



BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC KINH TẾ - KỸ THUẬT CÔNG NGHIỆP
KHOA CÔNG NGHỆ THÔNG TIN

Chủ biên PHÙNG THỊ THU HIỀN
VŨ THU UYÊN

TÀI LIỆU HỌC TẬP

MÔN TIN CƠ SỞ

Đối tượng: Sinh viên trình độ Đại học
Ngành đào tạo: Công nghệ thông tin

Năm 2019

MỤC LỤC

DANH MỤC CÁC HÌNH VẼ	vii
LỜI NÓI ĐẦU	ix
PHẦN 1: NHẬP MÔN TIN HỌC	1
CHƯƠNG 1: THÔNG TIN VÀ XỬ LÝ THÔNG TIN	1
1.1. Thông tin và xử lý thông tin	1
1.1.1. Khái niệm về thông tin (Information)	1
1.1.2. Đơn vị đo thông tin	2
1.1.3. Sơ đồ tổng quát của một quá trình xử lý thông tin	2
1.2.1. Các lĩnh vực nghiên cứu của tin học	3
1.2.2. Ứng dụng của tin học	3
1.3.1. Lịch sử phát triển của máy tính điện tử	3
1.3.2. Cấu trúc tổng quát của máy tính điện tử	4
1.3.3. Xử lý thông tin bằng máy tính điện tử	10
1.4. Thông tin trong tin học	10
1.4.1. Các hệ cơ số đếm	10
1.4.2. Chuyển đổi giữa các hệ đếm	12
CHƯƠNG 2: HỆ ĐIỀU HÀNH WINDOWS	26
2.1. Khái niệm về hệ điều hành	26
2.2. Tổ chức lưu trữ thông tin của hệ điều hành	26
2.2.1. Tập tin (File)	26
2.2.2. Thư mục (Folder/ Directory)	27
2.2.3. Ổ đĩa (Drive)	28
2.2.4. Đường dẫn (Path)	28
2.3. Giới thiệu hệ điều hành Windows	29
2.3.1. Lịch sử phát triển của Windows	29
2.3.2. Khởi động và thoát khỏi Windows 10	30
2.3.3. Màn hình nền của Windows	31
2.3.4. Hộp thoại trong Windows	33
2.4. File Explorer	33
2.4.1. Giới thiệu	33
2.4.2. Thao tác với các thư mục và tập tin	34
2.4.3. Thao tác với các Shortcuts	35

PHẦN 2: NGÔN NGỮ LẬP TRÌNH C++	40
CHƯƠNG 1: THUẬT GIẢI VÀ LƯU ĐỒ	40
1.1. Thuật giải (algorithm)	40
1.1.1. Khái niệm thuật giải	40
1.1.2. Chương trình	40
1.1.3. Các đặc trưng của thuật toán	40
1.1.4. Trình tự thực hiện các bước của thuật toán	41
1.2. Các cách diễn tả thuật toán	41
1.2.1. Diễn đạt bằng ngôn ngữ tự nhiên	41
1.2.2. Diễn đạt bằng ngôn ngữ lập trình	43
CHƯƠNG 2: TỔNG QUAN VỀ NGÔN NGỮ C++	49
2.1. Lịch sử hình thành và phát triển	49
2.2. Các tính chất đặc trưng của ngôn ngữ	49
2.3. Cấu trúc cơ bản của ngôn ngữ C++	50
2.3.1. Câu lệnh và biểu thức	50
2.3.2. Hàm (Functions)	51
2.3.3. Thư viện chuẩn C++ (C++ Standard Library)	51
2.3.4. Các bước viết chương trình	52
2.4. Môi trường làm việc của C++	55
2.4.1. Khởi động - thoát khỏi C++	55
2.4.2. Giao diện và cửa sổ soạn thảo	55
CHƯƠNG 3: CÁC YẾU TỐ CƠ BẢN CỦA NGÔN NGỮ C++	61
3.1. Các yếu tố cơ bản	61
3.1.1. Tập ký tự dùng trong ngôn ngữ C++	61
3.1.2. Từ khoá (keyword)	61
3.1.3. Định danh (identifier)	61
3.1.4. Chú thích trong chương trình	62
3.1.5. Các kiểu dữ liệu cơ bản	62
3.1.6. Khai báo biến	66
3.1.7. Khai báo hằng	67
3.2. Biểu thức và các toán tử	69
3.2.1. Biểu thức	69
3.2.2. Các toán tử (operator)	71
3.3. Nhập xuất dữ liệu cơ bản	73

3.3.1. Hàm cin	73
3.3.2. Hàm cout	74
3.3.3. Các lệnh trình bày màn hình	76
CHƯƠNG 4: CÁC LỆNH RỄ NHÁNH	79
4.1. Các cấu trúc điều khiển chương trình	79
4.1.1. Câu lệnh đơn (statement, instruction)	79
4.1.2. Câu lệnh ghép	79
4.2. Câu lệnh if	80
4.2.1. Câu lệnh if dạng 1	80
4.2.1. Câu lệnh if dạng 2 (if else)	81
4.3. Câu lệnh switch	83
CHƯƠNG 5: CÂU LỆNH LẶP	93
5.1. Câu lệnh lặp for	93
5.2. Câu lệnh lặp while	96
5.3. Câu lệnh lặp do while	100
5.4. Câu lệnh nhảy goto	103
5.5. Câu lệnh break và continue	104
CHƯƠNG 6: CON TRỞ VÀ CẤP PHÁT BỘ NHỚ ĐỘNG	109
6.1. Địa chỉ, phép toán &, toán tử tham chiếu *	109
6.1.1. Địa chỉ, phép toán &	109
6.1.2. Toán tử tham chiếu *	110
6.2. Con trỏ	112
6.2.1. Khái niệm con trỏ	112
6.2.2. Khai báo con trỏ	112
6.2.3. Tham chiếu và con trỏ trong C++	112
6.2.4. Sử dụng con trỏ	114
6.3. Các phép toán với con trỏ	115
6.3.1. Phép toán gán	115
6.3.2. Phép toán tăng giảm địa chỉ $p \pm n$	115
6.3.3. Phép toán tự tăng giảm	115
6.3.4. Hiệu của 2 con trỏ	116
6.3.5. Phép toán so sánh	116
6.4. Cấp phát bộ nhớ động	117
6.4.1. Toán tử new	117

6.4.2. Toán tử delete	118
CHƯƠNG 7: MẢNG VÀ XÂU KÝ TỰ	121
7.1. Mảng (Arrays)	121
7.1.1. Mảng một chiều	121
7.1.2. Mảng hai chiều	125
7.1.3. Mảng con trỏ	130
7.2. Xâu ký tự	135
7.2.1. Định nghĩa	135
7.2.2. Khai báo	135
7.2.3. Một số hàm xử lý xâu	136
7.2.4. Một số ví dụ	137
CHƯƠNG 8: KIỂU DỮ LIỆU CẤU TRÚC	141
8.1. Khái niệm và định nghĩa cấu trúc	141
8.2. Khai báo cấu trúc	141
8.3. Đặt tên kiểu dữ liệu bằng typedef	144
8.4. Thao tác trên biến cấu trúc	145
8.4.1. Khởi tạo giá trị ban đầu cho cấu trúc	145
8.4.2. Truy nhập đến thuộc tính của cấu trúc	147
CHƯƠNG 9: HÀM	151
9.1. Khai báo hàm	151
9.1.1. Giới thiệu	151
9.1.2. Khai báo hàm	151
9.1.3. Định nghĩa hàm	152
9.2. Tham số trong lời gọi hàm	153
9.3. Cấp phát và phạm vi hoạt động của các biến	155
9.4. Hàm đệ quy	158
9.4.1. Khái niệm	158
9.4.2. Cấu trúc chung của hàm đệ quy	160
TÀI LIỆU THAM KHẢO	162
PHỤ LỤC	1623

DANH MỤC CÁC HÌNH VẼ

Hình 1: Mô hình tổng quát quá trình xử lý thông tin	2
Hình 2: Sơ đồ các thành phần của máy vi tính.....	5
Hình 3: Màn hình thoát khỏi Windows	30
Hình 4: Màn hình nền (Desktop) của Windows.....	31
Hình 5: Giao diện của This PC.....	32
Hình 6: Lưu đồ thuật toán giải phương trình bậc nhất $ax + b = 0$	44
Hình 7: Lưu đồ thuật toán đổi chỗ.....	45
Hình 8: Lưu đồ thuật toán thuật toán tính tổng 2 số $n1$ và $n2$	45
Hình 9: Lưu đồ thuật toán tính $A = x^2 + y^2$	46
Hình 10: Sơ đồ hoạt động của lệnh if dạng 1.....	81
Hình 11: Sơ đồ hoạt động của lệnh if dạng 2.....	82
Hình 12: Sơ đồ hoạt động của lệnh switch.....	85
Hình 13: Sơ đồ hoạt động của lệnh for.....	94
Hình 14: Sơ đồ hoạt động của lệnh while	97
Hình 15: Sơ đồ hoạt động của lệnh do while.....	100

DANH MỤC CÁC BẢNG

Bảng 1: Mối quan hệ giữa các hệ đếm	12
Bảng 2: Các phép toán số học trong hệ nhị phân	15
Bảng 3: Các phép toán logic.....	16
Bảng 4: Mô tả các phép toán logic.	72

LỜI NÓI ĐẦU

Tin cơ sở là môn chuyên ngành trong chương trình đào tạo ngành công nghệ thông tin. Mục đích của môn học này là trang bị cho sinh viên những kiến thức cơ bản nhất về kỹ năng lập trình trong việc giải quyết bài toán bằng chương trình phần mềm trên máy tính.

Trên cơ sở các kiến thức tiếp thu được sinh viên có thể đi sâu tìm hiểu các ngôn ngữ lập trình bậc cao khác nhau và bước đầu thể hiện tính chuyên nghiệp trong lập trình. Để đáp ứng với yêu cầu học tập của sinh viên chuyên ngành Công nghệ thông tin, Trường Đại học Kinh tế - Kỹ thuật Công nghiệp tổ chức biên soạn tài liệu học tập “Tin cơ sở”. Đây là một học phần cơ bản của sinh viên chuyên ngành Công nghệ thông tin.

Tài liệu này được soạn theo đề cương chi tiết môn Tin cơ sở của Khoa Công nghệ thông tin. Mục tiêu của nó nhằm giúp các bạn sinh viên chuyên ngành có một tài liệu cô đọng dùng làm tài liệu học tập.

Tài liệu học tập được biên soạn theo đúng chương trình đào tạo và các quy định về cách trình bày của Nhà trường. Nội dung của tài liệu học tập bao gồm các chương, trong mỗi chương bao gồm các phần nội dung chủ yếu như sau:

- Mục tiêu của chương.
- Nội dung bài giảng lý thuyết.
- Câu hỏi thảo luận.
- Bài tập vận dụng.

Do thời gian và trình độ có hạn nên tài liệu học tập khó có thể tránh khỏi những thiếu sót nhất định. Chúng tôi luôn mong nhận được sự góp ý của bạn đọc để giáo trình được tái bản hoàn thiện hơn trong những lần sau.

Xin chân thành cảm ơn!

Biên soạn

PHẦN 1: NHẬP MÔN TIN HỌC

CHƯƠNG 1: THÔNG TIN VÀ XỬ LÝ THÔNG TIN

Mục tiêu của chương

Nắm vững:

- Các khái niệm về định nghĩa thông tin, vật mang tin và các lĩnh vực nghiên cứu của tin học.
- Cấu tạo và chức năng của máy vi tính.
- Các hệ đếm, cách chuyển đổi giữa các hệ đếm, và các phép toán số học và logic ứng dụng trong tin học.
- Vận dụng lý thuyết làm thành thạo các bài tập về chuyển đổi giữa các hệ đếm.

Nội dung của chương

Nghiên cứu một số khái niệm về thông tin, cách chuyển đổi giữa các hệ đếm, cấu trúc cơ bản của máy tính điện tử cũng như chức năng và hoạt động của thành phần trong máy tính.

1.1. Thông tin và xử lý thông tin

1.1.1. Khái niệm về thông tin (Information)

Thông tin là nguồn gốc của nhận thức, hiểu biết. Thông tin có thể phát sinh, mã hóa, truyền, tìm kiếm, xử lý, biến dạng... và được thể hiện dưới nhiều dạng khác nhau.

Thông tin được thể hiện dưới nhiều hình thức khác nhau như sóng ánh sáng, sóng điện từ, các ký hiệu viết trên giấy vv. Về nguyên tắc, bất kỳ cấu trúc vật chất nào hoặc bất kỳ dòng năng lượng nào cũng có thể mang thông tin. Chúng được gọi là những vật mang tin (giá mang tin).

Thông tin chứa đựng ý nghĩa, còn dữ liệu là các dữ kiện không có cấu trúc và không có ý nghĩa rõ ràng nếu nó không được tổ chức và xử lý. Cùng một thông tin, có thể được biểu diễn bằng những dữ liệu khác nhau. Cùng biểu diễn một đơn vị, nhưng trong chữ số thập phân ta dùng ký hiệu 1, còn trong hệ đếm La Mã lại dùng ký hiệu I. Mỗi dữ liệu lại có thể được thể hiện bằng những ký hiệu vật lý khác nhau.

Dữ liệu (data) là hình thức thể hiện của thông tin trong mục đích lưu trữ và xử lý nhất định. Khái niệm dữ liệu xuất hiện cùng với việc xử lý thông tin bằng máy tính. Vì thế trong nhiều tài liệu người ta định nghĩa dữ liệu là đối tượng xử lý của máy tính. Thông tin luôn mang một ý nghĩa xác định nhưng hình thức thể hiện của thông tin rõ ràng mang tính quy ước.

Tri thức (knowledge) có ý nghĩa khái quát hơn thông tin. Những nhận thức thu nhận được từ nhiều thông tin trong một lĩnh vực cụ thể nào đó, có tính hướng mục đích mới trở thành tri thức. Như vậy tri thức là mục đích của nhận thức trên cơ sở tiếp nhận thông tin. Quá trình xử lý thông tin chính là quá trình nhận thức để có tri thức.

1.1.2. Đơn vị đo thông tin

Trong kỹ thuật máy tính người ta dùng hai ký tự 0 và 1 để lưu trữ và xử lý thông tin. Ký tự 1 tương ứng với một bóng đèn sáng hoặc một phần tử kim loại được nhiễm từ tính, ký tự 0 tương ứng với một bóng đèn tắt hoặc một phần tử kim loại không nhiễm từ tính. Mỗi ký tự 0 và 1 tương ứng với một đơn vị nhỏ nhất của thông tin được gọi là bit. Ngoài ra còn các đơn vị khác như sau:

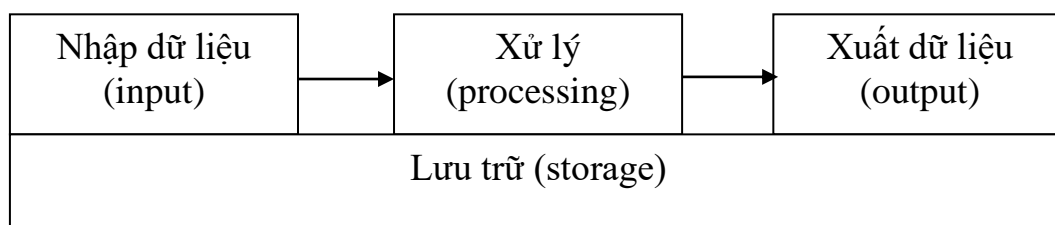
Đơn vị	Ký hiệu	Biểu diễn
Byte	B	1byte=8 bit
Kilo Byte	KB	1KB=2 ¹⁰ B
Mega Byte	MB	1MB=2 ¹⁰ KB
Giga Byte	GB	1GB=2 ¹⁰ MB
Tera Byte	TB	1TB=2 ¹⁰ GB
Peta Byte	PB	1PB=2 ¹⁰ TB

Để trao đổi thông tin trên máy vi tính, người ta dùng bộ mã chuẩn ASCII (American Standard Code for Information Interchange). Mỗi ký tự được mã hoá bằng một byte. Bảng mã ASCII có thể mã hoá 256 ký tự. Tuy nhiên số ký tự cơ bản chỉ gói gọn trong 128 số đầu. Còn 128 ký tự sau (từ 128 đến 255) được gọi là phần mã mở rộng và được dùng để mã hoá các ký tự riêng của một số ngôn ngữ (ví dụ như các ký tự có dấu của Tiếng Việt), các ký tự toán học, các ký tự đồ hoạ.

1.1.3. Sơ đồ tổng quát của một quá trình xử lý thông tin

Mọi quá trình xử lý thông tin bằng máy tính hay bằng con người đều được thực hiện theo một quy trình sau :

Dữ liệu (data) được nhập ở đầu vào (input). Máy tính hay con người sẽ thực hiện quá trình xử lý nào đó để nhận được thông tin ở đầu ra (output). Quá trình nhập dữ liệu, xử lý và xuất thông tin đều có thể được lưu trữ.



Hình 1: Mô hình tổng quát quá trình xử lý thông tin

1.2. Tin học

1.2.1. Các lĩnh vực nghiên cứu của tin học

Tin học (Informatics) được định nghĩa là ngành khoa học nghiên cứu các phương pháp, công nghệ và kỹ thuật xử lý thông tin tự động. Công cụ chủ yếu của tin học là máy tính điện tử và các thiết bị truyền tin khác. Việc nghiên cứu chính của tin học nhằm vào 2 kỹ thuật phát triển song song :

- Kỹ thuật phần cứng (hardware engineering): nghiên cứu chế tạo các thiết bị, linh kiện điện tử, công nghệ vật liệu mới... hỗ trợ cho máy tính và mạng máy tính đẩy mạnh khả năng xử lý toán học và truyền thông tin.

- Kỹ thuật phần mềm (software engineering): nghiên cứu phát triển các hệ điều hành, ngôn ngữ lập trình cho các bài toán khoa học kỹ thuật, mô phỏng, điều khiển tự động, tổ chức dữ liệu và quản lý hệ thống thông tin.

1.2.2. Ứng dụng của tin học

Tin học hiện đang được ứng dụng rộng rãi trong tất cả các ngành nghề khác nhau của xã hội từ khoa học kỹ thuật, y học, kinh tế, công nghệ sản xuất đến khoa học xã hội, nghệ thuật... như:

- Tự động hóa văn phòng
- Quản trị kinh doanh
- An ninh, quốc phòng
- Giáo dục, y học
- Nông nghiệp v.v....

1.3. Máy tính điện tử

1.3.1. Lịch sử phát triển của máy tính điện tử

Do nhu cầu cần tăng độ chính xác và giảm thời gian tính toán, con người đã quan tâm chế tạo các công cụ tính toán từ xưa: bàn tính tay của người Trung quốc, máy cộng cơ học của nhà toán học Pháp Blaise Pascal (1623 - 1662), máy tính cơ học có thể cộng trừ nhân chia của nhà toán học Đức Gottfried Wilhelm von Leibniz (1646 - 1716), máy sai phân để tính các đa thức toán học, máy phân giải điều khiển bằng phiếu đục lỗ của Charles Babbage (1792 - 1871) ...

Tuy nhiên, máy tính điện tử thực sự hình thành bắt đầu vào thập niên 1950 và đến nay đã trải qua 5 thế hệ được phân loại theo sự tiến bộ về công nghệ điện tử và vi điện tử cũng như các cải tiến về nguyên lý, tính năng và loại hình của nó.

- Thế hệ 1 (1950 - 1958): máy tính sử dụng các bóng đèn điện tử chân không, mạch riêng rẽ, vào số liệu bằng phiếu đục lỗ, điều khiển bằng tay. Máy có kích thước rất lớn, tiêu thụ năng lượng nhiều, tốc độ tính chậm khoảng 300 - 3.000 phép tính mỗi giây. Loại máy tính điển hình thế hệ 1 như EDVAC (Mỹ) hay BESM (Liên xô cũ).

- Thế hệ 2 (1958 - 1964): máy tính dùng bộ xử lý bằng đèn bán dẫn, mạch in. Máy đã có chương trình dịch như Cobol, Fortran và hệ điều hành đơn giản. Kích thước máy còn lớn, tốc độ tính khoảng 10.000 đến 100.000 phép/s. Điển hình như loại IBM-1070 (Mỹ) hay MINSK (Liên xô cũ).

- Thế hệ 3 (1965 - 1974): máy tính được gắn các bộ xử lý bằng vi mạch điện tử cỡ nhỏ có thể có được tốc độ tính khoảng 100.000 đến 1 triệu phép/s. Máy đã có các hệ điều hành đa chương trình, nhiều người dùng đồng thời hoặc theo kiểu chia thời gian. Kết quả từ máy tính có thể in ra trực tiếp ở máy in. Điển hình như loại IBM 360 (Mỹ) hay EC (Liên Xô cũ).

- Thế hệ 4 (1974 đến nay): máy tính bắt đầu có các vi mạch đa xử lý có tốc độ tính hàng chục triệu đến hàng tỷ phép/giây. Giai đoạn này hình thành 2 loại máy tính chính: máy tính cá nhân để bàn (Personal Computer - PC) hoặc xách tay (Laptop hoặc Notebook computer) và các loại máy tính chuyên nghiệp thực hiện đa chương trình, đa vi xử lý ... hình thành các hệ thống mạng máy tính (Computer Networks), và các ứng dụng phong phú đa phương tiện.

- Thế hệ 5 (1990 đến nay): bắt đầu có các nghiên cứu tạo ra các máy tính mô phỏng các hoạt động của não bộ và hành vi con người, có trí khôn nhân tạo với khả năng tự suy diễn phát triển các tình huống nhận được và những hệ quản lý kiến thức cơ sở để giải quyết các bài toán đa dạng.

Ngày nay, loài người đang bước vào cuộc cách mạng máy tính. Các máy tính, đặc biệt là máy vi tính xuất hiện khắp nơi, hoặc hỗ trợ hoặc thay thế hoàn toàn con người, thực hiện những việc do con người giao cho thông qua các chương trình do con người cài đặt cho chúng.

1.3.2. Cấu trúc tổng quát của máy tính điện tử

Một cách tổng thể thì máy tính điện tử (máy vi tính) bao gồm hai hệ thống: hệ thống các thiết bị được gọi là phần cứng và hệ thống các chương trình điều khiển, chương trình tiện ích được gọi là phần mềm.

Phần cứng của máy tính đảm nhận nhiệm vụ xử lý thông tin ở mức thấp nhất.

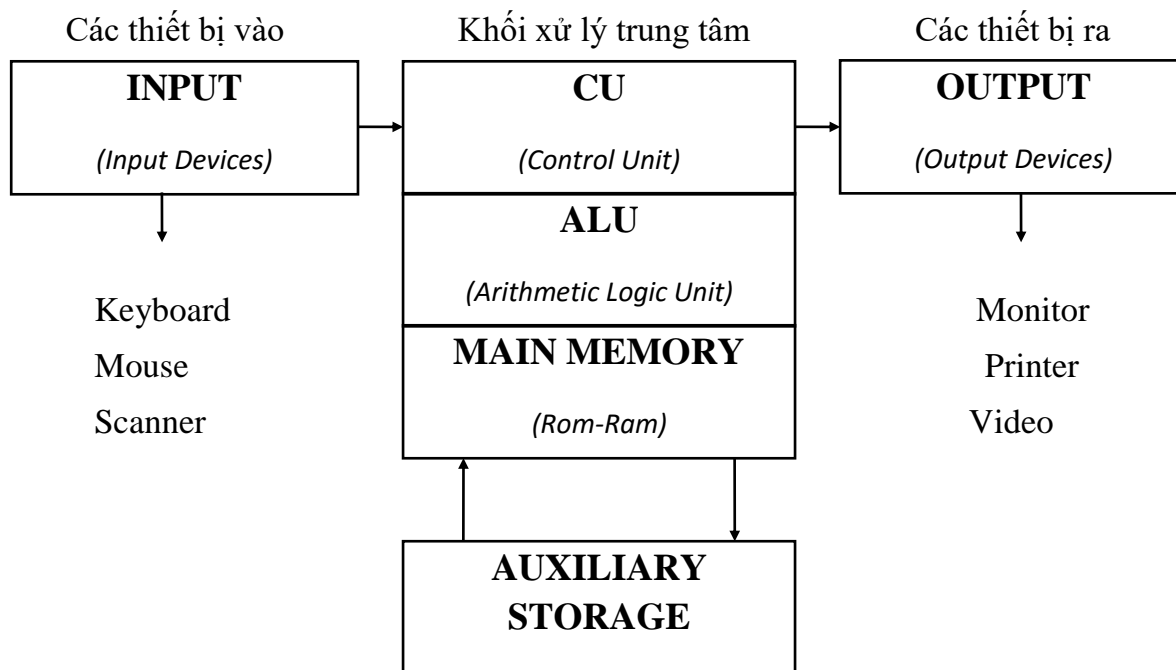
Phần mềm của máy tính đảm nhận các chức năng điều khiển, thực hiện quá trình giao tiếp giữa người và máy, góp phần làm cho việc sử dụng máy tính trở nên linh hoạt. Phần mềm là linh hồn của máy tính.

1.3.2.1. Phần cứng (hardware)

Phần cứng gồm các thiết bị máy có thể thực hiện các chức năng sau:

- * Nhập dữ liệu vào máy (input)
- * Xử lý dữ liệu (processing)
- * Xuất dữ liệu / thông tin (output)

Sơ đồ các thành phần của máy vi tính:



Hình 2: Sơ đồ các thành phần của máy vi tính.

a. Bộ nhớ

Bộ nhớ là thiết bị lưu trữ thông tin trong quá trình máy tính xử lý. Bộ nhớ bao gồm bộ nhớ trong và bộ nhớ ngoài.

◆ Bộ nhớ trong gồm ROM và RAM :

- **ROM** (Read Only Memory) là bộ nhớ chỉ đọc, có khả năng lưu trữ các thông số của nhà sản xuất, các chương trình hệ thống, chương trình điều khiển việc nhập xuất cơ sở mà người sử dụng không thể can thiệp trực tiếp vào được. Các chương trình này sẽ tự động hoạt động và kiểm tra các thiết bị mỗi lần vận hành. Ta chỉ có thể đọc thông tin trên ROM và không thể ghi hoặc xóa. Các thông tin trên ROM không bị mất đi sau khi tắt máy hoặc khi mất điện đột ngột.
- **RAM** (Random Access Memory) là bộ nhớ truy xuất ngẫu nhiên, được dùng để lưu trữ dữ kiện và chương trình trong quá trình thao tác và tính toán. Ta có thể đọc, ghi và xóa các thông tin lưu trên RAM. Muốn thông tin trên RAM không bị mất thì phải luôn có nguồn điện nuôi để lưu trữ nội dung thông tin. Do đó các thông tin trên RAM sẽ mất sau khi tắt máy hoặc khi mất điện đột ngột.

◆ Bộ nhớ ngoài: như đĩa từ, băng từ ... để lưu trữ thông tin và có thể chuyển các tin này qua máy tính khác, người ta sử dụng các đĩa, băng từ như là các bộ nhớ ngoài. Các bộ nhớ này có dung lượng chứa lớn, không bị mất đi khi không có nguồn điện. Trên các máy vi tính phổ biến hiện nay có các loại đĩa từ sau:

- **Đĩa cứng** (hard disk): có nhiều loại dung lượng từ vài trăm đến vài ngàn MB, đến nay đã có đĩa cứng 100 TB.

- **Đĩa mềm** (floppy disk) : phổ biến có 2 loại đĩa có đường kính 5.25 inches (dung lượng 360 KB hoặc 1.2 MB) và loại 3.5 inches (dung lượng 720 KB hoặc 1.44 MB)

- **Đĩa quang** (Compact disk): loại 4.72 inches có dung lượng vào khoảng 600 MB. Đĩa quang thường chỉ được đọc và không ghi được (CD-ROM) là thiết bị phổ biến với các phần mềm phong phú mang nhiều thông tin, hình ảnh, âm thanh không thể thiếu được trong các phương tiện đa truyền thông (multimedia).

Tốc độ quay của đĩa mềm 5.25 in. khoảng 300 vòng/phút, đĩa 3.5 in khoảng 600 vòng/phút. Tốc độ quay của đĩa cứng rất cao thường đạt trên 3600 vòng/phút. Vì vậy, thông tin chứa trên đĩa cứng sẽ được truy cập nhanh hơn trên đĩa mềm rất nhiều.

b. Bộ xử lý trung tâm (CPU)

Bộ xử lý trung tâm chỉ huy các hoạt động của máy tính theo lệnh và thực hiện các phép tính. CPU có 3 bộ phận chính: khối điều khiển, khối tính toán số học và logic, và một số thanh ghi.

- **Khối điều khiển** (CU: Control Unit) là trung tâm điều hành máy tính. Nó có nhiệm vụ giải mã các lệnh, tạo ra các tín hiệu điều khiển công việc của các bộ phận khác của máy tính theo yêu cầu của người sử dụng hoặc theo chương trình đã cài đặt.

- **Khối tính toán số học và logic** (ALU: Arithmetic-Logic Unit) bao gồm các thiết bị thực hiện các phép tính số học (cộng, trừ, nhân, chia, ...), các phép tính logic (AND, OR, NOT, XOR) và các phép tính quan hệ (so sánh lớn hơn, nhỏ hơn, bằng nhau, ...)

- **Các thanh ghi** (registers) được gắn chặt vào CPU bằng các mạch điện tử làm nhiệm vụ bộ nhớ trung gian. Các thanh ghi mang các chức năng chuyên dụng giúp tăng tốc độ trao đổi thông tin trong máy tính.

Ngoài ra, CPU còn được gắn với một đồng hồ (clock) hay còn gọi là bộ tạo xung nhịp. Tần số đồng hồ càng cao thì tốc độ xử lý thông tin càng nhanh. Thường thì đồng hồ được gắn tương xứng với cấu hình máy và có các tần số dao động (cho các máy PC 386 DX trở lên) là 33 MHz, 66 MHz, 100 MHz, 120 MHz, 133

c. Bộ nhập xuất

Bộ nhập xuất có tác dụng chuyển dữ liệu từ bên ngoài vào bên trong máy tính và ngược lại.

◆ Các thiết bị nhập thông tin chính:

- **Bàn phím:** Là thiết bị nhập chủ yếu trong máy tính (*keyboard*). Đây là thiết bị giúp người sử dụng đưa dữ liệu vào máy vi tính.

Có thể chia làm 3 nhóm phím chính:

✓ Nhóm phím đánh máy: gồm các phím chữ, phím số và phím các ký tự đặc biệt (~, !, @, #, \$, %, ^, &, ?, ...).

✓ Nhóm phím chức năng (function key): gồm các phím từ F1 đến F12 và các phím khác như ←↑→↓ (phím di chuyển từng điểm), phím PgUp (lên trang màn

hình), PgDn (xuống trang màn hình), Insert (chèn), Delete (xóa), Home (về đầu), End (về cuối), ...

✓ Nhóm phím đệm số (numeric keypad) như NumLock (cho các ký tự số), CapsLock (tạo các chữ in), ScrollLock (chế độ cuộn màn hình) thể hiện ở các đèn chỉ thị.

Ngoài ra ta còn các phím điều khiển quan trọng sau:

Caps Lock: Nếu đèn *Caps Lock* bật thì các phím ký tự sẽ cho chữ hoa, nếu đèn tắt thì cho chữ thường. Bật tắt đèn bằng cách ấn phím *Caps Lock*

Shift: Khi nhấn *Shift* kèm theo một phím có 2 ký tự sẽ được ký tự bên trên. Khi nhấn *Shift* cùng với một phím ký tự sẽ được chữ hoa nếu đèn *Caps Lock* tắt, chữ thường nếu đèn *Caps Lock* sáng

Num Lock: bật, tắt đèn *Num Lock*. Khi đèn *Num Lock* sáng, nhóm phím số ở bên tay phải sẽ hiển thị các số. Khi đèn *Num Lock* tắt, các phím này sẽ thực hiện chức năng thứ hai của nó.

Enter: Dùng để thực hiện một lệnh trong chế độ dòng lệnh hoặc xuống dòng trong chế độ soạn thảo

ESC: thường dùng để huỷ bỏ một công việc nào đó

BackSpace (←): xoá ký tự bên trái con trỏ

Insert: đổi chế độ viết chèn (*Insert*) thành chế độ viết đè (*Overwrite*) và ngược lại.

Ngoài bàn phím ra còn phải kể đến một số thiết bị nhập khác như:

- **Con chuột (Mouse):** là thiết bị cần thiết phổ biến hiện nay, nhất là các máy tính chạy trong môi trường Windows.

Con chuột có kích thước vừa nắm tay di chuyển trên một tấm phẳng (mouse pad) theo hướng nào thì đầu nháy hoặc mũi tên trên màn hình sẽ di chuyển theo hướng đó tương ứng với vị trí của cửa viên bi hoặc tia sáng (optical mouse) nằm dưới bụng của nó. Một số máy tính có con chuột được gắn trên bàn phím.

- **Máy quét (scanner):** là thiết bị dùng để nhập văn bản hay hình vẽ, hình chụp vào máy tính. Thông tin nguyên thủy trên giấy sẽ được quét thành các tín hiệu số tạo thành các tập tin ảnh (image file). Scanner đi kèm với phần mềm để nhận diện các tập tin ảnh hoặc văn bản.

- **Digitizer:** dùng để nhập dữ liệu đồ họa theo tọa độ X-Y vào máy tính, thường được dùng trong vẽ bản đồ.

- **Bút quang (Light pen):** dùng nhập điểm bằng cách chấm lên màn hình.

- **Touch screen:** màn hình đặc biệt có thể dùng ngón tay để chạm lên các điểm.

◆ Các thiết bị xuất thông tin chính:

- **Màn hình** (Screen hay Monitor): là thiết bị xuất chuẩn, dùng để thể hiện thông tin cho người sử dụng xem.

Mọi chữ hay ảnh trên màn hình mà ta thấy được đều tạo từ các điểm ảnh (pixel) thể hiện bởi một chấm nhỏ trên màn hình. Ngoài các tính năng giống như màn hình của máy thu hình thông thường cần phải kể đến các tính năng kỹ thuật có liên quan đến đặc thù của máy tính.

Một tính năng quan trọng của màn hình là độ phân giải (resolution) chỉ mật độ điểm ảnh trên màn hình đo khả năng thể hiện tinh tế của màn hình.

Một tính năng khác là khả năng thể hiện màu sắc. Thực ra cả hai tính năng trên không chỉ phụ thuộc vào chính màn hình mà còn phụ thuộc vào thiết bị điều khiển màn hình (video card).

Các màn hình Super VGA thông thường hiện nay cho độ phân giải tới 768 x 1024 điểm ảnh với từ 2^8 đến 2^{24} sắc độ màu khác nhau. Một tính năng khác mà hầu hết các màn hình ngày nay đều phải tính đến là khả năng tiết kiệm năng lượng. Khi ngừng làm việc với máy một thời gian đủ dài, các màn hình có thể tạm thời ngừng hoạt động để khỏi tiêu hao năng lượng vô ích.

Loại màn hình phổ biến nhất là đèn tia âm cực (đèn CRT) - chính là loại đèn hình dùng cho máy thu hình. Các điểm ảnh được tạo bởi các súng bắn điện tử trong đèn hình có phủ các vật liệu phát quang.

Ngày nay người ta còn dùng các màn hình mỏng dùng công nghệ tinh thể lỏng hay plasma. Các màn hình này thường dùng cho các máy tính xách tay (notebook) và bắt đầu dùng cho máy để bàn nhưng giá thành còn khá đắt.

- **Máy in** (Printer): Máy in là thiết bị đưa thông tin ra giấy in hoặc các thiết bị in ấn khác.

Hiện nay có các loại máy in phổ biến sau:

- ✓ Máy in kim (Dot): Máy in kim có một đầu đọc trên đó có các hàng kim đặt vuông góc với mặt giấy và châm từng chiếc kim xuống mặt giấy để ghi dữ liệu. Có 2 loại máy in chủ yếu là 9 kim và 24 kim. Máy có càng nhiều kim thì in càng nét.
- ✓ Máy in Laser (Laser Printer): Máy in Laser hoạt động theo nguyên tắc dùng tia Laser chiếu lên mặt trống bán dẫn tạo ra điện tích trên mặt trống. Trống quay hút bột mực từ ống mực rồi in ra giấy. Qua lỗ sấy mực sẽ bị nung chảy và bám vào mặt giấy. Máy in Laser có chất lượng in khá tốt, độ phân giải có thể đạt tới 1200 dpi, in được 6 đến 8 trang một phút và không gây tiếng ồn.
- ✓ Máy in phun (Jet Printer): Hoạt động tương tự như máy in kim tuy nhiên các đầu kim được thay bằng các ống mực với lỗ cực nhỏ cho phép độ phân

giải đạt tới 1440 dpi. Hiện nay máy in phun màu ngày càng phổ biến và dần thay thế các loại máy in kể trên.

- **Máy chiếu** (Projector): Có chức năng tương tự màn hình, thường được sử dụng thay cho màn hình trong các buổi hội nghị, báo cáo, thuyết trình trong hội thảo với bài trình bày để sẵn trong máy tính.

Hiện có hai loại máy chiếu phổ biến là máy chiếu dùng tinh thể lỏng (LCD) và máy chiếu DLP (Digital Light Processing) dùng vi gương.

Máy chiếu dùng màn hình tinh thể lỏng có nguyên lý làm việc giống như màn hình tinh thể lỏng. Tuy nhiên thay vì dùng đèn công suất thấp phía sau (back light) chỉ cốt để nhìn được thì máy chiếu dùng nguồn sáng cực mạnh phía sau để chiếu ảnh lên màn ảnh lớn.

Nguyên lý của DLP được Larry Hombek (làm việc tại Texas Instrument) phát minh vào năm 1987, nhưng tới cuối những năm 90 mới thực hiện được nhờ công nghệ nano. Để tạo nguồn ánh sáng, thiết bị tạo một chùm chớp sáng và dùng gương phản xạ chiếu lên màn ảnh. Gương trong DLP là linh kiện quang bán dẫn gọi là vi gương số (DMD – Digital Micro-mirror Device).

Chíp DMD chứa đến hơn 2 triệu vi gương, mỗi cái chỉ lớn 16 micro mét vuông (nhỏ bằng 1/5 sợi tóc) và xếp cách nhau 1 phần triệu mét. Vi gương được điều khiển đồng bộ với các điểm ảnh đồ họa.

Việc quay gương đi một góc nào đó sẽ giúp gương chiếu tia sáng vào màn ảnh hoặc chiếu tia sáng ra ngoài màn ảnh gây ra hiệu ứng tạo điểm ảnh hoặc tắt điểm ảnh trên màn ảnh. Gương có thể điều khiển thay đổi trạng thái tới 5000 lần/giây. Để tạo màu, người ta dùng các lọc màu theo ba màu cơ bản từ ánh sáng trắng để chiếu vào gương

1.3.2.2. Phần mềm (Software)

Phần mềm bao gồm các chương trình mà phần cứng có thể thực hiện được. Các chương trình chỉ được thực hiện sau khi đã được nạp vào bộ nhớ trong của máy.

Có thể phân loại các phần mềm dựa trên chức năng của nó:

- **Phần mềm hệ điều hành** (Operating System Software): là một bộ các câu lệnh để chỉ dẫn phần cứng máy tính và các phần mềm ứng dụng làm việc với nhau. Phần mềm hệ thống phổ biến hiện nay ở Việt nam là MS-DOS và Windows. Đối với mạng máy tính ta cũng có các phần mềm hệ điều hành mạng (Network Operating System) như Novell Netware, Unix, Windows NT, ...

- **Phần mềm ứng dụng** (Application Software): rất phong phú và đa dạng, bao gồm những chương trình được viết ra cho một hay nhiều mục đích ứng dụng cụ thể như soạn thảo văn bản, tính toán, phân tích số liệu, tổ chức hệ thống, bảo mật thông tin, vẽ đồ họa, chơi games...

Phần mềm ứng dụng có rất nhiều loại phục vụ cho các đối tượng sử dụng máy tính như :

- Phần mềm soạn thảo văn bản (Word Processing): Microsoft Word, EditPlus...
- Phần mềm quản trị dữ liệu: Visual Foxpro, Access, SQL Server...
- Phần mềm đồ họa: Corel Draw, PhotoShop, FreeHand , Illustrator...
- Phần mềm thiết kế: AutoCAD cho ngành xây dựng, cơ khí, Orcad cho ngành điện tử viễn thông.
- Phần mềm thiết kế trang Web: FrontPage, DreamWeaver...

1.3.3. Xử lý thông tin bằng máy tính điện tử

Thông tin là kết quả bao gồm nhiều quá trình xử lý các dữ liệu và thông tin có thể trở thành dữ liệu mới để theo một quá trình xử lý khác tạo ra thông tin mới hơn theo ý đồ của con người.

Con người có nhiều cách để có dữ liệu và thông tin. Người ta có thể lưu trữ thông tin qua tranh vẽ, giấy, sách báo, hình ảnh trong phim, băng từ ... Trong thời đại hiện nay, khi lượng thông tin đến với chúng ta càng lúc càng nhiều thì con người có thể dùng một công cụ hỗ trợ cho việc lưu trữ, chọn lọc và xử lý lại thông tin gọi là máy tính điện tử (computer).

Máy tính điện tử giúp con người tiết kiệm rất nhiều thời gian, công sức và tăng độ chính xác cao trong việc tự động hoá một phần hay toàn phần của quá trình xử lý dữ liệu hay thông tin.

1.4. Thông tin trong tin học

1.4.1. Các hệ cơ số đếm

Hệ đếm là tập hợp các ký hiệu và qui tắc sử dụng tập ký hiệu đó để biểu diễn và xác định các giá trị các số. Mỗi hệ đếm có một số ký số hữu hạn. Tổng số ký số của mỗi hệ đếm được gọi là cơ số (base hay radix), ký hiệu là b.

Hệ đếm mà hiện nay chúng ta thường sử dụng trong tính toán là:

Hệ đếm cơ số 2 (hệ nhị phân) là hệ chỉ dùng 2 chữ số 0 và 1 để biểu diễn các số

Hệ đếm cơ số 8 (hệ bát phân). Hệ này dùng 8 chữ số 0, 1, 2, 3, 4, 5, 6, 7 để biểu diễn các số.

Hệ đếm cơ số 10 (hệ thập phân). Hệ này dùng 10 chữ số 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 để biểu diễn các số.

Hệ đếm cơ số 16 (hệ thập lục phân) là hệ dùng 16 chữ số: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F để biểu diễn các số.

a. Hệ đếm nhị phân (Binary system)

Hệ đếm nhị phân còn gọi là đếm cơ số 2. Hệ này dùng hai ký hiệu số 0 và số 1 để biểu diễn, tính toán, đếm.

Mọi số nhị phân đều được biểu diễn dưới dạng tổng các số với lũy thừa cơ số 2.

Ví dụ: $11011.01 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$.

b. Hệ đếm bát phân (October system)

Hệ đếm bát phân còn gọi là đếm cơ số 8. Hệ này dùng các ký hiệu số (0, 1, 2, 3, 4, 5, 6, 7) để biểu diễn, tính toán, đếm.

Mọi số bát phân đều được biểu diễn dưới dạng tổng các số với lũy thừa cơ số 8.

Ví dụ: $213.4 = 2 \times 8^2 + 1 \times 8^1 + 3 \times 8^0 + 4 \times 8^{-1}$

c. Hệ đếm thập phân (Decimal system)

Hệ đếm thập phân còn gọi là hệ đếm cơ số 10. Hệ này dùng 10 ký hiệu số (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) để biểu diễn, đếm, tính toán.

Hệ đếm thập phân đều biểu diễn dưới dạng tổng các số với lũy thừa cơ số 10.

Ví dụ: $536.4 = 5 \times 10^2 + 3 \times 10^1 + 6 \times 10^0 + 4 \times 10^{-1}$.

d. Hệ đếm thập lục phân (Hexadecimal System)

Hệ đếm thập lục phân hay còn gọi là hệ đếm cơ số 16. Hệ này dùng 16 ký hiệu (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) để biểu diễn, đếm, tính toán.

Trong hệ này thì

- A Tương ứng với 10 của hệ thập phân
- B Tương ứng với 11 của hệ thập phân
- C Tương ứng với 12 của hệ thập phân
- D Tương ứng với 13 của hệ thập phân
- E Tương ứng với 14 của hệ thập phân
- F Tương ứng với 15 của hệ thập phân

Ví dụ: $12A.1E = 1 \times 16^2 + 2 \times 16^1 + A \times 16^0 + 1 \times 16^{-1} + E \times 16^{-2}$.

Hệ cơ số 10	Hệ cơ số 2	HỆ cơ số 8	Hệ cơ số 16
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

Bảng 1: Mối quan hệ giữa các hệ đếm

1.4.2. Chuyển đổi giữa các hệ đếm

A. Chuyển sang hệ 10

Ký hiệu $N(a)$ là biểu diễn của số N trong hệ cơ số a .

Trong hệ cơ số a , $N(a)$ có biểu diễn $b_n b_{n-1} \dots b_1 b_0$

Như vậy, trong hệ 10 số N sẽ có giá trị:

$$N = b_n * a^{n-1} + b_{n-1} * a^{n-2} + \dots + b_1 * a^1 + b_0 * a^0. \quad (1)$$

➤ Chuyển từ hệ đếm cơ số 10 sang các hệ khác

Muốn đổi số nguyên $N(10)$ sang cơ số a , hãy thực hiện theo các thao tác sau:

1. Thực hiện phép chia nguyên $N(10)$ cho a .
2. Nếu kết quả nhận được > 0 , tiếp tục đem kết quả chia nguyên cho a . Nếu kết quả bằng 0, chuyển sang bước 3.

3. Viết ngược thứ tự các số dư trong quá trình thực hiện các phép chia nguyên. Số nhận được chính là biểu diễn của số N trong hệ đếm cơ số a. Viết phần dư theo chiều từ dưới lên.

Ví dụ 1.1: Đổi số 156_{10} sang hệ nhị phân

$$\begin{array}{r} 156 : 2 = 78 \text{ dư } 0 \\ 78 : 2 = 39 \text{ dư } 0 \\ 39 : 2 = 19 \text{ dư } 1 \\ 19 : 2 = 9 \text{ dư } 1 \\ 9 : 2 = 4 \text{ dư } 1 \\ 4 : 2 = 2 \text{ dư } 0 \\ 2 : 2 = 1 \text{ dư } 0 \\ 1 : 2 = 0 \text{ dư } 1 \end{array}$$

Vậy $156_{10} = 10011100_2$

➤ **Muốn đổi phần thập phân N(10) sang hệ đếm cơ số a**, hãy thực hiện theo các thao tác sau:

1. Nhân số N(10) với a.
2. Kết quả nhận được đem tách làm hai phần: phần nguyên và phần lẻ. Nếu phần lẻ > 0 , tiếp tục đem phần lẻ nhân với a. Nếu phần lẻ bằng 0, chuyển sang bước 3.
3. Viết liên tục các giá trị phần nguyên nhận được trong quá trình thực hiện phép nhân trên ta sẽ có biểu diễn của N(10) trong hệ đếm cơ số a.

Ví dụ 1.2: đổi số $0,6875$ sang cơ số 2

Phần lẻ		Kết quả sau khi nhân	Phần nguyên	Hệ số
0,6875	x 2	1,375	1	$b_{-1}=1$
0,375	x 2	0,75	0	$b_{-2}=0$
0,75	x 2	1,5	1	$b_{-3}=1$
0,5	x 2	1,0	1	$b_{-4}=1$

Vậy $0,6875_{10} = 0,1011_2$

Chú ý: Nếu muốn dừng lại ở m chữ số thập phân, hãy thực hiện theo các thao tác sau:

Tính giá trị b_{-m+1}

Nếu b_{-m+1} lớn hơn 0, hãy cộng thêm 1 vào b_{-m} .

Viết liên tục các giá trị $b_{-1}b_{-2}...b_{-m}$ ta sẽ có biểu diễn của số N với m chữ số thập phân trong cơ số a.

Ví dụ 1.3: đổi số 0.63 sang cơ số 2 với 4 chữ số thập phân

Phần lẻ		Kết quả sau khi nhân	Phần nguyên	Hệ số
0,63	x 2	1,26	1	$b_{-1}=1$
0,26	x 2	0,52	0	$b_{-2}=0$
0,52	x 2	1,04	1	$b_{-3}=1$
0,04	x 2	0,08	0	$b_{-4}=0$
0,08	x 2	0,16	0	$b_{-5}=0$

Vì $b_{-5}=0$ nên giá trị b_{-4} vẫn giữ nguyên = 0.

Vậy biểu diễn của số 0.63 trong cơ số 2 với 4 chữ số thập phân là 0.10100.

B. Chuyển đổi từ hệ đếm cơ số 8, hệ đếm cơ số 16 sang hệ đếm cơ số 2

Để chuyển một số từ hệ 8 (16) sang hệ nhị phân: biểu thị từng trị số trong hệ 8 (16) thành từng nhóm 3 (4) số và ghép các nhóm đó lại.

Ví dụ 1.4:

$$5(\text{hệ } 8) = 4 + 1 + 0 = 1x2^2 + 0x2^1 + 1x2^0 = 101 (\text{hệ } 2)$$

$$6(\text{hệ } 8) = 4 + 2 + 2 = 1x2^2 + 1x2^1 + 0x2^0 = 110 (\text{hệ } 2)$$

$$\text{Vậy } 56_8 = 101\ 110_2$$

Ví dụ 1.5: Số 427_8 viết dưới dạng nhị phân sẽ là:

$$427_8 = 100\ 010\ 111_2$$

C. Chuyển đổi từ hệ đếm cơ số 2 sang hệ đếm cơ số 8 và hệ đếm cơ số 16

Để chuyển một số từ hệ nhị phân sang hệ 8 (16), ta chỉ cần chuyển từng nhóm 3 (4) chữ số của số này sang cơ số tương ứng.

Ví dụ 1.6: $1110111_2 = 167_8$

D. Chuyển đổi từ hệ đếm 2 sang hệ đếm 10

Ta áp dụng công thức (1), có thể đổi số nhị phân thành số thập phân.

$$\text{Ví dụ 1.7: } 1100_2 = 1x2^3 + 1x2^2 + 0x2^1 + 0x2^0 = 8 + 4 + 0 + 0 = 12_{10}$$

+ Đổi các số nhị phân lẻ:

Với những số nhị phân lẻ cách đổi cũng hoàn toàn tương tự:

Ví dụ 1.8: Đổi số 1011.01 ta viết

$$\begin{aligned} 1011.01_2 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 0 \times 2^{-2} \\ &= 8 + 0 + 2 + 1 + 0 + 0.25 = 11.25_{10} \end{aligned}$$

E. Chuyển đổi từ hệ đếm 8 sang hệ đếm 16

Ta có thể dùng hệ 10 hoặc hệ 2 làm trung gian để chuyển đổi hệ 8 sang hệ 16 và ngược lại. Thông thường dùng hệ 2 để trung chuyển có thuận lợi hơn.

Các bước thực hiện:

- Đổi từng số của hệ 8 sang hệ 2.
- Ghép các số nhị phân theo thứ tự.
- Chia số nhị phân vừa tạo ra thành từng nhóm 4 bit tính từ phải qua trái.
- Dùng bảng chuyển đổi đổi từng nhóm 4bit thành số của hệ 16, sau đó ghép kết quả lại theo thứ tự.

Ví dụ 1.9: đổi 35_8 sang hệ 16

Chuyển sang hệ 2, hoặc tra bảng ta có $3_8 = 011_2$; $5_8 = 101_2$;

Do đó $35_8 = 011101_2 = 0001 1101_2$

Tra tiếp bảng ta có $0001_2 = 1_{16}$; $1101_2 = D_{16}$

Vậy $35_8 = 1D_{16}$

F. Chuyển đổi từ hệ đếm 16 sang hệ đếm 8:

Quá trình đổi số hệ 16 sang hệ 8 là quá trình ngược lại của phần trên.

Ví dụ 1.10: đổi số $A2B_{16}$ sang hệ 8

Tra bảng ta có:

$A_{16} = 1010$; $2_{16} = 0010$, $B_{16} = 1011$

Do đó $A2B_{16} = 1010 0010 1011$

Chia số 1010 0010 1011 thành từng nhóm 3 bit ta có 101 000 101 011

Tra bảng $101_2 = 5_8$; $000_2 = 0_8$; $011_2 = 3_8$

Vậy $A2B_{16} = 5053_8$

Thử lại: $A2B_{16} = A \times 16^2 + 2 \times 16^1 + B \times 16^0 = 2560 + 32 + 11 = 2603_{10}$

$5053_8 = 5 \times 8^3 + 0 \times 8^2 + 5 \times 8^1 + 3 \times 8^0 = 2560 + 40 + 3 = 2603_{10}$

Chú ý: ngoài cách nêu trên còn có thể dùng phương pháp chuyển các số đã cho sang hệ 10 từ hệ 16 chuyển tiếp sang hệ đếm mới.

1.4.3. Các phép toán trong hệ nhị phân

1.4.3.1. Phép toán số học

CỘNG	TRỪ	NHÂN	CHIA
$0 + 0 = 0$	$0 - 0 = 0$	$0 * 0 = 0$	$0 / 1 = 0$
$0 + 1 = 1$	$1 - 0 = 1$	$0 * 1 = 0$	$1 / 1 = 1$
$1 + 0 = 1$	$1 - 1 = 0$	$1 * 0 = 0$	
$1 + 1 = 10$	$10 - 1 = 1$	$1 * 1 = 1$	

Bảng 2: Các phép toán số học trong hệ nhị phân

1.4.3.2. Phép toán logic

PHÉP PHỦ ĐỊNH	PHÉP HỢP	PHÉP GIAO
not 1 = 0	1 or 1 = 1	0 and 0 = 0
not 0 = 1	1 or 0 = 1	0 and 1 = 0
	0 or 1 = 1	1 and 0 = 0
	0 or 0 = 0	1 and 1 = 1

Bảng 3: Các phép toán logic

Câu hỏi thảo luận

1. Phân biệt giữa dữ liệu và thông tin.
2. Phân biệt khái niệm phần cứng và phần mềm của máy tính.
3. Trình bày chức năng của bộ nhớ trong.
4. So sánh sự khác nhau giữa ROM và RAM.
5. Trình bày chức năng bộ xử lý trung tâm (CPU).
6. Điền vào chỗ trống trong các phát biểu dưới đây:
 - a. Những sự kiện không có cấu trúc được lưu trữ trong máy tính gọi là
 - b. Quá trình xử lý dữ liệu sẽ sinh ra mà nó có ý nghĩa
 - c. Một máy tính xử lý dữ liệu thành
 - d. Dữ liệu đi vào trong máy tính khi
 - e. Thông tin đi ra khỏi máy tính khi
 - f. phân biệt một máy tính với một máy tính số học.
 - g. Dữ liệu và các lệnh của chương trình được chứa trong
 - h. Thành phần xử lý dữ liệu trong máy tính là
 - i. Các thành phần vật lý của một máy tính được gọi chung là
 - j. Các chương trình được nhận biết chung như là
7. Trình bày các bước chuyển từ hệ đếm 10 sang hệ đếm 2, hệ đếm 8.
8. Cho hai ví dụ minh họa chuyển từ hệ đếm 8 sang hệ đếm 16.
9. Trình bày bảng mối quan hệ giữa các hệ đếm.
10. Trình bày các bước chuyển từ hệ đếm 2 sang hệ đếm 8, hệ đếm 16.

Bài tập vận dụng

1. Khoanh vào câu trả lời đúng

Câu 1: 1 Byte = bao nhiêu bit?

- A. 10 bit
- B. 9 bit
- C. 8 bit
- D. 7 bit

Câu 2: 1 KB = bao nhiêu Byte?

- A. 1000 byte
- B. 1042 byte
- C. 1024 byte
- D. 1020 byte

Câu 3: Đổi số 11111_2 sang hệ 16:

- A. 3A
- B. 3B
- C. 3C
- D. 3F

Câu 4: Đổi số 0110102 sang hệ 10:

- A. 24
- B. 25
- C. 26
- D. 27

Câu 5: Đổi số $1E16$ sang hệ 2:

- A. 10111
- B. 11110
- C. 11010
- D. 11111

Câu 6: Đổi số 110012 sang hệ 8:

- A. 31
- B. 30
- C. 29
- D. 28

2. Chuyển đổi hệ đếm

a. Đổi các số sau sang hệ thập phân:

$$123AF_{16} =$$

$$1110011102 =$$

$$347654328 =$$

$$3476543216 =$$

b. Đổi các số sau sang hệ nhị phân:

$$AB97316 =$$

$$45728 =$$

$$12010 =$$

c. Đổi các số sau từ hệ thập phân sang hệ 2,16 :

$$38510 =$$

$$28910 =$$

d. Đổi các số sau từ hệ 8 sang hệ 10,2 :

$$1658 =$$

$$2128 =$$

3. Khoanh vào câu trả lời đúng

3.1. Các lĩnh vực nghiên cứu của tin học?

- A. Kỹ thuật về phần cứng và phần mềm.
- B. Kỹ thuật về phần cứng.
- C. Kỹ thuật về phần cứng, phần mềm và công nghệ.
- D. Kỹ thuật về phần mềm.

3.2. Chọn đơn vị đo thông tin lớn nhất?

- A. Byte (B)
- B. Megabyte (Mb)
- C. Terabyte (Tb)
- D. Gigabyte (Gb)

4. Thực hiện các phép toán sau:

$$0101 + 1110 = ?$$

$$111100 - 100 = ?$$

$$0101 \times 1110 = ?$$

$$10000 : 10 = ?$$

5, Khoanh vào câu trả lời đúng

5.1. Tác dụng của ROM BIOS

- A. Bộ nhớ truy cập ngẫu nhiên.
- B. Bộ nhớ ngoài, chứa các phần mềm hệ thống.
- C. Bộ nhớ chỉ đọc, quản lý hệ vào/ra cơ sở.
- D. Bộ nhớ trung tâm, xử lý tính toán.

5.2. Phần cứng (hardware) được gọi là:

- A. là các chương trình chạy trên máy tính
- B. là các thiết bị điện tử và cơ khí của máy tính.
- C. là thiết bị truyền dữ liệu.
- D. là thiết bị quản lý phần mềm.

5.3. Khi đèn Num Lock sáng

- A. Các phím số ở bên phải bàn phím sẽ hiển thị số.
- B. Các phím số ở bên trái bàn phím sẽ hiển thị số.
- C. Không soạn thảo được các phím chữ.
- D. Các phím số ở bên phải bàn phím sẽ không hiện số.

6, Khoanh vào câu trả lời đúng

Câu 1: Đơn vị đo nhỏ nhất trong thông tin?

- a. Megabyte (Mb)
- b. Kilobyte (Kb)
- c. Byte (B)
- d. Bit

Câu 2: Tác dụng của các thiết bị OUTPUT

- a. Đưa các dữ liệu vào máy tính
- b. Hiển thị kết quả ra màn hình hoặc máy in sau khi đã được CPU xử lý
- c. Đưa dữ liệu ra máy tính
- d. Là các thiết bị xuất

Câu 3: Thành phần xử lý các dữ liệu là

- a. RAM
- b. HDD
- c. USB
- d. CPU

Câu 4: Nêu vai trò của thông tin?

- a. Dùng để truyền tin hoặc thể hiện thông tin và được thể hiện bằng âm thanh, chữ viết, ký hiệu, biểu đồ ...
- b. Là công cụ để giao tiếp trong cuộc sống, căn cứ cho mọi quyết định và là phương tiện để lưu truyền cho thế hệ sau. Thông tin có ảnh hưởng lớn đối với kinh tế, xã hội của mọi quốc gia.
- c. Xử lý thông tin trên máy tương tự như xử lý thông tin của con người, đó là thu nhận, phân loại, sắp xếp, lưu trữ, tính toán so sánh và xử lý
- d. Là nguồn gốc của nhận thức hiểu biết, thông tin có thể phát sinh, mã hoá truyền tin và thể hiện ở nhiều vai trò khác nhau

Câu 5: Chức năng của RAM?

- a. Bộ nhớ chỉ đọc
- b. Bộ xử lý trung tâm
- c. Bộ nhớ truy cập ngẫu nhiên
- d. Chứa thông tin cấu hình

Câu 6: Tác dụng của CPU

- a. Bộ vi xử lý trung tâm
- b. Quản lý các thiết bị nhập xuất
- c. Quản lý các thiết bị phần cứng
- d. Bộ nhớ ngoài

Câu 7: Thiết bị không cần thiết đối với máy tính ngày nay

- a. USB
- b. DVD
- c. FDD
- d. HDD

Câu 8: Phím Page Up có tác dụng

- a. Lên một dòng
- b. Về đầu dòng
- c. Lên một trang
- d. Về cuối trang

Câu 9: Thiết bị nào quyết định trực tiếp đến version (đời) của máy tính

- a. RAM
- b. POWER (Thiết bị nguồn máy tính)
- c. CPU
- d. ROM

Câu 10: Phần cứng máy tính được chia làm mấy phần

- a. 1 b. 2 c. 3 d. 4

Câu 11: Khi mất điện, dữ liệu trên ROM sẽ

- a. tự động lưu cấu hình b. mất dữ liệu
c. tự động sao chép d. dữ liệu vẫn còn

Câu 12: Sự giống nhau giữa ROM và RAM

- a. Đều là bộ nhớ chính
b. Đều là bộ nhớ phụ
c. Đều là bộ nhớ quản lý thiết bị nhập/xuất
d. Là bộ nhớ truy cập ngẫu nhiên

Câu 13: CPU được chia làm mấy phần chính

- a. 1 b. 2 c. 3 d. 4

Câu 14: Thế hệ thứ I của máy tính có kích thước khoảng

- a. 50 m² b. 150 m² c. 200 m² d. 250 m²

Câu 15: Chọn câu trả lời sai.

Bộ nhớ ngoài bao gồm

- a. HDD b. USB c. ROM d. FDD

Câu 16: Đơn vị đo tốc độ xử lý của CPU

- a. Mb (Megabyte) b. Kb (Kilobyte)
c. Ghz (GigaHertz) d. Gb (Gigabyte)

Câu 17: Máy tính điện tử đầu tiên ra đời tên là gì

- a. IBM b. SONY c. ENIAC d. HP

Câu 18: Sự giống nhau giữa phím Shift và Caps Lock

- a. Đều xoá ký tự b. Đều tạo chữ hoa
c. Đều là phím chữ d. Đều là phím trợ giúp

Câu 19: Thứ tự cài phần mềm cho 01 máy tính

1. Virus
2. HĐH (Hệ điều hành)
3. Chương trình ứng dụng
4. Drive (nhận các thiết bị phần cứng)

- a. 1,2,3,4 b. 2,4,1,3 c. 3,4,1,2 d. 4,2,1,3

Câu 20: Tác dụng của phím ESC thường được dùng để

- a. Đóng cửa sổ
- b. Huỷ bỏ một công việc nào đó
- c. Chạy một thao tác nào đó
- d. Mở một cửa sổ mới

Câu 21: Tác dụng của thiết bị nhập (INPUT)

- a. Đưa các dữ liệu từ bên ngoài vào trong máy tính để xử lý
- b. Xử lý dữ liệu đầu vào
- c. Hiện thị kết quả sau khi đã được xử lý
- d. Xử lý dữ liệu đầu ra

Câu 22: Màn hình được chia làm mấy loại chính

- a. 1
- b. 2
- c. 3
- d. 4

Câu 23: Trong các phép tác động lên tin sau, phép nào quan trọng nhất

- a. Phép thu thập tin
- b. Phép truyền tin
- c. Phép trữ tin
- d. Phép xử lý tin

Câu 24: Chức năng của ROM

- a. Bộ nhớ truy cập ngẫu nhiên
- b. Bộ nhớ ngoài
- c. Bộ nhớ chỉ đọc
- d. Bộ nhớ chính

Câu 25: Khi bộ nhớ là....., nội dung của nó bị thay đổi

- a. RAM
- b. CPU
- c. ROM
- d. USB

Câu 26: Khi bộ nhớ là....., nội dung của nó không bị thay đổi

- a. RAM
- b. USB
- c. ROM
- d. HDD

Câu 27: Sự khác nhau giữa phím DELETE và BACKSPACE

- a. Đều là phím xoá dữ liệu
- b. Xoá phía sau dấu chò soạn thảo
- c. Xoá phía trước dấu chò soạn thảo
- d. Xoá phía trước và xoá phía sau dấu chò soạn thảo

Câu 28: Main frame là máy tính cỡ gì

- a. Cỡ vừa
- b. Cỡ nhỏ
- c. Cỡ lớn
- d. Cỡ vừa và nhỏ

Câu 29: Thành phần được ký hiệu CU trong bộ xử lý trung tâm (CPU) là gì

- a. Khối xử lý
- b. Khối điều khiển
- c. Khối tính toán số học và logic
- d. Khối kiểm tra

Câu 30: Thành phần được ký hiệu ALU trong bộ xử lý trung tâm (CPU) là gì

- a. Khối xử lý
- b. Khối điều khiển
- c. Khối tính toán số học và logic
- d. Khối logic

7, Chuyển đổi hệ đếm

Câu 1: Chuyển từ hệ 2 sang 8:

111000010101

Câu 2: Chuyển từ hệ 8 sang hệ 16:

3522

Câu 3: Chuyển đổi từ hệ 16 sang hệ 8:

A50D

Câu 4: Chuyển đổi từ hệ 16 sang hệ 8:

7AB5

Câu 5: Chuyển đổi từ hệ 8 sang hệ 16

55401

Câu 6: Chuyển đổi từ hệ 8 sang hệ 16

5002

Câu 7: Chuyển đổi từ hệ 16 sang hệ 8:

5002

Câu 8: Chuyển đổi từ hệ 16 sang hệ 8:

AC0D

Câu 9: Chuyển đổi từ hệ 16 sang hệ 8:

7DC5

Câu 10: Chuyển đổi từ hệ 2 sang hệ 10:

11011110000

Câu 11: Chuyển đổi từ hệ 2 sang hệ 10:

11000001111

Câu 12: Chuyển từ hệ 16 sang hệ 8:

AC06

Câu 13: Chuyển từ hệ 16 sang hệ 8:

CD58

Câu 14: Chuyển từ hệ 16 sang hệ 8:

B50DE

- Câu 15:** Chuyển từ hệ 16 sang hệ 8:
7ABC5
- Câu 16:** Chuyển từ hệ 2 sang hệ 10:
1101000100001
- Câu 17:** Chuyển đổi từ hệ 2 sang hệ 10:
111000000111
- Câu 18:** Chuyển đổi từ hệ 8 sang hệ 16:
2234765
- Câu 19:** Chuyển từ hệ 16 sang hệ 10:
45F67
- Câu 20:** Chuyển đổi từ hệ 2 sang hệ 10:
1000101001100010
- Câu 21:** Chuyển đổi từ hệ 16 sang hệ 8:
12AF4
- Câu 22:** Chuyển đổi từ hệ 16 sang hệ 8:
AB5C4
- Câu 23:** Chuyển đổi từ hệ 2 sang hệ 10:
110011001100011
- Câu 24:** Chuyển đổi từ hệ 8 sang hệ 16:
467321
- Câu 25:** Chuyển đổi từ hệ 10 sang hệ 2:
159441
- Câu 26:** Chuyển đổi từ hệ 16 sang hệ 8:
B6C4EF
- Câu 27:** Chuyển đổi từ hệ 8 sang hệ 16:
190477
- Câu 28:** Chuyển đổi từ hệ 10 sang hệ 2:
19047
- Câu 29:** Chuyển đổi từ hệ 16 sang hệ 8:
C2D4
- Câu 30:** Chuyển đổi từ hệ 2 sang hệ 10:
101110100000001101

- Câu 31:** Chuyển đổi từ hệ 16 sang hệ 8
FA1CD
- Câu 32:** Chuyển đổi từ hệ 2 sang hệ 10:
1110100010101010
- Câu 33:** Chuyển đổi từ hệ 10 sang hệ 2:
5239
- Câu 34:** Chuyển đổi từ hệ 16 sang hệ 8:
BE45F
- Câu 35:** Chuyển đổi từ hệ 2 sang hệ 10
101100011100011
- Câu 37:** Chuyển đổi từ hệ 16 sang hệ 8:
AC0D
- Câu 38:** Chuyển đổi từ hệ 16 sang hệ 8
7DC5
- Câu 39:** Chuyển đổi từ hệ 8 sang 16:
1234567
- Câu 40:** Chuyển đổi từ hệ 2 sang hệ 10:
1110010100001
- Câu 41:** Chuyển đổi từ hệ 16 sang hệ 8:
A12FC
- Câu 42:** Chuyển đổi từ hệ 10 sang hệ 2:
110
- Câu 43:** Chuyển đổi từ hệ 10 sang hệ 2: 235
- Câu 44:** Chuyển đổi từ hệ 10 sang hệ 2: 179
- Câu 45:** Chuyển đổi từ hệ 10 sang hệ 2: 91
- Câu 46:** Chuyển đổi từ hệ 10 sang hệ 2: 203
- Câu 47:** Chuyển đổi từ hệ 10 sang hệ 8: 205
- Câu 48:** Chuyển đổi từ hệ 10 sang hệ 8: 300
- Câu 49:** Chuyển đổi từ hệ 10 sang hệ 8: 261

CHƯƠNG 2: HỆ ĐIỀU HÀNH WINDOWS

Mục tiêu của chương

Nắm vững:

- Quy trình khởi động vào ra, cách sử dụng menu Start và thanh Taskbar trong hệ điều hành Windows 10.

- Sử dụng thành thạo các công cụ trợ giúp của Windows 10.

Nội dung của chương

- Khái niệm về hệ điều hành, cách tổ chức lưu trữ thông tin của hệ điều hành.

- Hệ điều hành Windows 10.

2.1. Khái niệm về hệ điều hành

Hệ điều hành (Operating System) là tập hợp các chương trình được tổ chức thành một hệ thống với nhiệm vụ đảm bảo tương tác giữa người sử dụng với máy tính, cung cấp các phương tiện và dịch vụ để điều phối việc thực hiện các chương trình, quản lý chặt chẽ các tài nguyên của máy, tổ chức khai thác chúng một cách thuận tiện và tối ưu.

Chức năng chính của hệ điều hành là:

- Thực hiện các lệnh theo yêu cầu của người sử dụng máy.

- Quản lý, phân phối và thu hồi bộ nhớ.

- Cung cấp các tài nguyên cho các chương trình và cách tổ chức thực hiện các chương trình đó.

- Kiểm tra và hỗ trợ bằng phần mềm cho các thiết bị ngoại vi để khai thác chúng một cách thuận tiện, hiệu quả.

- Cung cấp các dịch vụ tiện ích hệ thống.

Hiện nay có nhiều hệ điều hành khác nhau như MS-DOS, UNIX, LINUX, Windows 95, Windows 98, Windows 2000, Windows XP, Windows 7, Windows 10.

2.2. Tổ chức lưu trữ thông tin của hệ điều hành

2.2.1. Tập tin (File)

Tập tin là tập hợp thông tin/ dữ liệu được tổ chức theo một cấu trúc nào đó. Nội dung của tập tin có thể là chương trình, dữ liệu, văn bản,... Mỗi tập tin được lưu lên đĩa với một tên riêng phân biệt.

Mỗi hệ điều hành có quy ước đặt tên khác nhau, tên tập tin thường có 2 phần: phần tên (name) và phần mở rộng (extension). Phần tên là phần bắt buộc phải có của một tập tin, còn phần mở rộng thì có thể có hoặc không.

Quy tắc đặt tên: Tên tệp có dạng như sau:

<tên tệp> = <tên chính>[. <mở rộng>]

Trong đó:

- Phần tên do người tạo ra tập tin đặt. Với Msdos phần tên có tối đa là 8 ký tự, với Windows phần tên có thể đặt tối đa 128 ký tự.

- Phần tên không chứa dấu cách.

- Thông thường phần mở rộng do chương trình ứng dụng tạo ra tập tin tự đặt.

- Giữa phần tên và phần mở rộng có một dấu chấm (.) ngăn cách.

Ví dụ 2.1:

Tên đúng: luong.pas T1991.DBE H_dong.Txt QLNV

Tên sai: Can hai.Pas KH1990+1991.DBF Luong.DH TS

Chú ý: Phần mở rộng được đưa vào nhằm mục đích phân loại các tệp. Ví dụ các tệp có phần mở rộng:

COM hay EXE: các tệp chương trình bằng ngôn ngữ máy.

PAS, PRG, C: các tệp chương trình nguồn của Pascal, Foxpro, C.

TXT, DOC, TEX: các tệp văn bản, tài liệu.

BAK: các tệp lưu phòng hồ khi sửa chữa.

DBF, WKQ: các tệp cơ sở dữ liệu...

BMP, GIF, JPG, . . : các file hình ảnh.

MP3, DAT, WMA, ... : các file âm thanh, video.

Trong phần tên chính hay phần mở rộng của tệp có thể dùng dấu ? hay * (gọi là ký tự đại diện) để chỉ một nhóm tệp thay vì một tệp.

Dấu ? đại diện cho một ký tự.

Dấu * đại diện cho một nhóm ký tự.

Ví dụ 2.2:

?anh.doc có thể thay thế cho: hanh.doc ; vanh.doc

Data?.tt? có thể thay thế cho : Data1.tta Data2.ttb

lop*.txt có thể thay cho: lop.txt lophoc.txt

*.pas có thể thay cho : can2.pas Pt-b2 pas luong.pas

hoso.* có thể thay cho: hoso.txt hoso.doc hoso.pas.

Có thể có tập tin không chứa phần mở rộng.

2.2.2.Thư mục (Folder/ Directory)

Thư mục là hình thức phân vùng trên đĩa để việc lưu trữ các tệp có khoa học, hệ thống. Người sử dụng có thể phân một đĩa ra thành nhiều vùng riêng biệt, trong mỗi vùng có thể lưu trữ một phần mềm riêng hoặc các tệp riêng của từng cá nhân.

Mỗi vùng gọi là một thư mục (Directory).

Thư mục là nơi lưu giữ các tập tin theo một chủ đề nào đó theo ý người sử dụng. Đây là biện pháp giúp ta quản lý được tập tin, dễ dàng tìm kiếm chúng khi cần truy xuất. Các tập tin có liên quan với nhau có thể được xếp trong cùng một thư mục.

- Trên mỗi đĩa có một thư mục chung gọi là thư mục gốc.
- Thư mục gốc không có tên riêng và được ký hiệu là \.
- Dưới mỗi thư mục gốc có các tập tin trực thuộc và các thư mục con. Trong các thư mục con cũng có các tập tin trực thuộc và thư mục con của nó.
- Thư mục chứa thư mục con gọi là thư mục cha.
- Thư mục đang làm việc gọi là thư mục hiện hành.
- Tên của thư mục tuân thủ theo cách đặt tên của tập tin.

2.2.3. Ổ đĩa (Drive)

Ổ đĩa là thiết bị của máy tính dùng để đọc, ghi thông tin trên đĩa từ. Mỗi ổ đĩa có một tên khác nhau.

Hệ điều hành Msdos dùng các ký tự từ A đến Z kèm với dấu hai chấm (:) để gọi tên các ổ đĩa.

Các ổ đĩa thông dụng là:

- Ổ đĩa mềm: thường có tên là ổ đĩa A:, dùng cho việc đọc và ghi thông tin lên đĩa mềm.
- Ổ đĩa cứng: được đặt tên là ổ C:, D:,... có tốc độ truy xuất dữ liệu nhanh hơn ổ đĩa mềm nhiều lần. Một máy tính có thể có một hoặc nhiều ổ đĩa cứng.
- Ổ đĩa CD: có các loại như: loại chỉ có thể đọc gọi là ổ đĩa CD-ROM, loại khác còn có thể ghi dữ liệu ra đĩa CD gọi là ổ CD-RW, ngoài ra còn có ổ đĩa DVD.

2.2.4. Đường dẫn (Path)

Khi muốn truy xuất một tệp hay muốn làm việc với một thư mục người sử dụng phải xác định thật rõ ràng về vị trí của tệp hay thư mục đó trên đĩa. Để xác định vị trí của thư mục hay của tệp thông qua đường dẫn (path).

Đường dẫn là một chuỗi ký tự chỉ ra con đường tham chiếu đến thư mục hay tệp. Đường dẫn chứa các tên thư mục cách nhau bởi dấu \.

Đường dẫn tới thư mục có dạng sau:

[\tên thư mục][\tên thư mục]...[\tên thư mục].

Đường dẫn tới tệp có dạng sau:

[\tên thư mục][\tên thư mục]...[\tên thư mục][\tên tệp].

Trong đó tên tệp là tên của tệp cần tham chiếu tới.

2.3. Giới thiệu hệ điều hành Windows

2.3.1. Lịch sử phát triển của Windows

Windows là một bộ chương trình do hãng Microsoft sản xuất. Từ version 3.0 ra đời vào tháng 5 năm 1990 đến nay, Microsoft đã không ngừng cải tiến làm cho môi trường này ngày càng được hoàn thiện.

Vào cuối năm 1995, ở Việt nam đã xuất hiện một phiên bản mới của Windows mà chúng ta quen gọi là Windows 95. Hệ điều hành Windows 95 nhanh chóng trở thành một hiện tượng trên toàn thế giới. Đây là phiên bản đầu tiên xuất hiện các biểu tượng của Internet Explore, Recycle Bin và nút bấm Start.

Windows 98, Windows Me: là những phiên bản tiếp theo của Windows 95, những phiên bản này tiếp tục phát huy và hoàn thiện những tính năng ưu việt của Windows 95 và tích hợp thêm những tính năng mới về Internet và Multimedia.

Windows NT 4.0, Windows 2000, Windows XP, Windows 2003: là những hệ điều hành được phát triển cao hơn, được dùng cho các cơ quan và doanh nghiệp. Giao diện của những hệ điều hành này tương tự như Windows 98/ Windows Me. Điểm khác biệt là những hệ điều hành này có tính năng bảo mật cao, vì vậy nó được sử dụng cho môi trường có nhiều người dùng.

Năm 2012 ra mắt Windows 8 tuy nhiên do loại bỏ phím start và giao diện khó sử dụng cho người dùng.

Năm 2014 Windows 10 được ra mắt sửa được các lỗi gặp phải trong windows 8 trước đây cùng với sự nâng cấp và thêm mới của nhiều ứng dụng và liên tục được cập nhật cho đến tận ngày nay.

Tài liệu này sẽ trình bày dựa vào hệ điều hành Windows 10.

Windows là tập hợp các chương trình điều khiển máy tính thực hiện các chức năng chính như:

- Điều khiển phần cứng của máy tính. Ví dụ, nó nhận thông tin nhập từ bàn phím và gửi thông tin xuất ra màn hình hoặc máy in.
- Làm nền cho các chương trình ứng dụng khác chạy. Ví dụ như các chương trình xử lý văn bản, hình ảnh, âm thanh...
- Quản lý việc lưu trữ thông tin trên các ổ đĩa.
- Cung cấp khả năng kết nối và trao đổi thông tin giữa các máy tính.

Windows có giao diện đồ họa (GUI – Graphics User Interface). Nó dùng các phần tử đồ họa như biểu tượng (Icon), thực đơn (Menu) và hộp thoại (Dialog) chứa các lệnh cần thực hiện.

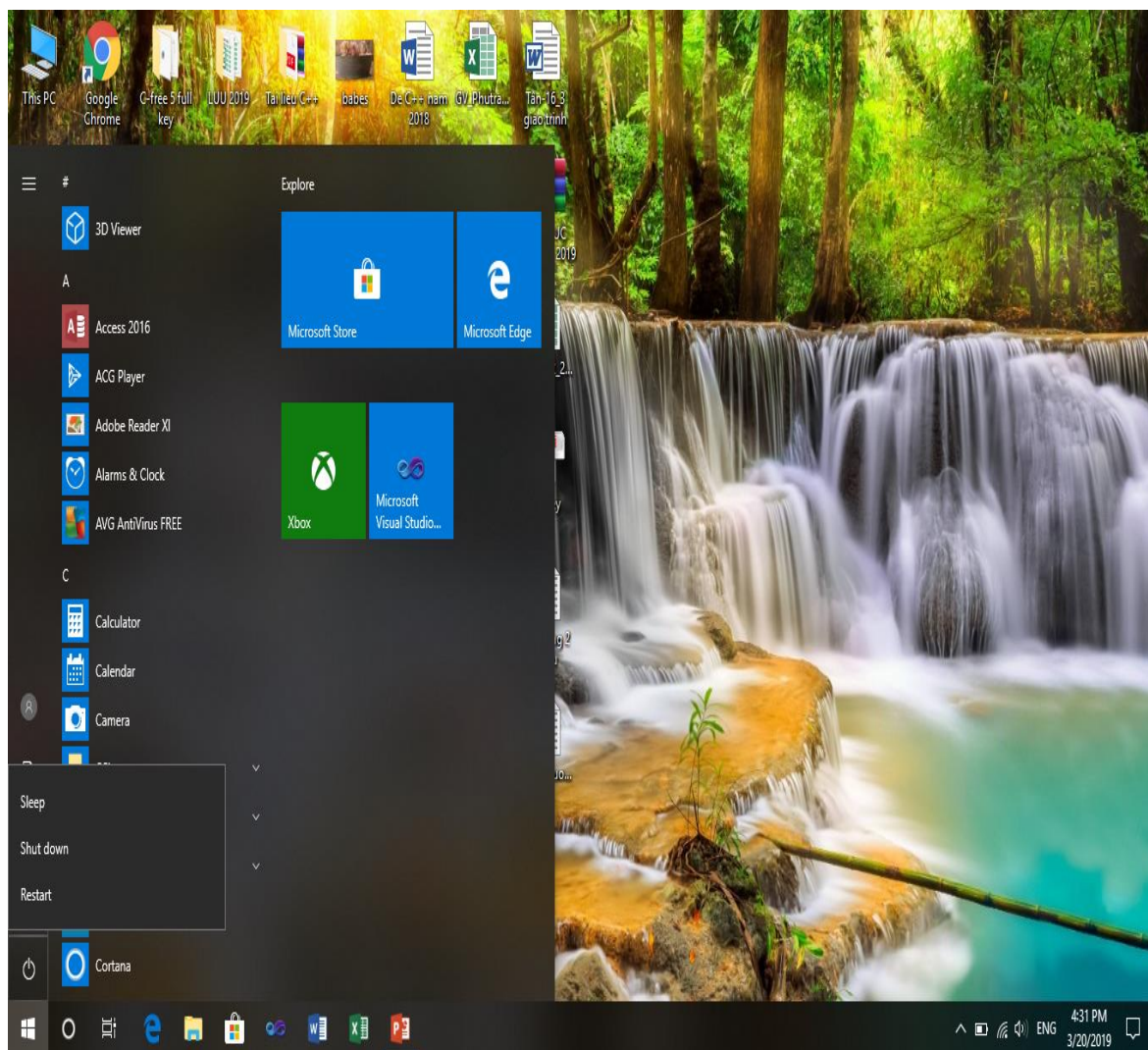
2.3.2. Khởi động và thoát khỏi Windows 10

a. Khởi động Windows

Windows được tự động khởi động sau khi bật máy. Sẽ có thông báo yêu cầu nhập vào tài khoản (User name) và mật khẩu (Password) của người dùng. Thao tác này gọi là đăng nhập.

b. Thoát khỏi Windows:

Bấm vào nút Start (hoặc nhấn tổ hợp phím Alt + F4), chọn mục Power, hộp thoại Power xuất hiện, bấm nút Shutdown.



Hình 3: Màn hình thoát khỏi Windows

Chú ý: Trước khi thoát khỏi Windows để tắt máy tính, người sử dụng nên thoát khỏi các ứng dụng đang chạy sau đó thoát khỏi Windows. Nếu tắt máy ngang có thể gây ra lỗi khi khởi động lại ở lần sử dụng tiếp theo.

2.3.3. Màn hình nền của Windows

a. Màn hình nền (Desktop)



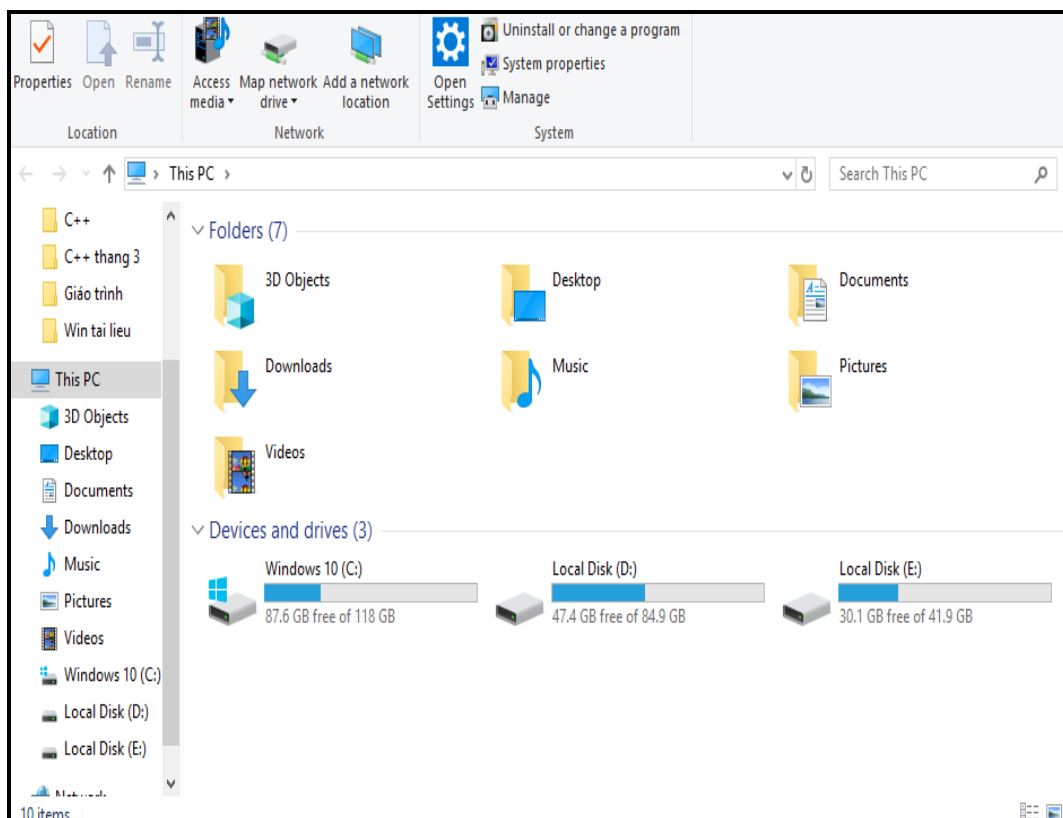
Hình 4: Màn hình nền (Desktop) của Windows.

Sau khi khởi động xong trên màn hình sẽ hiện ra:

- ✓ Màn hình nền (Desktop)
- ✓ Thanh công việc (Task bar)
- ✓ Các biểu tượng. (Icon)

Một số biểu tượng trên màn hình nền

- **This PC:** Cho phép duyệt nhanh tài nguyên trên máy tính. Khi mở biểu tượng This PC, cửa sổ xuất hiện như hình 2.3A.



Hình 5: Giao diện của This PC.

- Cho phép chọn thực hiện một số công việc hệ thống của máy.
- This PC chứa biểu tượng của tất cả các ổ đĩa mềm, ổ đĩa cứng, ổ đĩa CD, ...

Khi bấm vào các biểu tượng trong cửa sổ này sẽ có các cửa sổ cấp nhỏ hơn được mở. Do đó, bằng cách mở dần các cửa sổ từ ngoài vào người sử dụng có thể duyệt tất cả tài nguyên chứa trong máy tính.

- **My Network Places:** cho phép chia sẻ các tài nguyên, thông tin trên mạng. (Nếu người sử dụng là một thành viên trong mạng)
- **Recycle Bin**

Thùng rác là nơi chứa các tệp, các đối tượng bị xóa bỏ khi người sử dụng thao tác trong Windows. Các tệp này không thực sự bị loại bỏ khỏi đĩa cứng cho đến khi ta làm rỗng thùng rác.

Có thể phục hồi các tệp và biểu tượng chương trình bị xóa.

Các thao tác đối với thùng rác. Nháy đúp chuột lên biểu tượng Recycle Bin xuất hiện cửa sổ Recycle Bin, trong cửa sổ này có thể làm các thao tác:

- Phục hồi: Chọn tên tệp cần phục hồi, bấm phải chuột, một menu xuất hiện, chọn Restore, Windows sẽ tự động khôi phục tệp hay biểu tượng chương trình theo nguồn gốc ban đầu.
- Xóa khỏi thùng rác: chọn chương trình cần xóa bỏ, bấm phải chuột, một menu xuất hiện, chọn Delete.

- **Folder:** được gọi “biểu tượng nhóm” hay “thư mục”. Folder là nơi quản lý các Folder khác (cấp thấp hơn) và các tập tin.
- **Menu Start**

Khi Click lên nút Start trên thanh Taskbar, thực đơn Start sẽ được mở và sẵn sàng thi hành các chương trình ứng dụng. Ngoài ra trên thực đơn này người sử dụng còn có thể thực hiện các thao tác tìm kiếm và định cấu hình cho máy tính.

- **Các biểu tượng (icon):**

Biểu tượng là các hình vẽ nhỏ đặc trưng cho một đối tượng nào đó của Windows hoặc của các ứng dụng chạy trong môi trường Windows. Phía dưới biểu tượng là tên biểu tượng. Tên này mang một ý nghĩa nhất định, thông thường nó diễn giải cho chức năng được gán cho biểu tượng (ví dụ nó mang tên của 1 trình ứng dụng).

2.3.4. Hộp thoại trong Windows

Trong khi làm việc với Windows và các chương trình ứng dụng chạy dưới môi trường Windows thường gặp những hộp hội thoại. Các hộp thoại này xuất hiện khi nó cần thêm những thông số để thực hiện lệnh theo yêu cầu của người sử dụng.

Thông thường, trên một hộp hội thoại sẽ có các thành phần sau:

- Hộp văn bản (Text box): dùng để nhập thông tin.
- Hộp liệt kê (List box): liệt kê sẵn một danh sách có các mục có thể chọn lựa, nếu số mục trong danh sách nhiều không thể liệt kê hết thì sẽ xuất hiện thanh trượt để cuộn danh sách.
- Hộp liệt kê thả (Drop down list box/Combo box): khi nhấn chuột vào nút thả sẽ bung xuống một danh sách, trong đó liệt kê các mục và cho chọn.
- Hộp kiểm tra (Check box): cho phép chọn một hoặc nhiều mục.
- Nút tùy chọn (Option button): bắt buộc phải chọn một trong số các mục.
- Nút lệnh (Command) lệnh cần thực thi.
- Nút đóng hộp thoại thường nằm ở đầu bên phải thanh tiêu đề dùng để đóng hộp thoại.

2.4. File Explorer

2.4.1. Giới thiệu

File Explorer là một chương trình cho phép người sử dụng thao tác với các tài nguyên có trong máy tính như tập tin, thư mục, ổ đĩa và những tài nguyên khác có trong máy cũng như các máy tính trong hệ thống mạng (nếu máy tính của bạn có nối mạng).

Với File Explorer, các thao tác như sao chép, xóa, đổi tên thư mục và tập tin,... được thực hiện một cách thuận tiện và dễ dàng.

2.4.2. Thao tác với các thư mục và tập tin

2.4.2.1. Tạo thư mục

- Chọn nơi chứa thư mục cần tạo.
- Bấm phải chuột, chọn New sau đó chọn Folder.
- Nhập tên thư mục mới, sau đó gõ Enter để kết thúc.

2.4.2.2. Mở tập tin/ thư mục

Có thể thực hiện các cách sau:

- Cách 1: Bấm đúp lên biểu tượng của tập tin/thư mục.
- Cách 2: Bấm phải lên biểu tượng của tập tin/thư mục và chọn mục Open.
- Cách 3: Chọn tập tin/ thư mục và nhấn phím Enter.

2.4.2.3. Sao chép thư mục hoặc tập tin

Chọn thư mục hoặc tập tin cần sao chép.

Sau đó thực hiện một trong các cách sau:

- Cách 1: Nhấn giữ phím Ctrl và Drag đối tượng đã chọn đến nơi cần chép.
- Cách 2: Nhấn tổ hợp phím Ctrl + C.

(hoặc bấm phải chuột và chọn Copy), sau đó chọn nơi đến và nhấn tổ hợp phím Ctrl + V (hoặc bấm phải chuột chọn Paste).

2.4.2.4. Di chuyển thư mục và tập tin

Chọn thư mục hoặc tập tin cần di chuyển. Sau đó thực hiện một trong các cách sau

- Cách 1: Drag đối tượng đã chọn đến nơi cần di chuyển.
- Cách 2: Nhấn tổ hợp phím Ctrl + X (hoặc bấm phải chuột chọn Cut) để chép vào Clipboard, sau đó chọn nơi cần di chuyển đến và nhấn tổ hợp phím Ctrl + V (hoặc bấm phải chuột chọn Paste).

2.4.2.5. Xoá thư mục và tập tin

- Chọn các thư mục và tập tin cần xóa.
- Nhấn phím Delete (hoặc bấm phải chuột chọn Delete).

2.4.2.6. Chọn tập tin/thư mục

- Chọn một tập tin/thư mục: bấm lên biểu tượng tập tin/thư mục.
- Chọn một nhóm tập tin/thư mục: có thể thực hiện theo các cách:
 - Các đối tượng cần chọn liên tục nhau: bấm lên đối tượng đầu danh sách để chọn, sau đó nhấn giữ phím Shift và bấm lên đối tượng ở cuối danh sách.
 - Các đối tượng cần chọn nằm rời rạc nhau: nhấn giữ phím Ctrl và bấm chọn các đối tượng tương ứng.

2.4.2.7. Phục hồi thư mục và tập tin

Đối tượng bị xóa sẽ được đưa vào Recycle Bin. Nếu muốn phục hồi các đối tượng đã xóa, thực hiện các thao tác sau đây:

- Bấm đúp lên biểu tượng Recycle Bin.
- Chọn tên đối tượng cần phục hồi.
- Bấm phải và chọn mục Restore.

Chú ý: Nếu muốn xóa hẳn các đối tượng, ta thực hiện thao tác xóa một lần nữa đối với các đối tượng ở trong Recycle Bin.

2.4.2.8. Đổi tên thư mục và tập tin

- Chọn đối tượng muốn đổi tên.
- Bấm phải chuột và chọn mục Rename.
- Nhập tên mới, sau đó gõ Enter để kết thúc.

2.4.2.9. Thay đổi thuộc tính tập tin và thư mục

- Nhấn chuột lên đối tượng muốn thay đổi thuộc tính và chọn mục Properties.
- Thay đổi các thuộc tính.
- Chọn Apply để xác nhận thay đổi, ngược lại thì nhấn Cancel.

2.4.3. Thao tác với các Shortcuts

a. Tạo Shortcut trên màn hình nền

- Bấm phải chuột lên màn hình nền, chọn New/ Shortcut.
- Trong mục Type the location of the item, nhập đường dẫn của đối tượng cần tạo (ổ đĩa/ thư mục/ tập tin, ...) hoặc bấm lên nút Browse để chọn đường dẫn cho đối tượng. Chọn Next để qua bước kế tiếp.

- Nhập tên cho shortcut cần tạo.
- Bấm Finish để hoàn thành.

b. Đổi tên cho Shortcut.

- Chọn shortcut cần đổi tên, bấm phải chuột, chọn Rename.
- Nhập tên mới cho shortcut.
- Nhấn Enter để kết thúc.

c. Xóa shortcut

- Chọn shortcut cần xóa.
- Nhấn phím Delete hoặc bấm phải chuột phải lên shortcut, chọn Delete.
- Xác nhận có thực sự muốn xóa hay không (Yes/ No).

d. Thay đổi thuộc tính cho shortcut

- Chọn shortcut cần thay đổi thuộc tính, bấm phải chuột và chọn Properties.
- Lớp General: cho phép chọn thuộc tính chỉ đọc (Read-only), hay ẩn (Hidden).

- Lớp Shortcut: cho phép chọn thay đổi một số lựa chọn sau:
 - Shortcut key: gán phím nóng cho shortcut.
 - Run: chọn chế độ hiển thị màn hình khi mở (bình thường/ thu nhỏ/ phóng to).
 - Change icon: thay đổi biểu tượng của shortcut.

e. Phục hồi lại shortcut vừa bị xoá:

- Mở biểu tượng Recycle Bin.
- Chọn Shortcut cần khôi phục, nhấn phải chuột và chọn Restore.

Câu hỏi thảo luận

1. Nêu những ưu điểm của hệ điều hành win10.
2. Nêu cách tạo thư mục, cách đổi tên thư mục và xóa thư mục.
3. Cách tạo ShortCut trên Desktop.
4. Cách thay đổi thuộc tính và phục hồi shortcut.
5. Nêu các cách di chuyển thư mục và tập tin.
6. Trình bày sự khác nhau giữa hai khái niệm file và thư mục?

Bài tập vận dụng

1. Sắp xếp lại các Folder và ShortCut trên Desktop.
2. Điều chỉnh Background, thay đổi kiểu màn hình nền của desktop
3. Điều chỉnh ngày, tháng, năm cho hệ thống đúng với ngày tháng năm hiện hành (quan sát kết quả ở đồng hồ góc phải taskbar).
5. Mở cửa sổ This PC: bấm đúp vào biểu tượng This PC trên desktop.

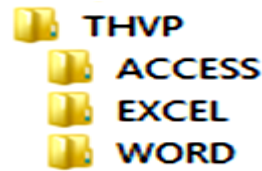
Thực hiện các công việc sau:

- Thu nhỏ cửa sổ This PC.
- Phóng to cửa sổ This PC.
- Phục hồi cửa sổ This PC.
- Thay đổi kích thước cửa sổ này tùy ý.
- Đóng cửa sổ This PC.

6. Mở cửa sổ Recycle bin trên desktop: bấm đúp vào biểu tượng Recycle bin trên desktop. Thực hiện các công việc sau:

- Thu nhỏ, phóng to và phục hồi cửa sổ Recycle bin.
- Chọn một tập tin, hoặc thư mục có trong cửa sổ này, thực hiện thao tác khôi phục tập tin, thư mục đã bị xóa.

7. Khởi động File Explorer và tạo cấu trúc thư mục (Folder) như hình dưới đây:



Thực hiện các thao tác:

- Chọn các thư mục (Folder) để nó trở thành thư mục (Folder) hiện hành và quan sát nội dung của nó.
- Xem thông tin của các thư mục ACCESS, WORD.
- Gán thuộc tính ẩn (Hidden) cho thư mục ACCESS, gán thuộc tính chỉ đọc (Read Only) cho thư mục WORD.
- Xóa bỏ các thuộc tính đã cài đặt cho thư mục ACCESS, WORD.

7. Khoanh vào câu trả lời đúng

Câu 1: Trong Window, phím hoặc tổ hợp phím dùng để chụp màn hình:

- a. Ctrl+C b. Ctrl+Ins c. Print Screen d. ESC

Câu 2: Trong Windows Explorer, để chọn nhiều tập tin không nằm liền nhau, ta dùng chuột và phím

- a. Shift b. Alt c. Tab d. Ctrl

Câu 3: Trong Window, Phím hoặc tổ hợp phím dùng để chụp cửa sổ hiện hành

- a. Alt+Print Screen b. Ctrl+Print Screen
c. Print Screen d. Shift+ Print Screen

Câu 4: Trong Window, phím tắt nào dùng để đóng cửa sổ hiện hành:

- a. F1 b. Alt+F4 c. F7 d. Esc

Câu 5: Trong Windows Explorer, để chọn nhiều tập tin nằm liền nhau, ta dùng chuột và phím


- a. Shift b. Alt c. F2 d. Tab

Câu 6: Trong Window, phím tắt để đổi tên tệp tin, thư mục

- a. F4 b. Ctrl+N c. F2 d. Alt+N

Câu 7: Tên của chương trình soạn thảo văn bản có sẵn trong Windows

- a. Ipad b. WordPad c. MS Word d. Paint

Câu 8: Trong Windows Explorer, chức năng của biểu tượng  là

- a. Hiện thị Desktop
b. Thoát khỏi Windows
c. Thoát khỏi Windows Explorer

d. Hiện thị thư mục trên 1 cấp (thư mục cha của thư mục hiện thời)

Câu 9: Trong Windows Explorer, chức năng của biểu tượng 

- a. Xóa đối tượng đang chọn
- b. Đóng ứng dụng
- c. Thoát Windows
- d. Copy

Câu 10: Phím tắt để sao chép một đối tượng vào Clipboard là:

- a. Ctrl+X
- b. Ctrl+C
- c. Ctrl+V
- d. Ctrl+Z

Câu 11: Phím tắt để dán một đối tượng từ Clipboard là:

- a. Ctrl+X
- b. Ctrl+C
- c. Ctrl+V
- d. Ctrl+Z

Câu 12: Xóa một tập tin mà không chuyển vào thùng rác, sử dụng tổ hợp phím

- a. Alt+F4
- b. Ctrl+Delete
- c. Shift+Delete
- d. Alt+Delete

Câu 13: Trong Window, phím tắt được sử dụng để hiển thị thông tin hệ thống (System Properties):

- a. Window+R
- b. Window+Pause/Break
- c. Window+S
- d. Window+Print Screens

Câu 14: Trong ứng dụng Windows Explorer, để chọn nhiều tập tin hay thư mục không liên tục ta thực hiện thao tác kết hợp phím ... với click chuột.

- a. Shift
- b. Alt
- c. Tab
- d. Ctrl

Câu 15: Thao tác: Chọn thư mục sau đó nhấn F2, có chức năng gì?

- a. Di chuyển thư mục
- b. Đổi tên thư mục
- c. Xóa thư mục
- d. Sao chép thư mục

Câu 16: Chọn thư mục, ấn tổ hợp phím Shift + Delete, điều gì sẽ xảy ra?

- a. Thư mục bị đưa vào thùng rác
- b. Thư mục bị xoá vĩnh viễn khỏi máy
- c. Phục hồi thư mục từ trong thùng rác
- d. Đổi tên thư mục

Câu 17: Phím tắt để cắt một đối tượng vào Clipboard là:

- a. Ctrl+X
- b. Ctrl+C
- c. Ctrl+V
- d. Ctrl+Z

Câu 18: Trong Windows, biểu tượng của thư mục thường có màu gì?

- a. Vàng
- b. Xanh
- c. Đỏ
- d. Tím

Câu 19: Muốn xóa một file trong Windows, có thể thực hiện theo cách nào sau đây?

- a. Kích chuột phải tại file, chọn Delete
- b. Mở file, chọn toàn bộ nội dung file, bấm phím Delete
- c. Mở file, chọn toàn bộ nội dung file, giữ phím Shift và bấm phím Delete
- d. Mở file, chọn toàn bộ nội dung file, giữ phím Shift và bấm phím Open

Câu 20: Khi tìm kiếm tập tin, thư mục, ký tự * dùng để làm gì?

- a. Thay thế cho 1 số ký tự
- b. Không thể thay thế cho ký tự
- c. Thay thế cho dấu cách
- d. Thay thế cho 1 ký tự

Câu 21: Ký tự nào sau đây được sử dụng để thay thế một ký tự trong tên file trong thao tác tìm kiếm?

- a. *
- b. ?
- c. %
- d. &

Câu 22: Tập hoặc thư mục có thuộc tính chỉ đọc là thuộc tính nào?

- a. Hidden
- b. Modify
- c. Delete
- d. Read Only

PHẦN 2: NGÔN NGỮ LẬP TRÌNH C++

CHƯƠNG 1: THUẬT GIẢI VÀ LƯU ĐỒ

Mục tiêu của chương

Nắm vững:

- Các khái niệm về thuật giải, chương trình, các cách diễn tả thuật toán.
- Trình tự thực hiện các bước của thuật toán ứng dụng trong các bài tập cụ thể.
- Vận dụng lý thuyết giải các bài tập về xây dựng sơ đồ thuật giải.

Nội dung của chương

Nghiên cứu một số khái niệm cơ bản về chương trình, các cách diễn tả thuật toán: bằng ngôn ngữ tự nhiên, diễn đạt bằng sơ đồ, và bằng ngôn ngữ lập trình C++.

1.1. Thuật giải (algorithm)

Khi cần giải quyết một bài toán trong thực tế với sự trợ giúp của máy tính điện tử, người sử dụng thường phải biết dữ liệu vào của bài toán (input) và yêu cầu dữ liệu ra (output) của bài toán. Bước tiếp theo, phải thiết lập được các bước thao tác cụ thể để từ input ta có được output. Công việc đó trong Tin học được gọi là xây dựng thuật giải.

1.1.1. Khái niệm thuật giải

Thuật giải (algorithm) là một tập hữu hạn các thao tác (các công việc, các phép toán) có thể đặt tên được và chúng được thực hiện theo một trình tự thích hợp trên một số đối tượng để đạt được mục đích mong muốn. "Hữu hạn" ở đây được hiểu cả về mặt thời gian thực hiện lẫn công cụ thực hiện.

Cùng một bài toán có thể có nhiều thuật toán khác nhau. Trong đó thuật toán nào đơn giản, dễ hiểu, có độ chính xác cao, được bảo đảm về mặt toán học và dễ triển khai trên máy nhất được gọi là thuật toán tối ưu.

Nghiên cứu thuật toán là một trong những vấn đề quan trọng nhất của Tin học.

1.1.2. Chương trình

Chương trình là tập hợp dãy các lệnh điều khiển máy tính thực hiện. Như vậy, có thể nói một chương trình là một cách diễn tả thuật giải trong một ngôn ngữ chính xác để máy tính có thể hiểu được.

Chương trình cho máy tính chính là sự mô tả cụ thể các thuật toán và sự biểu diễn một cách rõ ràng chính xác các dữ liệu.

1.1.3. Các đặc trưng của thuật toán

Để phân biệt thuật toán với các thuật ngữ khác như phương pháp tiến hành, chương trình hành động... thuật toán trong tin học phải có các tính chất sau:

- + Tính dừng: Thuật toán phải kết thúc sau một số hữu hạn bước.

+ Tính xác định: Các thao tác ở mỗi bước phải hết sức rõ ràng và chỉ được hiểu theo một nghĩa duy nhất. Trong cùng một điều kiện hai máy khác nhau hoặc hai lần thao tác khác nhau phải cho cùng một kết quả khi thực hiện cùng một thuật toán.

+ Tính hàng loạt: Thuật toán có hiệu lực như nhau đối với các bài toán cùng loại (có cùng miền áp dụng thuật toán).

+ Tính khả thi: Thuật toán phải bao gồm các thao tác mà máy có thể thực hiện được nghĩa là chỉ bao gồm những phép toán số học, các phép so sánh, các phép logic, các phép nhập xuất thông tin tiêu chuẩn.

+ Tính đầy đủ: Thuật toán phải vét được hết các tình huống, các khả năng có thể xảy ra, không bỏ sót bất kỳ một trường hợp nào.

1.1.4. Trình tự thực hiện các bước của thuật toán

Giải thuật được thiết kế theo ba cấu trúc suy luận cơ bản sau đây:

□ Tuần tự (Sequential): Các công việc được thực hiện một cách tuần tự, công việc này nối tiếp công việc kia.

□ Cấu trúc lựa chọn (Selection) : Lựa chọn một công việc để thực hiện căn cứ vào một điều kiện nào đó. Có một số dạng như sau:

- Cấu trúc 1: *Nếu <điều kiện> (đúng) thì thực hiện <công việc>*

- Cấu trúc 2: *Nếu <điều kiện> (đúng) thì thực hiện <công việc 1>, ngược lại (điều kiện sai) thì thực hiện <công việc 2>.*

- Cấu trúc 3: *Trường hợp <i> thực hiện <công việc i>.*

□ Cấu trúc lặp (Repeating): Thực hiện lặp lại một công việc nhiều lần căn cứ vào một điều kiện nào đó. Có hai dạng như sau:

- Lặp xác định: là loại lặp mà khi viết chương trình, người lập trình đã xác định được công việc sẽ lặp bao nhiêu lần.

- Lặp không xác định: là loại lặp mà khi viết chương trình người lập trình chưa xác định được công việc sẽ lặp bao nhiêu lần. Số lần lặp sẽ được xác định khi chương trình thực thi.

1.2. Các cách diễn tả thuật toán

Có rất nhiều cách diễn đạt thuật giải. Có thể diễn đạt thuật giải bằng một lưu đồ, bằng ngôn ngữ tự nhiên hoặc bằng một ngôn ngữ lập trình nào đó.

1.2.1. Diễn đạt bằng ngôn ngữ tự nhiên

Ví dụ 1.1: Mô tả thuật giải tìm ước chung lớn nhất (UCLN) của hai số a và b là:

Bước 1: Nhập vào hai số a và b.

Bước 2: So sánh 2 số a,b chọn số nhỏ nhất gán cho UCLN.

Bước 3: Nếu hai số a và b chia hết cho UCLN thì Thực hiện bước 5.

Bước 4: Giảm UCLN một đơn vị và quay lại bước 3

Bước 5: In UCLN - kết thúc.

Cách diễn đạt này tuy khá đơn giản và gần gũi với tư duy của con người nhưng phụ thuộc rất nhiều vào cách diễn đạt của người sử dụng.

Ví dụ 1.2: Mô tả thuật giải cho bài toán giải phương trình bậc hai.

Bước 1: Yêu cầu cho biết giá trị của 3 hệ số a, b, c.

Bước 2: Nếu a = 0, thông báo dữ liệu đầu vào không đảm bảo. Kết thúc giải thuật.

Bước 3: Nếu a ≠ 0:

3.1: Tính Delta = b² - 4ac.

3.2: Nếu Delta > 0 thì xuất thông báo phương trình có 2 nghiệm phân biệt là x₁, x₂.

Trong đó: $x_1 = \frac{-b + \sqrt{\Delta}}{2a}$, $x_2 = \frac{-b - \sqrt{\Delta}}{2a}$. Kết thúc giải thuật.

3.3: Nếu Delta = 0 thì xuất thông báo phương trình có nghiệm kép là $x_0 = \frac{-b}{2a}$.

Kết thúc giải thuật.

3.4: Nếu Delta < 0 thì xuất thông báo phương trình vô nghiệm. Kết thúc giải thuật.

Ví dụ 1.3: Mô tả thuật giải cho bài toán tính tổng $S = a_1 + a_2 + a_3 + \dots + a_n$ (với n nguyên, dương).

Bước 1: Nhập số các số hạng n.

Bước 2: Cho S=0 (lưu trữ số 0 trong S)

Bước 3: Cho i=1 (lưu trữ số 1 trong i)

Bước 4: Kiểm tra nếu i ≤ n thì thực hiện bước 5, ngược lại thực hiện bước 8.

Bước 5: Nhập a_i

Bước 6: Cho S=S+ a_i (lưu trữ giá trị S + a_i trong S)

Bước 7: Tăng i lên 1 đơn vị và quay lại bước 4.

Bước 8: In S và kết thúc chương trình.

Ví dụ 1.4: Mô tả thuật giải cho bài toán nhập vào 1 số n, sau đó lần lượt nhập vào n giá trị a₁, a₂, ..., a_n. Hãy tìm và in ra giá trị lớn nhất trong n số a₁, a₂, ..., a_n.

Bước 1: Nhập số n.

Bước 2: Nhập số thứ nhất a₁.

Bước 3: Gán max=a₁.

Bước 4: Gán i=2.

Bước 5: Nếu i ≤ n thì thực hiện bước 6, ngược lại thực hiện bước 9.

Bước 6: Nhập a_i.

Bước 7: Nếu max < a_i thì gán max=a_i.

Bước 8: Tăng i lên một đơn vị và quay lại bước 5.

Bước 9: In max - kết thúc.

Vi dụ 1.5: Mô tả thuật giải cho bài toán tính bình phương của một số

Bước 1: Nhập giá trị cho x

Bước 2: Tính giá trị $x*x$ và gán cho s

Bước 3: Trả về giá trị s .

Vi dụ 1.6: Mô tả thuật giải cho bài toán tăng lương hiện tại lên 5%.

Bước 1: Nhập giá trị cho lương cũ.

Bước 2: Tính giá trị lương cũ*1.05 và gán cho lương mới.

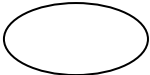
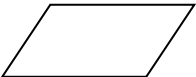

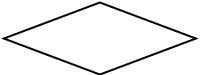


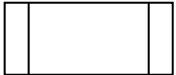

Bước 3: Trả về giá trị lương mới.

1.2.2. Diễn đạt bằng ngôn ngữ lập trình

Ngôn ngữ lập trình (language program) là ngôn ngữ do các chuyên gia tin học tạo ra chuyên dùng để viết chương trình cho máy tính. Nó được xây dựng khá đơn giản về chính tả và ngữ pháp khá gần gũi với ngôn ngữ khoa học kỹ thuật, quản lý.

a. Diễn đạt bằng lưu đồ

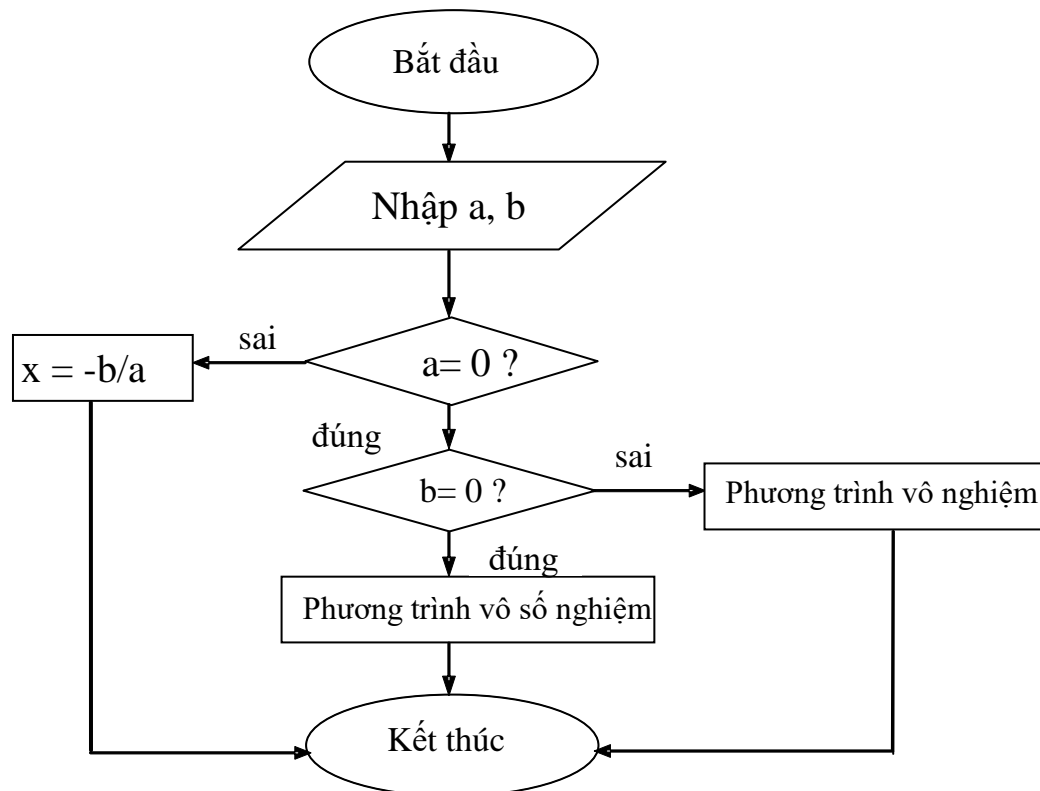
Một số ký hiệu hay dùng:

Ký hiệu	Mô tả
	Điểm bắt đầu và chấm dứt thuật toán
	Thao tác nhập hay xuất dữ liệu
	Khối xử lý công việc
	Khối quyết định lựa chọn
	Điểm nối
	Chuẩn bị
	Khối chương trình con
	Dòng tính toán, thao tác của chương trình

Ví dụ 1.7: Thuật toán giải phương trình bậc nhất $ax + b = 0$, gồm các bước:

- Bước 1: Nhập vào 2 hệ số a và b .
- Bước 2: Xét điều kiện $a = 0$?
Nếu đúng là $a = 0$, thì đi đến bước 3.
Nếu không, nghĩa là $a \neq 0$, thì đi đến bước 4.
- Bước 3: Xét điều kiện $b = 0$?
Nếu $b = 0$, thì báo phương trình có vô số nghiệm. Chuyển đến bước 5.
Nếu $b \neq 0$, thông báo phương trình vô nghiệm. Chuyển đến bước 5.
- Bước 4: Thông báo phương trình có một nghiệm duy nhất là $x = -b/a$.
- Bước 5: Kết thúc thuật toán.

Trong ví dụ trên ta có thể trình bày với lưu đồ sau:



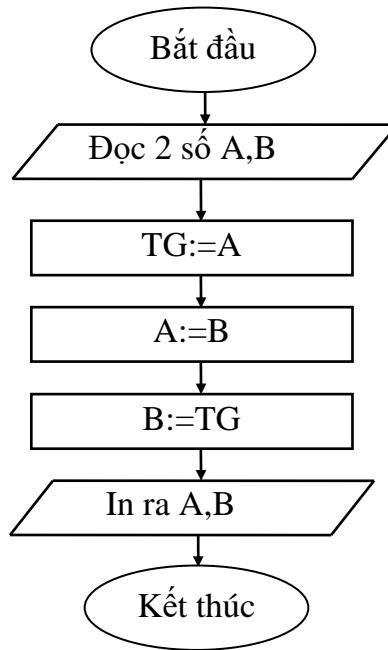
Hình 6: Lưu đồ thuật toán giải phương trình bậc nhất $ax + b = 0$.

Ví dụ 1.8: Vẽ lưu đồ cho thuật toán đổi chỗ

Viết thuật toán để nhập vào 2 số A, B từ bàn phím sau đó đổi giá trị của biến A cho biến B và ngược lại.

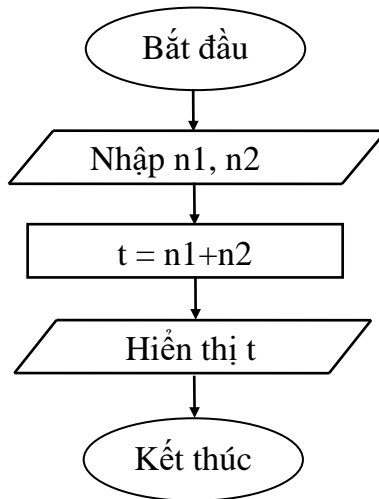
Để viết thuật toán cho bài toán này, chúng ta sẽ đưa thêm vào chương trình một biến trung gian (TG) sau đó tiến hành chuyển giá trị của biến A cho biến TG, chuyển giá trị của biến B cho biến A và cuối cùng chuyển giá trị của biến TG cho biến B .

Thuật toán được trình bày như sau:



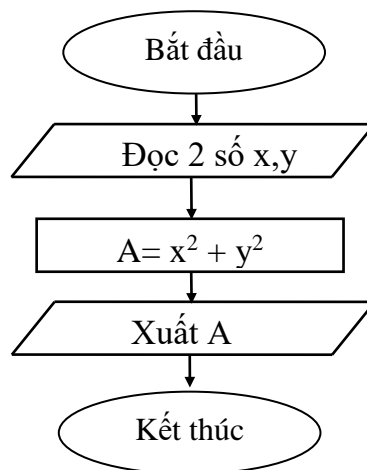
Hình 7: Lưu đồ thuật toán đổi chỗ.

Ví dụ 1.9: Vẽ lưu đồ cho thuật toán tính tổng 2 số n_1 và n_2



Hình 8: Lưu đồ thuật toán thuật toán tính tổng 2 số n_1 và n_2 .

Ví dụ 1.10: Vẽ lưu đồ cho thuật toán tính $A = x^2 + y^2$



Hình 9: Lưu đồ thuật toán tính $A = x^2 + y^2$.

1.2.2.2. Dùng mã giả

Phương pháp này sử dụng các cú pháp của một ngôn ngữ lập trình cụ thể để thể hiện giải thuật. Dùng mã giả vừa vận dụng được các khái niệm trong ngôn ngữ lập trình, vừa giúp người cài đặt dễ dàng nắm bắt được nội dung giải thuật. Tuy nhiên, trong mã giả vẫn dùng một phần ngôn ngữ tự nhiên.

Ví dụ 1.11: Một đoạn mã giả của giải thuật giải phương trình bậc hai, minh họa bằng ngôn ngữ Pascal.

```
Begin
  Nhập 3 hệ số a, b, c;
  If a=0 then Xuất kết quả: Hệ số a không hợp lệ
  Else Begin
    Delta := b*b-4*a*c;
    If Delta > 0 then Begin
      x1 := (-b-sqrt(delta))/(2*a);
      x2 := (-b+sqrt(delta))/(2*a);
      Xuất kết quả: Phương trình có 2 nghiệm là x1 và x2;
    End
    Else If Delta = 0 then
      Xuất kết quả: Phương trình có nghiệm kép là -b/(2*a)
    Else {Trường hợp Delta < 0}
      Xuất kết quả: Phương trình vô nghiệm;
    End;
  End.
```

Ví dụ 1.12: Một đoạn mã giả của giải thuật giải phương trình $ax^2 + bx + c = 0$ ($a \neq 0$), minh họa bằng ngôn ngữ C++.

```
int main() {
    float a, b, c; // khai bao cac he so
    float delta;
    float x1, x2; // khai bao 2 nghiem
    cout << "Nhap a, b, c:\n"; cin >> a >> b >> c; // qui uoc nhap a ≠ 0
    delta = b*b - 4*a*c;
    if (delta < 0) cout << "Phuong trinh vo nghiem\n";
    else if (delta==0) cout<<"Phuong trinh co nghiem kep:" << -b/(2*a) << '\n';
    else
    {
        x1 = (-b+sqrt(delta))/(2*a);
        x2 = (-b-sqrt(delta))/(2*a);
        cout << "nghiem 1 = " << x1 << " và nghiem 2 = " << x2;
    }
}
```

Câu hỏi thảo luận

1. Trình bày các đặc trưng cần có của một thuật toán? Các cách biểu diễn một thuật toán?
2. Trình tự thực hiện các bước của thuật toán.
3. Cho 3 ví dụ về mô tả và diễn đạt các thuật giải bằng ngôn ngữ tự nhiên.
4. Cho tam giác ABC có góc vuông A và cho biết cạnh a và góc B. Hãy viết thuật toán để tính góc C, cạnh b và cạnh c.
5. Trình bày tính chất xác định của thuật toán và nêu rõ nghĩa của tính chất này.
6. Xác định input và output cho các thuật toán sau đây:
 - Rút gọn một phân số.
 - Kiểm tra xem ba số cho trước a, b và c có thể là độ dài ba cạnh của một tam giác hay không?
 - Tính trung bình cộng của hai số.
 - Tìm chu vi và diện tích của hình tròn có bán kính cho trước.

Bài tập vận dụng

1. Viết thuật giải nhập 1 số từ bàn phím và in ra bình phương của số đó nếu số đó là số dương.
2. Viết thuật giải nhập từ bàn phím một số tự nhiên N và in ra các số nguyên trong phạm vi từ 1 đến N.

3. Viết thuật giải nhập từ bàn phím 2 số tự nhiên m, n ($m < n$) và in ra màn hình các số chia hết cho m trong khoảng từ 1 đến n .

Hướng dẫn: Để kiểm tra xem một số x có chia hết cho m không ta dùng biểu thức kiểm tra $X \text{ MOD } M = 0$. Nếu biểu thức này là đúng thì x chia hết cho m .

4. Viết thuật giải nhập 3 số từ bàn phím và in ra số lớn nhất trong ba số đó.

5. Viết thuật giải nhập 2 số từ bàn phím và in ra BCNN của hai số đó.

6. Biểu diễn thuật toán giải các bài toán sau bằng sơ đồ khối và giả mã:

- Giải hệ phương trình bậc nhất.

- Tính số ngày của một tháng một năm nào đó.

- Thuật toán tìm ước số chung lớn nhất.

7. Hãy xây dựng thuật toán để tính tổng các chữ số của một số nguyên N bất kỳ.

Ví dụ: $N = 2015$ thì thuật toán trả về 8 ($2+0+1+5$).

CHƯƠNG 2: TỔNG QUAN VỀ NGÔN NGỮ C++

Mục tiêu của chương

Nắm vững:

- Các cấu trúc cơ bản của ngôn ngữ C++, thư viện chuẩn trong C++.
- Các bước để viết một chương trình.
- Các thao tác soạn thảo và chạy code trên các chương trình phần mềm.

Nội dung của chương

Nghiên cứu các tính chất đặc trưng của ngôn ngữ C++, một số khái niệm cơ bản về cấu trúc cơ bản một chương trình C++.

2.1. Lịch sử hình thành và phát triển

Ngôn ngữ lập trình C++ được phát triển năm 1980 bởi Bjarne Stroustrup tại Bell Labs của AT&T (American Telephone & Telegraph), đặt tại Hoa Kỳ. Bjarne Stroustrup được biết đến như là người sáng lập ra ngôn ngữ C++.

Ngôn ngữ lập trình C++ được thiết kế với ưu tiên cho lập trình hệ thống và các hệ thống nhúng, tài nguyên hạn chế và lớn, với hiệu suất, hiệu quả và tính linh hoạt cao.

C++ với những điểm mạnh là cơ sở hạ tầng phần mềm phong phú và các ứng dụng hạn chế tài nguyên. bao gồm các ứng dụng dành cho máy tính để bàn, máy chủ và các ứng dụng quan trọng về hiệu suất.

C++ là một ngôn ngữ được biên dịch, có thể sử dụng trên nhiều nền tảng. Nhiều nhà cung cấp cung cấp các trình biên dịch C++, bao gồm tổ chức phần mềm tự do, Microsoft, Intel và IBM.

Năm 1985, tác phẩm The C++ Programming Language được xuất bản lần đầu tiên, cung cấp một tài liệu tham khảo quan trọng cho ngôn ngữ nhưng đó chưa là một tiêu chuẩn chính thức.

Năm 1989 phiên bản C++ 2.0 phát hành. Các tính năng mới bao gồm đa kế thừa, lớp trừu tượng, hàm tĩnh, hàm thành viên hằng, và thành viên bảo tồn.

Năm 1990, cuốn The Annotated C++ Reference Manual được xuất bản cung cấp nền tảng cho tiêu chuẩn tương lai.

Phiên bản xuất bản sau đó có thêm các chức năng tiêu bản, ngoại lệ, không gian tên, chuyển kiểu cho toán tử new, và kiểu boolean.

2.2. Các tính chất đặc trưng của ngôn ngữ

C++ là một ngôn ngữ lập trình bậc trung. Người sử dụng có thể sử dụng C++ để phát triển những ứng dụng bậc cao, và cả những chương trình bậc thấp hoạt động tốt trên phần cứng.

C++ là một ngôn ngữ lập trình hướng đối tượng. C++ được thiết kế với một cách tiếp cận hoàn toàn mới được gọi là lập trình hướng đối tượng, nơi mà người sử dụng

những đối tượng, các lớp và sử dụng các khái niệm như: thừa kế, đa hình, tính đóng gói, tính trừu tượng ...

C++ là một ngôn ngữ lập trình hướng cấu trúc giống ngôn ngữ C, nó có nghĩa là người sử dụng có thể tổ chức chương trình trên khái niệm functions.

C++ có thể chạy trên nhiều nền tảng khác nhau như Windows, Mac OS, một số biến thể của UNIX...

2.3. Cấu trúc cơ bản của ngôn ngữ C++

Một chương trình C++ cơ bản sẽ gồm 3 thành phần sau:

- ✓ Các câu lệnh và biểu thức (Statements and expressions).
- ✓ Hàm chính (Functions).
- ✓ Thư viện chuẩn C++ (Standard library)

2.3.1. Câu lệnh và biểu thức

a. Câu lệnh

- Câu lệnh trong C++ được thiết lập từ các từ khoá và các biểu thức ...
- Câu lệnh luôn luôn được kết thúc bằng dấu chấm phẩy ;
- Các câu lệnh được phép viết trên cùng một hoặc nhiều dòng.
- Câu lệnh gồm nhiều lệnh được bao bởi cặp dấu ngoặc {} và được gọi là khối lệnh.

b. Biểu thức

- Biểu thức là dãy kí hiệu kết hợp giữa các toán hạng, toán tử và cặp dấu () theo một qui tắc nhất định.
- Các toán hạng là hằng, biến, hàm.
- Biểu thức cung cấp cách thức tính giá trị mới dựa trên các toán hạng và toán tử trong biểu thức.

Ví dụ 2.1a:

```
int x;  
x = 1 + 2;  
cout << "Xin chao cac ban!";
```

Phân tích ví dụ:

Dòng 1: Câu lệnh khai báo biến có tên là x.

Dòng 2: Là câu lệnh gán giá trị cho biến x.

Ở đây biến x được gán giá trị bằng kết quả của biểu thức 1 + 2;

Dòng 3: Câu lệnh có nhiệm vụ đưa dữ liệu lên màn hình.

Ví dụ 2.1b:

```
(x + y) * 2 - 1 ;  
13 - x + sqrt(y) ;  
(-b + sqrt(delta)) / (2*a) ;
```

2.3.2. Hàm (Functions)

a. Định nghĩa: Hàm là một nhóm các câu lệnh được tập hợp lại để thực hiện một nhiệm vụ nào đó.

b. Cú pháp:

```
<kiểu kết quả> <tên hàm> (<tham số>) {  
    // nội dung thực hiện thuật toán  
}
```

Ví dụ 2.2:

```
int dientich(int dai, int rong) {  
    return dai * rong;  
}
```

Với bất cứ một chương trình C++ nào cũng đều phải có ít nhất là một hàm main().

2.3.3. Thư viện chuẩn C++ (C++ Standard Library)

Thư viện là một tập hợp các mã được biên dịch sẵn, được đóng gói lại để lập trình viên sử dụng, mà không cần phải viết lại. Ví dụ người sử dụng viết một chương trình tính toán, có thể include thư viện toán học...

Ví dụ 2.2:

```
#include <iostream>  
#include <math >  
#include <fstream>
```

Dòng 1: Khai báo thư viện có tên iostream. Thư viện này cung cấp cho người sử dụng chức năng nhập xuất dữ liệu cơ bản với chương trình.

Dòng 2: Khai báo thư viện toán học. Khi chúng ta cần dùng đến các phép toán ví dụ tính căn bậc 2 hay bình phương.....

Dòng 3: Khai báo thư viện fstream. Giúp người sử dụng có thể làm việc với các file nằm ngoài chương trình.

Thư viện là một phần không thể thiếu với người sử dụng. Có thư viện công việc của người sử dụng sẽ trở nên dễ dàng hơn.

Ví dụ 2.3: Một đoạn code đơn giản in ra “Ngon ngu lap trinh C++”.

```
// Chuong trinh don gian trong C++
#include <iostream>
using namespace std;
int main()
{
    cout << " Ngon ngu lap trinh C++"; // In dong chu Ngon ngu lap trinh C++
    return 0;
}
```

Chương trình trên có các phần sau:

- // Chuong trinh don gian trong C++
Đây là dòng chú thích. Tất cả các dòng bắt đầu bằng dấu //, dòng chú thích không có bất kì một ảnh hưởng nào đến hoạt động của chương trình. Người sử dụng dùng để giải thích bên trong mã nguồn của chương trình.
- Dòng **#include <iostream>**: đây là một thư viện vào ra chuẩn trong C++.
- Dòng **using namespace std**; khai báo sử dụng không gian tên std, định danh cout được định nghĩa trong không gian tên này.
- **int main()**: điểm bắt đầu quá trình thực hiện của các chương trình C++, tất cả các chương trình C++ đều có một hàm main.
- Theo sau **main** là một cặp ngoặc đơn bởi vì nó là một hàm. Trong C++, tất cả các hàm mà sau đó là một cặp ngoặc đơn () thì có nghĩa là nó có thể có hoặc không có tham số (không bắt buộc). Nội dung của hàm main tiếp ngay sau phần khai báo chính thức được bao trong các ngoặc nhọn { }.
- Dòng tiếp theo **cout << " Ngon ngu lap trinh C++"**; đây là một câu lệnh C++, làm nhiệm vụ in ra dòng chữ " Ngon ngu lap trinh C++".
- Dòng tiếp theo **return 0**; kết thúc hàm main() và trả về giá trị 0 cho hệ điều hành.

2.3.4. Các bước viết chương trình

a. Qui trình viết và thực hiện chương trình

Trước khi viết và chạy một chương trình thông thường chúng ta cần:

1. Xác định yêu cầu của chương trình. Nghĩa là xác định dữ liệu đầu vào (input) cung cấp cho chương trình và tập các dữ liệu cần đạt được tức đầu ra (output). Các tập hợp dữ liệu này ngoài các tên gọi còn cần xác định kiểu của nó.

Ví dụ để giải một phương trình bậc 2 dạng: $ax^2 + bx + c = 0$, cần báo cho chương trình biết dữ liệu đầu vào là a, b, c và đầu ra là nghiệm x1 và x2 của phương trình. Kiểu của a, b, c, x1, x2 là các số thực.

2. Xác định thuật toán giải.

3. Cụ thể hoá các khai báo kiểu và thuật toán thành dãy các lệnh, tức viết thành chương trình thông thường là trên giấy, sau đó bắt đầu soạn thảo vào trong máy. Quá trình này được gọi là soạn thảo chương trình nguồn.

4. Dịch chương trình nguồn để tìm và sửa các lỗi gọi là lỗi cú pháp.

5. Chạy chương trình, kiểm tra kết quả in ra trên màn hình. Nếu sai, sửa lại chương trình, dịch và chạy lại để kiểm tra. Quá trình này được thực hiện lặp đi lặp lại cho đến khi chương trình chạy tốt theo yêu cầu đề ra của người sử dụng.

b. Soạn thảo tệp chương trình nguồn

Gõ nội dung của chương trình vào trong máy và lưu lại nó lên đĩa.

Có thể soạn chương trình nguồn trên các bộ soạn thảo (editor) khác nhưng phải chạy trong môi trường tích hợp C++.

Mục đích của soạn thảo là tạo ra một văn bản chương trình và đưa vào bộ nhớ của máy. Các câu lệnh cần giống thẳng cột theo cấu trúc của lệnh. Các chú thích nên ghi ngắn gọn, rõ nghĩa.

c. Dịch chương trình

Sau khi đã soạn thảo xong chương trình nguồn, bước tiếp theo thường là dịch (ấn tổ hợp phím Alt-F9) để tìm và sửa các lỗi gọi là lỗi cú pháp.

Trong khi dịch C++ sẽ đặt con trỏ vào nơi gây lỗi (viết sai cú pháp) trong văn bản. Sau khi sửa xong một lỗi người sử dụng có thể dùng Alt-F8 để chuyển con trỏ đến lỗi tiếp theo hoặc dịch lại.

Để chuyển con trỏ về ngược lại lỗi trước đó có thể dùng Alt-F7.

Quá trình sửa lỗi, dịch được lặp lại cho đến khi văn bản đã được sửa hết lỗi cú pháp.

Sản phẩm sau khi dịch là một tệp mới gọi là chương trình đích có đuôi EXE tức là tệp mã máy để thực hiện.

Tệp này có thể lưu tạm thời trong bộ nhớ phục vụ cho quá trình chạy chương trình hoặc lưu lại trên đĩa tùy theo tùy chọn khi dịch của người sử dụng. Trong và sau khi dịch, C++ sẽ hiện một cửa sổ chứa thông báo về các lỗi (nếu có), hoặc thông báo chương trình đã được dịch thành công (không còn lỗi). Các lỗi này được gọi là lỗi cú pháp.

Sau khi đã soạn thảo xong chương trình nguồn, bước tiếp theo thường là dịch. Để dịch chương trình ta chọn menu \Compile\Compile hoặc \Compile\Make hoặc nhanh chóng hơn bằng cách ấn tổ hợp phím Alt-F9.

d. Chạy chương trình

Ấn Ctrl-F9 để chạy chương trình, nếu chương trình chưa dịch sang mã máy, máy sẽ tự động dịch lại trước khi chạy.

Kết quả của chương trình sẽ hiện ra trong một cửa sổ kết quả để người sử dụng kiểm tra. Nếu kết quả chưa được như mong muốn, quay lại văn bản để sửa và lại chạy lại chương trình.

Quá trình này được lặp lại cho đến khi chương trình chạy đúng như yêu cầu đã đề ra. Khi chương trình chạy, cửa sổ kết quả sẽ hiện ra tạm thời che khuất cửa sổ soạn thảo. Sau khi kết thúc chạy chương trình cửa sổ soạn thảo sẽ tự động hiện ra trở lại và che khuất cửa sổ kết quả.

Để xem lại kết quả đã hiện ấn Alt-F5. Sau khi xem xong để quay lại cửa sổ soạn thảo ấn phím bất kỳ.

Vi dụ 2.4: Viết chương trình nhập N , tính và in ra $\sqrt[3]{N}$.

```
#include <iostream>           // sử dụng thư viện vào/ra chuẩn
#include <math>                // sử dụng thư viện các hàm toán
using namespace std;         // sử dụng không gian tên chuẩn
int main() {                 // bắt đầu hàm main()
    double n, x;             // khai báo 2 biến n và x
    cout << "N = ";         // in ra màn hình "N = "
    cin >> n;                // nhập dữ liệu vào biến n
    x = sqrt(n);             // tính giá trị của x
    cout << "X = " << x;    // in ra màn hình: "X ="
}                             // kết thúc hàm main()
```

Vi dụ 2.5: Viết chương trình nhập x , tính $x*x$.

```
#include <iostream>           // sử dụng thư viện vào/ra chuẩn
#include <math>                // sử dụng thư viện các hàm toán
using namespace std;         // sử dụng không gian tên chuẩn
int main() {                 // bắt đầu hàm main()
    double x;                // khai báo x
    cin >> x;                // nhập x
    cout << x * x;           // tính  $x^2$  và in ra màn hình
    return 0;                // trả về 0 cho hệ thống
}                             // kết thúc hàm main()
```

Ví dụ 2.6: Viết chương trình tính diện tích và chu vi của một hình chữ nhật. Để chương trình có thể tính với các bộ giá trị khác nhau của chiều dài và rộng ta cần lưu giá trị này vào trong các biến (ví dụ cd, cr).

```
#include <iostream.h> // khai báo tệp nguyên mẫu để dùng được cin, cout
int main() // đây là hàm chính của chương trình
{
float cd, cr ; // khai báo các biến có tên cd, cr để chứa độ dài các cạnh
cout << "Hãy nhập chiều dài: " ; cin >> cd ; // nhập dữ liệu
cout << "Hãy nhập chiều rộng: " ; cin >> cr ;
cout << "Diện tích = " << cd * cr << '\n' ; // in kết quả
cout << "Chu vi = " << 2 * (cd + cr) << '\n';
return 0 ;
}
```

Gõ chương trình này và chạy. Khi chạy đến câu lệnh nhập, chương trình dừng để chờ nhận chiều dài và chiều rộng, người sử dụng nhập các giá trị cụ thể, chương trình sẽ tiếp tục thực hiện và in ra kết quả.

Thông qua câu lệnh nhập dữ liệu và 2 biến cd, cr người sử dụng có thể yêu cầu chương trình cho kết quả của một hình chữ nhật bất kỳ chứ không chỉ trong trường hợp hình có chiều dài 23 và chiều rộng 11 như trong ví dụ cụ thể trên.

2.4. Môi trường làm việc của C++

2.4.1. Khởi động - thoát khỏi C++

Khởi động C++ bằng cách nhấp đúp chuột lên biểu tượng của chương trình.

Khi chương trình được khởi động sẽ hiện ra giao diện gồm có menu công việc và một khung cửa sổ bên dưới phục vụ cho soạn thảo.

Một con trỏ nhấp nháy trong khung cửa sổ và chúng ta bắt đầu nhập nội dung (văn bản) chương trình vào trong khung cửa sổ soạn thảo này.

Ngôn ngữ lập trình C++ cho các sinh viên mới bắt đầu nên giáo trình chọn trình bày giao diện của các trình biên dịch quen thuộc là Turbo C++, hoặc Cfree5.0.

Để kết thúc làm việc với C++ (soạn thảo, chạy chương trình ...) và quay về môi trường Windows ấn Alt-X.

2.4.2. Giao diện và cửa sổ soạn thảo

Khi gọi chạy C++ trên màn hình sẽ xuất hiện một menu xổ xuống và một cửa sổ soạn thảo.

Trên menu gồm có các nhóm chức năng: File, Edit, Search, Run, Compile, Debug, Project, Options, Window, Help.

Để kích hoạt các nhóm chức năng, có thể ấn Alt+chữ cái biểu thị cho menu của chức năng đó (là chữ cái có gạch dưới). Ví dụ để mở nhóm chức năng File ấn Alt+F, sau đó dịch chuyển hộp sáng đến mục cần chọn rồi ấn Enter.

Để thuận tiện cho người sử dụng, một số các chức năng hay dùng còn được gắn với một tổ hợp các phím cho phép người dùng có thể chọn nhanh chức năng này mà không cần thông qua việc mở menu như đã mô tả ở trên.

Một số tổ hợp phím cụ thể đó sẽ được trình bày vào cuối phần này. Các bộ chương trình dịch hỗ trợ người lập trình một môi trường tích hợp tức ngoài chức năng soạn thảo, nó còn cung cấp nhiều chức năng, tiện ích khác giúp người lập trình vừa có thể soạn thảo văn bản chương trình vừa gọi chạy chương trình vừa gỡ lỗi ...

a. Các chức năng soạn thảo

Giống hầu hết các bộ soạn thảo văn bản, bộ soạn thảo của Turbo C hoặc Borland C cũng sử dụng các phím sau cho quá trình soạn thảo:

– **Dịch chuyển con trỏ:** các phím mũi tên cho phép dịch chuyển con trỏ sang trái, phải một kí tự hoặc lên trên, xuống dưới 1 dòng.

Để dịch chuyển nhanh có các phím như Home (về đầu dòng), End (về cuối dòng), PgUp, PgDn (lên, xuống một trang màn hình).

Để dịch chuyển xa hơn có thể kết hợp các phím này cùng phím Control (Ctrl, ^) như ^PgUp: về đầu tệp, ^PgDn: về cuối tệp.

– **Chèn, xoá, sửa:** Phím Insert cho phép chuyển chế độ soạn thảo giữa chèn và đè. Các phím Delete, Backspace cho phép xoá một kí tự tại vị trí con trỏ và trước vị trí con trỏ (xoá lùi).

– **Các thao tác với khối dòng:** Để đánh dấu khối dòng (thực chất là khối kí tự liền nhau bất kỳ) ta đưa con trỏ đến vị trí đầu ấn Ctrl-KB và Ctrl-KK tại vị trí cuối.

Cũng có thể thao tác nhanh hơn bằng cách giữ phím Shift và dùng các phím dịch chuyển con trỏ quét từ vị trí đầu đến vị trí cuối, khi đó khối kí tự được đánh dấu sẽ chuyển màu nền. Một khối được đánh dấu có thể dùng để cắt, dán vào một nơi khác trong văn bản hoặc xoá khỏi văn bản.

Để thực hiện thao tác cắt dán, đầu tiên phải đưa khối đã đánh dấu vào bộ nhớ đệm bằng nhóm phím Shift-Delete (cắt), sau đó dịch chuyển con trỏ đến vị trí mới cần hiện nội dung vừa cắt và ấn tổ hợp phím Shift-Insert.

Một đoạn văn bản được ghi vào bộ nhớ đệm có thể được dán nhiều lần vào nhiều vị trí khác nhau bằng cách lặp lại tổ hợp phím Shift-Insert tại các vị trí khác nhau trong văn bản. Để xoá một khối dòng đã đánh dấu mà không ghi vào bộ nhớ đệm, dùng tổ hợp phím Ctrl-Delete.

Khi một nội dung mới ghi vào bộ nhớ đệm thì nó sẽ xoá (ghi đè) nội dung cũ đã có, do vậy cần cân nhắc để sử dụng phím Ctrl-Delete (xoá và không lưu lại nội dung vừa xoá vào bộ đệm) và Shift-Delete (xoá và lưu lại nội dung vừa xoá) một cách phù hợp.

– Tổ hợp phím Ctrl-A rất thuận lợi khi cần đánh dấu nhanh toàn bộ văn bản.

b. Chức năng tìm kiếm và thay thế

Chức năng này dùng để dịch chuyển nhanh con trỏ văn bản đến từ cần tìm.

Để thực hiện tìm kiếm bấm Ctrl-QF, tìm kiếm và thay thế bấm Ctrl-QA.

Vào từ hoặc nhóm từ cần tìm vào cửa sổ Find, nhóm thay thế (nếu dùng Ctrl-QA) vào cửa sổ Replace và đánh dấu vào các tùy chọn trong cửa sổ bên dưới sau đó ấn Enter.

Các tùy chọn gồm: không phân biệt chữ hoa/thường, tìm từ độc lập hay đứng trong từ khác, tìm trong toàn văn bản hay chỉ trong phần được đánh dấu, chiều tìm đi đến cuối hay ngược về đầu văn bản, thay thế có hỏi lại hay không hỏi lại ... Để dịch chuyển con trỏ đến các vùng khác nhau trong một menu hay cửa sổ chứa các tùy chọn ta sử dụng phím Tab.

c. Các chức năng liên quan đến tệp

- **Ghi tệp lên đĩa:** Chọn menu File\Save hoặc phím F2. Nếu tên tệp chưa có (còn mang tên Noname.cpp) máy sẽ yêu cầu cho tên tệp. Phần mở rộng của tên tệp được mặc định là CPP.

- **Soạn thảo tệp mới:** Chọn menu File\New. Hiện ra cửa sổ soạn thảo và tên file tạm thời lấy là Noname.cpp.

- **Soạn thảo tệp cũ:** Chọn menu File\Open hoặc ấn phím F3, nhập tên tệp hoặc dịch chuyển con trỏ trong vùng danh sách tệp bên dưới đến tên tệp cần soạn rồi ấn Enter. Cũng có thể áp dụng cách này để soạn tệp mới khi không nhập vào tên tệp cụ thể.

- **Lưu tệp** đang soạn thảo lên đĩa với tên mới: Chọn menu File\Save As và nhập tên tệp mới, ấn Enter.

d. Chức năng dịch và chạy chương trình

- Ctrl-F9: Khởi động chức năng dịch và chạy toàn bộ chương trình.

- F4: Chạy chương trình từ đầu đến dòng lệnh hiện tại (đang chứa con trỏ)

- F7: Chạy từng lệnh một của hàm main(), kể cả các lệnh con trong hàm.

- F8: Chạy từng lệnh một của hàm main(). Khi đó mỗi lời gọi hàm được xem là một lệnh (không chạy từng lệnh trong các hàm được gọi).

Các chức năng liên quan đến dịch chương trình có thể được chọn thông qua menu Compile (Alt-C).

- Để dịch chương trình ta chọn menu \Compile\Compile hoặc \Compile\Make hoặc nhanh chóng hơn bằng cách ấn tổ hợp phím Alt-F9.

- Ấn Ctrl-F9 để chạy chương trình, nếu chương trình chưa dịch sang mã máy, máy sẽ tự động dịch lại trước khi chạy. Kết quả của chương trình sẽ hiện ra trong một cửa sổ kết quả để người sử dụng kiểm tra.

e. Tóm tắt một số phím nóng hay dùng

– Các phím kích hoạt menu: Alt+chữ cái đại diện cho nhóm menu đó.

Ví dụ Alt-F mở menu File để chọn các chức năng cụ thể trong nó như Open (mở file), Save (ghi file lên đĩa), Print (in nội dung văn bản chương trình ra máy in), ... Alt-C mở menu Compile để chọn các chức năng dịch chương trình.

- Các phím dịch chuyên con trỏ khi soạn thảo.
- F1: mở cửa sổ trợ giúp. Đây là chức năng quan trọng giúp người lập trình nhớ tên lệnh, cú pháp và cách sử dụng.
- F2: ghi tệp lên đĩa.
- F3: mở tệp cũ ra sửa chữa hoặc soạn thảo tệp mới.
- F4: chạy chương trình đến vị trí con trỏ.
- F5: Thu hẹp/mở rộng cửa sổ soạn thảo.
- F6: Chuyển đổi giữa các cửa sổ soạn thảo.
- F7: Chạy chương trình theo từng lệnh, kể cả các lệnh trong hàm con.
- F8: Chạy chương trình theo từng lệnh trong hàm chính.
- F9: Dịch và liên kết chương trình. Thường dùng chức năng này để tìm lỗi cú pháp của chương trình nguồn trước khi chạy.
- Alt-F7: Chuyển con trỏ về nơi gây lỗi trước đó.
- Alt-F8: Chuyển con trỏ đến lỗi tiếp theo.
- Ctrl-F9: Chạy chương trình.
- Ctrl-Insert: Lưu khối văn bản được đánh dấu vào bộ nhớ đệm.
- Shift-Insert: Dán khối văn bản trong bộ nhớ đệm vào văn bản tại vị trí con trỏ.
- Shift-Delete: Xoá khối văn bản được đánh dấu, lưu nó vào bộ nhớ đệm.
- Ctrl-Delete: Xoá khối văn bản được đánh dấu (không lưu vào bộ nhớ đệm).
- Alt-F5: Chuyển sang cửa sổ xem kết quả của chương trình vừa chạy xong.
- Alt-X: thoát C++ về lại Windows.

Câu hỏi thảo luận

1. Hãy khai báo một số thư viện chuẩn: thư viện cung cấp chức năng nhập xuất dữ liệu, thư viện toán học, thư viện làm việc với các file.
2. Trình bày một số thao tác soạn thảo cơ bản trong môi trường Turbo C++: chèn, xoá, sửa.
3. Trình bày một số thao tác làm việc với file trong Turbo C++: mở file, lưu file, xoá file.

Bài tập vận dụng

1. Cho đoạn chương trình sau, nêu ý nghĩa của các dòng lệnh 1, 2, 3, 4.

```
// Chương trình tính + - * /
```

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5. int a, b;
6. cout<<"Ban hay nhap 2 so nguyen: ";
7. cin>>a>>b;
8. cout<<"Tong cua 2 so vua nhap la: "<<a+b<<"\n";
9. cout<<"Hieu cua 2 so vua nhap la: "<<a-b<<"\n";
10. cout<<"Tich cua 2 so vua nhap la: "<<a*b<<"\n";
11. if (b!=0)
12. cout<<"Thuong cua 2 so vua nhap la: "<<a/b;
13. return 0;
14. }
```

2. Viết chương trình xuất ra đoạn thông báo:

“Đây là chương trình C++ đầu tiên.

Vui lòng nhấn phím Enter để kết thúc.”

3. Viết chương trình nhập xuất ra màn hình thông tin của mỗi sinh viên theo mẫu:

Ho ten: So dien thoai:

Ma so sinh vien: Gioi tinh:

Lop:

4. Viết chương trình nhập vào 1 số nguyên, xuất ra màn hình số nguyên vừa nhập.

5. Viết chương trình nhập vào 2 số nguyên, tính và xuất kết quả tổng, tích 2 số nguyên vừa nhập.

6. Viết chương trình nhập vào 2 số nguyên a, b. Xuất kết quả khi a chia cho b.

7. Viết chương trình cho phép nhập vào giờ, phút và giây, hãy đổi sang giây và in kết quả ra màn hình.

8. Tìm lỗi sai trong chương trình sau

```
/ Chuong trinh 3
include <iostream
usingnamespace std
int main {
cout << "Bài 2";
cout << "Cac khai niem co ban trong C++";
return 0;
```

9. Cho chương trình sau:

```
#include <iostream>
using namespace std;
int main()
{
cout << "Xin chao cac ban!" << endl;
}
```

a. Lưu chương trình thành file “xinchao.cpp”

b. Dịch và chạy chương trình (sửa lỗi nếu có).

10. Nhập điểm toán, điểm triết và điểm anh của một sinh viên. In ra điểm trung bình 3 môn học của học sinh đó.

CHƯƠNG 3: CÁC YẾU TỐ CƠ BẢN CỦA NGÔN NGỮ C++

Mục tiêu của chương

Nắm vững:

- Cách khai báo biến, khai báo hằng, cách đặt tên định danh, thứ tự ưu tiên của các phép toán.
- Các từ khoá, các kiểu dữ liệu cơ bản.
- Các lệnh trình bày màn hình trong ngôn ngữ C++.

Nội dung của chương

Nghiên cứu các kiểu dữ liệu cơ bản của ngôn ngữ C++, các lệnh nhập xuất dữ liệu cơ bản, viết code và chạy chương trình.

3.1. Các yếu tố cơ bản

3.1.1. Tập ký tự dùng trong ngôn ngữ C++

Bảng kí tự được phép dùng để tạo nên những câu lệnh của ngôn ngữ C++.

- Các kí tự chữ hoa: A, B, C, . . . , Z.
- Các kí tự chữ thường: a, b, c, . . . , z.
- Các chữ số: 0, 1, . . . , 9.
- Các kí tự dấu: , . ! ? : . . .
- Các kí tự trắng: ENTER, BACKSPACE, khoảng trắng.
- Các ký hiệu toán học: +, -, *, /, % , &, ||, !, >, <, =
- Các kí tự đặc biệt khác: ; :: [], {}, #, ...

3.1.2. Từ khoá (keyword)

Từ khoá là từ được qui định trước trong ngôn ngữ lập trình, mỗi từ có một ý nghĩa nhất định. Mọi ngôn ngữ lập trình đều có một bộ từ khóa riêng.

Người sử dụng có thể tạo ra những từ mới để chỉ các đối tượng của mình nhưng không được phép trùng với từ khoá.

Một đặc trưng của C++ là các từ khoá luôn luôn được viết bằng chữ thường.

Một số từ khóa thường gặp: auto, break, case, char, continue, default, do, double, else, externe, float, for, goto, if, int, long, register, return, short, sizeof, static, struct, switch, short, signed, sizeof, typedef, union, unsigned, while, namespace, new, operator, private, protected, public...

3.1.3. Định danh (identifier)

Dùng để đặt tên cho các đối tượng như hằng, biến, hàm, . .

Tên gọi do người sử dụng tự đặt phải tuân theo một số qui tắc sau:

- Là một dãy liên tiếp các chữ cái, chữ số và dấu gạch dưới (không chứa dấu cách).

- Phân biệt kí tự in hoa và thường.
- Không được trùng với từ khóa.
- Chiều dài của tên không bị giới hạn.

Ví dụ 3.1:

Các tên đúng: i, i1, j, delta, PT_Bac_2.

Các tên sai: Bai tap, 3abc, case.

Các tên sau đây là khác nhau: nam_dinh, Nam_dinh, NAM_DINH.

3.1.4. Chú thích trong chương trình

Vai trò của đoạn chú thích là làm cho chương trình dễ hiểu đối với người đọc, vì vậy đối với máy các đoạn chú thích sẽ được bỏ qua.

Có 2 loại chú thích:

// Chú thích theo dòng

/* Chú thích theo khối */

Chú thích theo dòng bắt đầu từ cặp dấu // cho đến cuối dòng.

Chú thích theo khối bắt đầu bằng /* và kết thúc bằng */ và có thể bao gồm nhiều dòng.

Ví dụ 3.2: Trong chương trình sau sử dụng ký hiệu chú thích:

```
#include <iostream.h>
using namespace std;
int main()
{
    int j;
    for (j = 0; j < 5 ; ++ j) // 0 - 5
        cout<<j<<"\n"; // In ra j
    return 0;
}
```

3.1.5. Các kiểu dữ liệu cơ bản

Khái niệm về kiểu dữ liệu: Thông thường dữ liệu hay dùng là số và chữ.

Tuy nhiên việc phân chia chỉ 2 loại dữ liệu là không đủ. Để dễ dàng hơn cho lập trình, hầu hết các ngôn ngữ lập trình đều phân chia dữ liệu thành nhiều kiểu khác nhau được gọi là các kiểu cơ bản hay chuẩn.

Trên cơ sở kết hợp các kiểu dữ liệu chuẩn, người sử dụng có thể tự đặt ra các kiểu dữ liệu mới để phục vụ cho chương trình giải quyết bài toán của mình.

Có nghĩa lúc đó mỗi đối tượng được quản lý trong chương trình sẽ là một tập hợp nhiều thông tin hơn và được tạo thành từ nhiều loại (kiểu) dữ liệu khác nhau. Dưới đây chúng ta sẽ xét đến một số kiểu dữ liệu chuẩn được qui định sẵn bởi C++.

Một biến như đã biết là một số ô nhớ liên tiếp nào đó trong bộ nhớ dùng để lưu trữ dữ liệu (vào, ra hay kết quả trung gian) trong quá trình hoạt động của chương trình.

Để quản lý chặt chẽ các biến, người sử dụng cần khai báo cho chương trình biết trước tên biến và kiểu của dữ liệu được chứa trong biến.

Việc khai báo này sẽ làm chương trình quản lý các biến dễ dàng hơn như trong việc phân bố bộ nhớ cũng như quản lý các tính toán trên biến theo nguyên tắc: chỉ có các dữ liệu cùng kiểu với nhau mới được phép làm toán với nhau. Do đó, khi đề cập đến một kiểu chuẩn của một ngôn ngữ lập trình, thông thường chúng ta sẽ xét đến các yếu tố sau:

- Tên kiểu: là một từ dành riêng để chỉ định kiểu của dữ liệu.
- Số byte trong bộ nhớ để lưu trữ một đơn vị dữ liệu thuộc kiểu này: Thông thường số byte này phụ thuộc vào các trình biên dịch và hệ thống máy khác nhau, ở đây ta chỉ xét đến hệ thống máy PC thông dụng hiện nay.
- Miền giá trị của kiểu: Cho biết một đơn vị dữ liệu thuộc kiểu này sẽ có thể lấy giá trị trong miền nào, ví dụ nhỏ nhất và lớn nhất là bao nhiêu.

Các giá trị này phụ thuộc vào số byte mà hệ thống máy qui định cho từng kiểu. người sử dụng cần nhớ đến miền giá trị này để khai báo kiểu cho các biến cần sử dụng một cách thích hợp.

a. Số nguyên

Loại dữ liệu	Tên kiểu	Số ô nhớ	Miền giá trị
Số nguyên	int (số nguyên)	2 byte	- 32768 .. 32767
	unsigned int (số nguyên không dấu)	2 byte	0 .. 65535
	short (số nguyên ngắn)	2 byte	- 32768 .. 32767
	long (số nguyên dài)	4 byte	- 2 ¹⁵ .. 2 ¹⁵ - 1

Các phép toán áp dụng được trên kiểu số nguyên:

- Các phép toán số học: +, -, *, /, %
- Các phép toán so sánh: <, <=, >, >=, ==, !=
- Các phép toán dịch chuyển số học: >>, và <<
- Các phép toán trên bit: ~ (not bit), & (and bit), | (or bit), ^ (xor bit).
- Các phép toán đặc biệt: tăng 1 đơn vị (++), giảm 1 đơn vị (--).

b. Số thực

Để sử dụng số thực ta cần khai báo kiểu float hoặc double mà miền giá trị của chúng được cho trong bảng. Các giá trị số kiểu double được gọi là số thực với độ chính xác gấp đôi vì với kiểu dữ liệu này máy tính có cách biểu diễn khác so với kiểu float để đảm bảo số số lẻ sau một số thực có thể tăng lên đảm bảo tính chính xác cao hơn so với số kiểu float.

Tuy nhiên, trong các bài toán thông dụng thường ngày độ chính xác của số kiểu float là đủ dùng. Việc in ấn số thực ta có một vài cách thiết đặt dạng in theo ý muốn, ví dụ độ rộng tối thiểu để in một số hay số số lẻ thập phân cần in ...vv

Loại dữ liệu	Tên kiểu	Số ô nhớ	Miền giá trị
Số thực	float	4 byte	$\pm 10^{-37} \dots \pm 10^{+38}$
	double	8 byte	$\pm 10^{-307} \dots \pm 10^{+308}$

Các phép toán áp dụng được trên kiểu số thực:

- Các phép toán số học: +, -, *, / (không có phép toán %)
- Các phép toán so sánh: <, <=, >, >=, ==, !=.
- Nhiều hàm toán học bổ sung (khai báo thư viện <cmath>): fabs, sqrt, pow, floor/ceil, exp, log, log10,...

Ví dụ 3.3a: Chương trình sau đây sẽ in diện tích và chu vi của một hình tròn có bán kính 2cm với 3 số lẻ.

```
#include <iostream.h>
#include <iomanip.h>
int main()
{
float r = 2 ; // r là tên biến dùng để chứa bán kính
cout << "Diện tích = " << setiosflags(ios::showpoint) ;
cout << setprecision(3) << r * r * 3.1416 ;
getch() ;
}
```

c. Ký tự

Một ký tự là một ký hiệu trong bảng mã ASCII. Như đã biết một số ký tự có mặt chữ trên bàn phím (ví dụ các chữ cái, chữ số) trong khi một số ký tự lại không (ví dụ ký tự biểu diễn việc lùi lại một ô trong văn bản, ký tự chỉ việc kết thúc một dòng hay kết thúc một văn bản).

Do vậy để biểu diễn một kí tự người ta dùng chính mã ASCII của kí tự đó trong bảng mã ASCII và thường gọi là giá trị của kí tự.

Ví dụ phát biểu "Cho kí tự 'A'" là cũng tương đương với phát biểu "Cho kí tự 65" (65 là mã ASCII của kí tự 'A'), hoặc "Xoá kí tự xuống dòng" là cũng tương đương với phát biểu "Xoá kí tự 13" vì 13 là mã ASCII của kí tự xuống dòng.

Như vậy một biến kiểu kí tự có thể được nhận giá trị theo 2 cách tương đương - chữ hoặc giá trị số:

Ví dụ giả sử c là một biến kí tự thì câu lệnh gán `c = 'A'` cũng tương đương với câu lệnh gán `c = 65`. Tuy nhiên để sử dụng giá trị số của một kí tự c nào đó ta phải yêu cầu đổi c sang giá trị số bằng câu lệnh `int(c)`.

Loại dữ liệu	Tên kiểu	Số ô nhớ	Miền giá trị
Kí tự	char	1 byte	- 128 .. 127
	unsigned char	1 byte	0 .. 255

Các phép toán áp dụng được trên kiểu char:

- Các phép toán số học: +, -, *, /, %
- Các phép toán so sánh: <, <=, >, >=, ==, !=.

Ví dụ 3.3b:

```
char c, d ; // c, d được phép gán giá trị từ -128 đến 127
unsigned e ; // e được phép gán giá trị từ 0 đến 255
c = 65 ; d = 179 ; // d có giá trị ngoài miền cho phép
e = 179; f = 330 ; // f có giá trị ngoài miền cho phép
cout << c << int(c) ; // in ra chữ cái 'A' và giá trị số 65
cout << d << int(d) ; // in ra là kí tự '|' và giá trị số -77
cout << e << int(e) // in ra là kí tự '|' và giá trị số 179
cout << f << int(f) // in ra là kí tự 'J' và giá trị số 74
```

Chú ý: Qua ví dụ trên ta thấy một biến nếu được gán giá trị ngoài miền cho phép sẽ dẫn đến kết quả không theo suy nghĩ thông thường.

Do vậy nên tuân thủ qui tắc chỉ gán giá trị cho biến thuộc miền giá trị mà kiểu của biến đó qui định.

Ví dụ nếu muốn sử dụng biến có giá trị từ 128 .. 255 ta nên khai báo biến dưới dạng kí tự không dấu (unsigned char), còn nếu giá trị vượt quá 255 ta nên chuyển sang kiểu nguyên (int) chẳng hạn.

d. Logic

Loại dữ liệu	Tên kiểu	Số ô nhớ	Miền giá trị
logic	bool	1 byte	True/False

Lưu trữ các giá trị đúng/sai (true/false)

Các phép toán áp dụng được trên kiểu logic:

- Là kết quả của các phép so sánh: >, >=, <, <=, ==, !=.
- Các phép toán logic: và (&&), hoặc (||), đảo (!), xor (^).

3.1.6. Khai báo biến

Biến là các tên gọi để lưu giá trị khi làm việc trong chương trình.

Các giá trị được lưu có thể là các giá trị dữ liệu ban đầu, các giá trị trung gian tạm thời trong quá trình tính toán hoặc các giá trị kết quả cuối cùng.

Khác với hằng, giá trị của biến có thể thay đổi trong quá trình làm việc bằng các lệnh đọc vào từ bàn phím hoặc gán. Hình ảnh cụ thể của biến là một số ô nhớ trong bộ nhớ được sử dụng để lưu các giá trị của biến.

Mọi biến phải được khai báo trước khi sử dụng. Một khai báo như vậy sẽ báo cho chương trình biết về một biến mới gồm có: tên của biến, kiểu của biến (tức kiểu của giá trị dữ liệu mà biến sẽ lưu giữ).

Thông thường với nhiều ngôn ngữ lập trình tất cả các biến phải được khai báo ngay từ đầu chương trình hay đầu của hàm, tuy nhiên để thuận tiện C++ cho phép khai báo biến ngay bên trong chương trình hoặc hàm, có nghĩa bất kỳ lúc nào người sử dụng thấy cần thiết sử dụng biến mới, họ có quyền khai báo và sử dụng nó từ đó trở đi.

Cú pháp khai báo biến gồm tên kiểu, tên biến và có thể có hay không khởi tạo giá trị ban đầu cho biến.

Cách đặt tên biến:

- Là một dãy liên tiếp các chữ cái, chữ số và dấu gạch dưới (không chứa dấu cách).
- Phân biệt kí tự in hoa và thường.
- Không được trùng với từ khóa.
- Chiều dài của tên không bị giới hạn.

Cú pháp khai báo biến gồm tên kiểu, tên biến và có thể có hay không khởi tạo giá trị ban đầu cho biến. Để khởi tạo hoặc thay đổi giá trị của biến ta dùng lệnh gán (=).

Cách khai báo biến:

tên_kiểu tên_biến;

Trong đó:

tên_kiểu: là một kiểu dữ liệu hợp lệ bất kỳ có trong C++.

tên_biến: là một hay nhiều biến, được phân cách bằng dấu ‘,’.

Có thể khai báo và đồng thời khởi tạo giá trị cho biến như sau:

<tên kiểu> <tên biến> = <giá trị>;

Chú ý: lệnh khai báo biến luôn được kết thúc bằng dấu ‘;’.

Ví dụ 3.3c: Khai báo biến

```
char ch; // khai báo biến kí tự
int tong = 10, a = 5, b = 7;
int a, b; // khai báo 2 biến kiểu nguyên
```

hoặc

```
int a;
int b;
```

Ví dụ 3.3.d:

```
const int n = 10 ;
int main ()
{
int i = 2, j , k = n + 5; // khai báo i và khởi tạo bằng 2, k bằng 15
float eps = 1.0e-6 ; // khai báo biến thực epsilon khởi tạo bằng 10-6
char c = 'Z'; // khai báo biến kí tự c và khởi tạo bằng ' Z '
char d[100] = "Toán học"; //khai báo xâu kí tự d chứa dòng chữ "Toán học"
...
}
```

3.1.7. Khai báo hằng

Là đại lượng có giá trị không thay đổi trong suốt thời gian tồn tại của nó.

Tên hằng phải là một danh hiệu hợp lệ và phải được khai báo trước khi sử dụng.

a. Hằng nguyên

- Kiểu short, int: 3, -7, ...
- Kiểu unsigned: 3, 123456, ...
- Kiểu long, long int: 3L, -7L, 123456L, ... (viết L vào cuối mỗi giá trị)

Các cách viết trên là thể hiện của số nguyên trong hệ thập phân, ngoài ra chúng còn được viết dưới các hệ đếm khác như hệ cơ số 8 hoặc hệ cơ số 16.

Hằng số nguyên: có thể được biểu diễn dưới dạng thập phân, nhị phân, bát phân, và thập lục phân.

b. Hằng số thực: có thể được biểu diễn dưới dạng kí pháp thông thường hoặc dạng kí pháp khoa học.

- Kí pháp thông thường (còn gọi là số thực dấu phẩy tĩnh) gồm 2 phần, được phân cách bởi dấu chấm thập phân.

Ví dụ 3.4a: 1234.56

- Kí pháp khoa học (còn gọi là số thực dấu phẩy động) gồm phần định trị (là một số thực) và phần mũ (là một số nguyên). Hai phần này được phân cách bởi chữ e hoặc E.

Ví dụ 3.4b: 1.2345 E+03.

c. Hằng kí tự

Có 2 cách để viết một hằng kí tự. Đối với các kí tự có mặt chữ thể hiện ta thường sử dụng cách viết thông dụng đó là đặt mặt chữ đó giữa 2 dấu nháy đơn như: 'A', '3', '' (dấu cách) ... hoặc sử dụng trực tiếp giá trị số của chúng.

Một số kí tự đặc biệt:

`\n` : biểu thị kí tự xuống dòng (cũng tương đương với endl)

`\t` : kí tự tab

`\a` : kí tự chuông (tiếng bíp)

`\b` : kí tự dành cho khoảng trống

`\r` : xuống dòng

`\f` : kéo trang

`\\` : dấu \

`\?` : dấu chấm hỏi ?

`\'` : dấu nháy đơn '

`\''` : dấu nháy kép "

`\kkk` : kí tự có mã là kkk trong hệ 8

`\xkk` : kí tự có mã là kk trong hệ 16

Khai báo hằng:

Một giá trị cố định (hằng) được sử dụng nhiều lần trong chương trình đôi khi sẽ thuận lợi hơn nếu ta đặt cho nó một tên gọi, thao tác này được gọi là khai báo hằng.

Ví dụ một chương trình quản lý sinh viên với giả thiết số sinh viên tối đa là 50. Nếu số sinh viên tối đa không thay đổi trong chương trình ta có thể đặt cho nó một tên gọi như sosv chẳng hạn. Trong suốt chương trình bất kỳ chỗ nào xuất hiện giá trị 50 ta đều có thể thay nó bằng sosv.

Tương tự C++ cũng có những tên hằng được đặt sẵn, được gọi là các hằng chuẩn và người sử dụng có thể sử dụng khi cần thiết. Ví dụ hằng π được đặt sẵn trong C++ với tên gọi PI.

Việc sử dụng tên hằng thay cho hằng có nhiều điểm thuận lợi như sau:

- Chương trình dễ đọc hơn, vì thay cho các con số ít có ý nghĩa, một tên gọi sẽ làm NSD dễ hình dung vai trò, nội dung của nó.

Ví dụ, khi gặp tên gọi sosv người sử dụng sẽ hình dung được chẳng hạn, "đây là số sinh viên tối đa trong một lớp", trong khi số 50 có thể là số sinh viên mà cũng có thể là tuổi của một sinh viên nào đó.

- Chương trình dễ sửa chữa hơn, ví dụ bây giờ nếu muốn thay đổi chương trình sao cho bài toán quản lý được thực hiện với số sinh viên tối đa là 60, khi đó ta cần tìm và thay thế hàng trăm vị trí xuất hiện của 50 thành 60.

Việc thay thế như vậy dễ gây ra lỗi vì có thể không tìm thấy hết các số 50 trong chương trình hoặc thay nhầm số 50 với ý nghĩa khác như tuổi của một sinh viên nào đó chẳng hạn. Nếu trong chương trình sử dụng hằng sosv, bây giờ việc thay thế trở nên chính xác và dễ dàng hơn bằng thao tác khai báo lại giá trị hằng sosv bằng 60. Lúc đó trong chương trình bất kỳ nơi nào gặp tên hằng sosv đều được chương trình hiểu với giá trị 60.

Để khai báo hằng ta dùng các câu khai báo sau:

```
#define tên_hằng giá_trị_hằng ;
```

hoặc:

```
const tên_hằng = giá_trị_hằng ;
```

Ví dụ 3.5a:

```
#define siso 50 ;
```

```
const siso = 50 ;
```

```
const pi=3.14;
```

Chú ý một giá trị hằng chưa nói lên kiểu sử dụng của nó vì vậy ta cần khai báo rõ ràng hơn bằng cách thêm tên kiểu trước tên hằng trong khai báo const, các hằng khai báo như vậy được gọi là hằng có kiểu.

Ví dụ 3.5b:

```
const int siso = 50 ;
```

```
const float diemtoan = 7.5 ;
```

3.2. Biểu thức và các toán tử

3.2.1. Biểu thức

Biểu thức là dãy kí hiệu kết hợp giữa các toán hạng, phép toán và cặp dấu () theo một qui tắc nhất định. Các toán hạng có thể là một biến, hằng, lời gọi hàm.

Biểu thức cung cấp một cách thức để tính giá trị mới dựa trên các toán hạng và toán tử trong biểu thức.

Có các loại biểu thức thông dụng sau: biểu thức gán, biểu thức số học, biểu thức logic.

a. Thứ tự ưu tiên của các phép toán

Ngôn ngữ C++ qui định trật tự tính toán theo các mức độ ưu tiên như sau:

1. Các biểu thức trong cặp dấu ngoặc ().
2. Các phép toán 1 ngôi (tự tăng, giảm, lấy địa chỉ, lấy nội dung con trỏ ...).
3. Các phép toán số học.
4. Các phép toán quan hệ, logic.
5. Các phép gán.

b. Phép chuyển đổi kiểu

Khi tính toán một biểu thức phần lớn các phép toán đều yêu cầu các toán hạng phải cùng kiểu.

- *Chuyển kiểu tự động*: về mặt nguyên tắc, khi cần thiết các kiểu có giá trị thấp sẽ được chương trình tự động chuyển lên kiểu cao hơn cho phù hợp với phép toán. Cụ thể phép chuyển kiểu có thể được thực hiện theo sơ đồ như sau:

char \leftrightarrow int \rightarrow long int \rightarrow float \rightarrow double

Ví dụ 3.6a:

```
int i = 10;
float f ;
f = i + 5;
```

Trong ví dụ trên i có kiểu nguyên và vì vậy i+5 cũng có kiểu nguyên trong khi f có kiểu thực. Tuy vậy phép toán gán này là hợp lệ vì chương trình sẽ tự động chuyển kiểu của i+5 (bằng 15) sang kiểu thực (bằng 15.0) rồi mới gán cho f.

- *Ép kiểu*: trong chuyển kiểu tự động, chương trình chuyển các kiểu từ thấp đến cao, tuy nhiên chiều ngược lại không thể thực hiện được vì nó có thể gây mất dữ liệu. Do đó nếu cần thiết người sử dụng phải ra lệnh cho chương trình.

Ví dụ 3.6b:

```
int x;
float y = 5 ; // tự động chuyển 5 thành 5.0 và gán cho y
x = y + 2 ; // sai vì mặc dù 5 + 2 = 7 nhưng không gán được cho x
```

Trong ví dụ trên để câu lệnh x= y+2 thực hiện được ta phải ép kiểu của biểu thức y+2 về thành kiểu nguyên.

Cú pháp tổng quát như sau:

tên_kiểu(biểu_thức)

Trong đó tên_kiểu là kiểu cần được chuyển sang. Như vậy câu lệnh trên phải được viết lại: x = int(y + 2) ;

3.2.2. Các toán tử (operator)

Các thành phần tên gọi tham gia trong phép toán được gọi là hạng thức hoặc toán hạng, các kí hiệu phép toán được gọi là toán tử. C++ có rất nhiều phép toán loại 1 ngôi, 2 ngôi và thậm chí cả 3 ngôi.

Ví dụ 3.7a:

Trong phép toán $x * y$;

x, y được gọi là toán hạng.

$*$ là toán tử.

Sau đây là một số toán tử thông dụng:

a. Các phép toán số học: +, -, *, /, %

- Các phép toán + (cộng), - (trừ), * (nhân) được hiểu theo nghĩa thông thường trong số học.

- Phép toán x / y được thực hiện theo kiểu của các toán hạng, tức nếu cả hai toán hạng là số nguyên thì kết quả của phép chia chỉ lấy phần nguyên, ngược lại nếu 1 trong 2 toán hạng là thực thì kết quả là số thực.

Ví dụ 3.7b:

$11/5 = 2$ // do 11 và 5 là 2 số nguyên

$11.0/5 = 11/5.0 = 11.0/5.0 = 2.2$ // do có ít nhất 1 toán hạng là thực

b. Toán tử quan hệ (so sánh)

Các phép toán so sánh

$==$ bằng nhau

$!=$ khác nhau

$>$ lớn hơn

$<$ nhỏ hơn

$>=$ lớn hơn hoặc bằng

$<=$ nhỏ hơn hoặc bằng

c. Toán tử logic

$\&\&$ và

$\|\|$ hoặc

$!$ không, phủ định

Hai toán hạng của loại phép toán này phải có kiểu logic tức chỉ nhận một trong hai giá trị "đúng" (được thể hiện bởi các số nguyên khác 0) hoặc "sai" (thể hiện bởi 0). Khi đó giá trị trả lại của phép toán là 1 hoặc 0 và được cho trong bảng sau:

x	y	x && y	x y	! x
1	1	1	1	0
1	0	0	1	0
0	1	0	1	1
0	0	0	0	1

Bảng 4: Mô tả các phép toán logic.

d. Toán tử gán

Toán tử gán dùng để thay đổi trị của một biến bằng trị của một biểu thức.

Kí hiệu: =

Biểu thức gán có dạng: **<biến> = <biểu thức>;**

Trong đó <biến> là một danh hiệu hợp lệ (nằm ở vế trái), và <biểu thức> (nằm ở vế phải) là biểu thức có cùng kiểu với kiểu của <biến>.

Cách tính trị của biểu thức gán: trình biên dịch tính trị của <biểu thức>, sau đó gán trị này cho <biến>. Toàn bộ biểu thức gán này cũng trả về một giá trị là trị của <biểu thức> vừa tính được.

Ví dụ 3.8:

a = 10; // gán trị 10 cho biến nguyên a.

a = 2*b; // tính trị 2*b, sau đó gán kết quả tính được cho biến a.

c = a + 2*b; // tính trị a + 2*b, sau đó gán kết quả tính được cho biến c.

Ngôn ngữ C++ cho phép viết gọn các biểu thức gán bằng các toán tử gán sau:

Dạng viết thông thường	Dạng viết thu gọn
i = i+<bt>	i += <bt>
i = i-<bt>	i -= <bt>
i = i*<bt>	i *= <bt>
i = i/<bt>	i /= <bt>
i = i%<bt>	i %= <bt>

...

e. Toán tử điều kiện (? :)

Là toán tử 3 ngôi, và có dạng:

<BT1> ? <BT2> : <BT3>

Trong đó:

<BT1> thường là một biểu thức so sánh hay một biểu thức logic.

<BT2>, và <BT3> là một biểu thức thông thường nào đó.

Kiểu của biểu thức điều kiện phụ thuộc vào kiểu của <BT2>, <BT3>.

Cách tính giá trị biểu thức điều kiện

B1: Tính trị của <BT1>

B2: Nếu <BT1> có trị !=0 thì toàn bộ biểu thức điều kiện sẽ nhận trị cho bởi <BT2>. Ngược lại thì chương trình dịch sẽ dùng trị của <BT3> làm trị của toàn bộ biểu thức điều kiện.

Ví dụ 3.9:

a. Biểu thức điều kiện trả về trị nhỏ nhất giữa 2 số nguyên cho trước x và y

$$(x < y) ? (x) : (y)$$

b. Biểu thức điều kiện trả về trị lớn nhất trong 3 số nguyên cho trước x, y, z

$$(x > y) ? (x > z ? x : z) : (y > z ? y : z)$$

f. Toán tử tăng (++), giảm (—) một đơn vị

Biểu thức có dạng

<biến>++ <biến>--

++<biến> --<biến>

Sự khác nhau giữa 2 dạng tiền tố và hậu tố chỉ nảy sinh khi biểu thức nằm trong một biểu thức khác như sau:

- Dạng tiền tố: trị của <biến> được thay đổi trước khi tham gia biểu thức chung
- Dạng hậu tố: biểu thức chung sử dụng trị cũ của <biến>, sau đó <biến> mới được thay đổi trị.

g. Một số hàm toán học

Một số hàm toán học trong thư viện cmath:

- sin(x), cos(x), tan(x), asin(x), acos(x), atan(x): các hàm lượng giác.
- exp(x): hàm mũ, trả lại giá trị e mũ x (ex).
- log(x), log10(x): trả lại lôgarit cơ số e và lôgarit thập phân của x (lnx, logx).
- pow(x, y): hàm mũ, trả lại giá trị x lũy thừa y (xy).
- sqrt(x): trả lại căn bậc 2 của x.
- abs(x), labs(x), fabs(x): trả lại giá trị tuyệt đối của x.
- ceil(x): trả lại giá trị làm tròn lên của x.
- floor(x): trả lại giá trị làm tròn xuống của x.

3.3. Nhập xuất dữ liệu cơ bản

3.3.1. Hàm cin

Nhập dữ liệu từ bàn phím:

Trong C++, ta dùng đối tượng cin và toán tử >> khai báo trong <iostream.h> như sau:

cin >> biến1 [>> biến2 >> biến3 ...];

biến_1, biến_2, biến_3 là các biến được sử dụng để lưu trữ các giá trị người sử dụng nhập vào từ bàn phím.

Ví dụ 3.10: Giả sử có các khai báo sau:

```
int x,y;  
double z;  
char ch;
```

Ta có thể viết các lệnh nhập dữ liệu:

```
cin >> x >> y;  
cin >> z;  
cin >> ch;
```

Thông thường, để nhập dữ liệu cho biến ta dùng cặp câu lệnh sau:

```
cout << "Nhập tri cho bien :"; // câu dẫn hướng  
cin >> var ; // nhập giá trị từ bàn phím cho biến var
```

Chú ý: Lệnh sau đây là câu lệnh hợp lệ: cin >> a, b, c; // ý nghĩa chỉ nhập trị cho biến x.

Để nhập đầy đủ trị cho 3 biến a, b và c: cin >> a >> b >> c;

3.3.2. Hàm cout

Để xuất giá trị của các biểu thức ra màn hình ta dùng câu lệnh sau:

cout<< BT1[<<BT2<<BT3 . . .];

Trong đó: BT1, BT2, . . . là các biểu thức hợp lệ.

Ví dụ 3.11a: để in câu "Chu vi hình vuông là 40 cm ", ta có thể sử dụng lệnh sau:

```
cout << " Chu vi hình vuông là 40 cm" ;
```

hoặc in ra diện tích của hình chữ nhật khi đã biết các cạnh là a và b bằng các câu lệnh sau:

```
cout << "Diện tích = " << a * b ;
```

Ví dụ 3.11b: lệnh cout

```
cout<<"Chao!!!";           // in chuỗi "Chao!!!".  
cout<<abc;                 // in ra giá trị của abc.  
cout<<15+16;              // tính giá trị 15+16 và in ra.  
cout<<"A = " <<a;         // in "A =", sau đó in giá trị a.  
cout<<endl;               // chuyển con trỏ xuống dòng mới.  
cout<<endl << endl;      // chuyển con trỏ xuống 2 dòng.
```

Ví dụ 3.12: Chương trình nhập vào hai số. Tính tổng và hiệu của hai số vừa nhập.

```
int main()
{
    int X, Y;
    cout<< "Nhap vao mot so X:";
    cin>>X;
    cout<< "Nhap vao mot so Y:";
    cin>>Y;
    cout<<"Tong cua chung:"<<X+Y<<"\n";
    cout<<"Hieu cua chung:"<<X-Y<<"\n";
    return 0;
}
```

Để thực hiện dòng xuất chúng ta sử dụng biến cout (console output) kết hợp với toán tử chèn (insertion operator) <<. Còn dòng nhập chúng ta sử dụng biến cin (console input) kết hợp với toán tử trích (extraction operator) >>.

Ví dụ 3.13a: Chương trình tính diện tích hình chữ nhật biết chiều dài là a và chiều rộng là b.

```
int main() // đây là hàm chính của chương trình
{
    float a, b ; // khai báo các biến có tên a, b để chứa độ dài các cạnh
    cout << "Hãy nhập chiều dài: " ;
    cin >> a ; // nhập dữ liệu
    cout << "Hãy nhập chiều rộng: " ;
    cin >> b ;
    cout << "Diện tích = " << a * b << "\n" ; // in kết quả
    cout << "Chu vi = " << 2 * (a + b) << "\n";
    return 0;
}
```

Ví dụ 3.13b: về kiểu số int

```
int a = 11;
int b = 3;
cout<< (a+b);    // in ra 14
cout<< (a-b);    // in ra 8
cout<< (a*b);    // in ra 33
cout<< (a/b);    // in ra 3
```

```

cout<< (a%b);    // in ra 2
cout<< ++a;      // tăng a lên 1, in ra 12
cout<< b++;      // in ra 3, b tăng lên 4
cout<< --b;      // giảm b đi 1, in ra 3
cout<< a--;      // in ra 12, giảm a đi 1

```

3.3.3. Các lệnh trình bày màn hình

Một số định dạng đơn giản, để sử dụng các định dạng này cần khai báo file nguyên mẫu <iomanip.h> ở đầu chương trình bằng chỉ thị #include <iomanip.h>.

- endl: Tương đương với kí tự xuống dòng '\n'.
- setw(n): Phương thức này qui định độ rộng dành để in ra các giá trị là n cột màn hình. Nếu n lớn hơn độ dài thực của giá trị, giá trị sẽ in ra theo lề phải, để trống phần thừa (dấu cách) ở trước.
- setprecision(n): Chỉ định số chữ số của phần thập phân in ra là n. Số sẽ được làm tròn trước khi in ra.
- setiosflags(ios::showpoint): Phương thức setprecision chỉ có tác dụng trên một dòng in. Để cố định các giá trị đã đặt cho mọi dòng in (cho đến khi đặt lại giá trị mới) ta sử dụng phương thức setiosflags(ios::showpoint).

Câu hỏi thảo luận

1. Trình bày các kiểu dữ liệu cơ bản.
2. Nêu ý nghĩa của lệnh cin và cout và cho ví dụ.
4. Thế nào là một định danh hợp lệ? Nguyên tắc sử dụng định danh?
5. Trình bày lệnh khai báo thư viện để sử dụng một số định dạng đơn giản như: setw(n), setprecision(n).

Bài tập vận dụng

1. Viết chương trình nhập vào 2 số thực. Tính và xuất kết quả tổng, hiệu, tích, thương của 2 số thực vừa nhập, kết quả lấy 1 số lẻ.
2. Viết chương trình nhập vào bán kính của hình tròn; xuất ra diện tích và chu vi của hình tròn đó.
3. Viết chương trình nhập vào đáy lớn, đáy nhỏ, chiều cao của hình thang; xuất ra diện tích của hình thang đó.
4. Nhập vào 4 số nguyên. Tính trung bình cộng 4 số đó.
5. Giá trị cuối cùng của a, b bằng bao nhiêu?

```
#include <iostream>
using namespace std;
int main ()
{
    int a, b; // a:?, b:?
    a = 10; // a:?, b:?
    b = 4; // a:?, b:?
    a = b; // a:?, b:?
    b += a; // a:?, b:?
    a += 7; // a:?, b:?
    cout << "a:" << a;
    cout << " b:" << b;
}
```

6. Viết chương trình tính giá trị của biểu thức, trong đó x là số nguyên nhập từ phím.

$$F(x) = 4x^2 + 4x + 1.$$

$$G(x) = 2x^4 - 5x^2 + 5x + 1.$$

7. Viết chương trình nhập vào điểm ba môn Toán, Anh1, Triết của một sinh viên. In ra điểm trung bình của học sinh đó với hai số lẻ thập phân.

8. Viết chương trình nhập vào ngày, tháng, năm. In ra ngày tháng năm theo dạng dd/mm/yy. (dd: ngày, mm: tháng, yy: năm. Ví dụ: 30/04/19).

9. Viết chương trình nhập vào độ dài 3 cạnh a, b, c của một tam giác. Tính chu vi và diện tích của tam giác theo công thức:

$$\text{Chu vi} = a+b+c$$

$$\text{Diện tích} = \sqrt{p*(p-a)*(p-b)*(p-c)}$$

Trong đó: p là nửa chu vi. In các kết quả lên màn hình.

10. Các giá trị sau thuộc kiểu dữ liệu chuẩn nào?

a) 12.0

b) 2019

- c) '?'
- d) $13 < 17$
- e) 15.172
- f) $11/5$

11. Chọn kiểu dữ liệu phù hợp cho các trường hợp sau:

- Tuổi của sinh viên
- Pi (3.14159265)
- Số trang của một văn bản
- Giá của một cổ phiếu (2 số sau dấu phẩy).

12. Tính giá trị các biểu thức sau:

- | | |
|---|--------------|
| a) $(5 > 3 \ \&\& \ 4 < 8)$ | e) $7 / 4$ |
| b) $(4 > 6 \ \&\& \ \text{true})$ | f) $14 \% 5$ |
| c) $(3 >= 3 \ \ \ \text{false})$ | g) $7 / -2$ |
| d) $(\text{true} \ \ \ \text{false}) ? 4 : 5$ | h) $-5 \% 2$ |

CHƯƠNG 4: CÁC LỆNH RỄ NHÁNH

Mục tiêu của chương

Nắm vững:

- Cách sử dụng câu lệnh đơn và câu lệnh ghép.
- Cú pháp các lệnh như: lệnh gán, lệnh rẽ nhánh, lệnh điều khiển để áp dụng vào các bài tập cụ thể.

Nội dung của chương

Nghiên cứu ý nghĩa cấu trúc các lệnh điều kiện, cách sử dụng các lệnh vào các bài tập.

4.1. Các cấu trúc điều khiển chương trình

4.1.1. Câu lệnh đơn (statement, instruction)

Câu lệnh đơn là một biểu thức được kết thúc bằng dấu ‘;’.

Câu lệnh đơn có dạng:

<biến> = <biểu thức>;

<biểu thức>;

<lời gọi hàm>;

Ví dụ 4.1: một số lệnh đơn

```
int x;  
++x;  
double d = 9.5;
```

4.1.2. Câu lệnh ghép

Câu lệnh ghép hay còn gọi là khối lệnh bao gồm từ 2 câu lệnh trở lên, và được đặt trong cặp ngoặc nhọn { } như sau:

```
{  
    <lệnh 1>;  
    <lệnh 2>;  
    ...  
    <lệnh n>;  
}
```

Ví dụ 4.2: Lệnh ghép

```
{ int min, x = 5, y = 10;  
  min = (x < y ? x : y);  
  cout << min << '\n';  
}
```

Câu lệnh điều khiển được xây dựng từ các cấu trúc điều khiển như:

- Cấu trúc rẽ nhánh: câu lệnh if ... else ...
- Cấu trúc lựa chọn: câu lệnh switch (...)
- Cấu trúc lặp: câu lệnh for (...), câu lệnh while(...), câu lệnh do...while(...).

Chú ý:

- Câu lệnh ‘;’ là một câu lệnh rỗng, nhưng hợp lệ.
- Khai báo biến, khai báo hàm nguyên mẫu cũng được kết thúc bằng dấu chấm phẩy ‘;’

4.2. Câu lệnh if

Một câu lệnh if cho phép chương trình có thể thực hiện khối lệnh này hay khối lệnh khác phụ thuộc vào một điều kiện được viết trong câu lệnh là đúng hay sai.

4.2.1. Câu lệnh if dạng 1

a. Cú pháp

```
if (<điều kiện>) { khối lệnh 1; }
```

Trong đó:

- <điều kiện> là một biểu thức logic tức nó có giá trị đúng (khác 0) hoặc sai (bằng 0).
- <khối lệnh 1> có thể là một câu lệnh đơn, một khối lệnh hay một câu lệnh phức.

b. Ý nghĩa

Kiểm tra điều kiện trước.

Nếu điều kiện đúng (True) thì thực hiện khối lệnh 1 theo sau biểu thức điều kiện.

Nếu điều kiện sai (False) thì bỏ qua khối lệnh 1 (những lệnh và khối lệnh sau đó vẫn được thực hiện bình thường vì nó không phụ thuộc vào điều kiện sau if).

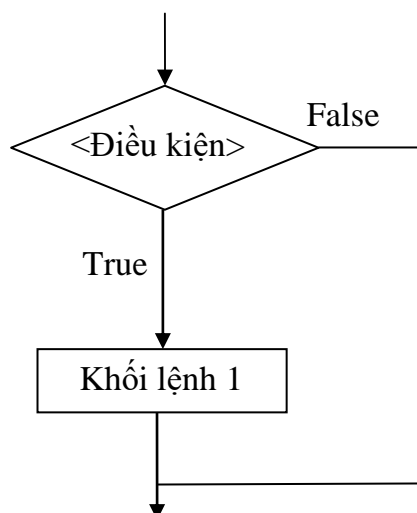
Ví dụ 4.3a: câu lệnh if dạng 1

```
if (x==10)
    cout << "x bằng 10";
```

Ví dụ 4.3b: câu lệnh if và khối lệnh

```
if (x==10)
{
    cout << " x bằng ";
    cout << x ;
}
```


c. Lưu đồ



Hình 10: Sơ đồ hoạt động của lệnh if dạng 1.

4.2.1. Câu lệnh if dạng 2 (if else)

a. Cú pháp

```
if (<điều kiện>) { khối lệnh 1; } else { khối lệnh 2; }
```

Trong đó

- <điều kiện> thường là biểu thức logic.
- Phần else là không bắt buộc phải có. Câu lệnh if không có phần else được gọi là câu lệnh “if thiếu”.
- <khối lệnh 1>, <khối lệnh 2> là câu lệnh hợp lệ bất kỳ: câu lệnh đơn, hoặc câu lệnh ghép, hoặc câu lệnh điều khiển.

b. Ý nghĩa

Nếu < điều kiện > thỏa mãn thì thực hiện <khối lệnh 1> ở phần if, còn ngược lại thì thực hiện < khối lệnh 2> ở phần else.

Như vậy chỉ < khối lệnh1> hoặc < khối lệnh 2> được thực hiện mà thôi.

Lệnh if else thực hiện như sau:

B1: Kiểm tra điều kiện.

B2: Nếu < điều kiện > đúng thì <khối lệnh 1> được thực hiện.

Còn ngược lại, nếu < điều kiện > sai thì < khối lệnh 2> được thực hiện (nếu có).

B3: Chuyển quyền điều khiển sang câu lệnh kế tiếp sau lệnh if.

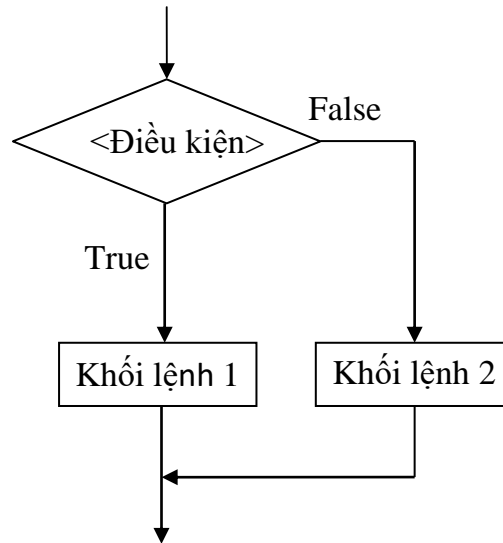
Ví dụ 4.4a: câu lệnh if dạng 2

```
if (x==10)
    cout << "x bằng 10";
else
    cout << "x khác 10";
```

Ví dụ 4.4b: câu lệnh if lồng nhau

```
if (x > 0)
    cout << "x la so duong";
else if (x < 0)
    cout << "x la so am";
else
    cout << "x bang 0";
```

c. Lưu đồ



Hình 11: Sơ đồ hoạt động của lệnh if dạng 2.

Chú ý:

Đặc điểm chung của các câu lệnh có cấu trúc là bản thân nó chứa các câu lệnh khác. Điều này cho phép các câu lệnh if có thể lồng nhau.

Nếu nhiều câu lệnh if (có else và không else) lồng nhau việc hiểu if và else nào đi với nhau cần phải chú ý. Quy tắc là else sẽ đi với if gần nó nhất mà chưa được ghép cặp với else khác.

Ví dụ 4.5: Viết chương trình nhập vào 2 số nguyên, in ra kết quả của các phép toán: cộng, trừ, nhân, chia của 2 số nguyên đó.

```
#include <iostream>
#include <conio.h>
#include <math.h>
using namespace std;
int main()
{
    int a, b;
    cout << "Ban hay nhap 2 so nguyen: ";
```

```

cin>>a>>b;
cout<<"Tong cua 2 so vua nhap la: "<<a+b<<"\n";
cout<<"Hieu cua 2 so vua nhap la: "<<a-b<<"\n";
cout<<"Tich cua 2 so vua nhap la: "<<a*b<<"\n";
if (b!=0)
cout<<"Thuong cua 2 so vua nhap la: "<<a/b;
return 0;
}

```

Ví dụ 4.6: Giải phương trình bậc hai: $ax^2 + bx + c = 0$ ($a \neq 0$).

```

#include <iostream>
#include <conio.h>
#include <math.h>
using namespace std;
int main()
{
float a, b, c;
float delta;
float x1, x2; // 2 nghiệm
cout << "Nhap a, b, c:\n" ; cin >> a >> b >> c ; // qui ước nhập a ≠ 0
delta = b*b - 4*a*c ;
if (delta < 0) cout << "phuong trinh vo nghiệm\n" ;
else if (delta==0) cout<<"phuong trinh co nghiệm kép:" << -b/(2*a) << '\n';
else
{
x1 = (-b+sqrt(delta))/(2*a);
x2 = (-b-sqrt(delta))/(2*a);
cout << "nghiệm 1 = " << x1 << " và nghiệm 2 = " << x2 ;
}
}

```

4.3. Câu lệnh switch

Câu lệnh switch cung cấp một câu lệnh cấu trúc khác cho phép chương trình có thể chọn một trong nhiều nhánh để thực hiện.

Để thực hiện câu lệnh switch đầu tiên chương trình tính giá trị của biểu thức điều khiển (btdk), sau đó so sánh kết quả của btdk với giá trị của các biểu_thức_i bên dưới lần lượt từ biểu thức đầu tiên (thứ nhất) cho đến biểu thức cuối cùng (thứ n), nếu giá trị của

btđk bằng giá trị của biểu thức thứ i đầu tiên nào đó thì chương trình sẽ thực hiện dãy lệnh thứ i và tiếp tục thực hiện tất cả dãy lệnh còn lại (từ dãy lệnh thứ i+1) cho đến hết (gặp dấu ngoặc đóng } của lệnh switch).

Nếu quá trình so sánh không gặp biểu thức (nhánh case) nào bằng với giá trị của btđk thì chương trình thực hiện dãy lệnh trong default và tiếp tục cho đến hết (sau default có thể còn những nhánh case khác).

Trường hợp câu lệnh switch không có nhánh default và biểu thức điều khiển không khớp với bất cứ nhánh case nào thì chương trình không làm gì, coi như đã thực hiện xong lệnh switch.

Nếu muốn lệnh switch chỉ thực hiện nhánh thứ i (khi biểu thức điều khiển = biểu_thức_i) mà không phải thực hiện thêm các lệnh còn lại thì cuối dãy lệnh thứ i thông thường ta đặt thêm lệnh break; đây là lệnh cho phép thoát ra khỏi một lệnh cấu trúc bất kỳ.

a. Cú pháp

switch (biểu thức điều khiển)

```
{  
    case biểu_thức_1: dãy_lệnh_1 ; break;  
    case biểu_thức_2: dãy_lệnh_2 ; break;  
    ..... ;  
    case biểu_thức_n: dãy_lệnh_n ; break;  
    default: dãy_lệnh;          break;  
}
```

Trong đó:

- Biểu thức điều khiển: phải có kiểu nguyên hoặc kí tự,
- Các biểu_thức_n: được tạo từ các hằng nguyên hoặc kí tự,
- Các dãy lệnh có thể rỗng. Không cần bao dãy lệnh bởi cặp dấu {},
- Nhánh default có thể có hoặc không và vị trí của nó có thể nằm bất kỳ trong câu lệnh (giữa các nhánh case), không nhất thiết phải nằm cuối cùng.

b. Ý nghĩa

Tùy theo giá trị của <biểu thức điều khiển > so trùng với <biểu thức> của nhãn case nào thì thực hiện <dãy lệnh> tương ứng cho đến khi gặp lệnh break thì thoát khỏi câu lệnh switch(...).

Lệnh switch thực hiện như sau:

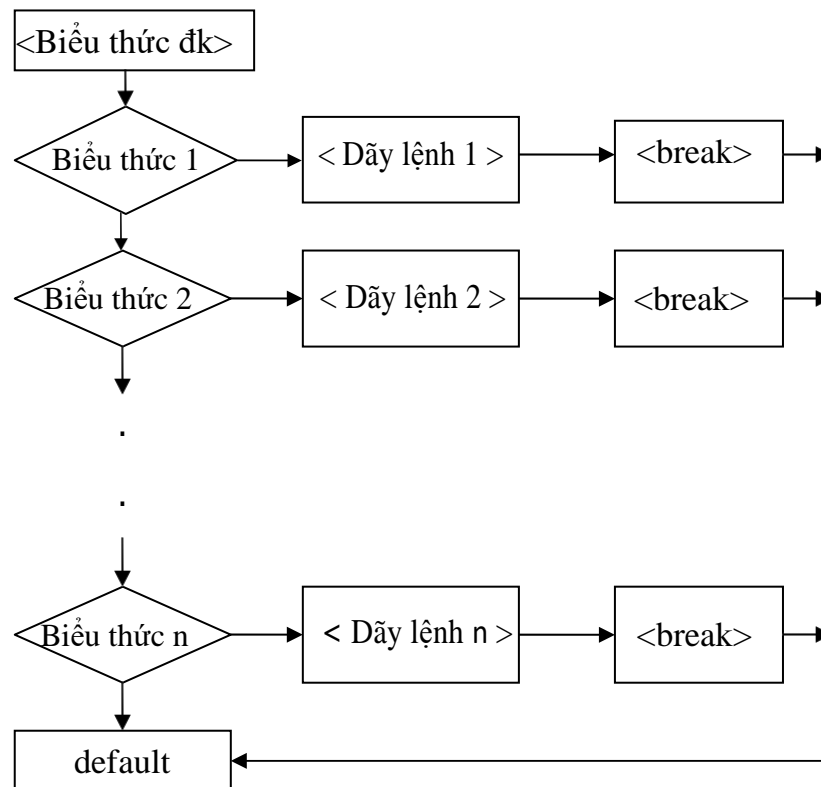
B1: < Biểu thức điều khiển > được tính giá trị.

B2: Lần lượt so sánh giá trị của < biểu thức điều khiển > vừa tính được với các giá trị < biểu thức 1>, < biểu thức 2>, ... sau nhãn case. Nếu so trùng thì <dãy lệnh n> tương ứng được thực hiện, cho đến khi gặp lệnh break.

Nếu không có nhãn giá trị nào so trùng với giá trị của < biểu thức điều khiển > thì <dãy lệnh> ở phần default được thực hiện (nếu có).

B3: Thoát khỏi lệnh switch(...), và chuyển quyền điều khiển sang câu lệnh kế tiếp.

c. Lưu đồ



Hình 12: Sơ đồ hoạt động của lệnh switch.

Ví dụ 4.7: Nhập tháng và cho biết tháng đó có bao nhiêu ngày.

```
#include <iostream>
#include <conio.h>
#include <math.h>
using namespace std;
int main()
{
    int thang;
    cout<<"Nhập vào thang: ";
    cin>>thang;
```

```

switch(thang)
{
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12:
        cout<<"31 ngay"; break;
    case 2:
        cout<<"28 hoac 29 ngay"; break;
    case 4:
    case 6:
    case 9:
    case 11:
        cout<<"30 ngay"; break;
}
}

```

Ví dụ 4.8: Nhập vào số và in ra thứ tương tự với số đó.

```

#include <iostream>
#include <conio.h>
#include <math.h>
using namespace std;
int main()
{
    char thu ;
    cout<<"Nhap vao thu: ";
    cin>>thu;
    switch(thu)
    {
        case 2:
            cout<<"Thu hai"; break;
        case 3:

```

```

        cout<<"Thu ba"; break;
case 4:
        cout<<"Thu tu"; break;
case 5:
        cout<<"Thu nam"; break;
case 6:
        cout<<"Thu sau"; break;
case 7:
        cout<<"Thu bay"; break;
case 8:
        cout<<"Chu nhat"; break;
default:
        cout<<"Thu ban nhap khong hop le!";
    }
}

```

Ví dụ 4.9: Nhập 2 số a và b vào từ bàn phím. Nhập kí tự thể hiện một trong bốn phép toán: cộng, trừ, nhân, chia. In ra kết quả thực hiện phép toán đó trên 2 số a, b.

```

#include <iostream>
#include <conio.h>
#include <math.h>
using namespace std;
int main ()
{
    float a, b, c ; // các toán hạng a, b và kết quả c
    char dau ; // phép toán được cho dưới dạng kí tự
    cout << "Hãy nhập 2 số a, b: " ; cin >> a >> b ;
    cout << "và dấu phép toán: " ; cin >> dau ;
    switch (dau)
    {
        case '+': c = a + b ; break ;
        case '-': c = a - b ; break ;
        case 'x': case '!': case '*': c = a * b ; break ;
        case ':': case '/': c = a / b ; break ;
    }
}

```

```

cout << setiosflags(ios::showpoint) << setprecision(3) ; // in 3 số lẻ
cout << "Kết quả là: " << c ;
}

```

Câu hỏi thảo luận

1. Trình bày cú pháp và công dụng của lệnh if và lệnh switch.
2. Trình bày thế nào và câu lệnh đơn, câu lệnh ghép cho 2 ví dụ minh họa.
3. So sánh lưu đồ của lệnh if dạng 1 và dạng 2.
4. Vẽ lưu đồ của lệnh switch và cho ví dụ minh họa.
5. Chuẩn bị lý thuyết chương tiếp theo.

Bài tập vận dụng

1. Viết chương trình nhập vào thông tin của một sinh viên bao gồm: mã số sinh viên, họ tên, quê quán, năm sinh, điểm trung bình các năm học; xuất ra thông tin của sinh viên vừa nhập.

2. Viết chương trình nhập vào ba cạnh a, b, c của một tam giác, tính và xuất ra diện tích của tam giác theo công thức $S = p(p - a)(p - b)(p - c)$ với p là nửa chu vi.

3. Một đoạn thẳng được biểu diễn bởi hai điểm trong mặt phẳng. Viết chương trình nhập vào hai điểm của một đoạn thẳng và xuất ra trung điểm của đoạn thẳng đó.

4. Nhập vào hai số nguyên a, b. In ra màn hình giá trị lớn nhất.

5. Cho ba số a, b, c nhập vào từ bàn phím. Hãy tìm giá trị lớn nhất của ba số trên và in ra kết quả.

6. Giải và biện luận phương trình: $ax + b = 0$.

7. Giải và biện luận phương trình: $ax^4 + bx^2 + c = 0$.

8. Viết chương trình xếp hạng học lực của sinh viên dựa trên các điểm bài kiểm tra, điểm thi giữa kỳ, điểm thi cuối kỳ. Nếu:

Điểm trung bình ≥ 8.5 là hạng A.

Điểm trung bình ≥ 7.0 và < 8.5 là hạng B.

Điểm trung bình ≥ 5.0 và < 7.0 là hạng C.

Điểm trung bình < 5.0 là hạng F.

9. Viết chương trình nhập vào một số nguyên n gồm ba chữ số. Xuất ra màn hình chữ số lớn nhất ở vị trí nào?

10. Viết chương trình tính diện tích và chu vi các hình: tam giác, hình vuông, hình chữ nhật và hình tròn với những thông tin cần được nhập từ bàn phím.

11. Viết một chương trình nhập lương nhân viên, tính thuế thu nhập và lương ròng (số tiền lương thực sự mà nhân viên đó nhận được). Với các thông số giả sử như sau

10% thuế thu nhập nếu lương là 15 triệu.

5% thuế thu nhập nếu lương từ 7 đến 15 triệu.

3% thuế thu nhập nếu lương dưới 7 triệu.

12. Viết chương trình nhập một số nguyên bất kỳ từ bàn phím và in kết quả ra màn hình để thông báo cho biết số đó lớn hay nhỏ hơn 200.

13. Viết chương trình tính cước điện thoại bàn cho một hộ gia đình với các thông số như sau: Phí thuê bao bắt buộc là 25 nghìn.

500 đồng cho mỗi phút gọi của 50 phút đầu tiên.

300 đồng cho mỗi phút gọi của 150 phút tiếp theo.

200 đồng cho bất kỳ phút gọi nào sau 200 phút đầu tiên.

14. Viết chương trình nhập vào một số. Xuất ra màn hình chuỗi “số chẵn” nếu số đó là số chẵn. Xuất ra màn hình chuỗi “số lẻ” nếu số đó là số lẻ.

15. Viết chương trình tính lương của nhân viên dựa theo thâm niên công tác (TNCT) như sau: Lương = hệ số * lương căn bản, trong đó lương căn bản là 1350000 đồng.

Nếu TNCT < 12 tháng: hệ số = 2.34.

Nếu 12 <= TNCT < 36 tháng: hệ số = 3.33.

Nếu 36 <= TNCT < 60 tháng: hệ số = 3.66.

Nếu TNCT >= 60 tháng: hệ số = 3.99.

16. Tìm lỗi sai trong các chương trình dưới đây:

16.1. Chương trình tính năm nhuận. Năm thứ n là nhuận nếu nó chia hết cho 4, nhưng không chia hết cho 100 hoặc chia hết 400.

(Chú ý: một số nguyên a là chia hết cho b nếu phần dư của phép chia bằng 0, tức $a \% b == 0$).

```
#include <iostream>
```

```
#include <conio.h>
```

```
#include <math.h>
```

```
using namespace std;
```

```
int main ()
```

```
{
```

```
    int nam;
```

```
    cout << “Nam = “ ; cin >> nam ;
```

```
    if (nam%4 != 0 && year%100 !=0 || nam%400 == 0)
```

```
        cout << nam << "la nam nhuan" ;
```

```
    else
```

```
        cout << nam << "la nam khong nhuan" ;
```

```
}
```

16.2. Chương trình nhập tuổi và in ra kết quả nếu tuổi học sinh đó không đủ điều kiện vào học lớp 10. Biết tuổi vào lớp 10 của học sinh là 16.

```
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    int tuoi1;
    cout << "Nhap tuoi hoc sinh : " << endl;
    cin>>tuoi;
    if(tuoi==16)
    {
        cout << "Tuoi cua hoc sinh la: " << tuoi << " tuoi." << endl;
        cout << "Hoc sinh du tuoi vao lop 10!" << endl;
    }
    else
    {
        cout << " Tuoi cua hoc sinh la: " << tuoi << " tuoi." << endl;
        cout << " Hoc sinh khong du tuoi vao lop 10!" << endl;
    }
}
return 0;
}
```

16.3. Chương trình nhập một số nguyên bất kỳ từ bàn phím và in kết quả ra màn hình để thông báo cho người dùng biết số đó lớn hay nhỏ hơn 100.

```
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    int a;
    cout << "Nhap so nguyen: " << endl;
    cin>>a;
    If (a>100)
    {
        cout << a << " lon hon 100." << endl;
    }
    else
```

```

    {
    cout << b << " nho hon 100." << endl;
    }
return 0;
}

```

16.4. Chương trình **nhập** vào 3 cạnh của một tam giác:

- Tính chu vi, diện tích của tam giác đó
- Kết luận tam giác đó là tam giác cân, tam giác thường hay tam giác đều.

```

#include<iostream>
#include<math.h>
using namespace std;
int main()
{
    int a, b, c, CV, S;
    int P;
    cout<<"\n canh a la:"<<cin>>a;
    cout<<"\n canh b la:"<<cin>>b;
    cout<<"\n canh c la:"<<cin>>c;
    P = (a+b+c)/2;
    Chuvi = a+b+c;
    S = sqrt(P*(P-a)*(P-b)*(P-c));
    cout<<"Chu vi tam giac la:"<<CV;
    cout<<"\nDien tich tam giac la:"<<S;
    if (a!=0 && b!=0 && c!=0 && a==b||b==c||c==a)
        {
            cout<<endl<<"la tam giac can";
        }
    else if (a!=0 && b!=0 && c!=0 && a==b==c)
        {
            cout<<endl<<"la tam giac deu";
        }
    else if (a!=0 && b!=0 && c!=0 && a!=b||b!=c||c!=a)
        {
            cout<<endl<<"la tam giac thuong";
        }
}

```

```
return 0;
```

16.5. Chương trình xếp hạng học lực của học sinh dựa trên các điểm bài kiểm tra, điểm thi giữa kỳ, điểm thi cuối kỳ. Nếu:

- Điểm trung bình ≥ 9.0 là hạng A.
- Điểm trung bình ≥ 7.0 và < 9.0 là hạng B.
- Điểm trung bình ≥ 5.0 và < 7.0 là hạng C.
- Điểm trung bình < 5.0 là hạng F.

```
#include <iostream>
#include<iomanip>
using namespace std;
int main()
{
    float a, b, c;
    int dtb;
    cout<<"Nhap diemkiem tra, diem giua ky, diem cuoi ky tuong ung:";
    cin>>a>>b;
    dtb=(a+b+c)/3;
    cout<<"Diem trung binh la: "<<dtb<<endl;
    if(dtb >= 9.0) cout<<"Hang A";
    else if((dtb>=7.0) && (dtb<9.0)) cout<< "Hang B";
    else if((dtb>=5.0) && (dtb<7.0)) cout<< " Hang C";
    else if(dtb<5.0) cout<<" Hang F";
    else cout<<"Diem khong hop le";
    cout<<"\n";
    return 0;
}
```