ, ế, ạ take 3 bytes. So a multiple-choice question asking “how many bytes does a Vietnamese character take” has the answer 1 to 3 bytes. For the byte count of a specific sentence, count letter by letter with a machine instead of relying only on the rule.

NHẬN XÉT / REMARK – Đề đếm byte của một câu tiếng Việt là chỗ dễ lệch nhất. Cách an toàn: đếm số kí tự trước, rồi cộng thêm 1 byte cho mỗi chữ đ, ă, â, ê, ô, ơ, ư và cộng thêm 2 byte cho mỗi nguyên âm mang dấu thanh.

Counting the bytes of a Vietnamese sentence is where answers most often go wrong. The safe way: count the characters first, then add 1 byte for each of đ, ă, â, ê, ô, ơ, ư and 2 bytes for each vowel carrying a tone mark.

PHÂN TÍCH ĐỀ / READING THE QUESTION – Đề cho một câu có chữ không dấu, chữ có dấu mũ và chữ có dấu thanh, và hỏi cả số kí tự lẫn số byte. Hai số này khác nhau, nên phải đếm riêng. Tra bảng ASCII cho chữ H, rồi lập luận theo UTF-8 cho các chữ còn lại.

The question gives a sentence with unaccented letters and letters with tone marks, and asks for both the number of characters and the number of bytes. The two numbers differ, so count them separately. Look up H in the ASCII table, then reason with UTF-8 for the other letters.

VÍ DỤ 3 (Đếm kí tự và đếm byte) / EXAMPLE 3 (Counting characters and bytes) – Câu “Học giỏi Tin” (kể cả hai dấu cách) gồm bao nhiêu kí tự và bao nhiêu byte khi lưu bằng UTF-8? Tìm mã nhị phân 8 bit của chữ H.

How many characters and how many bytes does the phrase “Học giỏi Tin” (spaces included) take in UTF-8? Find the 8-bit binary code of the letter H.

Hướng dẫn làm bài. / How to answer.

- Bước 1: đếm kí tự: H, ọ, c, dấu cách, g, i, ỏ, i, dấu cách, T, i, n. Tổng cộng 12 kí tự.
Step 1: count the characters: H, ọ, c, space, g, i, ỏ, i, space, T, i, n. In all, 12 characters.
- Bước 2: tìm kí tự mang dấu thanh: ọ và ỏ. Mỗi chữ chiếm 3 byte, hai chữ là 6 byte.
Step 2: find the characters with a tone mark: ọ and ỏ. Each takes 3 bytes, so 6 bytes for both.
- Bước 3: 10 kí tự còn lại là chữ không dấu và dấu cách, mỗi kí tự 1 byte, được 10 byte.
Step 3: the remaining 10 characters are unaccented letters and spaces, 1 byte each, giving 10 bytes.
- Bước 4: cộng lại: $6 + 10 = 16$ byte.
Step 4: add up: $6 + 10 = 16$ bytes.
- Bước 5: mã chữ H. H đứng sau A bảy vị trí (A, B, C, D, E, F, G, H), nên mã là $65 + 7 = 72$. Tra bảng: hàng 0100, cột 1000, ghép lại là 01001000.
Step 5: the code of H. H is seven places after A (A, B, C, D, E, F, G, H), so its code is $65 + 7 = 72$. From the table: row 0100, column 1000, joined to give 01001000.

```
xau = "Học giỏi Tin"
print(len(xau))
print(len(xau.encode("utf-8")))
```

Kết quả:

Output:

```
12
16
```

BÀI VIẾT MẪU / MODEL WRITING – Câu “Học giỏi Tin” có 12 kí tự nhưng chiếm 16 byte trong UTF-8. Lí do là hai chữ ọ và ỏ mang dấu thanh nên mỗi chữ cần 3 byte, còn 10 kí

tự kia (chữ không dấu và hai dấu cách) cần 1 byte mỗi kí tự. Tổng là $6 + 10 = 16$ byte. Mã nhị phân của chữ H là 01001000, ứng với mã thập phân 72.

The phrase “Học giỏi Tin” has 12 characters but takes 16 bytes in UTF-8. The reason is that ơ and ố carry tone marks, so each needs 3 bytes, while the other 10 characters (unaccented letters and two spaces) need 1 byte each. The total is $6 + 10 = 16$ bytes. The binary code of H is 01001000, which is 72 in decimal. ■

LỖI HAY GẶP / COMMON MISTAKE – Coi số kí tự bằng số byte. Với chữ có dấu, hai số này lệch nhau. Đề hỏi “số byte” thì phải cộng thêm phần của chữ có dấu.

Treating the number of characters as the number of bytes. With accented letters the two differ. If the question asks for bytes, add the extra for accented letters.

PHẢI THUỘC / MEMORISE – ASCII 7 bit: 128 mã. ASCII mở rộng 8 bit: 256 mã. Mã “A” là $65 = 01000001$, mã “a” là 97, mã chữ số “0” là 48. UTF-8: từ 1 đến 4 byte một kí tự. *ASCII 7-bit: 128 codes. Extended ASCII 8-bit: 256 codes. The code of “A” is $65 = 01000001$, “a” is 97, the digit “0” is 48. UTF-8: 1 to 4 bytes per character.*

Số hóa văn bản – Digitising text

Số hóa văn bản là đưa văn bản vào máy dưới dạng tệp. Việc này làm bằng phần mềm soạn thảo như Word hoặc Writer. Gần đây còn có cách đọc lời vào micro: máy nhận dạng tiếng nói và tạo ra văn bản. Tệp văn bản lưu ở bộ nhớ ngoài.

Digitising text means getting it into the machine as a file. This is done with word-processing software such as Word or Writer. More recently there is also speaking into a microphone: the machine recognises the speech and produces text. A text file is kept in external memory.

1.4. Cách đổi số thập phân sang nhị phân và ngược lại – How to convert between decimal and binary

Một công tắc đèn chỉ có hai vị trí: tắt hoặc bật. Ba công tắc cho $2 \times 2 \times 2 = 8$ cách bật tắt khác nhau. Máy tính được làm từ vô số linh kiện hai trạng thái như vậy, nên nó ghi mọi thứ, kể cả số, bằng hai chữ số 0 và 1. Cách ghi số bằng hai chữ số ấy gọi là hệ nhị phân.

A light switch has only two positions: off or on. Three switches give $2 \times 2 \times 2 = 8$ different patterns of on and off. A computer is built from countless such two-state components, so it records everything, numbers included, with the two digits 0 and 1. Writing numbers with those two digits is called the binary system.

Bắt đầu từ hệ thập phân quen thuộc. Số 472 là $4 \times 10^2 + 7 \times 10^1 + 2 \times 10^0$: mỗi chữ số đứng ở một hàng, và hàng đó là một lũy thừa của 10. Hệ nhị phân làm y hệt, chỉ thay 10 bằng 2.

Start from the familiar decimal system. The number 472 is $4 \times 10^2 + 7 \times 10^1 + 2 \times 10^0$: each digit sits in a position, and the position is a power of 10. The binary system does the same, with 2 in place of 10.

ĐẶC ĐIỂM 8 (Hệ nhị phân) / FEATURES 8 (Binary system) – Hệ nhị phân (hệ đếm cơ số 2) chỉ dùng hai chữ số 0 và 1, gọi là chữ số nhị phân. Mỗi số được biểu diễn bởi một dãy

chữ số nhị phân. Chữ số ở một hàng có giá trị gấp 2 lần chữ số đó ở hàng liền bên phải, nên chữ số 1 ở vị trí thứ k kể từ phải sang trái có giá trị 2^{k-1} . Khi cần ghi rõ cơ số, viết cơ số làm chỉ số dưới: $19_{10} = 10011_2$.

*The **binary system** (base 2) uses only the digits 0 and 1, called binary digits. Each number is represented by a string of binary digits. A digit in one position is worth twice the same digit one position to its right, so a 1 in the k -th place from the right is worth 2^{k-1} . When the base must be shown, write it as a subscript: $19_{10} = 10011_2$.*

Với n chữ số nhị phân, có 2^n dãy khác nhau. Một byte có 8 bit nên ghi được $2^8 = 256$ giá trị, từ 0 đến 255. Đó cũng là lí do bảng ASCII mở rộng có 256 mã.

With n binary digits there are 2^n different strings. A byte has 8 bits, so it can hold $2^8 = 256$ values, from 0 to 255. That is also why the extended ASCII set has 256 codes.

Đổi số thập phân sang nhị phân bằng phép chia liên tiếp cho 2 – *Decimal to binary by repeated division by 2*

Cần chuẩn bị: giấy nháp kẻ ba cột “Phép chia”, “Thương”, “Dư”. Mỗi lần chia cho 2, số dư chỉ có thể là 0 hoặc 1, và đó chính là một chữ số nhị phân của kết quả.

What you need: scrap paper ruled into three columns, “Division”, “Quotient”, “Remainder”. Each division by 2 leaves a remainder of 0 or 1, and that is one binary digit of the answer.

1. Viết số cần đổi vào ô đầu của cột “Phép chia”.

Write the number to convert in the first cell of the “Division” column.

2. Chia số đó cho 2. Ghi thương vào cột “Thương”, ghi số dư (0 hoặc 1) vào cột “Dư”.

Divide it by 2. Write the quotient in “Quotient” and the remainder (0 or 1) in “Remainder”.

3. Lấy thương vừa có làm số bị chia của dòng kế tiếp. Lặp lại bước 2.

Take that quotient as the dividend of the next row. Repeat step 2.

4. Dừng khi thương bằng 0. Dòng cuối cùng luôn có thương 0 và số dư 1.

Stop when the quotient is 0. The last row always has quotient 0 and remainder 1.

5. Đọc cột “Dư” từ dưới lên trên. Dãy đọc được là số nhị phân. Nếu đề cần đủ 8 bit, thêm số 0 vào bên trái.

Read the “Remainder” column from bottom to top. The string you read is the binary number. If 8 bits are needed, pad with zeros on the left. ■

$$\begin{array}{rcl}
 37 : 2 & = & 18 \quad \text{dur } 1 \\
 18 : 2 & = & 9 \quad \text{dur } 0 \\
 9 : 2 & = & 4 \quad \text{dur } 1 \\
 4 : 2 & = & 2 \quad \text{dur } 0 \\
 2 : 2 & = & 1 \quad \text{dur } 0 \\
 1 : 2 & = & 0 \quad \text{dur } 1
 \end{array}$$

$$37 = 100101 \text{ (nhị phân / binary)}$$



Hình 2. Đổi 37 sang nhị phân: chia liên tiếp cho 2 đến khi thương bằng 0, đọc số dư từ dưới lên
Figure 2. Converting 37 to binary: divide repeatedly by 2 until the quotient is 0, then read the remainders from bottom to top

Đổi số nhị phân sang thập phân – Binary to decimal

- Viết số nhị phân ra giấy. Đánh số vị trí cho từng chữ số, từ phải sang trái, bắt đầu từ 0.

Write the binary number down. Number the positions of its digits from right to left, starting at 0.

- Với mỗi chữ số 1, viết giá trị 2 mũ vị trí của nó: vị trí 0 là 1, vị trí 1 là 2, vị trí 2 là 4, vị trí 3 là 8, và cứ thế gấp đôi.

For each digit 1, write the value 2 to the power of its position: position 0 is 1, position 1 is 2, position 2 is 4, position 3 is 8, doubling each time.

- Bỏ qua các chữ số 0. Cộng các giá trị đã viết. Tổng là số thập phân.

Ignore the digits 0. Add up the values written. The total is the decimal number. ■

Cách thứ hai để đổi sang nhị phân, hợp với đề cần đủ 8 bit: lập hàng 128, 64, 32, 16, 8, 4, 2, 1. Đi từ trái sang phải, nếu số còn lại không nhỏ hơn giá trị ở hàng thì ghi 1 và trừ đi giá trị đó, ngược lại ghi 0.

A second way to convert to binary, handy when 8 bits are required: write the row 128, 64, 32, 16, 8, 4, 2, 1. Go from left to right; if the remaining number is not smaller than the value in that place, write 1 and subtract that value, otherwise write 0.

PHÂN TÍCH ĐỀ / READING THE QUESTION – Đề có hai chiều đổi, nên làm riêng từng chiều và kiểm bằng chiều ngược lại. Kết quả sai thường lộ ngay khi kiểm: số thập phân tính lại không khớp số ban đầu. Đề cho 8 chữ số nhị phân bắt đầu bằng 1, nên kết quả phải nằm giữa 128 và 255.

The question has two directions, so do each separately and check with the reverse. A wrong answer usually shows at once: the decimal recomputed does not match the original. The question gives 8 binary digits starting with 1, so the result must lie between 128 and 255.

VÍ DỤ 4 (Đổi cơ số hai chiều) / EXAMPLE 4 (Converting in both directions) – Đổi 94 sang hệ nhị phân (đủ 8 bit). Đổi 11010110_2 sang hệ thập phân. Kiểm tra cả hai kết quả.
Convert 94 to binary (8 bits). Convert 11010110_2 to decimal. Check both results.

Hướng dẫn làm bài. / How to answer.

- Bước 1: lập bảng chia liên tiếp cho 2 với số 94 (bảng bên dưới).
Step 1: build the repeated-division table for 94 (the table below).
- Bước 2: đọc cột Dư từ dưới lên: 1011110. Thêm một số 0 bên trái cho đủ 8 bit: 01011110.
Step 2: read the Remainder column from bottom to top: 1011110. Pad one zero on the left to make 8 bits: 01011110.
- Bước 3: đổi 11010110_2 . Các chữ số 1 ở vị trí 7, 6, 4, 2, 1. Giá trị 128, 64, 16, 4, 2.
Step 3: convert 11010110_2 . The digits 1 are at positions 7, 6, 4, 2, 1, worth 128, 64, 16, 4, 2.
- Bước 4: cộng $128 + 64 + 16 + 4 + 2 = 214$.
Step 4: add $128 + 64 + 16 + 4 + 2 = 214$.
- Bước 5: kiểm tra. Đổi 1011110 về thập phân: $64 + 16 + 8 + 4 + 2 = 94$. Khớp.
Step 5: check. Convert 1011110 back: $64 + 16 + 8 + 4 + 2 = 94$. It matches.

```
n = 94
print(bin(n))
print(format(n, "08b"))
print(int("11010110", 2))
```

Kết quả:

Output:

```
0b1011110
01011110
214
```

Hàm bin viết số bằng nhị phân kèm tiền tố 0b. Hàm format với “08b” viết số bằng nhị phân đủ 8 chữ số, thiếu thì thêm số 0 bên trái. Hàm int với đối số thứ hai là 2 đổi một dãy nhị phân sang số thập phân. Máy in 214, khớp với phép tính tay.

The function bin writes a number in binary with the prefix 0b. The function format with “08b” writes a number in binary with 8 digits, padding with zeros on the left if needed. The function int with 2 as its second argument converts a binary string to a decimal number. The machine prints 214, matching the hand calculation.

BÀI VIẾT MẪU / MODEL WRITING – Muốn đổi 94 sang nhị phân, chia liên tiếp cho 2 và ghi lại số dư. 94 chia 2 được 47 dư 0. 47 chia 2 được 23 dư 1. 23 chia 2 được 11 dư 1. 11 chia 2 được 5 dư 1. 5 chia 2 được 2 dư 1. 2 chia 2 được 1 dư 0. Cuối cùng 1 chia 2 được 0 dư 1. Phép chia dừng khi thương bằng 0, vì khi đó không còn chữ số nào để lấy ra. Các số dư theo thứ tự xuất hiện là 0, 1, 1, 1, 1, 0, 1. Số dư đầu tiên là chữ số hàng đơn vị, số dư cuối cùng là chữ số lớn nhất, nên phải đọc ngược từ dưới lên. Ta được 1011110. Số này chỉ có 7 chữ số. Khi lưu trong một byte phải thêm một số 0 ở bên trái, thành 01011110. Để chắc kết quả, kiểm ngược. Đánh số các chữ số của 1011110 từ phải sang trái, bắt đầu từ 0. Chữ số 1 nằm ở các vị trí 1, 2, 3, 4 và 6, ứng với 2, 4, 8, 16 và 64. Cộng lại được 94, khớp với số ban đầu.

Chiều ngược lại, đổi 11010110 sang thập phân bằng chính phép cộng ấy. Các chữ số 1 nằm ở vị trí 7, 6, 4, 2 và 1, ứng với 128, 64, 16, 4 và 2. Tổng là 214. Có thể đoán trước độ lớn: số có 8 chữ số, chữ số đầu là 1, nên nằm giữa 128 và 255, và 214 hợp lí.

Bước kiểm ngược nên làm với mọi bài, vì chỉ mất một phút và bắt được lỗi đọc số dư sai thứ tự. Hai cách làm là hai chiều của một quy tắc: mỗi chữ số nhị phân cho biết lũy thừa 2 ở hàng đó có mặt trong số hay không. Chia dư cho 2 tìm ra các chữ số ấy, còn phép cộng các lũy thừa ráp chúng lại thành số.

To convert 94 to binary, divide repeatedly by 2 and record the remainders. 94 divided by 2 is 47 remainder 0. 47 divided by 2 is 23 remainder 1. 23 divided by 2 is 11 remainder 1. 11 divided by 2 is 5 remainder 1. 5 divided by 2 is 2 remainder 1. 2 divided by 2 is 1 remainder 0. Finally 1 divided by 2 is 0 remainder 1. The division stops when the quotient is 0, because then no digit is left to take out.

The remainders in order of appearance are 0, 1, 1, 1, 1, 0, 1. The first remainder is the units digit and the last is the largest digit, so they must be read from the bottom up. This gives 1011110. It has only 7 digits. Stored in one byte, a zero is added on the left, giving 01011110.

To be sure of the result, check backwards. Number the digits of 1011110 from right to left, starting at 0. The digits 1 are at positions 1, 2, 3, 4 and 6, worth 2, 4, 8, 16 and 64. Adding gives 94, matching the original number.

In the other direction, convert 11010110 to decimal with the same addition. The digits 1 are at positions 7, 6, 4, 2 and 1, worth 128, 64, 16, 4 and 2. The total is 214. The size can be predicted: the number has 8 digits and starts with 1, so it lies between 128 and 255, and 214 is reasonable.

The backward check is worth doing on every question, because it takes a minute and catches remainders read in the wrong order. The two methods are the two directions of one rule: each binary digit says whether the power of 2 at that place is present in the number. Dividing with remainders finds those digits, and adding the powers puts them back together into the number. ■

Phép chia <i>Division</i>	Thương <i>Quotient</i>	Dư <i>Remainder</i>
94 : 2	47	0
47 : 2	23	1
23 : 2	11	1
11 : 2	5	1
5 : 2	2	1
2 : 2	1	0
1 : 2	0	1

Dòng cuối có thương 0, nên phép chia dừng ở đó. Đọc cột Dư từ dưới lên được 1, 0, 1, 1, 1, 1, 0.

The last row has quotient 0, so the division stops there. Reading the Remainder column from bottom to top gives 1, 0, 1, 1, 1, 1, 0.

LỖI HAY GẶP / COMMON MISTAKE – Đọc cột Dư từ trên xuống. Số dư đầu tiên là chữ số nhỏ nhất, nên đọc xuôi cho ra số bị đảo ngược. Kiểm bằng cách đổi ngược về thập phân.

Reading the Remainder column from top to bottom. The first remainder is the smallest digit, so reading forwards gives the number reversed. Check by converting back to decimal.

LỖI HAY GẶP / COMMON MISTAKE – Quên thêm số 0 bên trái khi đề yêu cầu đủ 8 bit. Số 01011110 và 1011110 cùng giá trị, nhưng đề đã nói “8 bit” thì phải viết đủ 8 chữ số.
Forgetting to pad zeros on the left when 8 bits are required. 01011110 and 1011110 have the same value, but when the question says “8 bits” all 8 digits must be written.

Biểu diễn số nguyên trong máy tính, kể cả số âm – *Representing integers, including negatives*

Số nguyên không dấu được lưu đúng bằng dạng nhị phân của nó, thêm số 0 bên trái cho tròn từng byte. Số 45 là 00101101 (một byte). Số 300 là 100101100 có 9 chữ số, nên phải dùng hai byte và thêm 7 số 0: 0000000100101100.

An unsigned integer is stored as its binary form, padded with zeros on the left to whole bytes. The number 45 is 00101101 (one byte). The number 300 is 100101100, with 9 digits, so two bytes are needed with 7 zeros added: 0000000100101100.

KHÁI NIỆM 9 (Số nguyên có dấu: mã thuận, mã bù 1, mã bù 2) / CONCEPT 9 (Signed integers: sign-magnitude, 1's and 2's complement) – Số nguyên có dấu dùng bit ngoài cùng bên trái làm bit dấu: 0 là dấu cộng, 1 là dấu trừ. Có ba cách mã hóa. Với số dương, cả ba giống nhau: bit dấu 0, rồi dạng nhị phân của số. Với số âm: **mã thuận** giữ phần độ lớn và đổi bit dấu thành 1; **mã bù 1** đảo mọi bit của mã số dương (0 thành 1, 1 thành 0); **mã bù 2** là mã bù 1 cộng thêm 1.

*A signed integer uses the leftmost bit as the sign bit: 0 for plus, 1 for minus. There are three encodings. For a positive number all three agree: sign bit 0, then the binary form of the number. For a negative number: **sign-magnitude** keeps the magnitude and sets the sign bit to 1; **1's complement** flips every bit of the positive code (0 becomes 1, 1 becomes 0); **2's complement** is the 1's complement plus 1.*

Thử với số 6 trong 8 bit: 6 là 00000110. Số -6 có mã thuận 10000110, mã bù 1 là 11111001, mã bù 2 là 11111010. Mã bù 2 đúng bằng mã bù 1 cộng 1. Với 8 bit, mã bù 1 của số n cũng bằng $255 - n$, và mã bù 2 bằng $256 - n$.

Try the number 6 in 8 bits: 6 is 00000110. The number -6 has sign-magnitude code 10000110, 1's complement 11111001 and 2's complement 11111010. The 2's complement is exactly the 1's complement plus 1. With 8 bits, the 1's complement of n also equals $255 - n$ and the 2's complement equals $256 - n$.

PHÂN TÍCH ĐỀ / READING THE QUESTION – Đề hỏi số âm trong 8 bit và nêu tên ba mã, nên phải trả lời ba dòng. Bước nào cũng bắt đầu từ mã của số dương tương ứng. Cách kiểm nhanh: mã thuận và mã bù 1 của một số âm luôn bắt đầu bằng bit 1.

The question asks for a negative number in 8 bits and names three codes, so give three lines. Every step starts from the code of the matching positive number. A quick check: the sign-magnitude and 1's complement codes of a negative number always begin with bit 1.

VÍ DỤ 5 (Số âm trong 8 bit) / EXAMPLE 5 (A negative number in 8 bits) – Viết -45 bằng 8 bit theo mã thuận, mã bù 1 và mã bù 2. Cộng 45 với mã bù 2 của -45 để kiểm tra.
Write -45 in 8 bits as sign-magnitude, 1's complement and 2's complement. Add 45 to the 2's complement of -45 as a check.

Hướng dẫn làm bài. / How to answer.

- Bước 1: đổi 45 sang nhị phân đủ 8 bit: 00101101 ($45 = 32 + 8 + 4 + 1$).
Step 1: convert 45 to 8-bit binary: 00101101 ($45 = 32 + 8 + 4 + 1$).
- Bước 2: mã thuận, đổi bit dấu thành 1: 10101101.
Step 2: sign-magnitude, set the sign bit to 1: 10101101.
- Bước 3: mã bù 1, đảo mọi bit của 00101101: 11010010.
Step 3: 1's complement, flip every bit of 00101101: 11010010.
- Bước 4: mã bù 2, lấy mã bù 1 cộng 1: $11010010 + 1 = 11010011$.
Step 4: 2's complement, add 1 to the 1's complement: $11010010 + 1 = 11010011$.
- Bước 5: kiểm tra, $00101101 + 11010011 = 100000000$. Máy lưu số trong một ô nhớ (chỗ giữ một số) chỉ rộng 8 bit, nên giữ 8 bit bên phải, được 00000000, tức $45 + (-45) = 0$.
Step 5: check, $00101101 + 11010011 = 100000000$. The machine stores the number in a memory cell (a place that holds one number) only 8 bits wide, so keep the right-hand 8 bits to get 00000000, that is $45 + (-45) = 0$.

```
n = 45
print(format(n, "08b"))
print(format(255 - n, "08b"))
print(format(256 - n, "08b"))
```

Kết quả:*Output:*

```
00101101
11010010
11010011
```

Ba dòng máy in lần lượt là mã của 45, mã bù 1 của -45 ($255 - 45$) và mã bù 2 của -45 ($256 - 45$), khớp với tay.

The three lines are the code of 45, the 1's complement of -45 ($255 - 45$) and the 2's complement of -45 ($256 - 45$), matching the hand work. ■

Cách mã hóa <i>Encoding</i>	+45	-45
Mã thuận <i>Sign-magnitude</i>	00101101	10101101
Mã bù 1 <i>1's complement</i>	00101101	11010010
Mã bù 2 <i>2's complement</i>	00101101	11010011

Cột +45 giống nhau ở cả ba dòng, chỉ cột -45 khác nhau.

The +45 column is the same in all three rows; only the -45 column differs.

NHẬN XÉT / REMARK – Với 8 bit không dấu, các giá trị đi từ 0 đến 255. Mã thuận và mã bù 1 dành 1 bit cho dấu nên còn 7 bit độ lớn, ghi được từ -127 đến $+127$. Mã bù 2 ghi được

từ -128 đến $+127$.

With 8 unsigned bits the values run from 0 to 255. Sign-magnitude and 1's complement spend 1 bit on the sign, leaving 7 bits of magnitude, so they cover -127 to $+127$. 2's complement covers -128 to $+127$.

LỖI HAY GẶP / COMMON MISTAKE – Quên đảo cả bit dấu khi lập mã bù 1. Mã của số dương có bit dấu 0, đảo thì thành 1, và bit 1 ấy đúng là dấu trừ.

Forgetting to flip the sign bit too when forming the 1's complement. A positive code has sign bit 0; flipped it becomes 1, and that 1 is exactly the minus sign.

1.5. Cách cộng và nhân hai số nhị phân – How to add and multiply binary numbers

Khi cộng $7 + 5$ trên giấy, bạn viết 2 và nhớ 1 sang hàng chục. Cộng nhị phân làm đúng như vậy, chỉ khác là hàng đầy khi đạt 2 chứ không phải 10. Vì thế phép cộng hai số nhị phân có thể làm bằng bảng cộng bốn dòng.

When you add $7 + 5$ on paper, you write 2 and carry 1 to the tens. Binary addition does exactly that, except a place fills up at 2 rather than 10. So adding two binary numbers needs only a four-row addition table.

x	y	$x + y$	$x \times y$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	10 (viết 0, nhớ 1) 10 (write 0, carry 1)	1

Dòng cuối của bảng là chỗ cần nhớ: $1 + 1 = 10$ trong nhị phân, tức là viết 0 và nhớ 1 sang hàng bên trái.

The last row of the table is the one to remember: $1 + 1 = 10$ in binary, which means write 0 and carry 1 to the left.

Các bước cộng – Steps for addition

- Viết hai số thẳng hàng theo chữ số cuối. Số ngắn hơn thì thêm số 0 vào bên trái cho cùng độ dài.

Write the two numbers aligned on their last digits. Pad the shorter one with zeros on the left to the same length.

- Kẻ thêm một dòng “nhớ” phía trên. Cộng từng cột từ phải sang trái: hai chữ số của cột cộng với số nhớ đưa sang từ cột bên phải.

Add a “carry” row above. Add column by column from right to left: the two digits of the column plus the carry from the column to its right.

- Tổng của cột bằng 0 hoặc 1: ghi xuống. Tổng bằng 2 (10): ghi 0, nhớ 1. Tổng bằng 3 (11): ghi 1, nhớ 1.

A column total of 0 or 1: write it down. A total of 2 (10): write 0, carry 1. A total of 3 (11): write 1, carry 1.

4. Hết cột mà còn nhớ 1 thì ghi thêm 1 ở đầu kết quả.*If a carry of 1 is left after the last column, write an extra 1 at the front of the result.***5. Kiểm tra: đổi hai số hạng và tổng sang thập phân, xem tổng có đúng không.***Check: convert both addends and the sum to decimal and see whether the sum is right. ■*

nhớ / carry	1	1	1	1	
a =	0	1	0	1	1
b =	0	0	1	1	1
tổng / sum	1	0	0	1	0

$1011 + 111$
= 18 (thập phân / decimal)

Hình 3. Cộng 01011 với 00111: cột nào tổng đạt 2 hoặc 3 thì ghi 0 hoặc 1 và nhớ 1 sang cột bên trái*Figure 3. Adding 01011 and 00111: where a column total reaches 2 or 3, write 0 or 1 and carry 1 to the next column on the left***Các bước nhân – Steps for multiplication**

Nhân nhị phân giống nhân thập phân, nhưng dễ hơn vì mỗi chữ số của thừa số thứ hai chỉ là 0 hoặc 1. Nhân với 0 cho dãy toàn số 0, nhân với 1 cho chính thừa số thứ nhất.

Binary multiplication is like decimal multiplication but easier, since each digit of the second factor is only 0 or 1. Multiplying by 0 gives a row of zeros; multiplying by 1 gives the first factor itself.

1. Viết thừa số thứ nhất, bên dưới viết thừa số thứ hai.*Write the first factor, and the second factor below it.***2. Lấy từng chữ số của thừa số thứ hai, từ phải sang trái. Chữ số 1 thì chép lại thừa số thứ nhất, chữ số 0 thì viết dãy số 0.***Take each digit of the second factor from right to left. For a digit 1 copy the first factor; for a digit 0 write a row of zeros.***3. Mỗi dòng sau dịch sang trái thêm một chỗ so với dòng trước, như khi nhân thập phân.***Shift each following row one place further left than the one before, as in decimal multiplication.***4. Cộng tất cả các dòng bằng phép cộng nhị phân.***Add all the rows with binary addition.***5. Kiểm tra bằng cách nhân hai số thập phân tương ứng.***Check by multiplying the two matching decimal numbers. ■*

Cả hai phép tính nằm trong một quy trình ba bước: đổi các số hạng sang nhị phân, tính trong hệ nhị phân, rồi đổi kết quả về thập phân. Máy tính làm đúng như thế, và bạn dùng chính quy trình đó để tự kiểm tra bài.

Both operations fit one three-step routine: convert the operands to binary, calculate in binary, then convert the result back to decimal. A computer does just that, and you can use the same routine to check your own work.

PHÂN TÍCH ĐỀ / READING THE QUESTION – Đề yêu cầu cả cộng lẫn nhân và dẫn kiểm tra bằng thập phân. Hai số cộng có độ dài khác nhau (7 và 6 chữ số) nên phải thêm số 0 bên trái. Bảng cộng có dòng “nhớ” giúp không bỏ sót.

The question asks for both addition and multiplication and tells you to check in decimal. The two numbers to add have different lengths (7 and 6 digits), so pad with a zero on the left. A table with a “carry” row prevents omissions.

VÍ DỤ 6 (Cộng và nhân nhị phân) / EXAMPLE 6 (Binary addition and multiplication) –

Tính $1011011_2 + 110111_2$ và $1011_2 \times 110_2$ trong hệ nhị phân. Kiểm tra bằng thập phân.
Calculate $1011011_2 + 110111_2$ and $1011_2 \times 110_2$ in binary. Check in decimal.

Hướng dẫn làm bài. / How to answer.

- Bước 1:** thêm số 0 cho thẳng hàng: 1011011 và 0110111 . Đây là các cột từ phải sang trái; bảng bên dưới ghi số nhớ, hai số hạng và tổng.
Step 1: pad with a zero so the numbers align: 1011011 and 0110111 . The columns run from right to left; the table below records the carries, the two addends and the sum.
- Bước 2:** cột 0: $1 + 1 = 10$, ghi 0, nhớ 1. Cột 1: $1 + 1 + 1 = 11$, ghi 1, nhớ 1. Cột 2: $0 + 1 + 1 = 10$, ghi 0, nhớ 1. Cột 3: $1 + 0 + 1 = 10$, ghi 0, nhớ 1.
Step 2: column 0: $1 + 1 = 10$, write 0, carry 1. Column 1: $1 + 1 + 1 = 11$, write 1, carry 1. Column 2: $0 + 1 + 1 = 10$, write 0, carry 1. Column 3: $1 + 0 + 1 = 10$, write 0, carry 1.
- Bước 3:** cột 4: $1 + 1 + 1 = 11$, ghi 1, nhớ 1. Cột 5: $0 + 1 + 1 = 10$, ghi 0, nhớ 1. Cột 6: $1 + 0 + 1 = 10$, ghi 0, nhớ 1. Cột 7: $0 + 0 + 1 = 1$, ghi 1. Tổng 10010010 .
Step 3: column 4: $1 + 1 + 1 = 11$, write 1, carry 1. Column 5: $0 + 1 + 1 = 10$, write 0, carry 1. Column 6: $1 + 0 + 1 = 10$, write 0, carry 1. Column 7: $0 + 0 + 1 = 1$, write 1. The sum is 10010010 .
- Bước 4:** kiểm tra, $1011011_2 = 91$, $110111_2 = 55$, $91 + 55 = 146$, và $10010010_2 = 128 + 16 + 2 = 146$. Khớp.
Step 4: check, $1011011_2 = 91$, $110111_2 = 55$, $91 + 55 = 146$, and $10010010_2 = 128 + 16 + 2 = 146$. It matches.
- Bước 5:** nhân 1011 với 110 . Chữ số cuối là 0 nên dòng đầu là 0000. Chữ số giữa là 1 nên chép 1011 dịch sang trái một chỗ: 10110 . Chữ số đầu là 1 nên chép 1011 dịch hai chỗ: 101100 .
Step 5: multiply 1011 by 110 . The last digit is 0, so the first row is 0000. The middle digit is 1, so copy 1011 shifted one place left: 10110 . The first digit is 1, so copy 1011 shifted two places: 101100 .
- Bước 6:** cộng $0000 + 10110 + 101100 = 1000010$. Kiểm tra: $11 \times 6 = 66$ và $1000010_2 = 64 + 2 = 66$. Khớp.
Step 6: add $0000 + 10110 + 101100 = 1000010$. Check: $11 \times 6 = 66$ and $1000010_2 = 64 + 2 = 66$. It matches.

```
a = int("1011", 2)
b = int("110", 2)
print(format(a * b, "b"))
print(format(int("1011011", 2) + int("110111", 2), "b"))
```

Kết quả:

Output:

```
1000010
10010010
```

Hàng Row	7	6	5	4	3	2	1	0
Nhớ Carry	1	1	1	1	1	1	1	
Số thứ nhất First	0	1	0	1	1	0	1	1
Số thứ hai Second	0	0	1	1	0	1	1	1
Tổng Sum	1	0	0	1	0	0	1	0

Dòng nhớ có số 1 ở các cột từ 1 đến 7. Cột 0 đến cột 6 đều cộng ra từ 2 trở lên nên đều nhớ sang cột bên trái, và cột 7 nhận số nhớ cuối cùng.

The carry row has a 1 in columns 1 to 7. Columns 0 to 6 each total 2 or more, so each carries to the column on its left, and column 7 receives the last carry.

Máy tính làm mọi phép số học trực tiếp trên hệ nhị phân. Phép cộng làm từng cột từ phải sang trái, có nhớ. Phép nhân chia thành các dòng toàn số 0 hoặc dòng sao chép thừa số, dịch trái rồi cộng.

A computer does all its arithmetic directly in the binary system. Addition goes column by column from right to left with carries. Multiplication splits into rows of zeros or copies of the factor, shifted left and added.

LỖI HAY GẶP / COMMON MISTAKE – Viết $1 + 1 = 2$ trong bảng nhị phân. Trong hệ này không có chữ số 2: $1 + 1 = 10$, viết 0 và nhớ 1.

Writing $1 + 1 = 2$ in a binary table. This system has no digit 2: $1 + 1 = 10$, write 0 and carry 1.

LỖI HAY GẶP / COMMON MISTAKE – Quên số nhớ ở cột có hai chữ số 0 hoặc có 0 và 1. Số nhớ cộng vào cột kế tiếp bất kể cột ấy có chữ số nào. Dòng “nhớ” kẻ riêng giúp tránh sót.

Forgetting the carry in a column with two 0s or a 0 and a 1. A carry joins the next column whatever its digits. A separately ruled “carry” row prevents this.

NHẬN XÉT / REMARK – Nhân một số nhị phân với 10_2 (tức 2) chỉ cần thêm một số 0 vào bên phải, giống nhân thập phân với 10. Dùng nó để kiểm nhanh: $1011_2 \times 10_2 = 10110_2$, đúng bằng $11 \times 2 = 22$.

Multiplying a binary number by 10_2 (that is, 2) just appends a zero on the right, like multiplying a decimal by 10. Use it as a quick check: $1011_2 \times 10_2 = 10110_2$, which is $11 \times 2 = 22$.

1.6. Dữ liệu logic, phép toán logic và cách lập bảng chân trị – Logical data, logical operations and how to build a truth table

Bố nói: “Con được xem phim nếu đã làm xong bài VÀ đã dọn bàn.” Có bốn khả năng: làm bài rồi và dọn rồi, làm bài rồi mà chưa dọn, chưa làm mà dọn rồi, chưa làm cũng chưa dọn. Chỉ khả năng đầu tiên được xem phim. Chữ “VÀ” trong câu nói là một phép toán logic.

Dad says: “You may watch the film if you have finished your homework AND tidied your desk.” There are four possibilities: homework done and desk tidy, homework done but desk not tidy, homework not done but desk tidy, neither done. Only the first lets you watch. The word “AND” in the sentence is a logical operation.

Mệnh đề là một khẳng định hoặc đúng hoặc sai, ví dụ “9 là số nguyên tố” là mệnh đề sai. “Đúng” và “Sai” là các **giá trị chân lí** (giá trị logic) của mệnh đề, thường ghi gọn là 1 và 0. **Đại lượng logic** là đại lượng chỉ nhận giá trị logic. Các biểu thức so sánh trong toán, như $3 > 5$, đều là mệnh đề.

*A **proposition** is a statement that is either true or false; for example “9 is a prime number” is a false proposition. “True” and “False” are the **truth values** (logical values) of a proposition, usually written briefly as 1 and 0. A **Boolean quantity** is one that takes only logical values. The comparison expressions of mathematics, such as $3 > 5$, are all propositions.*

ĐẶC ĐIỂM 10 (Các phép toán logic AND, OR, XOR, NOT) / FEATURES 10 (Logical operations AND, OR, XOR, NOT) – Bốn phép toán logic cơ bản trên hai mệnh đề p và q:

p AND q (nhân logic): đúng khi cả p và q đều đúng.

p OR q (cộng logic): sai chỉ khi cả p và q đều sai.

p XOR q (cộng loại trừ, eXclusive OR): đúng khi p và q có giá trị khác nhau.

NOT p (phủ định): đúng nếu p sai, sai nếu p đúng.

Kết quả chỉ phụ thuộc giá trị 1 hoặc 0 của các mệnh đề, không phụ thuộc chữ nghĩa của chúng, nên cũng áp dụng được cho từng bit.

Four basic logical operations on two propositions p and q:

p AND q (logical product): true when both p and q are true.

p OR q (logical sum): false only when both p and q are false.

p XOR q (exclusive OR): true when p and q have different values.

NOT p (negation): true if p is false, false if p is true.

The result depends only on the 1 or 0 of the propositions, not on their wording, so the operations also apply bit by bit.

Bảng chân trị liệt kê kết quả của mỗi phép toán cho mọi cặp giá trị của p và q. Dòng 1 1 có AND bằng 1, OR bằng 1, XOR bằng 0, NOT p bằng 0.

A truth table lists the result of each operation for every pair of values of p and q. In the row 1 1, AND is 1, OR is 1, XOR is 0, and NOT p is 0.

p	q	p AND q	p OR q	p XOR q	NOT p
1	1	1	1	0	0
1	0	0	1	1	0
0	1	0	1	1	1
0	0	0	0	0	1

AND chỉ có một dòng bằng 1, OR chỉ có một dòng bằng 0, và XOR bằng 1 đúng ở hai dòng mà p khác q.

AND equals 1 on only one row, OR equals 0 on only one row, and XOR equals 1 on exactly the two rows where p differs from q.

Thứ tự tính trong một biểu thức logic theo sách giáo khoa: dấu ngoặc trước, rồi NOT, còn AND và OR ngang nhau và tính từ trái sang phải. Khi viết bằng Python, hãy dùng ngoặc cho mọi biểu thức có cả and lẫn or, vì Python tính and trước or. Thử hai dòng sau.

The textbook order of evaluation in a logical expression: brackets first, then NOT, while AND and OR rank equally and go from left to right. In Python, use brackets for every expression containing both and and or, because Python evaluates and before or. Try the two lines below.

```
print(1 or 0 and 0)
print((1 or 0) and 0)
```

Kết quả:

Output:

```
1
0
```

Dòng đầu Python tính 0 and 0 trước, được 0, rồi 1 or 0, được 1. Dòng hai có ngoặc nên tính 1 or 0 trước, được 1, rồi 1 and 0, được 0. Hai dòng cho kết quả khác nhau, nên ngoặc quyết định đáp án.

On the first line Python evaluates 0 and 0 first, getting 0, then 1 or 0, getting 1. The second line has brackets, so 1 or 0 comes first, giving 1, then 1 and 0, giving 0. The two lines differ, so the brackets decide the answer.

Python coi số 0 là giá trị Sai và mọi số khác 0 là giá trị Đúng. Các phép and, or và các dấu &, |, ^ làm việc với bit. Dấu & là AND, dấu | là OR, dấu mũ ^ là XOR.

Python treats the number 0 as False and every nonzero number as True. The operations and, or and the signs &, |, ^ work with bits. The sign & is AND, | is OR, and the caret ^ is XOR.

```
print(bool(0), bool(7))
a = int("10110110", 2)
mat_na = int("00001111", 2)
print(format(a & mat_na, "08b"))
print(format(a | mat_na, "08b"))
print(format(a ^ mat_na, "08b"))
```

Kết quả:

Output:

```
False True
00000110
10111111
10111001
```

Hàm bool đổi một số thành giá trị logic, nên dòng đầu in False True (0 là Sai, 7 là Đúng). Số thứ hai là mặt nạ, tức số dùng để chọn bit cần giữ. Ba dòng sau làm AND, OR, XOR cho từng cặp bit tương ứng của hai byte. Với AND, chỗ mặt nạ có 0 thì bit kết quả là 0, còn chỗ mặt nạ có 1 thì giữ nguyên bit của số gốc.

The function bool turns a number into a logical value, so the first line prints False True (0 is False, 7 is True). The second number is a mask, a number used to choose which bits to keep. The next three lines apply AND, OR and XOR to each matching pair of bits of the two bytes. With AND, where the mask has 0 the result bit is 0, and where the mask has 1 the bit of the original number is kept.

Trong đời sống, mọi cặp trạng thái đối lập như sáng – tối, bật – tắt, có – không đều là dữ liệu logic. Chỉ cần 1 bit là đủ: 1 là Đúng, 0 là Sai. Nhưng một số ngôn ngữ lập trình có quy ước riêng và không dùng đúng một bit cho giá trị logic.

In daily life every pair of opposite states such as light – dark, on – off, yes – no is logical data. One bit is enough: 1 is True, 0 is False. But some programming languages have their own conventions and do not use exactly one bit for a logical value.

Cách lập bảng chân trị cho biểu thức có ba biến – *Building a truth table for an expression with three variables*

1. Đếm số biến n . Bảng có 2^n dòng: hai biến thì 4 dòng, ba biến thì 8 dòng.
Count the variables n . The table has 2^n rows: 4 for two variables, 8 for three.
2. Viết các cột biến p, q, r . Điền theo thứ tự đếm nhị phân từ 000 đến 111, để không sót và không lặp dòng nào.
Write the variable columns p, q, r . Fill them in binary counting order from 000 to 111, so no row is missed or repeated.
3. Thêm một cột cho mỗi biểu thức con (một phần của biểu thức, như $p \text{ OR } q$ trong $(p \text{ OR } q) \text{ AND NOT } r$), bắt đầu từ biểu thức trong ngoặc trong cùng, rồi đến NOT, rồi đến phép toán ngoài.
Add one column for each sub-expression (a part of the expression, such as $p \text{ OR } q$ in $(p \text{ OR } q) \text{ AND NOT } r$), starting with the innermost brackets, then NOT, then the outer operation.
4. Điền từng dòng một. Mỗi ô chỉ làm một phép toán, dựa vào bảng của phép đó.
Fill row by row. Each cell performs one operation, using the table for that operation.
5. Đọc cột cuối: đếm số dòng bằng 1. Muốn so hai biểu thức thì đặt hai cột cuối cạnh nhau: giống nhau ở cả 8 dòng thì hai biểu thức tương đương (luôn cho cùng kết quả).
Read the last column: count the rows equal to 1. To compare two expressions put their last columns side by side: identical on all 8 rows means the two are equivalent (always give the same result). ■

PHÂN TÍCH ĐỀ / READING THE QUESTION – Đề cho ba điều kiện và một quy tắc có cả OR lẫn NOT. Dịch quy tắc thành biểu thức trước, rồi đặt ngoặc đúng chỗ. Đề còn đòi so với một biểu thức khác, nên bảng cần thêm các cột của biểu thức thứ hai.

The question gives three conditions and a rule containing both OR and NOT. Translate the rule into an expression first, then place the brackets correctly. It also asks for a comparison with another expression, so the table needs extra columns for the second expression.

VÍ DỤ 7 (Bảng chân trị ba biến) / EXAMPLE 7 (A three-variable truth table) – Robot hút bụi dừng dọn khi (pin yếu HOẶC bình chứa đầy) VÀ KHÔNG ở gần trạm sạc. Gọi p: pin yếu, q: bình đầy, r: ở gần trạm sạc. Lập bảng chân trị của $E = (p \text{ OR } q) \text{ AND NOT } r$. Nó có tương đương với $(p \text{ AND NOT } r) \text{ OR } (q \text{ AND NOT } r)$ không?

A robot vacuum cleaner stops cleaning when (battery low OR bin full) AND NOT near the charging station. Let p: battery low, q: bin full, r: near the charging station. Build the truth table of $E = (p \text{ OR } q) \text{ AND NOT } r$. Is it equivalent to $(p \text{ AND NOT } r) \text{ OR } (q \text{ AND NOT } r)$?

Hướng dẫn làm bài. / How to answer.

- Bước 1:** có 3 biến, nên bảng có 8 dòng. Các cột biến đếm từ 000 đến 111.
Step 1: there are 3 variables, so the table has 8 rows. The variable columns count from 000 to 111.
- Bước 2:** thêm cột $p \text{ OR } q$ và cột $\text{NOT } r$, rồi cột E bằng $(p \text{ OR } q) \text{ AND NOT } r$.
Step 2: add the columns $p \text{ OR } q$ and $\text{NOT } r$, then the column E equal to $(p \text{ OR } q) \text{ AND NOT } r$.
- Bước 3:** thêm các cột $p \text{ AND NOT } r$, $q \text{ AND NOT } r$ và cột E2 bằng phép OR của hai cột ấy.
Step 3: add the columns $p \text{ AND NOT } r$, $q \text{ AND NOT } r$ and the column E2, the OR of those two.
- Bước 4:** điền từng dòng (bảng bên dưới), rồi so cột E với cột E2.
Step 4: fill each row (the table below), then compare column E with column E2.

BÀI VIẾT MẪU / MODEL WRITING – Bảng chân trị của E có 3 dòng bằng 1: $p = 0, q = 1, r = 0$; $p = 1, q = 0, r = 0$; $p = 1, q = 1, r = 0$. Cả ba dòng đều có $r = 0$, nghĩa là robot chỉ dừng dọn khi không ở gần trạm sạc, và trong lúc đó phải có ít nhất một trong hai điều: pin yếu hoặc bình đầy. Cột E2 trùng với cột E ở cả 8 dòng. Vì vậy hai biểu thức tương đương. Đây là quy tắc phân phối của AND qua OR, giống phép nhân phân phối qua phép cộng: $(p \text{ OR } q) \text{ AND NOT } r$ bằng $(p \text{ AND NOT } r) \text{ OR } (q \text{ AND NOT } r)$.

The truth table of E has 3 rows equal to 1: $p = 0, q = 1, r = 0$; $p = 1, q = 0, r = 0$; $p = 1, q = 1, r = 0$. All three have $r = 0$, meaning the robot stops cleaning only when it is not near the charging station, and at that moment at least one of two things holds: battery low or bin full. Column E2 matches column E on all 8 rows. So the two expressions are equivalent. This is the distributive rule of AND over OR, like multiplication distributing over addition: $(p \text{ OR } q) \text{ AND NOT } r$ equals $(p \text{ AND NOT } r) \text{ OR } (q \text{ AND NOT } r)$. ■

p	q	r	$p \text{ OR } q$	$\text{NOT } r$	E	$p \text{ AND NOT } r$	$q \text{ AND NOT } r$	E2
0	0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0	0
0	1	0	1	1	1	0	1	1

p	q	r	p OR q	NOT r	E	p AND NOT r	q AND NOT r	E2
0	1	1	1	0	0	0	0	0
1	0	0	1	1	1	1	0	1
1	0	1	1	0	0	0	0	0
1	1	0	1	1	1	1	1	1
1	1	1	1	0	0	0	0	0

Hai cột E và E2 giống nhau từng dòng một, và cả hai bằng 1 đúng ở ba dòng có $r = 0$ cùng với p hoặc q bằng 1.

Columns E and E2 are identical row by row, and both equal 1 on exactly the three rows with $r = 0$ and p or q equal to 1.

NHẬN XÉT / REMARK – Hai biểu thức chỉ tương đương khi hai cột kết quả giống nhau ở mọi dòng. Chỉ cần một dòng khác là chúng khác nhau, nên không được dừng sớm sau vài dòng giống.

Two expressions are equivalent only if their result columns agree on every row. A single differing row makes them different, so do not stop early after a few matching rows.

LỖI HAY GẶP / COMMON MISTAKE – Nhầm OR với XOR khi đề viết “hoặc”. Trong đề logic, OR vẫn đúng khi cả hai cùng đúng. Chỉ đề nói rõ “một trong hai, không cả hai” mới là XOR.

Confusing OR with XOR when the question says “or”. In a logic question OR is still true when both hold. Only a question saying “one of the two, not both” means XOR.

PHẢI THUỘC / MEMORISE – AND đúng khi cả hai đúng. OR sai khi cả hai sai. XOR đúng khi hai giá trị khác nhau. NOT đảo giá trị. Ba biến có 8 dòng, thứ tự đếm 000 đến 111.

AND is true when both are true. OR is false when both are false. XOR is true when the two values differ. NOT flips the value. Three variables give 8 rows, counted 000 to 111.

Bảng chiến lược: nhìn dấu hiệu, chọn cách làm – *Strategy table: read the sign, choose the method*

Dấu hiệu trong đề <i>Sign in the question</i>	Cách làm <i>Approach</i>	Ví dụ <i>Example</i>
Cho dung lượng ở đơn vị này, hỏi đơn vị khác <i>A size in one unit, asked in another</i>	Đếm bậc trên thang B, KB, MB, GB, TB rồi nhân hoặc chia 1024 mỗi bậc <i>Count steps on the ladder B, KB, MB, GB, TB, then multiply or divide by 1024 per step</i>	1,5 GB ra MB: $1,5 \times 1024$
Có cả chữ b nhỏ và B hoa, hỏi thời gian tải <i>Both small b and capital B, asking download time</i>	Nhân dung lượng với 8 rồi chia cho tốc độ <i>Multiply the size by 8, then divide by the speed</i>	600 MB, 40 Mb/s: $4.800 : 40$

Dấu hiệu trong đề <i>Sign in the question</i>	Cách làm <i>Approach</i>	Ví dụ <i>Example</i>
Hỏi số byte của một câu tiếng Việt <i>Byte count of a Vietnamese sentence</i>	Đếm kí tự, cộng thêm byte cho chữ có dấu <i>Count characters, add bytes for accented letters</i>	Học giỏi Tin: 12 kí tự, 16 byte
“Đổi sang nhị phân”, “đủ 8 bit” <i>“Convert to binary”, “8 bits”</i>	Chia liên tiếp cho 2, đọc dư từ dưới lên, thêm 0 bên trái <i>Divide repeatedly by 2, read remainders upward, pad zeros on the left</i>	$94 = 01011110$
Số âm, nêu mã thuận, bù 1, bù 2 <i>A negative number, naming the three codes</i>	Mã số dương, rồi đổi bit dấu, hoặc đảo bit, hoặc đảo bit cộng 1 <i>The positive code, then set the sign bit, or flip bits, or flip and add 1</i>	$-45 = 11010011$ (bù 2)
Cộng hoặc nhân hai số nhị phân <i>Add or multiply two binary numbers</i>	Viết thẳng cột, kẻ dòng nhớ; nhân thì chép và dịch trái rồi cộng <i>Align columns, rule a carry row; for multiplication copy, shift left, then add</i>	$1011 \times 110 = 1000010$
Biểu thức có p, q, r và AND, OR, NOT <i>An expression with p, q, r and AND, OR, NOT</i>	Lập bảng 2^n dòng, mỗi biểu thức con một cột <i>Build a 2^n-row table, one column per sub-expression</i>	$(p \text{ OR } q) \text{ AND NOT } r$

Đề có dấu hiệu nào thì làm theo cách ở cột giữa. Mỗi ví dụ ở cột phải đã được giải trong chương, nên bạn tra lại được lời giải.

When a question shows a sign, follow the method in the middle column. Each example in the right column was worked earlier in the chapter, so you can look up its solution.

Bảng tóm tắt kiến thức – *Summary of knowledge*

Nội dung <i>Topic</i>	Quy tắc cần nhớ <i>Rule to remember</i>	Kiểm nhanh <i>Quick check</i>
Byte và đơn vị <i>Byte and units</i>	1 byte = 8 bit; mỗi bậc hơn kém 1024 lần <i>1 byte = 8 bits; each level differs by 1024</i>	1 KB = 1024 B
Bảng mã <i>Character sets</i>	ASCII 7 bit có 128 mã, mở rộng 8 bit có 256 mã; UTF-8 từ 1 đến 4 byte <i>ASCII 7-bit has 128 codes, extended 8-bit has 256; UTF-8 uses 1 to 4 bytes</i>	$A = 65 = 01000001$
Hệ nhị phân <i>Binary</i>	n bit có 2^n giá trị; 8 bit không dấu từ 0 đến 255 <i>n bits give 2^n values; 8 unsigned bits run 0 to 255</i>	$2^8 = 256$
Số âm <i>Negative numbers</i>	Bù 1 là đảo bit; bù 2 là bù 1 cộng 1 <i>1's complement flips bits; 2's complement adds 1</i>	$-45: 11010010, 11010011$

Nội dung <i>Topic</i>	Quy tắc cần nhớ <i>Rule to remember</i>	Kiểm nhanh <i>Quick check</i>
Cộng nhị phân <i>Binary addition</i>	$1 + 1 = 10$: ghi 0, nhớ 1 $1 + 1 = 10$: write 0, carry 1	$1011 + 0111 = 10010$
Lôgic <i>Logic</i>	1 là Đúng, 0 là Sai; AND, OR, XOR, NOT; n biến có 2^n dòng 1 is True, 0 is False; AND, OR, XOR, NOT; n variables give 2^n rows	$1 \text{ XOR } 1 = 0$

Cột trái liệt kê sáu nhóm kiến thức của chương. Học cột giữa trước, rồi dùng cột phải để tự thử.

The left column lists the chapter's six groups of knowledge. Learn the middle column first, then use the right column to test yourself.

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
Thông tin	Information	tất cả những gì mang lại hiểu biết cho con người <i>everything that gives people understanding</i>
Dữ liệu	Data	thông tin đã đưa vào máy tính để máy xử lý <i>information put into a computer for processing</i>
Bit	Bit	một chữ số nhị phân, 0 hoặc 1 <i>one binary digit, 0 or 1</i>
Byte	Byte	đơn vị gồm 8 bit <i>a unit of 8 bits</i>
Thiết bị số	Digital device	thiết bị làm việc với thông tin số <i>a device working with digital information</i>
Kiểu dữ liệu	Data type	nhóm dữ liệu có cùng cách biểu diễn và phép xử lý <i>a group of data sharing a representation and operations</i>
Bảng mã	Character set	danh sách kí tự và mã của chúng <i>a list of characters and their codes</i>
ASCII	ASCII	bảng mã chuẩn của Mỹ, 7 bit hoặc 8 bit <i>the American standard code, 7-bit or 8-bit</i>
Unicode	Unicode	bảng mã chung cho mọi ngôn ngữ <i>one code set for all languages</i>
UTF-8	UTF-8	cách mã hóa Unicode dài 1 đến 4 byte <i>a Unicode encoding of 1 to 4 bytes</i>
Hệ nhị phân	Binary system	hệ đếm cơ số 2, chỉ dùng 0 và 1 <i>base-2 counting using only 0 and 1</i>

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
Mã bù 2	2's complement	mã bù 1 cộng thêm 1 <i>the 1's complement plus 1</i>
Mệnh đề	Proposition	khẳng định hoặc đúng hoặc sai <i>a statement that is true or false</i>
Giá trị chân lí	Truth value	Đúng hoặc Sai, ghi là 1 hoặc 0 <i>True or False, written 1 or 0</i>
Phép toán logic	Logical operation	AND, OR, XOR, NOT <i>AND, OR, XOR, NOT</i>
Bảng chân trị	Truth table	bảng kết quả với mọi giá trị của các biến <i>a table of results for all values of the variables</i>

1.7. Đề tự luyện – Practice questions

Mức 1 – Học sinh giỏi cấp tỉnh – Level 1 – Provincial gifted-student contest

Đề 1.1 / Question 1.1 – Câu lạc bộ ảnh của lớp quay các clip, mỗi clip nặng 180 MB, lưu vào một thẻ nhớ 32 GB còn trống. (a) Thẻ chứa được tối đa bao nhiêu clip nguyên vẹn và còn trống bao nhiêu MB? (b) Tải một clip qua đường truyền 24 Mb/s mất bao nhiêu giây? (c) Tải liên tiếp 5 clip như thế mất mấy phút? Thang điểm: ý a 1,5; ý b 1,5; ý c 1,0. Tổng 4,0 điểm.
The class photography club films clips of 180 MB each and saves them to an empty 32 GB memory card. (a) How many whole clips fit at most, and how many MB are left? (b) How many seconds does it take to download one clip over a 24 Mb/s line? (c) How many minutes to download 5 such clips one after another? Marking: part a 1.5; part b 1.5; part c 1.0. Total 4.0 marks.

Đề 1.2 / Question 1.2 – Hồ sơ một thẻ thư viện lưu sáu trường: số thẻ 00417; họ tên “Mai Thị Ánh”; số sách đang mượn 3; tiền phạt 12,5 (nghìn đồng); đã gia hạn (có hoặc không); ảnh thẻ. (a) Chọn kiểu dữ liệu cho từng trường. (b) Vì sao số thẻ nên lưu là văn bản? (c) Họ tên chiếm bao nhiêu byte trong UTF-8, gồm cả hai dấu cách? (d) Số “3” trong trường số sách mượn là dữ liệu hay thông tin, và khi nào nó mang thông tin sai? Thang điểm: a 1,5; b 0,5; c 1,0; d 1,0. Tổng 4,0 điểm.
A library card record stores six fields: card number 00417; full name “Mai Thị Ánh”; books on loan 3; fine 12.5 (thousand dong); renewed (yes or no); card photo. (a) Choose a data type for each field. (b) Why should the card number be stored as text? (c) How many bytes does the full name take in UTF-8, spaces included? (d) Is the “3” in the books-on-loan field data or information, and when does it carry wrong information? Marking: a 1.5; b 0.5; c 1.0; d 1.0. Total 4.0 marks.

Mức 2 – Học sinh giỏi quốc gia – Level 2 – National gifted-student contest

Đề 1.3 / Question 1.3 – Dùng 8 bit. (a) Đổi 173 sang nhị phân. (b) Đổi 01011100_2 sang thập phân. (c) Viết -57 theo mã thuận, mã bù 1 và mã bù 2. (d) Cộng 57 với mã bù 2 của -57 bằng phép cộng nhị phân và cho biết kết quả khi ô nhớ chỉ có 8 bit. Thang điểm: a 1,0; b 0,5; c 1,5; d 1,0. Tổng 4,0 điểm.
Use 8 bits. (a) Convert 173 to binary. (b) Convert 01011100_2 to decimal. (c) Write -57 in sign-

magnitude, 1's complement and 2's complement. (d) Add 57 to the 2's complement of -57 by binary addition and state the result when the cell has only 8 bits. Marking: a 1.0; b 0.5; c 1.5; d 1.0. Total 4.0 marks.

Đề 1.4 / Question 1.4 – (a) Tính $11101_2 + 10111_2$. (b) Tính $10110_2 \times 101_2$. (c) Từ kết quả câu b, viết nhanh $10110_2 \times 1010_2$ mà không nhân lại. (d) Hai số 150 và 140 đều đủ trong 1 byte, nhưng tổng của chúng có đủ trong 1 byte không? Giải thích bằng nhị phân. Thang điểm: a 1,0; b 1,5; c 0,5; d 1,0. Tổng 4,0 điểm.

(a) Calculate $11101_2 + 10111_2$. (b) Calculate $10110_2 \times 101_2$. (c) From the result of part b, write $10110_2 \times 1010_2$ quickly without multiplying again. (d) The numbers 150 and 140 each fit in 1 byte, but does their sum fit in 1 byte? Explain in binary. Marking: a 1.0; b 1.5; c 0.5; d 1.0. Total 4.0 marks.

Mức 3 – Chuyên sâu: lập trình và dự án – Level 3 – Advanced: programming and projects

Đề 1.5 / Question 1.5 – Hệ thống báo động của một căn phòng có ba cảm biến: c (cửa đang mở), n (nhiệt độ cao), p (có người trong phòng). Chuông kêu khi (c AND NOT p) OR (n AND NOT c). (a) Lập bảng chân trị đủ 8 dòng. (b) Có bao nhiêu dòng làm chuông kêu? Nêu một tình huống đời thường ứng với một dòng. (c) Dùng bảng chân trị chứng minh NOT (c OR p) tương đương với (NOT c) AND (NOT p). (d) Khi viết biểu thức ở câu a bằng Python, vì sao cần giữ đủ các dấu ngoặc? Thang điểm: a 2,0; b 1,0; c 1,5; d 0,5. Tổng 5,0 điểm.

A room alarm system has three sensors: c (door open), n (high temperature), p (someone in the room). The bell rings when (c AND NOT p) OR (n AND NOT c). (a) Build the full 8-row truth table. (b) How many rows make the bell ring? Give an everyday situation matching one row. (c) Use a truth table to prove NOT (c OR p) is equivalent to (NOT c) AND (NOT p). (d) When writing the expression of part a in Python, why must all the brackets be kept? Marking: a 2.0; b 1.0; c 1.5; d 0.5. Total 5.0 marks.

Đề 1.6 / Question 1.6 – Một nhà trường cần mã nhị phân độ dài cố định để đánh dấu 45 loại sách. Ngoài ra mỗi phòng học có mã gồm tầng (từ 1 đến 5) và số phòng trong tầng (từ 1 đến 12). (a) Cần tối thiểu bao nhiêu bit để mã 45 loại sách? Giải thích. (b) So sánh hai phương án mã phòng: phương án A tách tầng và phòng thành hai trường; phương án B đánh số chung 60 phòng bắt đầu từ 0. Mỗi phương án dùng bao nhiêu bit và ưu nhược điểm gì? (c) Viết mã phòng 9 của tầng 4 theo cả hai phương án. (d) Viết chương trình Python in hai mã đó. Thang điểm: a 1,0; b 1,5; c 1,5; d 1,0. Tổng 5,0 điểm.

A school needs fixed-length binary codes to label 45 kinds of book. Each classroom also has a code made of the floor (1 to 5) and the room number on that floor (1 to 12). (a) What is the minimum number of bits to code 45 kinds of book? Explain. (b) Compare two schemes for room codes: scheme A splits floor and room into two fields; scheme B numbers all 60 rooms together starting from 0. How many bits does each use and what are the pros and cons? (c) Write the code of room 9 on floor 4 in both schemes. (d) Write a Python program that prints both codes. Marking: a 1.0; b 1.5; c 1.5; d 1.0. Total 5.0 marks.

1.8. Đáp án và hướng dẫn chấm bài tự luyện – Answers and marking guide

Đề 1.1 – Question 1.1

Phân tích: đề cho thẻ nhớ ở GB, clip ở MB, nên đưa về MB. Tốc độ ở Mb/s (b nhỏ), clip ở MB (B hoa), nên đổi clip sang Mb bằng cách nhân 8.

Analysis: the card is in GB and the clip in MB, so bring both to MB. The speed is in Mb/s (small b) and the clip in MB (capital B), so convert the clip to Mb by multiplying by 8.

1. Ý a (1,5 điểm). $32 \text{ GB} = 32 \times 1024 = 32.768 \text{ MB}$ (0,5 điểm). $32.768 : 180 = 182,04\dots$, nên chứa được 182 clip (0,5 điểm). $182 \times 180 = 32.760 \text{ MB}$ đã dùng, còn trống $32.768 - 32.760 = 8 \text{ MB}$ (0,5 điểm).
Part a (1.5 marks). $32 \text{ GB} = 32 \times 1024 = 32,768 \text{ MB}$ (0.5 marks). $32,768 : 180 = 182.04\dots$, so 182 clips fit (0.5 marks). $182 \times 180 = 32,760 \text{ MB}$ used, leaving $32,768 - 32,760 = 8 \text{ MB}$ (0.5 marks).
2. Ý b (1,5 điểm). $180 \text{ MB} \times 8 = 1.440 \text{ Mb}$ (0,75 điểm). $1.440 : 24 = 60$ giây (0,75 điểm). Kiểm tra: 24 Mb/s là 3 MB mỗi giây, $180 : 3 = 60$.
Part b (1.5 marks). $180 \text{ MB} \times 8 = 1,440 \text{ Mb}$ (0.75 marks). $1,440 : 24 = 60 \text{ seconds}$ (0.75 marks). Check: 24 Mb/s is 3 MB per second, and $180 : 3 = 60$.
3. Ý c (1,0 điểm). $5 \times 60 = 300$ giây = 5 phút.
Part c (1.0 mark). $5 \times 60 = 300 \text{ seconds} = 5 \text{ minutes}$.

BÀI VIẾT MẪU / MODEL WRITING – Thẻ nhớ 32 GB bằng 32.768 MB. Mỗi clip nặng 180 MB nên thẻ chứa được 182 clip, dùng hết 32.760 MB và còn trống 8 MB, không đủ cho clip thứ 183. Tốc độ 24 Mb/s đo theo bit trong khi clip đo theo byte, nên clip 180 MB tương ứng 1.440 Mb. Tải mất 60 giây, và năm clip mất 300 giây, tức 5 phút.

The 32 GB card equals 32,768 MB. Each clip is 180 MB, so the card holds 182 clips, using 32,760 MB and leaving 8 MB, not enough for a 183rd clip. The speed of 24 Mb/s is measured in bits while the clip is measured in bytes, so a 180 MB clip corresponds to 1,440 Mb. The download takes 60 seconds, and five clips take 300 seconds, that is 5 minutes. ■

Đề 1.2 – Question 1.2

Phân tích: ý a hỏi phân loại, ý b hỏi lí do dùng phép thử “cộng có nghĩa không”, ý c hỏi đếm byte nên phải đếm kí tự rồi cộng thêm cho chữ có dấu, ý d hỏi tính toàn vẹn.

Analysis: part a asks for classification, part b for the reason using the “does adding make sense” test, part c is a byte count so count characters and add for accented letters, part d is about integrity.

1. Ý a (1,5 điểm, mỗi trường 0,25 điểm): số thẻ là văn bản; họ tên là văn bản; số sách mượn là số nguyên; tiền phạt là số thực; đã gia hạn là logic; ảnh thẻ là hình ảnh.
Part a (1.5 marks, 0.25 per field): card number is text; full name is text; books on loan is an integer; fine is a real number; renewed is logical; card photo is an image.
2. Ý b (0,5 điểm): không ai cộng hai số thẻ với nhau, và chữ số 0 ở đầu phải được giữ nguyên. Số 00417 lưu như số sẽ thành 417 và đổi mất số thẻ.
Part b (0.5 marks): nobody adds two card numbers, and the leading zeros must be kept. 00417 stored as a number would become 417 and change the card number.
3. Ý c (1,0 điểm): “Mai Thị Ánh” có 11 kí tự. Chữ i chiếm 3 byte, chữ Á chiếm 2 byte, 9 kí tự còn lại mỗi kí tự 1 byte. Tổng $3 + 2 + 9 = 14$ byte.
Part c (1.0 mark): “Mai Thị Ánh” has 11 characters. i takes 3 bytes, Á takes 2 bytes, and the other 9 characters take 1 byte each. Total $3 + 2 + 9 = 14$ bytes.

```
ten = "Mai Thị Ánh"
print(len(ten))
print(len(ten.encode("utf-8")))
```

Kết quả:

Output:

11
14

4. Ý d (1,0 điểm): số 3 là dữ liệu. Thông tin là “bạn đọc đang mượn 3 cuốn”. Nếu hồ sơ không ghi số 3 đếm cho thẻ nào, hoặc không ghi sách đã trả hay chưa, thì con số không đủ dữ liệu và thông tin có thể sai hoặc không xác định được.

Part d (1.0 mark): the number 3 is data. The information is “the reader has 3 books on loan”. If the record does not say which card the 3 counts for, or whether books were returned, the number lacks data and the information may be wrong or undetermined.

BÀI VIẾT MẪU / MODEL WRITING – Số thẻ 00417 phải lưu dạng văn bản vì số thẻ chỉ dùng để nhận diện, không dùng để tính toán, và chữ số 0 đứng đầu là một phần của mã. Nếu coi là số, máy bỏ các số 0 đầu và hai thẻ 00417 và 417 bị lẫn thành một.

The card number 00417 must be stored as text because it only identifies and is never used in calculation, and the leading zeros are part of the code. Treated as a number, the machine drops the leading zeros and the cards 00417 and 417 are confused as one. ■

Đề 1.3 – Question 1.3

Phân tích: ý a và b là hai chiều đổi cơ số, mỗi chiều kiểm bằng chiều kia. Ý c bắt đầu từ mã của 57. Ý d kiểm tra tính chất của mã bù 2.

Analysis: parts a and b are the two conversion directions, each checked by the other. Part c starts from the code of 57. Part d tests the property of 2's complement.

1. Ý a (1,0 điểm). Chia liên tiếp cho 2: 173 chia 2 được 86 dư 1; 86 chia 2 được 43 dư 0; 43 chia 2 được 21 dư 1; 21 chia 2 được 10 dư 1; 10 chia 2 được 5 dư 0; 5 chia 2 được 2 dư 1; 2 chia 2 được 1 dư 0; 1 chia 2 được 0 dư 1. Đọc dư từ dưới lên: 10101101.

Kiểm tra: $128 + 32 + 8 + 4 + 1 = 173$.

Part a (1.0 mark). Repeated division by 2: 173 divided by 2 is 86 remainder 1; 86 gives 43 remainder 0; 43 gives 21 remainder 1; 21 gives 10 remainder 1; 10 gives 5 remainder 0; 5 gives 2 remainder 1; 2 gives 1 remainder 0; 1 gives 0 remainder 1. Reading the remainders from the bottom: 10101101. Check: $128 + 32 + 8 + 4 + 1 = 173$.

2. Ý b (0,5 điểm). Các chữ số 1 ở vị trí 6, 4, 3, 2 ứng với 64, 16, 8, 4. Tổng $64 + 16 + 8 + 4 = 92$.

Part b (0.5 marks). The digits 1 are at positions 6, 4, 3, 2, worth 64, 16, 8, 4. The total is $64 + 16 + 8 + 4 = 92$.

3. Ý c (1,5 điểm). $57 = 32 + 16 + 8 + 1 = 00111001$ (0,5 điểm). Mã thuận của -57 : 10111001 (0,5 điểm). Mã bù 1: 11000110 (0,25 điểm). Mã bù 2: 11000111 (0,25 điểm).

Part c (1.5 marks). $57 = 32 + 16 + 8 + 1 = 00111001$ (0.5 marks). Sign-magnitude of -57 : 10111001 (0.5 marks). 1's complement: 11000110 (0.25 marks). 2's complement: 11000111 (0.25 marks).

4. Ý d (1,0 điểm). $00111001 + 11000111 = 100000000$, có 9 chữ số. Ô nhớ chỉ có 8 bit nên giữ 8 bit bên phải: 00000000, tức $57 + (-57) = 0$.

Part d (1.0 mark). $00111001 + 11000111 = 100000000$, with 9 digits. The cell has only 8 bits, so keep the right-hand 8: 00000000 , that is $57 + (-57) = 0$.

```
n = 57
print(format(n, "08b"))
print(format(255 - n, "08b"))
print(format(256 - n, "08b"))
print(format(int("00111001", 2) + int("11000111", 2), "b"))
```

Kết quả:

Output:

```
00111001
11000110
11000111
100000000
```

BÀI VIẾT MẪU / MODEL WRITING – Số 57 viết bằng 8 bit là 00111001. Mã thuận của -57 chỉ đổi bit dấu, thành 10111001. Mã bù 1 đảo mọi bit, thành 11000110. Mã bù 2 cộng thêm 1 vào mã bù 1, thành 11000111. Cộng 57 với mã bù 2 của -57 cho 100000000. Kết quả có 9 chữ số, nhưng ô nhớ chỉ giữ 8 bit, nên phần còn lại là 00000000. Đúng là 57 cộng -57 bằng 0, và đó là lí do mã bù 2 hợp để máy làm phép cộng với số âm.

The number 57 in 8 bits is 00111001. The sign-magnitude code of -57 only changes the sign bit, giving 10111001. The 1's complement flips every bit, giving 11000110. The 2's complement adds 1 to the 1's complement, giving 11000111. Adding 57 to the 2's complement of -57 gives 100000000. The result has 9 digits, but the cell keeps only 8 bits, so what remains is 00000000. Indeed 57 plus -57 is 0, and that is why 2's complement suits a machine doing addition with negative numbers. ■

Đề 1.4 – Question 1.4

Phân tích: cả bốn ý kiểm tra được bằng thập phân. Ý c dùng nhận xét nhân với 10_2 chỉ thêm một số 0. Ý d là câu về giới hạn của một byte.

Analysis: all four parts can be checked in decimal. Part c uses the remark that multiplying by 10_2 just appends a zero. Part d is about the limit of one byte.

- Ý a (1,0 điểm). Cột 0: $1 + 1 = 10$, ghi 0 nhớ 1. Cột 1: $0 + 1 + 1 = 10$, ghi 0 nhớ 1. Cột 2: $1 + 1 + 1 = 11$, ghi 1 nhớ 1. Cột 3: $1 + 0 + 1 = 10$, ghi 0 nhớ 1. Cột 4: $1 + 1 + 1 = 11$, ghi 1 nhớ 1. Còn nhớ 1 nên ghi thêm 1. Kết quả 110100. Kiểm tra: $29 + 23 = 52 = 32 + 16 + 4$.

Part a (1.0 mark). Column 0: $1 + 1 = 10$, write 0 carry 1. Column 1: $0 + 1 + 1 = 10$, write 0 carry 1. Column 2: $1 + 1 + 1 = 11$, write 1 carry 1. Column 3: $1 + 0 + 1 = 10$, write 0 carry 1. Column 4: $1 + 1 + 1 = 11$, write 1 carry 1. A carry is left, so write an extra 1. Result 110100. Check: $29 + 23 = 52 = 32 + 16 + 4$.

- Ý b (1,5 điểm). Chữ số cuối 1: chép 10110. Chữ số giữa 0: dòng toàn số 0. Chữ số đầu 1: chép 10110 dịch hai chỗ, được 1011000. Cộng $10110 + 0 + 1011000 = 1101110$. Kiểm tra: $22 \times 5 = 110 = 64 + 32 + 8 + 4 + 2$.

Part b (1.5 marks). Last digit 1: copy 10110. Middle digit 0: a row of zeros. First digit 1: copy 10110 shifted two places to get 1011000. Add $10110 + 0 + 1011000 = 1101110$. Check: $22 \times 5 = 110 = 64 + 32 + 8 + 4 + 2$.

- Ý c (0,5 điểm). $1010_2 = 101_2 \times 10_2$, nên kết quả bằng 1101110 thêm một số 0 bên phải: 11011100. Kiểm tra: $22 \times 10 = 220$.

Part c (0.5 marks). $1010_2 = 101_2 \times 10_2$, so the result is 1101110 with a zero appended on the right: 11011100. Check: $22 \times 10 = 220$.

4. Ý d (1,0 điểm). $150 = 10010110$, $140 = 10001100$. Tổng $290 = 100100010$, có 9 chữ số. Một byte có 8 bit, ghi tối đa 255, nên tổng 290 không đủ trong 1 byte, phải dùng 2 byte.

Part d (1.0 mark). $150 = 10010110$, $140 = 10001100$. The sum $290 = 100100010$ has 9 digits. One byte has 8 bits and holds at most 255, so 290 does not fit in 1 byte; 2 bytes are needed.

```
print(format(int("11101", 2) + int("10111", 2), "b"))
print(format(int("10110", 2) * int("101", 2), "b"))
print(format(150 + 140, "b"))
```

Kết quả:

Output:

```
110100
1101110
100100010
```

BÀI VIẾT MẪU / MODEL WRITING – Hai số 150 và 140 đều nhỏ hơn 256 nên mỗi số vừa một byte, lần lượt là 10010110 và 10001100. Tổng của chúng là 290. Một byte chỉ ghi được từ 0 đến 255, trong khi 290 viết bằng nhị phân là 100100010, có chín chữ số. Vì vậy kết quả cần hai byte: hai số hạng vừa một byte vẫn có thể cho tổng không vừa.

The numbers 150 and 140 are both below 256, so each fits one byte, as 10010110 and 10001100.

Their sum is 290. One byte holds only 0 to 255, while 290 in binary is 100100010, with nine digits. So the result needs two bytes. Two addends that fit one byte can still give a sum that does not. ■

Đề 1.5 – Question 1.5

Phân tích: ba biến nên bảng có 8 dòng. Ý c là một biểu thức hai biến, 4 dòng. Ý d nhắc quy tắc Python tính and trước or.

Analysis: three variables give an 8-row table. Part c is a two-variable expression, 4 rows. Part d recalls the Python rule that and is evaluated before or.

1. Ý a (2,0 điểm): bảng sinh bằng chương trình bên dưới, mỗi dòng là c, n, p rồi kết quả (1 là chuông kêu). Mỗi dòng đúng 0,25 điểm.

Part a (2.0 marks): the table is produced by the program below; each line is c, n, p and then the result (1 means the bell rings). Each correct row earns 0.25 marks.

```
for c in (0, 1):
    for n in (0, 1):
        for p in (0, 1):
            chuong = (c and not p) or (n and not c)
            print(c, n, p, int(chuong))
```

Kết quả:

Output:

```

0 0 0 0
0 0 1 0
0 1 0 1
0 1 1 1
1 0 0 1
1 0 1 0
1 1 0 1
1 1 1 0

```

2. Ý b (1,0 điểm): bốn dòng làm chuông kêu: $(c, n, p) = (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 1, 0)$. Ví dụ dòng $(1, 0, 0)$: cửa mở, nhiệt độ bình thường, không có người, tức có người lạ vào lúc phòng trống nên chuông kêu.

Part b (1.0 mark): four rows make the bell ring: $(c, n, p) = (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 1, 0)$. For example the row $(1, 0, 0)$: door open, normal temperature, nobody inside, so an intruder enters an empty room and the bell rings.

3. Ý c (1,5 điểm): lập bảng $(c, p, c \text{ OR } p, \text{NOT } (c \text{ OR } p), \text{NOT } c, \text{NOT } p, (\text{NOT } c) \text{ AND } (\text{NOT } p))$. Dòng 0 0: 0, 1, 1, 1, 1. Dòng 0 1: 1, 0, 1, 0, 0. Dòng 1 0: 1, 0, 0, 1, 0. Dòng 1 1: 1, 0, 0, 0, 0. Hai cột $\text{NOT } (c \text{ OR } p)$ và $(\text{NOT } c) \text{ AND } (\text{NOT } p)$ lần lượt là 1, 0, 0, 0 và 1, 0, 0, 0, giống nhau ở cả 4 dòng nên hai biểu thức tương đương.

Part c (1.5 marks): build the table $(c, p, c \text{ OR } p, \text{NOT } (c \text{ OR } p), \text{NOT } c, \text{NOT } p, (\text{NOT } c) \text{ AND } (\text{NOT } p))$. Row 0 0: 0, 1, 1, 1, 1. Row 0 1: 1, 0, 1, 0, 0. Row 1 0: 1, 0, 0, 1, 0. Row 1 1: 1, 0, 0, 0, 0. The columns $\text{NOT } (c \text{ OR } p)$ and $(\text{NOT } c) \text{ AND } (\text{NOT } p)$ are 1, 0, 0, 0 and 1, 0, 0, 0, identical on all 4 rows, so the expressions are equivalent.

4. Ý d (0,5 điểm): Python tính and trước or, nên bỏ ngoặc có thể đổi thứ tự tính và đổi kết quả (xem chương trình ở mục 1.6). Giữ ngoặc thì thứ tự rõ ràng.

Part d (0.5 marks): Python evaluates and before or, so dropping brackets can change the order and the result (see the program in section 1.6). Keeping brackets makes the order clear.

BÀI VIẾT MẪU / MODEL WRITING – Chuông kêu ở bốn trong tám dòng của bảng chân trị. Trường hợp cửa mở và không có người luôn làm chuông kêu, dù nhiệt độ thế nào, còn trường hợp cửa đóng thì chuông chỉ kêu khi nhiệt độ cao. Dòng cửa mở, có người, nhiệt độ bình thường cho chuông im: người trong phòng mở cửa không phải là báo động.

The bell rings on four of the eight rows of the truth table. A door open with nobody inside always rings the bell, whatever the temperature, while with the door closed the bell rings only if the temperature is high. The row with the door open, someone inside and normal temperature leaves the bell silent: a person in the room opening the door is not an alarm. ■

Đề 1.6 – Question 1.6

Phân tích: n bit có 2^n mã, nên cần tìm n nhỏ nhất sao cho 2^n không nhỏ hơn số đối tượng. Phương án A dùng trường riêng cho từng thông tin, phương án B gộp thành một số thứ tự.

Analysis: n bits give 2^n codes, so find the smallest n with 2^n not below the number of objects. Scheme A gives each piece of information its own field; scheme B merges them into one sequence number.

1. Ý a (1,0 điểm): $2^5 = 32$ nhỏ hơn 45, còn $2^6 = 64$ không nhỏ hơn 45, nên cần tối thiểu 6 bit. Còn dư $64 - 45 = 19$ mã chưa dùng.

Part a (1.0 mark): $2^5 = 32$ is below 45, while $2^6 = 64$ is not below 45, so at least 6 bits are needed. There are $64 - 45 = 19$ unused codes left.

2. Ý b (1,5 điểm): phương án A cần 3 bit cho tầng (đến 5 tầng, vì $2^2 = 4$ chưa đủ) và 4 bit cho phòng (đến 12 phòng, vì $2^3 = 8$ chưa đủ), tổng 7 bit. Ưu điểm: đọc ra tầng và phòng ngay từ các nhóm bit. Phương án B cần 6 bit vì 60 phòng không vượt quá 64, tiết kiệm 1 bit, nhưng muốn biết tầng và phòng phải chia và lấy dư cho 12.

Part b (1.5 marks): scheme A needs 3 bits for the floor (up to 5 floors, since $2^2 = 4$ is not enough) and 4 bits for the room (up to 12 rooms, since $2^3 = 8$ is not enough), 7 bits in all. Its advantage: floor and room can be read straight from the bit groups. Scheme B needs 6 bits because 60 rooms do not exceed 64, saving 1 bit, but to find the floor and room one must divide and take the remainder by 12.

3. Ý c (1,5 điểm): phương án A: tầng 4 là 100, phòng 9 là 1001, mã 1001001 (0,75 điểm). Phương án B: số thứ tự $(4 - 1) \times 12 + (9 - 1) = 44 = 101100$ (0,75 điểm).

Part c (1.5 marks): scheme A: floor 4 is 100, room 9 is 1001, code 1001001 (0.75 marks).

Scheme B: sequence number $(4 - 1) \times 12 + (9 - 1) = 44 = 101100$ (0.75 marks).

```
tang = 4
phong = 9
a = format(tang, "03b") + format(phong, "04b")
b = format((tang - 1) * 12 + (phong - 1), "06b")
print(a)
print(b)
```

Kết quả:

Output:

```
1001001
101100
```

BÀI VIẾT MẪU / MODEL WRITING – Ba bit cho tầng ghi được tối đa 8 giá trị, bốn bit cho phòng ghi được 16 giá trị. Phương án A vì thế đủ cho 5 tầng, 12 phòng nhưng để dư nhiều mã trống. Phương án B dùng 6 bit cho 60 phòng, gọn hơn một bit. Với mã 1001001, người đọc tách ngay tầng 100 và phòng 1001. Với mã 101100 phải đổi ra số 44 rồi chia cho 12 mới biết tầng và phòng. Nếu trường thêm tầng, phương án A chỉ cần thêm bit ở trường tầng, còn phương án B phải đánh số lại toàn bộ.

Three bits for the floor hold at most 8 values, and four bits for the room hold 16. Scheme A therefore covers 5 floors and 12 rooms but leaves many spare codes. Scheme B uses 6 bits for 60 rooms, one bit shorter. With the code 1001001 a reader splits off floor 100 and room 1001 at once. With the code 101100 one must convert to 44 and divide by 12 to find the floor and room. If the school adds floors, scheme A only needs more bits in the floor field, while scheme B must renumber everything. ■

Chương 2. Dữ liệu âm thanh, hình ảnh và thiết bị số – Chapter 2.

Sound and image data, digital devices

Mục tiêu chương – Chapter objectives

- Nhận ra một thiết bị có phải thiết bị thông minh hay không và nói được vai trò của tin học với xã hội bằng ví dụ cụ thể.
Tell whether a device is a smart device and describe the role of informatics in society with concrete examples.
- Giải thích được cách số hóa âm thanh: lấy mẫu, chọn thang giá trị mẫu, ghi dãy số.
Explain how sound is digitised: sampling, choosing a value scale, recording a list of numbers.
- Tính được dung lượng tệp âm thanh chưa nén từ số mẫu mỗi giây, số bit mỗi mẫu, số kênh và thời gian.
Work out the size of an uncompressed sound file from samples per second, bits per sample, channels and time.
- Tính được dung lượng ảnh chưa nén từ kích thước ảnh và độ sâu màu, rồi đổi đúng sang KB, MB, GB.
Work out the size of an uncompressed image from its pixel size and colour depth, then convert correctly to KB, MB, GB.
- Dùng thiết bị số cá nhân để lưu, chép, chia sẻ tệp và biết ước lượng tệp đó nặng bao nhiêu.
Use a personal digital device to store, copy and share files, and estimate how large those files are.

2.1. Thiết bị thông minh và Internet vạn vật – Smart devices and the Internet of Things

Đèn hành lang bật khi bạn bước tới, tắt khi bạn đi qua. Bóng đèn đó tự làm việc theo quy tắc ai đó đã cài sẵn. Nếu nó còn báo được lên điện thoại của bảo vệ là đèn đã cháy hay chưa, thì nó đang làm một việc nữa: nói chuyện với thiết bị khác.

A corridor light switches on as you step towards it and off after you pass. It follows a rule someone installed in advance. If it can also tell the guard's phone whether the lamp has burnt out, it is doing one more thing: talking to another device.

KHÁI NIỆM 2.1 (Thiết bị thông minh) / CONCEPT 2.1 (Smart device) – Thiết bị thông minh là thiết bị điện tử tự chủ làm việc ở một mức nhất định nhờ phần mềm điều khiển cài sẵn, tự thích ứng với hoàn cảnh và kết nối được với thiết bị khác để trao đổi dữ liệu. Thường gặp nhất là điện thoại thông minh và máy tính bảng.

A smart device is an electronic device that works on its own to some extent through pre-installed control software, adapts to its surroundings and connects to other devices to exchange data. The most common are smartphones and tablets.

Một số thiết bị thông minh còn “bắt chước” cách làm việc hoặc cách nghĩ của con người ở mức độ còn hạn chế. Ví dụ: người máy hiểu và nói chuyện bằng ngôn ngữ tự nhiên, xe

tự hành đoán trước nguy cơ và chậm để giảm tốc. Những khả năng đó gọi chung là trí tuệ nhân tạo (AI, viết tắt của Artificial Intelligence).

Some smart devices also ‘imitate’ how people act or think, in a limited way. For example, a robot understands and speaks natural language, and a self-driving car predicts a possible collision and slows down. These abilities are called artificial intelligence (AI).

KHÁI NIỆM 2.2 (Internet vạn vật) / CONCEPT 2.2 (Internet of Things) – Internet vạn vật (IoT, viết tắt của Internet of Things) là việc nối các thiết bị thông minh với nhau để thu thập và xử lý thông tin tự động, tức thời, trên diện rộng. Ví dụ: giám sát giao thông, cảnh báo thiên tai, lái xe tự động. IoT là một yếu tố cơ bản của cuộc cách mạng công nghiệp lần thứ tư, trong đó thiết bị thông minh là thành phần chủ chốt.

The Internet of Things (IoT) connects smart devices so they collect and process information automatically and instantly over a wide area. Examples: traffic monitoring, disaster warning, self-driving. IoT is a basic element of the fourth industrial revolution, and smart devices are its key component.

Cuộc cách mạng công nghiệp lần thứ tư (gọi tắt là 4.0) dựa trên công nghệ số và kết hợp với các công nghệ thông minh để tạo cách sản xuất mới. Nó dùng AI, IoT, điện toán đám mây (cloud computing, tức thuê chỗ lưu trữ và sức tính qua mạng) và dữ liệu lớn (big data, tức khối dữ liệu cực lớn, liên tục đổi). Sản phẩm vẫn làm ra ở thế giới thật, nhưng việc tính toán, thiết kế, tạo mẫu diễn ra trong không gian số.

The fourth industrial revolution (called 4.0 for short) rests on digital technology combined with smart technologies to create new ways of producing. It uses AI, IoT, cloud computing (renting storage and computing power over the network) and big data (huge, constantly changing data sets). Products are still made in the physical world, but calculation, design and prototyping happen in digital space.

Cách nhận ra thiết bị thông minh – *How to recognise a smart device*

Đề thường đưa một thiết bị và hỏi nó có thông minh không. Bạn trả lời bằng ba câu hỏi, không cần nhớ danh sách thiết bị.

A question usually names a device and asks whether it is smart. You answer with three questions, with no need to memorise a list of devices.

Cần chuẩn bị gì – *What you need*

- Mô tả của đề về thiết bị: nó làm gì, nó có nối mạng không, ai điều khiển nó.

The question’s description of the device: what it does, whether it connects to a network, who controls it.

- Định nghĩa thiết bị thông minh (khái niệm 2.1) để đối chiếu.

The definition of a smart device (concept 2.1) to check against.

Các bước làm – *Steps*

1. Gạch chân hành động của thiết bị trong đề. Hỏi: nó tự làm theo quy tắc cài sẵn, hay phải có người bấm từng việc?

Underline what the device does in the question. Ask: does it act by itself on a built-in rule, or does a person press for each action?

2. Gạch chân chỗ đề nói về kết nối (wifi, bluetooth, mạng). Hỏi: nó trao đổi dữ liệu với thiết bị khác không?
Underline where the question mentions connection (wifi, bluetooth, network). Ask: does it exchange data with other devices?
3. Hỏi: nó có tiếp nhận dữ liệu, xử lý rồi đưa kết quả không? Nếu chỉ hiển thị một giá trị cố định thì chưa đủ.
Ask: does it take in data, process it and give a result? Merely showing a fixed value is not enough.
4. Cả ba câu đều “có” thì kết luận là thiết bị thông minh. Thiếu một câu thì nói rõ thiếu câu nào.
If all three answers are ‘yes’, it is a smart device. If one is missing, say which one. ■

PHÂN TÍCH ĐỀ / READING THE QUESTION – Đề hỏi “có phải thiết bị thông minh không” là hỏi kết luận kèm lí do. Ghi kết luận trước, rồi viết hai lí do lấy từ đúng mô tả của đề: tự chủ ở chỗ nào, kết nối ở chỗ nào.

‘Is it a smart device?’ asks for a conclusion plus reasons. Give the conclusion first, then two reasons taken from the question’s own description: where it acts on its own and where it connects.

VÍ DỤ 2.1 (Phân loại một thiết bị) / EXAMPLE 2.1 (Classifying a device) – Nhà kính trồng rau có bộ điều khiển. Cảm biến (bộ phận đo một đại lượng, ở đây là độ ẩm) đặt trong đất. Khi đất khô, bộ điều khiển tự mở van tưới. Mỗi giờ nó gửi số liệu độ ẩm lên điện thoại của người trồng qua wifi. Bộ điều khiển có phải thiết bị thông minh không? Cả hệ thống này có thuộc IoT không?

A greenhouse has a controller. A sensor (a part that measures a quantity, here moisture) sits in the soil. When the soil is dry the controller opens the watering valve by itself. Every hour it sends the moisture figures to the grower’s phone over wifi. Is the controller a smart device? Does the whole system belong to IoT?

Hướng dẫn làm bài. / How to answer.

1. Bước 1: “tự mở van khi đất khô” là làm theo quy tắc cài sẵn, không cần người bấm.
Câu hỏi 1: có.
Step 1: ‘opens the valve when the soil is dry’ is a built-in rule, no one presses anything. Question 1: yes.
2. Bước 2: “gửi số liệu lên điện thoại qua wifi” là trao đổi dữ liệu với thiết bị khác. Câu hỏi 2: có.
Step 2: ‘sends figures to the phone over wifi’ is exchanging data with another device. Question 2: yes.
3. Bước 3: nó nhận dữ liệu từ cảm biến, xử lý bằng cách so với ngưỡng, đưa kết quả là lệnh mở van và số liệu gửi đi. Câu hỏi 3: có.
Step 3: it takes data from the sensor, processes it by comparing with a threshold, and outputs the valve command and the data sent. Question 3: yes.
4. Bước 4: cả ba “có”, nên bộ điều khiển là thiết bị thông minh. Nhiều thiết bị như vậy nối với nhau và với điện thoại để thu thập, xử lý thông tin tự động, nên hệ thống thuộc IoT.

Step 4: all three are 'yes', so the controller is a smart device. Several such devices linked together and to the phone gather and process information automatically, so the system is IoT.

BÀI VIẾT MẪU / MODEL WRITING – Bộ điều khiển là thiết bị thông minh. Nó tự mở van tưới khi cảm biến báo đất khô, tức làm việc theo phần mềm cài sẵn mà không cần người can thiệp. Nó cũng nối wifi để gửi số liệu độ ẩm lên điện thoại, tức trao đổi dữ liệu với thiết bị khác. Vì vậy cả hệ thống cảm biến, bộ điều khiển và điện thoại là một hệ thống IoT.

The controller is a smart device. It opens the valve by itself when the sensor reports dry soil, so it works from pre-installed software without human action. It also connects over wifi to send moisture figures to a phone, so it exchanges data with another device. The whole system of sensor, controller and phone is therefore an IoT system. ■

LỖI HAY GẶP / COMMON MISTAKE – Cho rằng thiết bị nào có màn hình hoặc chạy điện đều là thiết bị thông minh. Một cái cân điện tử chỉ hiện số cân, không nối mạng, chưa trả lời “có” ở câu hỏi 2.

Assuming any device with a screen or a power supply is smart. An electronic scale that only shows a weight and has no network fails question 2.

NHẬN XÉT / REMARK – Một thiết bị thông minh làm việc một mình vẫn là thiết bị thông minh. Chỉ khi nhiều thiết bị nối với nhau để thu thập và xử lý thông tin trên diện rộng thì mới gọi là IoT.

A smart device working alone is still a smart device. Only when many devices link up to collect and process information over a wide area do we call it IoT.

2.2. Tin học với xã hội: ứng dụng và các mốc phát triển – *Informatics and society: applications and milestones*

Bạn đặt vé xem phim qua điện thoại, thầy cô nhập điểm vào phần mềm, ngân hàng báo số dư ngay khi bạn chuyển tiền. Ba việc ấy đều là tin học đang làm việc. SGK nhìn thành tựu tin học ở hai phía: ứng dụng vào đời sống và sự phát triển của chính ngành tin học.

You book a cinema ticket on your phone, a teacher enters marks into software, a bank shows your balance the moment you transfer money. All three are informatics at work. The textbook looks at its achievements from two sides: applications in life, and the growth of the field itself.

Năm lĩnh vực ứng dụng – *Five areas of application*

- **Quản lý:** trường học quản lý học tập, ngân hàng quản lý tài khoản và giao dịch, doanh nghiệp quản lý sản xuất, bán hàng, kế toán. Làm nhanh, chính xác, tiết kiệm chi phí.

Management: schools track learning, banks track accounts and transactions, firms run production, sales and accounting. Fast, accurate, cheaper.

- **Tự động hóa:** thiết bị thông minh chạy theo chương trình làm thay người nhiều việc. Robot làm được cả ở nơi nguy hiểm như nhà máy điện hạt nhân hoặc dưới nước.

Automation: smart devices run programs and take over many tasks. Robots can work even in dangerous places such as nuclear plants or underwater.

- **Bài toán khoa học kĩ thuật:** tính toán nhanh, chính xác để mô phỏng, kiểm nghiệm, dự báo thời tiết, giải mã gen, làm bản đồ số.
Science and engineering problems: fast, accurate calculation for simulation, testing, weather forecasting, gene decoding and digital maps.
- **Cách làm việc của nhiều nghề:** dạy học và mua hàng trực tuyến; in sắp chữ thay bằng chế bản trên máy tính; chụp ảnh phim thay bằng chụp ảnh số.
How many jobs are done: online teaching and shopping; typesetting replaced by computer layout; film photography replaced by digital photography.
- **Giao tiếp cộng đồng:** thư điện tử, diễn đàn, mạng xã hội giúp trao đổi nhanh và trực quan.
Community communication: email, forums and social networks make exchange quick and visual. ■

Các mốc phát triển của chính ngành tin học – *Milestones in the growth of informatics itself*

Với mỗi thành tựu, nhớ năm (nếu có) và việc nó đổi được. Đề hay hỏi ngược: cho một việc, bạn chọn thành tựu tương ứng.

For each achievement, remember its year (if given) and what it changed. Questions often run the other way: given a task, you pick the matching achievement.

Thành tựu <i>Achievement</i>	Mốc <i>Date</i>	Việc nó giải quyết <i>What it solved</i>
Hệ điều hành OS/360 (máy IBM/360) <i>The OS/360 operating system (IBM/360)</i>	1964	Đưa ra hầu hết nguyên lí của hệ điều hành hiện đại: quản lí thông tin, phần cứng, các việc đang chạy, giao diện người dùng. <i>Set out most principles of modern operating systems: managing information, hardware, running tasks and the user interface.</i>
Ngôn ngữ lập trình bậc cao FORTRAN <i>The high-level language FORTRAN</i>	1957	Người lập trình chỉ nêu cách giải (thuật toán), chương trình dịch đổi sang mã máy. <i>Programmers state the solution (the algorithm); a translator converts it to machine code.</i>
Giao thức TCP/IP <i>The TCP/IP protocol</i>	1983	Đặt bộ quy tắc (giao thức) trao đổi dữ liệu để kết nối Internet toàn cầu. <i>Set the rules (the protocol) for exchanging data so the Internet could connect globally.</i>
World Wide Web (WWW) <i>The World Wide Web</i>	1992	Cách truy cập Internet dễ dàng và nhất quán, giúp Internet phổ cập. <i>An easy, consistent way to reach the Internet, which spread its use.</i>

Thành tựu <i>Achievement</i>	Mốc <i>Date</i>	Việc nó giải quyết <i>What it solved</i>
Hệ quản trị cơ sở dữ liệu (phần mềm quản lý kho dữ liệu) <i>Database management systems (software that manages a data store)</i>	cuối thập niên 60; dạng bảng sau 1970 <i>the late sixties; tables after 1970</i>	Tổ chức, cập nhật, truy cập dữ liệu không phụ thuộc bài toán cụ thể. <i>Organise, update and access data independently of any specific task.</i>

NHẬN XÉT / REMARK – Hai mốc 1983 và 1992 hay bị đảo. Nhớ theo thứ tự cần có: quy tắc trao đổi dữ liệu (1983) đến trước, cách truy cập dễ dàng (1992) đến sau.

The dates 1983 and 1992 are often swapped. Remember the order of need: rules for exchanging data (1983) come first, the easy way to reach it (1992) comes after.

LỖI HAY GẶP / COMMON MISTAKE – Trả lời “vai trò của tin học” bằng một câu chung chung. Đề chấm ví dụ: nêu lĩnh vực, nói máy tính làm việc gì trong lĩnh vực đó, đổi được điều gì.

Answering ‘the role of informatics’ with one vague sentence. Marks go to examples: name the area, say what the computer does there, say what changed.

2.3. Số hóa âm thanh: lấy mẫu và thang giá trị mẫu – *Digitising sound: sampling and the sample-value scale*

Bạn nói vào micro điện thoại, rồi nghe lại giọng mình. Giữa hai việc đó, giọng nói đã thành một dãy số và từ dãy số được phát lại. Dãy số ấy được tạo ra theo ba bước, nêu ở khái niệm 2.3.

You speak into your phone’s microphone, then listen to your own voice. In between, your voice became a list of numbers and was played back from them. That list is made in three steps, set out in concept 2.3.

Âm thanh truyền đi bằng sóng âm. Đồ thị của sóng âm là một đường liên tục theo thời gian, gọi là tín hiệu tương tự (analog). Độ cao của đường tại mỗi lúc là biên độ. Máy tính chỉ lưu được các số, nên phải đổi đường liên tục đó thành dãy số. Việc đổi gọi là số hóa.

Sound travels as a sound wave. Its graph is a continuous line over time, called an analogue signal. The height of the line at each moment is the amplitude. A computer can only store numbers, so the continuous line must be turned into a list of numbers. That conversion is digitisation.

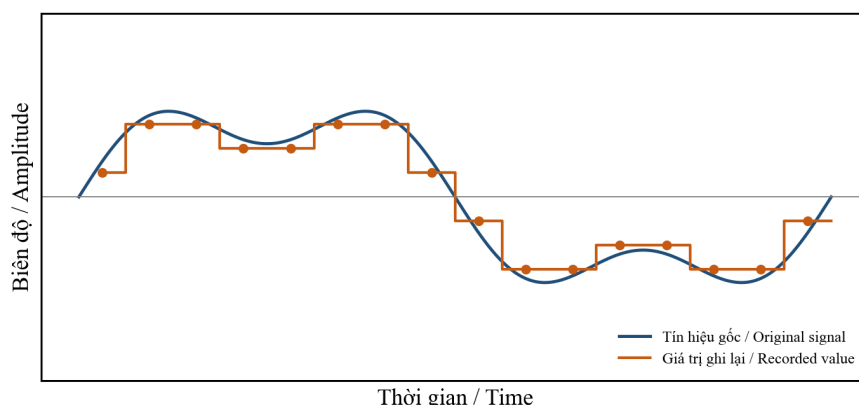
NGUYÊN TẮC 2.3 (Số hóa âm thanh bằng PCM) / PRINCIPLE 2.3 (Digitising sound with PCM) – Số hóa âm thanh bằng PCM (điều chế mã xung, Pulse Code Modulation) gồm ba bước. Bước 1, lấy mẫu: đo biên độ tín hiệu tại những thời điểm cách đều nhau; khoảng cách giữa hai lần đo là chu kỳ lấy mẫu. Bước 2, biểu diễn giá trị mẫu: chọn một thang gồm một số mức đều nhau, đổi biên độ về thang đó và làm tròn. Bước 3, ghi dãy giá trị đã đổi làm biểu diễn của âm thanh.

Digitising sound with PCM (Pulse Code Modulation) has three steps. Step 1, sampling: measure the signal’s amplitude at evenly spaced moments; the gap between two measurements is the sampling

period. Step 2, representing sample values: pick a scale of equally spaced levels, convert the amplitude to that scale and round. Step 3, record the converted values as the representation of the sound.

Thang 256 mức (2^8 mức) cho giá trị mẫu từ 0 đến 255, tức từ 00000000 đến 11111111 trong hệ nhị phân (hệ chỉ dùng hai kí hiệu 0 và 1), vừa một byte. Một cách tổng quát: thang có 2 lũy thừa k mức thì mỗi mẫu cần k bit. Thang 16 mức cần 4 bit vì $2^4 = 16$. Thang 65.536 mức cần 16 bit vì $2^{16} = 65.536$.

A scale of 256 levels (2^8 levels) gives sample values from 0 to 255, that is 00000000 to 11111111 in binary (a system with only the symbols 0 and 1), exactly one byte. In general, a scale of 2 to the power k levels needs k bits per sample. A 16-level scale needs 4 bits because $2^4 = 16$. A 65,536-level scale needs 16 bits because $2^{16} = 65,536$.



Hình 2.1. Lấy mẫu một sóng âm: mỗi chấm là một lần đo, đường bậc thang là giá trị được ghi lại sau khi làm tròn lên thang

Figure 2.1. Sampling a sound wave: each dot is one measurement, the step line is the recorded value after rounding to the scale

KHÁI NIỆM 2.4 (Tốc độ bit) / CONCEPT 2.4 (Bit-rate) – Tốc độ bit (bit-rate) là số bit cần để biểu diễn một giây âm thanh. Đơn vị thường gặp là bit trên giây (b/s). Chu kì lấy mẫu càng nhỏ và thang càng chi tiết thì tốc độ bit càng lớn, âm thanh càng trung thực, tệp càng nặng.

*The **bit-rate** is the number of bits needed to represent one second of sound. The usual unit is bits per second (b/s). The smaller the sampling period and the finer the scale, the higher the bit-rate, the more faithful the sound and the larger the file.*

Thiết bị ghi âm có mạch ADC (Analog to Digital Converter, bộ đổi tín hiệu tương tự sang số). Loa và tai nghe cần mạch DAC (Digital to Analog Converter, bộ đổi ngược từ số về tương tự) để phát lại. Tệp PCM nặng, nên có hai cách làm nhẹ: nén không làm mất chất lượng (lossless), hoặc bỏ bớt một phần thông tin nhưng chất lượng còn chấp nhận được. Định dạng Mp3 làm tệp nhỏ đi khoảng 10 lần so với wav của PCM.

A recording device has an ADC (Analog to Digital Converter). Speakers and headphones need a DAC (Digital to Analog Converter) to play the sound back. PCM files are large, so there are two ways to shrink them: compress without losing quality (lossless), or drop some information while

keeping quality acceptable. The Mp3 format makes a file about 10 times smaller than the wav format of PCM.

Cách đọc đề về lấy mẫu – How to read a sampling question

1. Tìm trong đề thang giá trị mẫu. Đề cho số mức thì viết số mức dưới dạng 2 lũy thừa k , lấy k làm số bit mỗi mẫu.
Find the sample-value scale in the question. If it gives a number of levels, write it as 2 to the power k and take k as the bits per sample.
2. Tìm chu kì lấy mẫu. Số mẫu mỗi giây bằng 1 chia cho chu kì (tính bằng giây).
Find the sampling period. Samples per second equals 1 divided by the period (in seconds).
3. Đổi từng giá trị mẫu sang nhị phân (cách đổi ở Chương 1) đúng k bit, thêm số 0 ở đầu nếu thiếu.
Write each sample value in binary (conversion is in Chapter 1) with exactly k bits, adding leading zeros if short.
4. Nhân số mẫu mỗi giây với k để được tốc độ bit.
Multiply samples per second by k to get the bit-rate. ■

PHÂN TÍCH ĐỀ / READING THE QUESTION – Đề cho thang 16 mức và chu kì 1/8.000 giây. Hai con số ấy cho $k = 4$ và 8.000 mẫu mỗi giây. Mọi câu hỏi sau đó chỉ là phép nhân hai số này với thời gian.

The question gives a 16-level scale and a period of 1/8,000 second. Those give $k = 4$ and 8,000 samples per second. Every later part is just multiplying these with time.

VÍ DỤ 2.2 (Từ thang mức đến dãy bit) / EXAMPLE 2.2 (From a scale to a bit sequence)

– Một mạch số hóa dùng thang 16 mức, đánh số từ 0 đến 15, chu kì lấy mẫu 1/8.000 giây. Dãy giá trị mẫu đầu tiên: 3, 9, 14, 15, 11, 6. a) Mỗi mẫu cần mấy bit? b) Viết dãy bit của sáu mẫu và cho biết dãy dài mấy bit. c) Tốc độ bit là bao nhiêu? d) Giải thích bằng khoảng 300 chữ vì sao chu kì lấy mẫu nhỏ hơn và thang nhiều mức hơn làm âm thanh trung thực hơn nhưng tệp nặng hơn.

A digitiser uses a 16-level scale numbered 0 to 15, with a sampling period of 1/8,000 second. The first sample values are 3, 9, 14, 15, 11, 6. a) How many bits does each sample need? b) Write the bits of the six samples and say how long the sequence is. c) What is the bit-rate? d) Explain in about 300 words why a smaller sampling period and more levels make the sound more faithful but the file larger.

Hướng dẫn làm bài. / How to answer.

1. a) $16 = 2^4$ nên mỗi mẫu cần 4 bit.
a) $16 = 2^4$, so each sample needs 4 bits.
2. b) Đổi từng mẫu sang 4 bit: 3 là 0011; 9 là 1001; 14 là 1110; 15 là 1111; 11 là 1011; 6 là 0110. Ghép lại được 0011 1001 1110 1111 1011 0110, dài $6 \times 4 = 24$ bit, tức 3 byte.
b) Write each in 4 bits: 3 is 0011; 9 is 1001; 14 is 1110; 15 is 1111; 11 is 1011; 6 is 0110. Joined: 0011 1001 1110 1111 1011 0110, which is $6 \times 4 = 24$ bits, or 3 bytes.
3. c) Số mẫu mỗi giây là $1 \div (1/8.000) = 8.000$. Tốc độ bit = $8.000 \times 4 = 32.000$ bit trên giây.

c) Samples per second is $1 \div (1/8,000) = 8,000$. Bit-rate = $8,000 \times 4 = 32,000$ bits per second.

4. d) Viết theo ba ý: sóng gốc liên tục và ta chỉ giữ một số điểm; chu kì nhỏ thì nhiều điểm hơn và đường bậc thang bám sát; thang nhiều mức thì mỗi điểm ít bị làm tròn sai; cả hai làm số bit mỗi giây tăng.

d) Write three points: the original wave is continuous and we keep only some points; a smaller period gives more points and the step line follows closely; a finer scale means each point is rounded less; both raise the bits per second.

BÀI VIẾT MẪU / MODEL WRITING – Sóng âm là một đường liên tục theo thời gian, nhưng máy tính chỉ lưu được dãy số. Vì vậy ta chỉ giữ lại biên độ của sóng tại những thời điểm cách đều nhau. Khoảng cách giữa hai thời điểm là chu kì lấy mẫu. Giữa hai lần đo, máy coi biên độ không đổi, nên đường liên tục bị thay bằng một đường bậc thang. Chu kì càng nhỏ thì các bậc càng hẹp, đường bậc thang càng bám sát sóng gốc. Ví dụ, với chu kì 1/8.000 giây, mỗi giây có 8.000 điểm đo; nếu chu kì giảm một nửa thì mỗi giây có 16.000 điểm, và những chỗ sóng đổi nhanh được ghi lại kĩ hơn. Mỗi lần đo cho một số thực, nhưng thang giá trị chỉ có một số mức đều nhau, nên số thực phải được làm tròn về mức gần nhất. Thang 16 mức làm tròn thô: hai biên độ khác nhau khá nhiều vẫn có thể rơi vào cùng một mức. Thang 65.536 mức mịn hơn nhiều, sai số làm tròn nhỏ hơn, tiếng ghi lại gần tiếng thật hơn. Hai điều chỉnh này độc lập: có thể lấy mẫu dày mà thang thô, hoặc lấy mẫu thưa mà thang mịn. Cái giá phải trả là số bit. Thang 16 mức cần 4 bit mỗi mẫu, thang 65.536 mức cần 16 bit. Tốc độ bit bằng số mẫu mỗi giây nhân số bit mỗi mẫu, nên giảm chu kì hay thêm mức đều làm tốc độ bit tăng. Một giây âm thanh với thang 16 mức và 8.000 mẫu cần 32.000 bit, còn với thang 65.536 mức và 16.000 mẫu cần 256.000 bit, gấp 8 lần. Cùng một đoạn âm thanh, tốc độ bit lớn hơn thì tệp nặng hơn. Vì thế người ta cân nhắc: muốn trung thực thì chịu tệp nặng, muốn nhẹ thì nén hoặc chấp nhận bỏ bớt thông tin.

A sound wave is a continuous line over time, but a computer can only store a list of numbers. So we keep the wave's amplitude only at evenly spaced moments. The gap between two moments is the sampling period. Between two measurements the machine treats the amplitude as unchanged, so the continuous line is replaced by a step line. The smaller the period, the narrower the steps and the closer the step line follows the original wave. For example, with a period of 1/8,000 second there are 8,000 measurements a second; halve the period and there are 16,000, and places where the wave changes fast are recorded in more detail. Each measurement gives a real number, but the scale has only a fixed number of equal levels, so the number is rounded to the nearest level. A 16-level scale rounds coarsely: two amplitudes that differ quite a lot may fall on the same level. A 65,536-level scale is much finer, the rounding error is smaller and the recording sounds closer to the real thing. The two adjustments are independent: you can sample densely on a coarse scale, or sparsely on a fine one. The price is bits. A 16-level scale needs 4 bits per sample, a 65,536-level scale needs 16 bits. The bit-rate equals samples per second times bits per sample, so shrinking the period or adding levels both raise it. One second of sound on a 16-level scale at 8,000 samples needs 32,000 bits, while on a 65,536-level scale at 16,000 samples it needs 256,000 bits, 8 times as many. For the same stretch of sound, a higher bit-rate means a larger file. So people weigh it up: for faithfulness accept a large file; for a small file compress or accept dropping some information. ■

LỖI HAY GẶP / COMMON MISTAKE – Bỏ số 0 ở đầu của mẫu nhỏ, ví dụ viết 3 là 11 thay vì 0011. Dãy bit ghép lại mất 2 bit, máy đọc sai toàn bộ các mẫu phía sau.

Writing 3 as 11 instead of 0011. Joined up the sequence loses 2 bits and the machine misreads every later sample.

PHẢI THUỘC / MEMORISE – Thang 2 lũy thừa k mức thì k bit cho mỗi mẫu. Thang 256 mức thì 8 bit, đúng một byte.

A scale of 2 to the power k levels needs k bits per sample. A 256-level scale needs 8 bits, exactly one byte.

2.4. Cách tính dung lượng tệp âm thanh chưa nén – *How to calculate the size of an uncompressed sound file*

Bạn ghi âm một buổi phỏng vấn và cần biết thẻ nhớ còn đủ không. Phép tính chỉ gồm nhân và chia, nhưng sai đơn vị là sai cả bài, nhất là khi nhầm 1.000 với 1.024.

You record an interview and need to know if the memory card has room. The calculation is only multiplying and dividing, but a wrong unit spoils the whole answer, above all when 1,000 and 1,024 get mixed up.

NGUYÊN TẮC 2.5 (Quy ước đổi đơn vị lưu trữ) / PRINCIPLE 2.5 (Storage-unit convention) – **Quy ước đổi đơn vị lưu trữ.** SGK Bài 1 ghi các đơn vị đo dữ liệu hơn kém nhau $2^{10} = 1.024$ lần: 1 KB = 1.024 B, 1 MB = 1.024 KB, 1 GB = 1.024 MB. Một byte gồm 8 bit. Sách này dùng quy ước đó cho mọi đề. Nếu một đề tự ghi “1 KB = 1.000 B” thì đề đó đổi theo đề, và bài làm phải nêu rõ đã đổi theo quy ước nào.

Storage-unit convention. Textbook Lesson 1 states the units of data differ by $2^{10} = 1,024$ times: 1 KB = 1,024 B, 1 MB = 1,024 KB, 1 GB = 1,024 MB. One byte is 8 bits. This book uses that convention in every question. If a question states ‘1 KB = 1,000 B’, convert as the question says, and your answer must state which convention you used.

NGUYÊN TẮC 2.6 (Dung lượng âm thanh chưa nén) / PRINCIPLE 2.6 (Uncompressed sound size) – **Dung lượng âm thanh chưa nén.** Mỗi giây có một số mẫu; mỗi mẫu có số bit nhất định; mỗi kênh (một đường tiếng, như loa trái hoặc loa phải) có dãy mẫu riêng. Vì vậy tốc độ bit = số mẫu mỗi giây $\times$ số bit mỗi mẫu $\times$ số kênh, và dung lượng (bit) = tốc độ bit $\times$ thời gian (giây). Chia cho 8 để ra byte, rồi chia cho 1.024 để lên KB, MB. Số mẫu mỗi giây tính cho từng kênh. Âm thanh một kênh gọi là mono, hai kênh (trái và phải) gọi là stereo. Khái niệm số kênh không có trong SGK (ngoài SGK): sách lấy từ tài liệu kỹ thuật của Microsoft về định dạng âm thanh wave. SGK Bài 6 chỉ trình bày một dãy mẫu, nên khi đề không nói số kênh, sách này hiểu là một kênh.

Uncompressed sound size. Each second has some samples; each sample has a set number of bits; each channel (one line of sound, such as the left or right speaker) has its own sample sequence. So bit-rate = samples per second $\times$ bits per sample $\times$ channels, and size (bits) = bit-rate $\times$ time (seconds). Divide by 8 for bytes, then by 1,024 to move up to KB and MB. The samples per second apply to each channel. One-channel sound is called mono, two-channel sound (left and right) is called stereo. The number of channels is not in the textbook (outside the textbook): this book takes it from Microsoft’s technical documentation of the wave audio format. Textbook Lesson 6 shows only one sample sequence, so when a question gives no number of channels, this book takes it as one channel.

Cần chuẩn bị gì – *What you need*

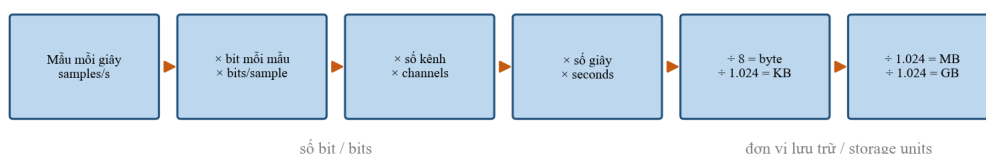
- Giấy nháp, bút, máy tính bỏ túi hoặc ứng dụng máy tính trên điện thoại.
Scrap paper, a pen, a calculator or a phone calculator app.

- Bốn số rút từ đề: số mẫu mỗi giây, số bit mỗi mẫu, số kênh, thời gian. Kẻ một dòng có đủ bốn ô, điền dần.

Four numbers from the question: samples per second, bits per sample, channels, time. Draw one row with four boxes and fill them in.

Các bước làm – Steps

1. Đổi thời gian ra giây. Ví dụ 3 phút 20 giây là $3 \times 60 + 20 = 200$ giây.
Convert the time to seconds. For example 3 minutes 20 seconds is $3 \times 60 + 20 = 200$ seconds.
2. Đổi thang mức ra số bit mỗi mẫu (xem mục 2.3). Đề cho sẵn số bit thì bỏ qua bước này.
Turn the scale into bits per sample (see section 2.3). If the question gives the bits, skip this step.
3. Viết tốc độ bit = số mẫu mỗi giây $\times$ số bit mỗi mẫu $\times$ số kênh. Tính ra bit trên giây.
Write bit-rate = samples per second $\times$ bits per sample $\times$ channels. Calculate it in bits per second.
4. Nhân tốc độ bit với số giây để được dung lượng bằng bit.
Multiply the bit-rate by the seconds to get the size in bits.
5. Chia cho 8 để ra byte. Chia tiếp cho 1.024 để ra KB, chia thêm lần nữa để ra MB. Ghi rõ đơn vị trên từng dòng.
Divide by 8 for bytes. Divide by 1,024 for KB, once more for MB. Write the unit on every line.
6. Đọc lại đề: câu hỏi cần KB hay MB, làm tròn mấy chữ số. Kiểm nhanh: kết quả MB nhỏ hơn KB một nghìn lần chừng, không phải một triệu lần.
Reread the question: does it want KB or MB, and how many decimals. Quick check: the MB figure is roughly a thousand times smaller than the KB figure, not a million. ■



Hình 2.2. Chuỗi phép tính dung lượng âm thanh chưa nén, đọc từ trái sang phải
Figure 2.2. The chain of steps for uncompressed sound size, read left to right

PHÂN TÍCH ĐỀ / READING THE QUESTION – Đề nêu ba con số về cách ghi (16.000 mẫu mỗi giây, thang 65.536 mức, một kênh) và một con số thời gian. Đề hỏi “dung lượng bằng MB” nên bài phải kết thúc bằng hai lần chia 1.024 sau khi chia 8.

The question gives three figures about the recording (16,000 samples a second, a 65,536-level scale, one channel) and one for time. It asks for ‘size in MB’, so the work must end with two divisions by 1,024 after dividing by 8.

VÍ DỤ 2.3 (Dung lượng ghi âm một kênh) / EXAMPLE 2.3 (Size of a one-channel recording)

– Một máy ghi âm giọng nói thu buổi phỏng vấn dài 3 phút 20 giây với 16.000 mẫu mỗi giây, thang 65.536 mức, một kênh, chưa nén. Tính dung lượng tệp ra byte, KB và MB (làm tròn MB đến hai chữ số thập phân). Số liệu luyện tập.

A voice recorder captures an interview of 3 minutes 20 seconds at 16,000 samples a second, a 65,536-level scale, one channel, uncompressed. Find the file size in bytes, KB and MB (MB to two decimals). Practice data.

Hướng dẫn làm bài. / How to answer.

- Bước 1:** 3 phút 20 giây = $3 \times 60 + 20 = 200$ giây.
Step 1: 3 min 20 s = $3 \times 60 + 20 = 200$ seconds.
- Bước 2:** $65.536 = 2^{16}$ nên mỗi mẫu 16 bit.
Step 2: $65,536 = 2^{16}$, so 16 bits per sample.
- Bước 3:** tốc độ bit = $16.000 \times 16 \times 1 = 256.000$ bit trên giây.
Step 3: bit-rate = $16,000 \times 16 \times 1 = 256,000$ bits per second.
- Bước 4:** dung lượng = $256.000 \times 200 = 51.200.000$ bit.
Step 4: size = $256,000 \times 200 = 51,200,000$ bits.
- Bước 5:** $51.200.000 \div 8 = 6.400.000$ byte. $6.400.000 \div 1.024 = 6.250$ KB. $6.250 \div 1.024 = 6,1035\dots$ MB, làm tròn 6,10 MB.
Step 5: $51,200,000 \div 8 = 6,400,000$ bytes. $6,400,000 \div 1,024 = 6,250$ KB. $6,250 \div 1,024 = 6,1035\dots$ MB, rounded 6.10 MB.
- Bước 6:** kiểm: 6.250 KB cỡ một nghìn lần 6,10 MB. Đúng. Đáp số: 6.400.000 byte; 6.250 KB; khoảng 6,10 MB.
Step 6: check: 6,250 KB is about a thousand times 6.10 MB. Correct. Answer: 6,400,000 bytes; 6,250 KB; about 6.10 MB. ■

VÍ DỤ 2.4 (Hai kênh và hai quy ước đơn vị) / EXAMPLE 2.4 (Two channels and two unit conventions)

– Một bộ ghi thu hai kênh (trái và phải ghi riêng) với 24.000 mẫu mỗi giây, 16 bit mỗi mẫu, trong 64 giây, chưa nén. a) Tính dung lượng ra KB theo quy ước 1 KB = 1.024 B. b) Một nhóm bạn làm theo quy ước 1 KB = 1.000 B ghi trong đề của nhóm. Họ ra bao nhiêu KB? c) Vì sao hai đáp số khác nhau mà cả hai đều đúng? Số liệu luyện tập.

A recorder captures two channels (left and right recorded separately) at 24,000 samples a second, 16 bits per sample, for 64 seconds, uncompressed. a) Find the size in KB using 1 KB = 1,024 B. b) A group follows the convention 1 KB = 1,000 B stated in their own question. How many KB do they get? c) Why do the two answers differ although both are right? Practice data.

Hướng dẫn làm bài. / How to answer.

- Tốc độ bit** = $24.000 \times 16 \times 2 = 768.000$ bit trên giây (nhân 2 vì hai kênh).
Bit-rate = $24,000 \times 16 \times 2 = 768,000$ bits per second (times 2 for two channels).
- Dung lượng** = $768.000 \times 64 = 49.152.000$ bit = 6.144.000 byte.
Size = $768,000 \times 64 = 49,152,000$ bits = 6,144,000 bytes.
- a) $6.144.000 \div 1.024 = 6.000$ KB.
a) $6,144,000 \div 1,024 = 6,000$ KB.

4. b) $6.144.000 \div 1.000 = 6.144 \text{ KB}$.

b) $6,144,000 \div 1,000 = 6,144 \text{ KB}$.

5. c) Cùng một lượng byte, nhưng “KB” trong hai đề là hai cỡ khác nhau (1.024 B và 1.000 B). Hai đáp số đều đúng với quy ước của đề mình. Bài làm phải ghi quy ước để giám khảo chấm đúng.

c) *The same number of bytes, but ‘KB’ in the two questions is two different sizes (1,024 B and 1,000 B). Both answers are right under each question’s own convention. Your answer must state the convention so the marker can follow. ■*

VÍ DỤ 2.5 (Tính ngược thời gian ghi) / EXAMPLE 2.5 (Working backwards to the recording time) – Một bộ nhớ còn trống 3 MB (theo quy ước 1.024). Ghi âm một kênh với

tốc độ bit 64.000 bit trên giây. Ghi được bao lâu? Số liệu luyện tập.

A memory has 3 MB free (convention 1,024). A one-channel recording runs at 64,000 bits per second. How long can it record? Practice data.

Hướng dẫn làm bài. / How to answer.

1. Bước 1: đổi chỗ trống ra bit: $3 \times 1.024 \times 1.024 \times 8 = 25.165.824 \text{ bit}$.

Step 1: convert the free space to bits: $3 \times 1,024 \times 1,024 \times 8 = 25,165,824 \text{ bits}$.

2. Bước 2: chia cho tốc độ bit: $25.165.824 \div 64.000 = 393,216 \text{ giây}$.

Step 2: divide by the bit-rate: $25,165,824 \div 64,000 = 393.216 \text{ seconds}$.

3. Bước 3: đổi ra phút: $393,216 \div 60 = 6,5536 \text{ phút}$, tức 6 phút 33 giây (làm tròn xuống).

Step 3: convert to minutes: $393.216 \div 60 = 6.5536 \text{ minutes}$, i.e. 6 minutes 33 seconds (rounded down). ■

LỖI HAY GẶP / COMMON MISTAKE – Quên chia 8: để nguyên số bit rồi gọi là byte. Đáp số lớn gấp 8 lần.

Forgetting to divide by 8: leaving the bits and calling them bytes. The answer is 8 times too large.

LỖI HAY GẶP / COMMON MISTAKE – Quên số kênh. Đề nói “hai kênh” hoặc “trái và phải” mà chỉ tính một dãy mẫu, đáp số chỉ bằng một nửa.

Forgetting the channels. The question says ‘two channels’ or ‘left and right’ but only one sample sequence is counted, so the answer is half.

LỖI HAY GẶP / COMMON MISTAKE – Nhầm KB 1.000 với 1.024 khi đề đã nói rõ. Đề ghi 1 KB = 1.000 B mà vẫn chia 1.024 là mất điểm đáp số.

Mixing 1,000 with 1,024 when the question has said which to use. Dividing by 1,024 when it states 1 KB = 1,000 B loses the answer mark.

NHẬN XÉT / REMARK – SGK Bài 6 ghi tốc độ bit theo Kb/s (cũng viết Kbps) mà không nói 1 K bằng bao nhiêu. Vì vậy sách này luôn ghi tốc độ bit bằng bit trên giây, không dùng Kb/s để tính.

Textbook Lesson 6 gives bit-rates in Kb/s (also written Kbps) without saying how much a K is. So this book always gives bit-rates in bits per second and never calculates with Kb/s.

PHẢI THUỘC / MEMORISE – Mẫu mỗi giây $\times$ bit mỗi mẫu $\times$ số kênh $\times$ số giây = số bit. Chia 8 ra byte, chia 1.024 mỗi bậc lên KB, MB.

Samples per second × bits per sample × channels × seconds = bits. Divide by 8 for bytes, by 1,024 for each step up to KB, MB.

2.5. Số hóa hình ảnh và cách tính dung lượng ảnh chưa nén – Digitising images and how to calculate the size of an uncompressed image

Ảnh chụp bằng điện thoại nặng vài MB. Bạn thử phóng to một ảnh lên thật nhiều: ảnh vỡ thành những ô vuông nhỏ, mỗi ô một màu. Mỗi ô ấy là một điểm ảnh, và dung lượng ảnh chính là số ô nhân với số bit lưu màu của một ô.

A phone photo weighs a few MB. Zoom in on one far enough and it breaks into small squares, each a single colour. Each square is a pixel, and the image size is the number of squares times the bits that store one square's colour.

KHÁI NIỆM 2.7 (Điểm ảnh, ảnh bitmap, độ sâu màu) / CONCEPT 2.7 (Pixel, bitmap image, colour depth) – Điểm ảnh (pixel, viết từ picture element) là phần tử nhỏ nhất của ảnh số. Ảnh lưu thông tin màu theo từng điểm ảnh gọi là **ảnh bitmap**. **Độ sâu màu** (bit depth) là số bit mã hóa màu của một điểm ảnh. Độ sâu càng lớn thì màu càng tinh tế. Ảnh màu thông dụng theo hệ RGB (đỏ, xanh lá, xanh dương) có độ sâu 24 bit: mỗi màu cơ bản 8 bit, 256 sắc độ từ 0 (tắt hẳn) đến 255 (đậm nhất). Ảnh xám thông dụng 8 bit, 256 mức xám. Ảnh đen trắng độ sâu 1 bit.

*A pixel (from picture element) is the smallest element of a digital image. An image that stores colour per pixel is a **bitmap image**. **Colour depth** (bit depth) is the number of bits that encode the colour of one pixel. The greater the depth, the finer the colours. A common colour image uses the RGB system (red, green, blue) with 24-bit depth: 8 bits per primary colour, 256 shades from 0 (off) to 255 (strongest). A common greyscale image is 8-bit with 256 grey levels. A black-and-white image has a depth of 1 bit.*

Màu trắng có mã (255, 255, 255), đỏ (255, 0, 0), xanh lá (0, 255, 0), xanh dương (0, 0, 255), đen (0, 0, 0). Ba màu cơ bản mỗi màu 256 sắc độ nên có 256^3 tổ hợp, khoảng 16,7 triệu màu khác nhau. Số mức xám hoặc số màu bằng 2 lũy thừa độ sâu: độ sâu 1, 2, 4, 8 cho 2, 4, 16, 256 mức xám.

White has code (255, 255, 255), red (255, 0, 0), green (0, 255, 0), blue (0, 0, 255), black (0, 0, 0). Each of three primaries has 256 shades, giving 256^3 combinations, about 16.7 million different colours. The number of grey levels or colours is 2 to the power of the depth: depths 1, 2, 4, 8 give 2, 4, 16, 256 grey levels.

Ảnh bitmap nguyên gốc lưu trong tệp .bmp và rất tốn bộ nhớ. Hai cách làm nhẹ: nén tệp mà khi xem giải nén không mất chất lượng, hoặc bỏ bớt thông tin và chịu mất một phần chất lượng. Định dạng .jpeg nén có mất chất lượng, tệp khá nhỏ. Định dạng .png nén tốt, không mất chất lượng, có thể có nền trong suốt.

An original bitmap image is stored in a .bmp file and uses a lot of memory. There are two ways to slim it: compress so that decompressing loses no quality, or drop some information and accept some loss. The .jpeg format compresses with some loss and gives a fairly small file. The .png format compresses well without loss and can have a transparent background.

NGUYÊN TẮC 2.8 (Dung lượng ảnh chưa nén) / PRINCIPLE 2.8 (Uncompressed image size) – Dung lượng ảnh chưa nén. Số điểm ảnh = chiều rộng × chiều cao (tính bằng điểm ảnh). Dung lượng (bit) = số điểm ảnh × độ sâu màu. Chia 8 ra byte, chia 1.024 mỗi bậc lên KB, MB, GB. Đề hỏi “ảnh chưa nén” là chỉ tính dữ liệu điểm ảnh bằng công thức này, không tính nén.

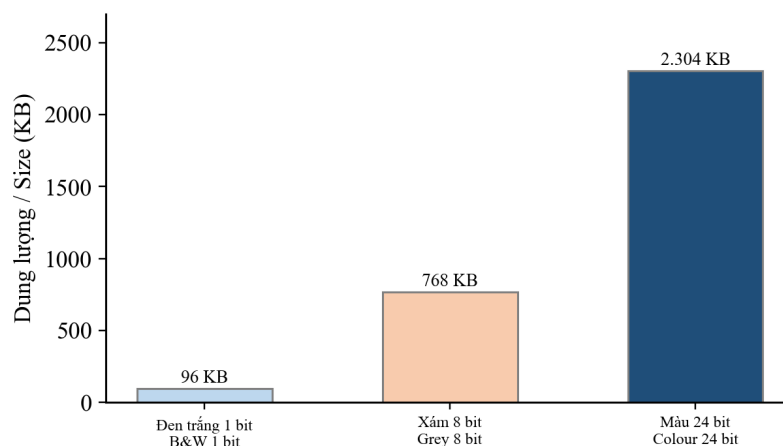
Uncompressed image size. Number of pixels = width × height (in pixels). Size (bits) = number of pixels × colour depth. Divide by 8 for bytes, by 1,024 for each step up to KB, MB, GB. ‘Uncompressed image’ means count only the pixel data with this formula, with no compression.

Cần chuẩn bị gì – *What you need*

- Ba số rút từ đề: chiều rộng và chiều cao (điểm ảnh), loại ảnh (màu, xám, đen trắng) hoặc độ sâu màu.
Three items from the question: width and height (pixels) and the type of image (colour, grey, black-and-white) or the colour depth.
- Bảng độ sâu: màu 24 bit, xám 8 bit, đen trắng 1 bit.
The depth table: colour 24 bits, grey 8 bits, black-and-white 1 bit.

Các bước làm – *Steps*

1. Viết chiều rộng × chiều cao để có số điểm ảnh. Ghi số này ra giấy.
Write width × height to get the number of pixels. Note it down.
2. Xác định độ sâu màu: đề nói “ảnh màu” là 24, “ảnh xám” là 8, “đen trắng” là 1. Đề cho số màu thì tính độ sâu bằng cách viết số màu thành 2 lũy thừa.
Find the colour depth: ‘colour image’ is 24, ‘grey image’ is 8, ‘black-and-white’ is 1. If the question gives a number of colours, write it as 2 to a power to get the depth.
3. Nhân số điểm ảnh với độ sâu để có số bit.
Multiply pixels by depth to get bits.
4. Chia 8 ra byte. Chia 1.024 ra KB. Nếu cần, chia 1.024 nữa ra MB.
Divide by 8 for bytes, by 1,024 for KB, and again by 1,024 for MB if needed.
5. Câu hỏi “vừa được bao nhiêu ảnh”: đổi dung lượng bộ nhớ ra byte, chia cho dung lượng một ảnh, bỏ phần thập phân.
For ‘how many images fit’: convert the memory to bytes, divide by the size of one image, drop the decimals. ■



Hình 2.3. Cùng một ảnh 1.024×768 điểm ảnh, chưa nén: ba độ sâu màu cho ba dung lượng khác nhau

Figure 2.3. The same $1,024 \times 768$ pixel image, uncompressed: three colour depths give three sizes

PHÂN TÍCH ĐỀ / READING THE QUESTION – Đề cho kích thước 1.600×1.200 điểm ảnh và nói “ảnh màu”. Hai dữ kiện đó cho số điểm ảnh và độ sâu 24 bit. Câu hỏi MB nên nhớ chia 8, rồi chia 1.024 hai lần.

The question gives $1,600 \times 1,200$ pixels and says ‘colour image’. These give the pixel count and a 24-bit depth. For MB remember to divide by 8, then by 1,024 twice.

VÍ DỤ 2.6 (Dung lượng ảnh màu) / EXAMPLE 2.6 (Size of a colour image) – Một ảnh màu chưa nén có kích thước 1.600×1.200 điểm ảnh. Tính dung lượng ra byte, KB và MB (MB làm tròn đến hai chữ số thập phân). Ảnh có thể có bao nhiêu màu khác nhau? Số liệu luyện tập.

An uncompressed colour image measures $1,600 \times 1,200$ pixels. Find its size in bytes, KB and MB (MB to two decimals). How many different colours can it hold? Practice data.

Hướng dẫn làm bài. / How to answer.

- Bước 1: số điểm ảnh $= 1.600 \times 1.200 = 1.920.000$.
Step 1: pixels $= 1,600 \times 1,200 = 1,920,000$.
- Bước 2: ảnh màu có độ sâu 24 bit.
Step 2: a colour image has a depth of 24 bits.
- Bước 3: số bit $= 1.920.000 \times 24 = 46.080.000$ bit.
Step 3: bits $= 1,920,000 \times 24 = 46,080,000$ bits.
- Bước 4: $46.080.000 \div 8 = 5.760.000$ byte. $\div 1.024 = 5.625$ KB. $\div 1.024 = 5,4932...$ MB, làm tròn 5,49 MB.
Step 4: $46,080,000 \div 8 = 5,760,000$ bytes. $\div 1,024 = 5,625$ KB. $\div 1,024 = 5.4932... MB$, rounded 5.49 MB.
- Bước 5: 24 bit cho $2^{24} = 16.777.216$ màu (đúng với “khoảng 16,7 triệu” SGK nêu).
Step 5: 24 bits give $2^{24} = 16,777,216$ colours (matching the textbook’s ‘about 16.7 million’).

■

VÍ DỤ 2.7 (Ba độ sâu màu và số ảnh vừa thẻ nhớ) / EXAMPLE 2.7 (Three colour depths and images per card) – Một ảnh 1.024×768 điểm ảnh, chưa nén. a) Tính dung lượng khi là ảnh xám 8 bit (KB). b) Tính khi là ảnh đen trắng (KB). c) Ảnh màu 24 bit cùng kích thước nặng bao nhiêu MB, gấp mấy lần ảnh xám? d) Một thẻ nhớ 1 GB chứa tối đa bao nhiêu ảnh màu như vậy? Số liệu luyện tập.

An uncompressed image of $1,024 \times 768$ pixels. a) Find its size as an 8-bit grey image (KB). b) As a black-and-white image (KB). c) How many MB does a 24-bit colour image of the same size take, and how many times the grey image is it? d) At most how many such colour images does a 1 GB card hold? Practice data.

Hướng dẫn làm bài. / How to answer.

- Số điểm ảnh = $1.024 \times 768 = 786.432$.
Pixels = $1,024 \times 768 = 786,432$.
- a) Xám 8 bit: $786.432 \times 8 \div 8 = 786.432$ byte = 768 KB.
a) Grey 8-bit: $786,432 \times 8 \div 8 = 786,432$ bytes = 768 KB.
- b) Đen trắng 1 bit: $786.432 \times 1 \div 8 = 98.304$ byte = 96 KB.
b) Black-and-white 1-bit: $786,432 \times 1 \div 8 = 98,304$ bytes = 96 KB.
- c) Màu 24 bit: $786.432 \times 24 \div 8 = 2.359.296$ byte = 2.304 KB = 2,25 MB. Gấp $24 \div 8 = 3$ lần ảnh xám.
c) Colour 24-bit: $786,432 \times 24 \div 8 = 2,359,296$ bytes = 2,304 KB = 2.25 MB. That is $24 \div 8 = 3$ times the grey image.
- d) 1 GB = $1.024 \times 1.024 \times 1.024 = 1.073.741.824$ byte. $1.073.741.824 \div 2.359.296 = 455,1\dots$ nên chứa được 455 ảnh. Không làm tròn lên vì 0,1 ảnh không đủ chỗ cho một ảnh nữa.
d) 1 GB = $1,024 \times 1,024 \times 1,024 = 1,073,741,824$ bytes. $1,073,741,824 \div 2,359,296 = 455.1\dots$ so it holds 455 images. Do not round up because 0.1 of an image is not room for another. ■

LỖI HAY GẶP / COMMON MISTAKE – Quên độ sâu: lấy số điểm ảnh làm dung lượng. 1.920.000 điểm ảnh là chưa phải 1.920.000 byte.

Forgetting the depth: taking the pixel count as the size. 1,920,000 pixels is not 1,920,000 bytes.

LỖI HAY GẶP / COMMON MISTAKE – Làm tròn lên khi hỏi số ảnh vừa thẻ nhớ. 455,1 ảnh thì chứa được 455, không phải 456.

Rounding up when asked how many images fit. 455.1 images means 455 fit, not 456.

NHẬN XÉT / REMARK – Ảnh .jpeg hoặc .png thật trong máy thường nhẹ hơn con số tính ra ở đây, vì hai định dạng này nén. Công thức cho dung lượng ảnh chưa nén, không cho kích thước tệp .jpeg.

Real .jpeg or .png files are usually smaller than the figure computed here, because those formats compress. The formula gives the uncompressed size, not the size of a .jpeg file.

PHẢI THUỘC / MEMORISE – Dung lượng ảnh (bit) = rộng $\times$ cao $\times$ độ sâu màu. Màu 24, xám 8, đen trắng 1. Độ sâu k cho 2 lũy thừa k màu hoặc mức xám.

Image size (bits) = width × height × colour depth. Colour 24, grey 8, black-and-white 1. Depth k gives 2 to the power k colours or grey levels.

2.6. Thiết bị số cá nhân và cách quản lý tệp trên điện thoại – *Personal digital devices and managing files on a phone*

Chiếc điện thoại của bạn là lịch, đồng hồ báo thức, máy ảnh, máy ghi âm và sổ danh bạ trong một vật. Những thiết bị gom nhiều việc như vậy gọi là trợ thủ số cá nhân.

Your phone is a calendar, an alarm clock, a camera, a recorder and an address book in one object. Devices that bundle so many jobs are called personal digital assistants.

KHÁI NIỆM 2.9 (Trợ thủ số cá nhân) / CONCEPT 2.9 (Personal digital assistant) – Trợ thủ số cá nhân (PDA, Personal Digital Assistant) là thiết bị số tích hợp nhiều chức năng và phần mềm ứng dụng hữu ích cho người dùng. Đặc điểm quan trọng là nhỏ gọn và có khả năng kết nối mạng. Ví dụ: điện thoại thông minh, máy tính bảng, đồng hồ thông minh, máy đọc sách.

A personal digital assistant (PDA) is a digital device that combines many useful functions and applications for the user. Its key features are being compact and able to connect to a network. Examples: smartphones, tablets, smartwatches, e-readers.

Ban đầu PDA chỉ có sổ lịch, đồng hồ, sổ danh bạ, danh sách việc cần làm, sổ ghi nhớ, máy tính bỏ túi. Ngày nay chúng còn nghe nhạc, ghi âm, xem phim, gọi điện, chụp ảnh, quay phim, tìm đường, điều khiển thiết bị từ xa. Chúng có cổng USB, khe thẻ nhớ, kết nối bluetooth và wifi. Phần lớn điện thoại và máy tính bảng chạy hai hệ điều hành phổ biến là iOS của Apple và Android của Google.

At first a PDA had only a calendar, clock, address book, to-do list, notes and calculator. Today they also play music, record sound, play films, make calls, take photos, shoot video, give directions and control devices remotely. They have a USB port, memory card slot, bluetooth and wifi. Most phones and tablets run one of two common operating systems: iOS from Apple and Android from Google.

Làm quen với điện thoại thông minh – *Getting to know a smartphone*

- Hai bên thân máy thường có nút khóa (bật máy hoặc tắt màn hình) và nút tăng giảm âm lượng.

The sides of the phone usually carry a lock button (switch on or turn the screen off) and volume buttons.

- Màn hình chính có thanh trạng thái (tình trạng kết nối, giờ, phần trăm pin), các biểu tượng ứng dụng (app), thanh truy cập nhanh cho ứng dụng hay dùng.

The home screen has a status bar (connection, time, battery percentage), application icons (apps) and a quick-access bar for frequently used apps.

- Máy Android thường có thanh điều hướng với nút ảo: nút Quay lại (Back) và nút Tổng quan (Overview) hiện danh sách ứng dụng đang chạy.

Android phones usually have a navigation bar with virtual buttons: Back, and Overview, which lists the apps now running.

- Ứng dụng có sẵn: gọi điện, nhắn tin, danh bạ, chụp ảnh, trình duyệt, email, lịch, báo thức. Ứng dụng khác tải từ chợ phần mềm: học trực tuyến như Zoom, MS Teams, Google Meet; lưu trữ đám mây như OneDrive, Google Drive.

Built-in apps: calls, messages, contacts, camera, browser, email, calendar, alarm. Others come from the app store: online learning such as Zoom, MS Teams, Google Meet; cloud storage such as OneDrive, Google Drive. ■

Cách quản lý tệp trên điện thoại – *How to manage files on a phone*

Bộ nhớ của máy, thẻ nhớ và dịch vụ đám mây đều giống các ổ đĩa trên máy tính. Tệp nằm trong thư mục, thư mục nằm trong thư mục khác, bắt đầu từ thư mục gốc. Mỗi hãng đặt tên ứng dụng quản lý tệp khác nhau (File Manager, Files, My Files...), nên bạn tìm theo tên gần giống.

The phone's memory, the memory card and a cloud service are all like the drives of a computer. Files sit in folders, folders sit in other folders, starting from the root folder. Each maker names its file-manager app differently (File Manager, Files, My Files...), so look for a similar name.

Cần chuẩn bị gì – *What you need*

- Điện thoại có ứng dụng quản lý tệp và ít nhất một tệp ảnh.
A phone with a file-manager app and at least one image file.
- Biết tệp ấy đang nằm ở bộ nhớ trong, thẻ nhớ hay đám mây.
Know whether the file is in internal memory, on the memory card or in the cloud.

Các bước làm – *Steps*

1. Mở ứng dụng quản lý tệp. Vào thư mục chứa ảnh, ví dụ thư mục ảnh chụp ở bộ nhớ trong.
Open the file-manager app. Go to the folder holding the photos, for example the camera folder in internal memory.
2. Chạm một lần để mở xem tệp. Đọc tên tệp, ngày và dung lượng nếu máy hiển thị.
Tap once to open and view the file. Read the name, date and size if the phone shows them.
3. Chạm và giữ lâu một chút để chọn tệp. Các nút điều khiển hiện ra: di chuyển, sao chép, chia sẻ, xóa.
Touch and hold a little to select the file. Control buttons appear: move, copy, share, delete.
4. Chọn việc cần làm. Nếu chép sang thẻ nhớ hay đám mây, mở thư mục đích rồi xác nhận. Kiểm tra tệp đã nằm đúng chỗ trước khi xóa bản cũ.
Choose what to do. To copy to the card or the cloud, open the destination folder and confirm. Check the file is in the right place before deleting the old copy. ■

NHẬN XÉT / REMARK – Dung lượng một tệp ảnh trên điện thoại thường nhỏ hơn con số tính cho ảnh chưa nén ở mục 2.5, vì ảnh chụp đã được nén bằng .jpeg. Muốn biết thẻ nhớ còn chứa được bao nhiêu ảnh, hãy lấy dung lượng thật của vài tệp trong máy làm trung bình, đừng dùng công thức chưa nén.

A photo file on a phone is usually smaller than the uncompressed figure from section 2.5, because phone photos are already compressed as .jpeg. To estimate how many photos a card still holds, average the real sizes of a few files on the phone instead of using the uncompressed formula.