TRƯỜNG ĐẠI HỌC NÔNG NGHIỆP I
BỘ MÔN CÔNG NGHỆ PHẦN MỀM
****
Các Tác giả: Đỗ Thị mơ (Chủ biên)
TS. Dương Xuân Thành, Ths. Nguyễn Thị Thuỷ, Ths. Ngô Tuấn Anh

GIÁO TRÌNH
TIN HỌC ĐẠI CƯƠNG

DÙNG CHO KHỐI A

Hà Nội - 2006

MỤC LỤC

Phần I: Đại cương về Tin học 3
Chương I: Giới thiệu chung 3
1. Thông tin và tin học 3
1.1 Khái niệm thông tin 3
1.2 Khái niệm tin học 3
2. Hệ đếm trong máy vi tính 3
2.1 Hệ 10 3
2.2 Hệ 2 4
2.3 Hệ 16 4
2.4 Biểu diễn số trong máy tính và các đơn vị thông tin 5
2.5 Chuyển đổi giữa các hệ đếm 6
2.6 Các phép toán trong hệ 2 8
3. Tệp và thư mục 10
3.1 Tệp 10
3.2 Thư mục 10
4. Mã hoá 11
4.1 Khái niệm mã hoá 11
4.2 Mã ASCII 12
5. Đại số logic 12
5.1 Khái niệm mệnh đề Logic, biến Logic, hàm Logic 12
5.2 Các toán tử Logic cơ bản 13
Bài tập -Câu hỏi ôn tập 15
Chương II: Cấu trúc máy vi tính 16
1. Chức năng và sơ đồ cấu trúc của máy vi tính 16
1.1 Chức năng của máy vi tính 16
1.2 Sơ đồ cấu trúc máy vi tính 16
2. Các bộ phận cơ bản của máy vi tính 17
2.1 Bộ xử lý trung tâm 17
2.2 Bộ nhớ 18
2.3 Thiết bị ngoại vi 18
3. Tổ chức một phòng máy tính và cách bảo quản 21
3.1 Tổ chức một phòng máy 21
3.2 Bảo quản đĩa từ, màn hình, bàn phím 21
Bài tập - Câu hỏi ôn tập 22

Chương III: Hệ điều hành 23
1. Khái niệm hệ điều hành 23
1.1 Khái niệm hệ điều hành 23
1.2 Phân loại hệ điều hành 23
1.3 Một số hệ điều hành thông dụng 23
2. Hệ điều hành Microsoft windows 98/2000 25
2.1 Khởi động máy và các thành phần cơ bản 25
2.2 Tạo, thay đổi các biểu tượng hoặc mục chọn cho một nội dung 28
2.3 Sử dụng hộp thoại CONTROL PANEL 29
2.4 Sử dụng chương trình Windows Explorer 35
Bài tập - Câu hỏi ôn tập 39
Chương IV: Hệ soạn thảo văn bản Microsoft Word 40
1. Giới thiệu chương trình MS-Word 40
1.1 Giới thiệu chung 40
1.2 Cách sử dụng chuột 40
1.3 Khởi động và ra khỏi Word 41
1.4 Mô tả các thành phần trong màn hình Word 42
2. Các phím gõ tắt (Shortcut-Key) 47
2.1 Các phím gõ tắt bắt đầu bằng phím ALT 47
2.2 Các phím gõ tắt bắt đầu bằng phím CTRL 47
2.3 Các phím gõ tắt bắt đầu bằng phím SHIFT 47
3. Các thao tác định dạng 48
3.1 Phương pháp tiếng Việt 48
3.2 Định dạng kí tự 49
3.3 Định dạng đoạn văn bản 51
3.4 Định dạng trang giấy 51
4. Kẻ bảng biểu 47
4.1 Chèn bảng biểu và văn bản 52
4.2 Một số thao tác sửa chữa bảng biểu 54
4.3 Đánh số thứ tự trong một cột 55
4.4 Tạo bảng băng bút chì 56
5. Các thao tác thường gặp 56
5.1 Viết chữ cái đầu tiên 56
5. 2 Chia trang hoặc đoạn văn bản thành nhiều cột 56
5. 3 Thuật ngữ viết tắt 57
5. 4 Tạo đoạn văn bản mẫu 58
5. 5 Viết số mũ và chỉ số 59

5. 6 Chèn vào văn bản một kí tự đặc biệt 59
6. Tự động hoá quá trình soạn thảo 59
6.1 Tạo dòng TAB. 59
6.2. Tạo các Style 60
6.3 Tạo Macro. 62
7. Đồ hoạ 65
7.1 Các thao tác vẽ cơ bản 65
7.2 Chữ nghệ thuật 68
8. Trộn tài liệu tạo thành văn bản mới 71
8.1 Tạo mẫu giấy mời 71
8.2 Tạo danh sách 71
9. Các thao tác hỗ trợ 72
9.1 Khôi phục các thanh công cụ 72
9.2 Hiện hoặc dấu các thanh công cụ 73
9.3 Thêm bớt chức năng trên thanh công cụ 74
9.4 Bảo mật văn bản 75
Bài tập - Câu hỏi ôn tập 78
Chương V: Internet và cách sử dụng Internet 83
1. Giới thiệu chung 83
1.1 Mạng máy tính 83
1.2 Nhà cung cấp dịch vụ Internet 84
1.3 Kết nối Internet 85
2. Sử dụng trình duyệt Internet Explorer (IE) 86
2.1 Khởi động Internet Explorer 86
2.2 Ra khỏi Internet Explorer 86
2.3 Tìm kiếm thông tin 86
2 4 Cách lưu trữ các địa chỉ và cách tổ chức lại các địa chỉ. 87
2.5 Một số tuỳ chọn Internet Explorer 87
2.6 Lưu trang Web vào máy tính 88
2.7 Lưu ảnh 88
2.8 In trang Web 88
3. Thư điện tử (E-mail) 89
3.1 Giới thiệu Email 89
3.2 Các mô hình hoạt động của dịch vụ Email 90
3.3 Tạo hòm thư miễn phí với Yahoo 90
3.4 Gửi và nhận thư với Yahoo 90
3.5 Gửi nhận thư bằng Outlook Express 92

3.6 Download và một số địa chỉ Internet..... 94
Bài tập - Câu hỏi ôn tập 95
Chương VI: Giải thuật 96
1. Khái niệm giải thuật 96
2. Các yêu cầu với giải thuật..... 97
3. Các cách diễn tả giải thuật 97
4. Thiết kế giải thuật..... 99
5. Giải thuật sắp xếp.....100
6. Giải thuật tìm kiếm.....101
7.Giải thuật đệ quy 102
Bài tập chương VI. 105

Phần II: Ngôn ngữ lập trình Turbo Pascal.....106
Chương I: Cấu trúc của chương trình Pascal.....106

1. Giới thiệu về ngôn ngữ lập trình 106
2. Yêu cầu phần mềm và cách khởi động.....107
3. Những quy định về cách viết biểu thức và chương trình 108
4. Cấu trúc chung của một chương trình Pascal 109
Câu hỏi ôn tập chương I 109
Chương II: Các kiểu dữ liệu cơ sở và cách khai báo 111
1. Khái niệm dữ liệu, kiểu dữ liệu 111
2. Các kiểu dữ liệu đơn giản.....111
3. Cách khai báo.....115
4. Biểu thức và câu lệnh.....117
Bài tập chương II 117
Chương III:.Các thủ tục vào ra dữ liệu và các cấu trúc điều khiển 118
1. Phép gán 118
2. Các thủ tục Vào-Ra dữ liệu.....118
3. Các câu lênh điều kiện.....123
4. Các lệnh lặp 126
Bài tập chương III.....134
Chương IV: Kiểu dữ liệu có cấu trúc: kiểu mảng, kiểu xâu kí tự, kiểu tập hợp 135
1. Kiểu mảng.....135
2. Kiểu xâu kí tự.....142

3. Kiểu tập.....146
Bài tập chương IV. 150
Chương V: Chương trình con: Hàm và thủ tục 151
1. Cấu trúc của hàm và thủ tục.....151
2. Biến toàn cục, biến cục bộ và truyền dữ liệu.....154
3. Tính đệ quy của chương trình con 154
4. Một số chương trình con của Turbo Pascal 156
Bài tập chương V. .....
Chương VI: Kiểu dữ liệu có cấu trúc: Kiểu bản ghi và kiểu tệp. 157
1. Kiểu bản ghi 157
2. Kiểu tệp.....159
Bài tập chương VI. 171
Chương VII:.Đồ họa và âm thanh 172
1. Đồ họa 172
2. Âm thanh 182
Bài tập chương VII. 185
Tài liệu tham khảo. 186

MỞ ĐẦU

Tin học là một ngành khoa học được dạy trong tất cả các trường đại học, tuy nhiên nội dung dạy có khác nhau. Tuỳ theo yêu cầu của từng ngành mà chỉ định số tiết, ngành nông nghiệp được chia ra làm hai khối, khối A và khối B, thường dạy 60 tiết cho khối A và 45 tiết cho khối B.

Giáo trình Tin học đại cương này được viết cho sinh viên Đại học Nông nghiệp thuộc các chuyên ngành khối A. Nội dung viết theo đề cương môn học trong khung chương trình đào tạo kỹ sư Nông nghiệp của Bộ Giáo dục và Đào tạo năm 2004.

Giáo trình này cung cấp cho sinh viên những kiến thức cơ bản về Tin học, kiến thức lập trình trong ngôn ngữ Pascal và một số phần mềm được ứng dụng rộng rãi nhất hiện nay, từ đó giúp sinh viên nâng cao hiểu biết về tin học và vận dụng kiến thức Tin học vào các môn học chuyên ngành, cũng như vận dụng Tin học vào các công việc của mình.

Giáo trình này viết cho người học, do đó khi dạy các giáo viên cần tham khảo thêm sách viết về cấu trúc máy tính, hệ điều hành, hệ soạn thảo văn bản Mirosoft Word, mạng máy tính, Internet, cấu trúc dữ liệu và thuật giải, lập trình trong Turbo Passcal.

Để nắm vững kiến thức trình bày trong sách không có cách nào tốt hơn là xem kỹ các thí dụ, làm đủ các bài tập, trả lời các câu hỏi ôn tập và tăng cường thực hành trên máy tính, ngoài ra còn phải có trình độ tiếng Anh tối thiểu là trình độ B.

Nội dung của giáo trình chia 2 phần; Phần I bao gồm 6 chương nêu lên những kiến thức đại cương về Tin học. Phần 2 gồm 7 chương nêu lên cách lập trình trong Turbo Pascal. Sau mỗi chương có phần bài tập hoặc câu hỏi ôn tập. Các bài tập đã được chọn lọc ở mức độ trung bình để sinh viên qua đó hiểu thêm bài giảng và thu hoạch thêm một số nội dung mở rộng gắn với thực tế.

Chương I - Phần 1 trình bày các khái niệm chung trong Tin học. Chương này giới thiệu các khái niệm cơ bản cần thiết nhất trong Tin học như: Thông tin và Tin học, hệ đếm sử dụng trong máy tính, mã hoá, đại số Logic, tệp và thư mục. Các khái niện được trình bày một cách cơ bản và ngắn gọn nhất. Chỉ nêu những phần cần thiết trong Tin học để phù hợp với thời lượng của môn học. Cuối chương là các bài tập và câu hỏi ôn tập để củng cố kiến thức.

Chương II - Phần 1 trình bày cấu trúc của máy vi tính. Chương này giới thiệu các thành phần cơ bản của máy vi tính đó là: Bộ xử lý trung tâm, bộ nhớ và các thiết bị ngoại vi. Ngoài ra còn giới thiệu cách tổ chức và bảo quản một phòng máy tính. Chương này chỉ giới thiệu các khối chính của máy vi tính bao gồm cấu trúc và chức năng, tránh đi vào cấu tạo chi tiết làm cho sinh viên khó hiểu vì sinh viên không thuộc chuyên ngành Tin học nên thiếu các kiến thức về kỹ thuật điện tử và kỹ thuật số.

Chương III - Phần 1 giới thiệu hệ điều hành. Trước tiên là khái niệm hệ điều hành, phân loại hệ điều hành, trình bày khái quát các hệ điều hành thông dụng hiện nay. Phần chính của chương này trình bày cụ thể về hệ điều hành WINDOWS 98/2000 trong đó có chương trình WINDOWS EXPLORER. Trong chương này nên chú trọng phần thực hành.

Chương IV - Phần 1 trình bày về hệ soạn thảo văn bản Microsoft Word 2000. Trong đó quan trọng nhất là các lệnh về soạn thảo văn bản, trình bày văn bản, kẻ bảng biểu, các lệnh trên tệp, các lệnh vẽ. Chú trọng thực hành nhiều trên máy để củng cố các kiến thức và tăng cường các kỹ năng soạn thảo.

Chương V - Phần 1 trình bày về Internet. Chương này cung cấp cho người học kiến thức về Internet, cách sử dụng Internet, cách sử dụng thư điện tử và các dịch vụ khác trên Internet.

Chương VI - Phần 1 trình bày về giải thuật. Chương này cung cấp cho người học kiến thức về giải thuật, các giải thuật cơ bản về sắp xếp và tìm kiếm.

Chương I - Phần 2 trình bày cấu trúc của chương trình trong Turbo Pascal. Chương này nêu lên cấu trúc của chương trình và các bước cơ bản khi lập trình.

Chương II - Phần 2 trình bày các kiểu dữ liệu cơ sở . Chương này cung cấp các kiến thức về các kiểu dữ liệu và đi sâu vào các kiểu dữ liệu đơn giản.

Chương III - Phần 2 nêu ra những kiến thức tổ chức các thủ tục vào, ra dữ liệu và các cấu trúc điều khiển trong Pascal.

Chương IV - Phần 2 trình bày các kiểu dữ liệu có cấu trúc, bao gồm kiểu mảng, kiểu xâu kí tự, kiểu tập hợp.

Chương V - Phần 2 nêu ra cách tổ chức chương trình con là hàm và thủ tục, cách sử dụng biến toàn cục và biến cục bộ, vấn đề truyền dữ liệu trong và tính đệ quy trong chương trình con.

Chương VI - Phần 2 trình bày các kiểu dữ liệu có cấu trúc bao gồm kiểu bản ghi và kiểu tệp.

Chương VII - Phần 2 trình bày các kiến thức về đồ họa và âm thanh trong Pascal.

Khi biên soạn chúng tôi đã cố gắng đưa ra một bố cục của cuốn sách sao cho bám sát được đề cương chi tiết của môn học, với hy vọng rằng giáo trình này sẽ rất bổ ích và cần thiết đối với các sinh viên đại học Nông nghiệp và các ngành khác.

Giáo trình này được viết bởi các thầy cô trong khoa Công nghệ thông tin. Thầy Dương Xuân Thành viết chương IV phần 1 và chương VII phần 2. Cô Nguyễn Thị Thủy viết chương I, chương II và chương III phần 2. Thầy Nguyễn Tuấn Anh viết chương VI phần 1. Cô Đỗ Thị Mơ viết chương I, chương II, chương III, chương V phần 1 và chương IV, chương V, chương VI phần 2.

Trong quá trình biên soạn sách chúng tôi đã nhận được nhiều ý kiến đóng góp quý báu của các giáo viên trong bộ môn Công nghệ phần mềm và bộ môn khoa học máy tính - Khoa công nghệ thông tin trường đại học Nông nghiệp I Hà nội. Chúng tôi xin chân thành cảm ơn.

Tuy đã hết sức cố gắng nhưng chúng tôi nghĩ rằng cuốn sách này còn nhiều thiếu sót, rất mong sự góp ý của bạn đọc. Xin chân thành cảm ơn.

Hà nội, ngày 20 tháng 2 năm 2006

Các tác giả

PHẦN I: ĐẠI CƯƠNG VỀ TIN HỌC

CHƯƠNG I: GIỚI THIỆU CHUNG

Để tìm hiểu sâu hơn về các phần trong giáo trình này trước tiên phải có những kiến thức cơ bản ban đầu. Chương này cung cấp những khái niệm, những kiến thức cơ bản nhất trong Tin học. Nội dung chính của chương này bao gồm các phần: Khái niệm Thông tin và Tin học, hệ đếm được sử dụng trong máy tính, mã hoá, đại số logic, tệp và thư mục.

1 – Thông tin và Tin học

1.1 - Khái niệm Thông tin

* Thông tin là tập hợp các dấu hiệu, các đặc điểm, các tính chất cho ta hiểu biết về một đối tượng.

* Thông tin có thể được biểu diễn dưới nhiều dạng phong phú như biểu tượng, kí tự, hình ảnh, âm thanh v.v... . Thi dụ biển đăng kí xe ô tô, xe máy bao gồm các kí tự dạng chữ và số, cho ta biết các thông tin về xe như xe đó thuộc tỉnh nào, xe công hay xe tư, ... Trên các bưu kiện có hình cái cốc và hình cái ô để báo cho biết trong thùng có vật dễ vỡ và phải che để không bị thấm nước.

* Nói tới thông tin phải đề cập tới quá trình xử lý thông tin. Ngày nay do xã hội phát triển nên lượng thông tin cần xử lý rất lớn, độ phức tạp của quá trình xử lý rất cao, do vậy xử lý thông tin bằng thủ công, cơ giới không thể đáp ứng được yêu cầu tự động hoá quá trình xử lý thông tin. Tin học ra đời để đáp ứng yêu cầu đó

1.2 – Khái niệm Tin học

Từ "Tin học" (Informatique) được người Pháp tên là Phillipe Dreyfus dùng đầu tiên vào năm 1962 để định nghĩa cho một môn khoa học mới mẻ trong lĩnh vực xử lí thông tin. Sau đó vào năm 1966 viện hàn lâm khoa học Pháp đã đưa ra định nghĩa sau: "Tin học là môn khoa học về xử lí hợp lí các thông tin, đặc biệt bằng các thiết bị tự động, các thông tin đó chứa đựng kiến thức của loài người trong các lĩnh vực kĩ thuật, kinh tế và xã hội "

Vậy ta có thể coi môn học Tin học là một môn học nghiên cứu việc tự động hoá quá trình xử lý thông tin.

Định nghĩa trên cho phép ta phân Tin học thành hai lĩnh vực sau:

- Phần mềm (Soft Ware): Xây dựng các thuật toán, các chương trình máy tính để xử lý thông tin.

- Phần cứng (Hard Ware): Thiết kế, lắp đặt, bảo trì các thiết bị tự động để xử lí thông tin.

Ngày nay, các kĩ thuật viên tin học vừa có khả năng hoạt động trên lĩnh vực phần cứng cũng như phần mềm.

2- Hệ đếm trong máy vi tính

2.1- Hệ 10 (Hệ thập phân: Decimal)

Hệ 10 hay hệ thập phân là hệ đếm được sử dụng để đếm và tính toán trong đời sống hàng ngày.

Hệ 10 sử dụng 10 kí hiệu chữ số 0,1,2,3,4,5,6,7,8,9 để biểu diễn các số. Do sử dụng bộ 10 chữ số nên hệ đếm này có cơ số là 10.

Để phân biệt số trong các hệ đếm khác nhau người ta thường viết số trong hệ 10 kèm theo cơ số dạng sau: Nb (số N trong hệ đếm cơ số b) hoặc viết chữ D vào sau số

Ví dụ: 2092_{10}; $789,12_{10}$; 1027D; 125,47D

Ta có thể biểu diễn số theo cơ số của hệ đếm. Ví dụ biểu diễn số N trong hệ 10 theo cơ số của nó như sau:

$8623,56_{10} = 8\times10^3 + 6\times10^2 + 2\times10^1 + 3\times10^0 + 5\times10^{-1} + 6\times10^{-2}$

Tổng quát biểu diễn một số trong hệ đếm cơ số b qua cơ số của hệ đếm như sau:

Giả sử có số N_b có n+1 chữ số phần nguyên là $a_na_{n-1} \dots a_1a_0$ và m chữ số phần lẻ là $c_1c_2 \dots c_m$. Số này được viết tổng quát như sau $N_b = a_na_{n-1} \dots a_1a_0, c_1c_2 \dots c_m$

Số Nb được biểu diễn theo cơ số b như sau:

$N_b = a_na_{n-1} \dots a_1a_0, c_1c_2 \dots c_n = a_n\times b^n + a_{n-1}\times b^{n-1} + \dots + a_1\times b^1 + a_0\times b^0 + c_1\times b^{-1} + c_2\times b^{-2} + \dots + c_m\times b^{-m}$ (*)

Công thức (*) nêu trên sẽ được sử dụng để chuyển đổi số giữa các hệ đếm.

2.2- Hệ 2 (Hệ nhị phân: Binary)

Hệ 2 hay hệ nhị phân là hệ đếm sử dụng 2 chữ số 0, 1 để biểu diễn các số.

Bit là đơn vị cơ bản của thông tin theo hệ thống số nhị phân (Binary digit). Các mạch điện tử trong máy tính sẽ phát hiện sự khác nhau giữa hai trạng thái (dòng điện mức cao và dòng điện mức thấp) và biểu diễn các trạng thái đó dưới dạng một trong hai số nhị phân 1 hoặc 0. Vì việc chế tạo một mạch điện tin cậy có thể phân biệt được sự khác nhau giữa 1 và 0 là tương đối dễ dàng và rẻ tiền, cho nên máy tính có khả năng xử lý nội bộ các thông tin nhị phân một cách rất chính xác, theo tiêu chuẩn, nó mắc ít hơn một lỗi nội bộ trong 100 tỉ thao tác xử lý

Do chỉ sử dụng tập 2 chữ số nên hệ 2 có cơ số là 2. Số N trong hệ 2 được kí hiệu N_2 hoặc viết chữ B vào sau số

Ví dụ: 10011_2 ; 11010111_2 ; 1110011B ; 10100001B

Biểu diễn tương đương hệ 10 và hệ 2 như sau:

Hệ 10	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	…
Hệ 2	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	…

Trong máy tính sử dụng hệ 2. Như vậy mọi thông tin đưa vào máy tính đều được mã hoá và chuyển sang hệ 2. Hệ 2 trong máy tính được sử dụng do lý do kỹ thuật: chế tạo các lịnh kiện có 2 trạng thái (ứng với số 0 và số 1) thì đơn giản và có tính ổn định cao. Nếu sử dụng hệ 10 thì các linh kiện phải có 10 trang thái tương ứng với 10 chữ số như vậy sẽ rất khó khăn và phức tạp.

2.3- Hệ 16 (Hexadecimal)

Hệ 16 sử dụng 16 kí hiệu 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F để biểu diễn các số. Do sử dụng 16 kí hiệu nên hệ này có cơ số là 16. Mỗi chữ số của hệ 16 tương ứng với 1 nhóm 4 bít trong hệ 2. Số N trong hệ 16 được kí hiệu N_{16} hoặc viết chữ H vào sau số

Ví dụ: $10EF_{16}$; $AE9F_{16}$; 2EFBH ; 45CDH

Biểu diễn tương đương hệ 10, hệ 16 và hệ 2 như sau:

Hệ 10	1	2	3	4	5	6	7	8	9	10	11	12
Hệ 16	1	2	3	4	5	6	7	8	9	A	B	C
Hệ 2	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100

Hệ 10	13	14	15	16	17	…
Hệ 16	D	E	F	10	11	…
Hệ 2	1101	1110	1111	0001 0000	00010001	…

Hệ 16 được dùng để ghi địa chỉ các ô nhớ trong bộ nhớ của máy tính, địa chỉ các cổng vào / ra của máy tính.

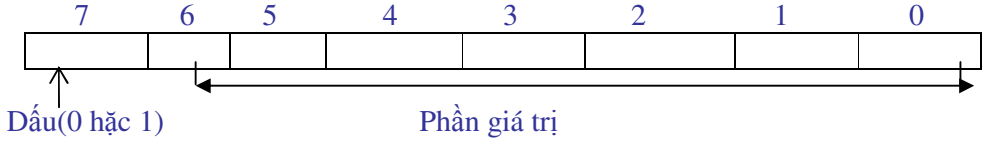
2.4- Biểu diễn số trong máy tính và các đơn vị thông tin

a) Biểu diễn số trong máy tính

* Trong máy tính các số được biểu diễn theo một khuôn thống nhất, độ dài khuôn có thể 8 bít hoặc 16 bít hoặc 32 bít, …

Có thể biểu diễn số theo dạng dấu phẩy tĩnh hoặc dạng dấu phẩy động. Trong phần này ta chỉ xét cách biểu diễn số nguyên theo dạng dấu phẩy tĩnh.

* Biểu diễn số nguyên dấu phẩy tĩnh: Số được biểu diễn theo khuôn thống nhất, bít trái nhất (bít cao nhất) dùng để biểu diễn dấu: dấu dương (+) ứng với 0, dấu âm (-) ứng với 1. Các bít còn lại dùng để biểu diễn giá trị của số. Giả sử dùng khuôn 8 bít như sau:



Ví dụ: Biểu diễn số $+6_{10}$ trong khuôn 8 bít. Trước tiên phải đổi 6_{10} sang hệ 2 là 110_2 , kết quả như sau:

0	0	0	0	0	1	1	0

* Biểu diễn số nguyên âm dấu phẩy tĩnh theo phương pháp bù 2:

Qui tắc biểu diễn số nguyên âm theo phương pháp bù 2: Trước tiên biểu diễn số nguyên dương có giá trị tuyệt đối bằng với số đó trong khuôn qui ước, sau đó đảo từng bít (1 đổi thành 0, 0 đổi thành 1), cuối cùng công với +1 (sẽ giải thích sau ở phần phép cộng trong hệ 2).

Ví dụ: biểu diễn số – 6 trong khuôn 8 bít như sau:

- Biểu diễn + 6 :

0	0	0	0	0	1	1	0

- Đảo từng bít:

1	1	1	1	1	0	0	1

- Công +1

0	0	0	0	0	0	0	1

- Kết quả - 6 :

1	1	1	1	1	0	1	0

b) Đơn vị thông tin

* BIT là đơn vị nhỏ nhất của thông tin, nó biểu thị một phần tử nhớ của máy tính. Các thiết bị máy tính đều xây dựng bằng các linh kiện điện tử chỉ có hai trạng thái khác nhau và được mã hoá bằng hai số 0/1 (gọi là con số nhị phân), nếu qui ước gọi trạng thái này là 1 thì trạng thái kia là 0 (thí dụ công tắc điện đóng=1/ngắt=0 ; bóng đèn sáng=1/tắt=0).

Các thông tin đưa vào máy phải được chuyển hoá thành các xung điện có mức điện thế cao (gọi là mức Logic1) hoặc mức điện thế thấp (mức Logic 0). Mỗi phần tử nhớ của máy tính chỉ có thể được thiết lập ở trạng thái 1 hoặc 0 nên mỗi phần tử đó gọi là một BIT (Binary digiT=chữ số nhị phân).

* BYTE (đọc là Bai) là một nhóm 8 BIT. Máy tính dùng 8 phần tử nhớ để ghi nhớ một kí tự. Vậy mỗi kí tự ứng với một BYTE (8 bit). Bít thấp nhất có số thứ tự là 0

Ví dụ: Kí tự "A" có mã ASCII 65_{10} tương ứng với 1000001_2, được biểu diễn trong 1 byte như sau

0	1	0	0	0	0	0	1

* WORD (Từ) là một nhóm gồm 2 hoặc 4 Byte (với các máy tính dùng mạch vi xử lí 80286 trở về trước mỗi từ đơn gồm 2 Byte, phù hợp với loại thanh ghi 16 Bit ; với các máy tính dùng mạch vi xử lí từ 80386 về sau mỗi từ kép gồm 4 Byte phù hợp với thanh ghi 32 Bit).

* Cũng như các đơn vị đo lường khác, đơn vị đo lường thông tin cũng có các đơn vị bội như sau:

1 Bít = 0 hoặc 1
1 Byte= 8 Bít
1 Kilobyte (KB) = 2^{10} Bytes= 1024 Bytes
1 MegaByte (MB) = 1024 KB
1 GigaByte (GB) = 1024 MB
1 TeraByte (TB) = 1024 GB

2.5- Chuyển đổi số giữa các hệ đếm

a) Chuyển đổi số từ hệ đếm cơ số b sang hệ 10

* Qui tắc: Muốn chuyển đổi số từ hệ đếm cơ số b sang hệ 10 ta dùng công thức (*) trong mục 2.1 của chương này như sau:

$N_b = a_n a_{n-1} \dots a_1 a_0, c_1 c_2 \dots c_m$

$N_b = a_n \times b^n + a_{n-1} \times b^{n-1} + \dots + a_1 \times b^1 + a_0 \times b^0 + c_1 \times b^{-1} + c_2 \times b^{-2} + \dots + c_m \times b^{-m}$

Ví dụ 1 : Đổi số 110101_2 sang hệ 10, ở đây b=2

$N_{10} = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2 + 1 \times 2^0 = 32+16+4+1 = 53_{10}$

Ví dụ 2: Đổi số $110101,11_2$ sang hệ 10

$N_{10} = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 52 + 0,5 + 0,25 = 52,75_{10}$

Ví dụ 3: : Đổi số $10F_{16}$ sang hệ 10, ở đây b=16

$N_{10} = 1 \times 16^2 + 0 \times 16 + 15 \times 16^0 = 256+15 = 271_{10}$

b) Chuyển đổi số từ hệ đếm 10 sang hệ đếm cơ số b

* Chuyển đổi số nguyên dương từ hệ 10 sang hệ đếm cơ số b:

- Qui tắc: Lấy số hệ 10 chia nguyên liên tiếp cho cơ số b, kết quả số trong hệ đếm cơ số b là các số dư của phép chia lấy theo thứ tự ngược lại (số dư của phép chia cuối cùng là số lớn nhất).

Chú ý: Chia nguyên liên tiếp cho cơ số b có nghĩa là thương của phép chia trước lại được lấy chia nguyên tiếp cho b. . . và chỉ dừng lại khi thương = 0.

Ví dụ 1: Đổi số 30_{10} sang hệ 2, ta nên tạo ra bảng chia nguyên liên tiếp để theo dõi số dư cho dễ dàng như sau

Số bị chia	Số chia (b)	Thương	Số dư
30	2	15	**0**
15	2	7	**1**
7	2	3	**1**
3	2	1	**1**
1	2	0	**1**

Kết quả $N_2 = 11110_2$

Ví dụ 2: Đổi số 172_{10} sang hệ 16, ở đây b=16

Số bị chia	Số chia (b)	Thương	Số dư
172	16	10	**12**
10	16	0	**10**

Kết quả $N_{16} = AC_{16}$ (10 ứng với A, 12 ứng với C)

* Chuyển đổi phần lẻ từ hệ 10 sang hệ đếm cơ số b :

- Qui tắc: Lấy phần lẻ nhân liên tiếp với cơ số b, kết quả lấy phần nguyên của phép nhân theo đúng thứ tự thực hiện. Nếu phần lẻ bằng 0 thì dừng; nếu phần lẻ khác 0, muốn lây bao nhiêu chữ số ta thực hiên bây nhiêu phép nhân.

Ví dụ1: Đổi số $6,25_{10}$ sang hệ 2, ở đây b=2

6_{10} đổi sang hệ 2 là 110_2

$0,25_{10}$ đổi sang hệ 2 bằng cách thực hiện phép nhân liên tiếp như sau:

$0,25 \times 2 =$ **0**,50

$0,50 \times 2 =$ **1**,0

Như vậy $0,25_{10}$ thì hệ 2 sẽ là $0,01_2$

Kết quả $6,25_{10}$ đổi sang hệ 2 là: $110,01_2$

Ví dụ 2: Đổi số 7, 37_{10} sang hệ 2, ở đây b=2

7_{10} đổi sang hệ 2 là 111_2

$0,37_{10}$ đổi sang hệ 2 như sau:

$0,37 \times 2 =$ **0**, 74

$0,74 \times 2 =$ **1**,48

$0,48 \times 2 =$ **0**,96

$0,96 \times 2 =$ **1**,92

Ta dừng ở 4 phép nhân với 4 chữ số phần lẻ, như vây $0,37_{10}$ sang hệ 2 là $0,0101_2$

Kết quả $7,37_{10}$ đổi sang hệ 2 là $111,0101_2$

c) Chuuyển đổi số từ hệ 16 sang hệ 2 và từ hệ 2 sang hệ 16

* Chuyển đổi số từ hệ 16 sang hệ 2:

- Quy tắc: Thay 1 chữ số hệ 16 bằng một nhóm 4 bít hệ 2 tương ứng.

Ví dụ: Đổi số $105EF_{16}$ hệ 2

$N_2 = 0001\ 0000\ 0101\ 1110\ 1111_2$

* Chuyển đổi số từ hệ 2 sang hệ 16:

- Quy tắc: Thay một nhóm 4 bít hệ 2 bằng1 chữ số hệ 16 tương ứng, việc nhóm các bít hệ 2 được thực hiện từ phải qua trái, nhóm cuối cùng không đủ 4 bit thì sẽ thêm các bít 0 vào trước. Vì hệ 16 dùng để ghi địa chỉ nên chỉ dùng các số nguyên không có phần lẻ.

Ví dụ: Đổi số 1110010101110_2 sang hệ 16.

Ta thêm các số 0 vào trước cho đủ các nhóm 4 bít như sau: $N_2 = 0001\ 1100\ 1010\ 1110$

tương ứng với $N_{16} = 1CAE_{16}$

2.6- Các phép toán trong hệ 2

a) Phép cộng

* Quy tắc cộng 2 bít như sau: Thực hiện theo bảng cộng (còn gọi là bảng chân lý) sau đây, trong Carry là bít nhớ chuyển sang bít cao hơn.

A	B	A+B	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- Cách thức thực hiện phép cộng hai số hệ 2 có nhiều bít như sau: đầu tiên cộng từng cặp bít có cùng thứ tự của hai số với nhau, sau đó cộng bít kết quả vừa thực hiện với bít nhớ chuyển sang từ bít thấp hơn.

Ví dụ : A =0 0 1 0 1 1 0

B =0 1 0 1 1 0 1

A+B = 1 0 0 0 0 1 1

b) Phép trừ

* Quy tắc trừ 2 bít như sau: Thực hiện theo bảng trừ sau đây, trong đó Carry là bít nhớ chuyển sang bít cao hơn của số trừ.

A	B	A-B	Carry
0	0	0	0
1	0	1	0
1	1	0	0
0	1	1	1

- Cách thức thực hiện phép trừ hai số hệ 2 có nhiều bít ta thực hiện trừ từng cặp bít theo bảng trừ trên, làm tương tự như trong hệ 10.

Ví dụ: A=1101011

B=1001101

A-B=0011110

Cách khác thực hiện phép trừ: Lấy số bị trừ cộng với số trừ biểu diễn ở dạng số âm.

a- b = a + (-b)

Ví dụ: Thực hiên phép tính $15_{10} - 6_{10}$ trong khuôn 8 bít.
Trong ví dụ này phải biểu diễn số -6_{10} trong khuôn 8 bít
(lấy kết quả ở ví dụ mục 2.4 a)

15_{10} biểu diễn trong khuôn 8 bít:

0	0	0	0	1	1	1	1

-6_{10} biểu diễn trong khuôn 8 bít :

1	1	1	1	1	0	1	0

Kết quả cộng 2 byte trên ($15_{10} - 6_{10}$) :

0	0	0	0	1	0	0	1

Khi cộng hệ 2 ta theo quy tắc ở trên, chú ý bít trái nhất là bít dấu nên khi cộng không nhớ sang bít cao hơn.

c) Phép nhân

* Qui tắc nhân 2 bít: Thực hiện theo bảng nhân sau đây

A	B	A×B
0	0	0
1	0	0
0	1	0
1	1	1

- Cách thức thực hiện phép nhân hai số hệ 2 có nhiều bít ta thực hiện tương tự như trong hệ 10 với bảng nhân trên, phép nhân được thực hiện kết hợp giữa phép dịch trái và phép cộng.

Ví dụ: a= 10011_2 b= 1011_2, thực hiện a×b như sau:

```
            10011
      ×      1011
      -----------
            10011
    +      10011
          00000
         10011
      -----------
Kết quả: 1101 0 0 01
```

d) Phép chia

Trong phần này chỉ xét phép chia nguyên.
* Cách thực hiện: Phép chia được thực hiện tương tự như trong hệ 10.
Ví dụ: A=1100 B=100 A: B = 11
Cách khác thực hiện phép chia: Lấy số bị chia trừ liên tiếp cho số chia, nếu hiệu là số dương hoặc là 0 thì thương được cộng với 1. Dừng thực hiện khi hiệu là số âm hoặc là 0.

Ví dụ: $a=12_{10}$ $b=6_{10}$ thực hiện a: b trong hệ 2 với khuôn 8 bít
$a=12_{10}$ biểu diễn trong khuôn 8 bít như sau: 00001100
$b=6_{10}$ vậy -6_{10} biểu diễn như sau: 11111010
Thực hiện phép trừ liên tiếp và cho thương như sau:

Bước 1: 12=	00001100	thương=0	
-6=	11111010		
Bước 2: Hiệu	00000110	thương= 0+1=1	
-6=	11111010		
Hiệu	00000000	thương=1 +1=10	Dừng thực hiện

Vậy 00001100 : 00000110 = 10

3- Tệp và thư mục

3.1- Tệp (File)

* Định nghĩa tệp: Tệp là tập hợp thông tin có liên quan với nhau và được lưu trữ trong bộ nhớ ngoài, bộ nhớ ngoài thông dụng hiện nay là đĩa từ. Các thông tin ta đưa vào trong máy tính như chương trình, văn bản, dữ liệu,... đều được lưu trữ trên đĩa từ dưới dạng tệp.

Ví dụ: Một bài thơ, một bảng điểm, một chương trình được lưu trữ trên đĩa từ dưới dạng tệp.

Mỗi tệp được đặt một tên duy nhất, hai tệp trong một mục không được trùng tên.

* Tên tệp: Tên tệp gồm hai phần là tên chính và đuôi cách nhau bằng dấu chấm (.)

- Tên chính là một nhóm ký tự bất kỳ thường không chứa dấu chấm (.), tên chính dùng để phân biệt tệp, tên tệpthường được chọn phù hợp với nội dung tệp. Bắt buộc phải có tên chính trong tên tệp.

- Phần đuôi tệp (phần mở rộng tệp: Extension) thường chứa 3 kí tự không chứa dấu chấm(.). Đuôi tệp được dùng để phân loại tệp. Tệp không nhất thiết phải có đuôi.

Các tệp có đuôi	EXE , COM , BAT , PRG ...	là các tệp kiểu chương trình.
	DAT , DBF ...	là các tệp dữ liệu
	TXT , DOC	là các tệp kiểu văn bản.
	PIC , GIF ...	là các tệp đồ hoạ và ảnh .

Ví dụ tên tệp : DIEM.DBF ; BAOCAO.DOC ; TINHLUONG.PRG

- Trong tên tệp cho phép dùng kí hiệu gộp (kí hiệu thay thế) : dấu? thay thế cho một kí tự bất kỳ và dấu * thay thế cho một nhóm kí tự bất kỳ. Các kí hiệu gộp này chỉ dùng trong một số lệnh như tìm kiếm, thay thế, xoá,…

Ví dụ : Trong lệnh tìm kiếm ta muốn tìm các tệp có tên là LOP1.DAT, LOP2.DAT, LOP3.DAT, có thể được đặt bằng một tên gộp là LOP?.DAT trong đó dấu ? đại diện cho 1 kí tự là 1,2 và 3 đứng sau chữ LOP.

*.DOC là tên nhóm các tệp văn bản trong WORD

. là tên nhóm đại diện cho tất cả mọi tệp.

3.2 Thư mục (Directory hay folder)

Để tìm kiếm thông tin nhanh và quản lý tốt các tệp trên các thiết bị lưu trữ thông tin của máy tính người ta tổ chức thông tin theo thư mục.

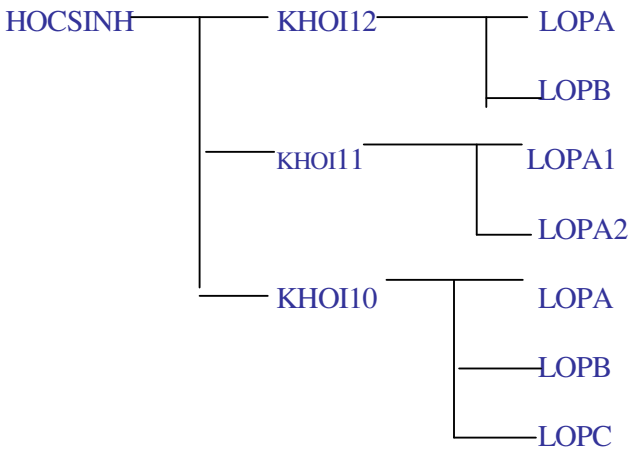
* Thư mục là một ngăn logic chứa tệp trong các thiết bị lưu trữ thông tin của máy tính (đĩa từ, băng từ hoặc đĩa CD). Giống như một tổ chức thư viện, các thông tin cũng được sắp xếp vào từng ngăn.

Mỗi thư mục được đặt một tên duy nhất và 2 thư mục không được trùng tên. Tên thư mục đặt giống như phần tên chính của tệp.

* Tổ chức thư mục trên đĩa từ: Thư mục được tổ chứa theo mô hình phân cấp (mô hình cây : TREE): Mỗi ngăn chứa gọi là một thư mục (Directory), ngăn to nhất bao trùm trên toàn bộ đĩa gọi là thư mục gốc (ROOT), trong thư mục gốc có chứa các thư mục con (ngăn con) và các tệp, trong thư mục con lại có chứa các thư mục con của nó (thư mục cháu) và các tệp,...

Một mô tả đầy đủ cấu trúc của tổ chức thư mục kể từ thư mục gốc đến các thư mục con bên trong gọi là cây thư mục (TREE).

Ví dụ: Thư mục HOCSINH có cấu trúc như sau



4. Mã hoá

4.1- Khái niệm mã hoá

* Mã hoá là thuật toán nhằm gán cho mỗi đối tương một nhóm số.

Ví dụ: Đánh số báo danh cho các thí sinh trong phòng thi là mã hoá, kết quả là mỗi thí sinh sẽ nhận được một số báo danh, chỉ cần biết số báo danh của thí sinh ta sẽ tìm được mọi thông tin về thí sinh đó. Mỗi trường sẽ có cách đánh số báo danh riêng hay cách mã hoá riêng.

* Như phần trên ta đã biết các thông tin trong máy tính đều được biểu diễn ở dạng hệ 2. Các thông tin đưa vào máy vi tính bao gồm các chữ cái la tinh A..Z, a..z, các số 0..9 và các dấu (gọi chung là kí tự - Character), bảng kí tự đó sẽ được mã hoá tương ứng với một bảng mã số. Số lượng các kí tự nhỏ hơn hoặc bằng 256, nếu dùng mã nhị phân (số hệ 2) thì chỉ cần dùng khuôn 8 bít là đủ (có thể biểu diễn số từ 0 đến $2^8 = 255$). Do vậy để mã hoá cho các kí tự được đưa vào máy vi tính người ta dùng mã nhị phân trong khuôn 8 bít.

4.2- Mã ASCII (American Standard Code for Information Interchange)

* Bảng mã ASCII là bảng mã chuẩn của Mỹ và được sử dụng thông dụng hiện nay. Mã ASCII dùng mã nhi phân trong khuôn 8 bít. Bảng mã này đánh số thứ tự cho các kí tự như sau:

Các kí tự	Mã số
Các kí tự điều khiển :	0 ---> 31
Khoảng cách (Space Bar) :	32
Dấu ! " # $ % & ' ( ) + , - . /	33 ---> 47
Các số từ 0 đến 9 :	48 ---> 57
Dấu : ; < = > ? @	58 ---> 64
Các chữ cái to A ---> Z :	65 ---> 90
Dấu [\ ] ^ _ `	91 ---> 96
Các chữ cái nhỏ a --> z :	97 ---> 122
Dấu { | } ~ <DEL>	123 ---> 127
Các kí tự đặc biệt , kí tự kẻ khung :	128 ---> 255

ở đây ta sử dụng số hệ 10 để viết cho gọn và dễ nhớ, trong máy mỗi số sẽ được biểu diễn bằng một nhóm 8 bít trong hệ 2.

Ví dụ: Kí tự "A" có mã 65_{10}, biểu diễn trong máy tính là 001000001_2

Kí tự "B" có mã 66_{10}, biểu diễn trong máy tính là 001000010_2

5. Đại số logic

5.1- Khái niệm mệnh đề logic, biến logic, hàm logic

* Mệnh đề logic: Mệnh đề logic là một câu nói hoặc câu viết có tính chất khẳng định hoặc phủ định một sự kiện. Mỗi mệnh đề logic đều có thể đặt được câu hỏi có đúng không hoặc có sai không.

- Các câu cảm thán, các câu mệnh lệnh hoặc các khẩu hiệu không gọi là mệnh đề logic. Các câu chung chung chẳng đúng mà cũng chẳng sai cũng không gọi là mệnh đề logic.

Ví dụ : "Tôi đã già" ; "Anh A học giỏi" là mệnh đề logic

"Hãy làm cho tôi việc này" không phải là mệnh đề logic

"Ôi hôm nay trời đẹp quá" là câu cảm thán không gọi là mệnh đề logic

- Mỗi mệnh đề chỉ nhận một giá trị hoặc "đúng" (TRUE) hoặc "sai" (FALSE) tương tự như một mạch điện hoặc "đóng" hoặc "tắt". Môn logic mệnh đề không quan tâm đến câu cú ngữ pháp của mệnh đề mà chỉ xét đến tính đúng sai của mệnh đề.

- Từ các mệnh đề đơn giản ta có thể xây dựng lên các mệnh đề phức tạp hơn nhờ các phép liên kết "Không", "Và" , "Hoặc"

Ví dụ: Các mệnh đề đơn A = "Hà nội đông dân "

B = "Hà nội có nhiều cây xanh "

X = "An là con liệt sĩ "

Y = "An là bôi đội "

Phép "Và" liên kết A và B cho C = "Hà nội đông dân và có nhiều cây xanh",

Phép "Không" với X cho Z= " Không phải An là con liệt sĩ".

Phép "Hoặc" liên kết X và Y cho Z = "An là con liệt sĩ hoặc An là bộ đội"

- Các phép "Không", "Và", "Hoặc" cùng với các mệnh đề làm thành một đại số gọi là đại số logic hay đại số mệnh đề

* Hai giá trị TRUE (đúng) và FALSE (sai) là 2 hằng logic, với TRUE > FALSE.

* Biến logic: Là biến chỉ có thể nhận 1 trong 2 giá trị logic là đúng (TRUE) hoặc sai (FALSE).

Ví dụ: X= "Số m là số âm " , trong ví dụ này X là một biến logic vì nó có thể nhận giá trị TRUE hoặc FALSE tuỳ theo giá trị của m, giả sử m=-5 thì X nhận giá trị TRUE, nếu m=7 thì X nhận giá trị FALSE.

* Hàm logic:

- Những bài toán Logic thường được phát biểu dưới dạng các câu nói hoặc câu viết xác định các yêu cầu và các ràng buộc đối với hệ thống mà bài toán giải quyết. Ta có thể biểu diễn sự liên kết giữa các mệnh đề bằng một biểu thức Logic hoặc được gọi là hàm Logic.

- Hàm logic là một hàm của các biến logic. Kết quả của hàm trả về 1 giá trị logic.

Ví dụ: Biến X="Sinh viên có hộ khẩu Hà nội"

Biến Y="Sinh viên có tuổi > 20 "

Hàm F= X "và" Y hay F= X AND Y có nghĩa F = "Sinh viên có hộ khẩu Hà nội và có tuổi > 20 ". Như vậy ứng với những giá trị X, Y khác nhau thì hàm F sẽ cho các giái trị khác nhau được mô tả theo bảng sau:

X	Y	X AND Y
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

Các giá trị có thể của một hàm logic được biểu diễn dưới dạng một bảng, bảng này được gọi là bảng chân lý. Hàm logic có n biến thì bảng chân lý sẽ có 2^n giá trị có thể của hàm.

5.2- Các toán tử logic cơ bản

Các phép toán trong đại số logic gọi là các toán tử logic. Sau đây sẽ trình bày 4 toán tử logic cơ bản theo thứ tự ưu tiên của các toán tử, các toán tử logic khác đều có thể được biểu diễn qua 4 toán tử cơ bản này.

* Toán tử NOT (phủ định hay đảo)

Bảng chân lý:

X	NOT X
FALSE	TRUE
TRUE	FALSE

* Toán tử AND (và)

Bảng chân lý:

X	Y	X AND Y
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

* Toán tử OR (hoặc)

Bảng chân lý:

X	Y	X OR Y
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

* Toán tử XOR (hoặc loại trừ)

Bảng chân lý:

X	Y	X XOR Y
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	FALSE

* Thứ tự ưu tiên của 4 toán tử trên là: NOT, AND, OR, XOR

* Biểu thức logic: Là sự kết hợp giữa các hằng logic, biến logic, hàm logic và các toán tử logic. Kết quả của biểu thức logic là 1 hằng logic (TRUE hoặc FALSE).

Có thể tính gía trị của biểu thức logic theo thứ tự sau:

- Thay giá trị vào các biến nếu có.
- Thực hiện các phép tính số học, các phép tính so sánh nếu có.
- Thực hiện các toán tử logic theo thư tự ưu tiên đã nêu.

Các phép tính số học và phép so sánh các số ta đã biết, chú ý hơn về so sánh 2 xâu kí tự.

* So sánh 2 xâu kí tự: Khi so sánh 2 xâu kí tự ta tiến hành so sánh mã ASCII của từng cặp kí tự tương ứng từ 2 xâu, kết quả của phép so sánh theo 3 trường hợp sau:

- Nếu gặp một cặp có mã khác nhau thì xâu chứa kí tự có mã nhỏ hơn là xâu nhỏ hơn.
- Nếu hai xâu có tất cả các cặp kí tự có mã bằng nhau thì 2 xâu bằng nhau.
- Nếu trong qúa trình so sánh một xâu đã hết các kí tự, một xâu vẫn còn kí tự thì xâu ít kí tự hơn là xâu nhỏ hơn.

Ví dụ: “ABCD” < “AX”
“ABCD” = ”ABCD”
“ABC” < ”ABCDE”

Ví dụ: Tính giá trị biểu thức logic sau

(“AB” > “1234”) OR (sinx >2) AND NOT $((X^2 + Y^2 + 2XY) \geq 0)$

= TRUE OR FALSE AND NOT TRUE

= TRUE OR FALSE AND FALSE

= TRUE OR FALSE = TRUE

Bài tập chương I

Bài 1: Chuyển đổi số trong các hệ đếm

a, Chuyển 33_{10} , 35_{10} sang số hệ 2,hệ 16
b, Chuyển 10111001_2, 1001101_2 hệ 10, hệ 16
c, Chuyển $20A_{16}$, $3EB_{16}$ sang hệ 2, hệ 10

Bài 2: Thực hiện các phép tính trong hệ 2

a, $x=36_{10}$, $y=18_{10}$ Tính x+y, x-y, x*y, x/y
b, $x=20_{10}$, $y=5_{10}$ Tính x+y, x-y, x*y, x/y

Bài 3: Tính giá trị của các biểu thức logic sau

a, NOT (LOAI < 'C') AND (GIA >= 500000) OR (LOAI <= 'B') AND (GIA <=200000)
Tính khi : 1- LOAI='D' , GIA= 500000
2- LOAI='A' , GIA= 300000
3- LOAI='B' , GIA= 100000

b, NOT (DT>8) AND (KV=3) AND (DIEM>=16) OR (DT>8) AND (KV=1) AND (DIEM>=15)
Tính khi : 1- DT=6, KV=3, DIEM=17
2- DT=9, KV=1, DIEM=16
3- DT=6, KV=3, DIEM=15

c, (TEN='H*') AND (TINH='T*') OR NOT(TEN='D*') AND (TINH < > 'H*')
Chú ý: Dấu * thay cho một dãy kí tự bất kỳ.
Tính khi : 1- TEN='HA' , TINH= 'THAI BINH'
2- TEN='DUNG', TINH=' HA NOI'
3- TEN='MAI', TINH='THAI NGUYEN'

d, $(SIN^2 X < 2)$ OR NOT('ABC' > 'AXY') AND ($(X+Y)^2 \geq 2XY$)
e, ((COS X + SIN X) > 2) OR ('ABC' < 'XY') AND NOT ((3<7) = (‘1’>’a’))
f, NOT ((lg100 + 1!) >2) AND ((3!+2)<6) AND NOT ((‘123’<'ABC') > (‘abc’<’ABC’))

Câu hỏi ôn tập chương I

1. Tin học là gì?
2. Trong máy tính dùng các hệ đếm nào?
3. Biểu diễn số trong máy tính như thế nào?
4. Trình bày các đơn vị thông tin sử dụng hiện nay.
5. Trình bày cách chuyển đổi số giữa các hệ đếm.
6. Trình bày các phép tính trong hệ 2.
7. Trình bày cách đặt tên tệp, tên thư mục.
8. Bảng mã thông dụng trong tin học hiện nay là bảng mã nào? Trình bày về bảng mã đó.
9. Trình bày các toán tử logic cơ bản theo thứ tự ưu tiên.
10. Trình bày cách so sánh 2 xâu kí tự.

CHƯƠNG II: CẤU TRÚC MÁY VI TÍNH

Để sử dụng tốt máy vi tính phục vụ cho chuyên môn của mình trước tiên phải hiểu chức năng và cấu tạo của máy vi tính. Đây là một giáo trình Tin học dạy cho sinh viên không phải chuyên ngành Tin học do đó không đi sâu vào cấu tạo chi tiết máy vi tính mà chỉ giới thiệu cấu trúc của máy vi tính hay các khối chính của máy vi tính. Chương này cung cấp các kiến thức chính như sau: Chức năng và sơ đồ cấu trúc của máy vi tính, CPU, bộ nhớ, các thiết bị ngoại vi, cách tổ chức một phòng máy.

1- Chức năng và sơ đồ cấu trúc của máy vi tính

1.1- Chức năng của máy vi tính

Máy vi tính là máy vi điện tử hoạt động theo chương trình và có các chức năng cơ bản sau đây:

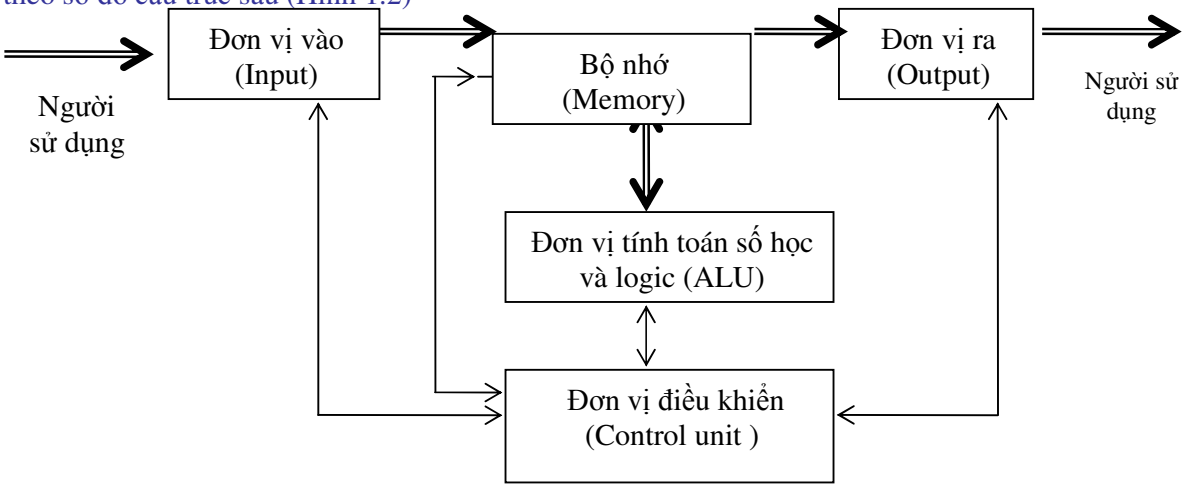
* Tiếp nhận thông tin vào và đưa thông tin ra (Vào / Ra : Input / Output). Với chức năng này người sử dụng giao tiếp được với máy vi tính.

* Xử lý thông tin hay biến đổi thông tin, đây là chức năng quan trọng nhất của máy vi tính, thực hiện nhiệm vụ tự động hoá xử lý thông tin thay cho con người.

* Lưu trữ thông tin. Các thông tin sử dụng trên máy vi tính nếu cần sẽ được lưu trữ để trao đổi và sử dụng cho nhiều lần sau.

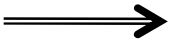
1.2- Sơ đồ cấu trúc máy vi tính

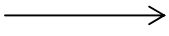
Để đảm nhận được các chức năng cơ bản trên, máy vi tính được thiết kế với các khối chính theo sơ đồ cấu trúc sau (Hình 1.2)



Hình 1.2

Trong sơ đồ trên dùng 2 kí hiệu:


Tín hiệu thông tin


Tín hiệu điều khiển

* Các tín hiệu thông tin từ người sử dụng qua đơn vị vào (bàn phím, chuột, ổ đĩa,...) được đưa vào bộ nhớ, từ bộ nhớ các thông tin được chuyển vào đơn vị tính toán số học và logic để xử lý, xử lý xong kết quả được chuyển vào bộ nhớ, từ bộ nhớ chuyển đến đơn vị ra (màn hình, máy in, máy vẽ, ...) và tới người sử dụng.

* Các tín hiệu điều khiển được thiết lập giữa đơn vị điều khiển và các khối khác trong dàn máy vi tính để chuyển các tín hiệu điều khiển của đơn vị điều khiển tới các khối khác và chuyển các tín hiệu phản hồi từ các khối khác về đơn vị điều khiển.

* Các khối trong máy vi tính được nối với nhau bằng các cáp truyền dẫn (BUS), gồm 3 loại cáp đó là cáp địa chỉ (Bus Address), cáp Dữ liệu (Bus Data) và cáp điều khiển (Bus Control).

* Trong máy vi tính 2 khối quan trọng nhất là đơn vị tinh toán số học và logic, đơn vị điều khiển nằm trong cùng một vỏ của mạch vi xử lí (Micro Processor), được gọi là đơn vị xử lý trung tâm (CPU : Central Processor Unit) hay bộ vi xử lý(Micro Processor).

* CPU và bộ nhớ nằm trong thân máy. Đơn vị vào/ ra nằm bên ngoài máy được gọi chung là thiết bị ngoại vi, ngày nay các thiết bị ngoại vi của máy vi tính rất phong phú và đa dạng.

* Ngoài các khối cơ bản kể trên, để cho máy vi tính hoạt động cần có một bộ nguồn nối với máy.

2- Các bộ phận cơ bản của máy vi tính

2.1- Bộ xử lý trung tâm(CPU: Central Processor Unit)

a) Chức năng của CPU

CPU là bộ chỉ huy của máy vi tính, có các chức năng sau:
- Thực hiện việc nhận lệnh, giải mã lệnh và điều khiển các khối khác thực hiện lệnh.
- Thực hiện các phép tính số học, logic và các phép tính khác.
- Sinh ra các tín hiệu địa chỉ trên máy.

b) Cấu tạo của CPU

CPU bao gồm các phần sau:

* Đơn vị điều khiển (Control unit): Thực hiện việc nhận lệnh, giải mã lệnh và điều khiển các khối khác thực hiện lệnh và sinh ra các tín hiệu địa chỉ trên máy để quản lý bộ nhớ.

* Đơn vị tính toán số học và logic (Arithmeric – Logic Unit): Bao gồm các vi mạch tính để thực hiện các phép tính số học, logic và các phép tính khác.

* Thanh ghi (Register) là một cấu trúc gồm 16 bít (hoặc 32 bit) nhớ liền kề nhau được thiết lập ngay trong mạch vi xử lý. Các thanh ghi này được phân thành 4 nhóm theo mục đích sử dụng sau:

- Nhóm 1: 4 thanh ghi đa năng kí hiệu là AX,BX,CX,DX được sử dụng cho nhiều mục đích khác nhau.

- Nhóm 2: 4 thanh ghi địa chỉ đoạn (Segment) bộ nhớ, thanh ghi CS (Code segment) chứa điạc chỉ đoạn mã lệnh, thanh ghi DS (Data Segment) chứa địa chỉ đoạn dữ liệu, thanh ghi SS (Stack Segment) chứa địa chỉ đoạn ngăn xếp, thanh ghi ES (Extra Segment) chứa địa chỉ đoạn dữ liệu mở rộng.

- Nhóm 3: 5 thanh ghi địa chỉ tương đối (offset), thanh ghi IP (Instruction Pointer) trỏ tới ô chứa lệnh trong đoạn CS; thanh ghi SI (Source Index) trỏ tới ô chứa dữ liệu nguồn trong

đoạn DS, Thanh ghi DI (Destination Index) trỏ tới ô chứa dữ liệu đích trong đoạn DS, Thanh ghi SP (Stack Pointer) trỏ tới ô chứa dữ liệu trong đoạn ngăn xếp SS và thanh ghi BP (Base Pointer) trỏ tới ô dữ liệu trong đoạn chứa dữ liệu mở rộng ES.

Nhóm 4: 1 thanh ghi cờ (Flag) để ghi các trạng thái (ta gọi là cờ) xảy ra trong các phép toán số học.

2.2- Bộ nhớ (Memory)

Bộ nhớ hay còn gọi là bộ nhớ trong (bộ nhớ) chính của máy vi tính.

* Chức năng của bộ nhớ: Bộ nhớ dùng để chứa các thông tin cần thiết như chương trình, dữ liệu trong quá trình máy hoạt động.

* Tổ chức của bộ nhớ:

- Các thông tin được cất trong bộ nhớ dưới dạng mã nhị phân (các bít 0 hoặc 1).

- Một nhóm các bít có thể là 16 bít hoặc 32 bít,... gọi là một ô nhớ. Mỗi ô nhớ được đánh một số thứ tự trong hệ 16 gọi là địa chỉ của ô nhớ. Thông tin ghi bên trong ô nhớ là nội dung của ô nhớ. Dùng địa chỉ để truy nhập vào ô nhớ để đọc nội dung hoặc ghi nôi dung vào ô nhớ.

- Kích thước của bộ nhớ hay dung lượng của bộ nhớ là số lượng thông tin mà bộ nhớ có khả năng chứa được, thường dùng đơn vị MB hoặc GB.

* Phân loại bộ nhớ: Theo tính chất thông tin chứa trong bộ nhớ người ta chia thành bộ nhớ ROM và RAM.

- ROM (Read Only Memory): là bộ nhớ cố định cho phép chỉ đọc thông tin mà không ghi thông tin vào được. ROM là bộ nhớ cứng do hãng chế tạo cài đặt sẵn các chương trình bên trong, bao gồm các chương trình kiểm tra và các chương trình cơ sở cốt lõi nhất của máy vi tính. Các thông tin trong ROM sẽ không bị mất đi khi ta tắt nguồn của máy.

- RAM (Random Access Memory): Là bộ nhớ mềm, có thể thay đổi, truy nhập một cách ngẫu nhiên. RAM làm từ các mạch vi mạch (gọi là Chip nhớ). RAM dùng để ghi chương trình của hệ điều hành nạp vào từ đĩa khởi động, chương trình và dữ liệu của người sử dụng. Có thể đọc và ghi thông tin vào RAM. Thông tin trong RAM sẽ bị mất đi khi ta tắt nguồn của máy, do vậy ta phải lưu trữ thông tin ra bộ nhớ ngoài.

2.3- Thiết bị ngoại vi

a) Bàn phím (*Keyboard*)

* Bàn phím là thiết bị vào thông dụng của máy vi tính. Dùng bàn phím có thể đưa vào máy các lệnh điều khiển, chương trình, dữ liệu.

* Nguyên tắc của bàn phím: ấn một phím thì gây ra sự tiếp điện giữa một dây dọc và một dây ngang tạo ra một xung điện , xung điện này qua chương trình điều khiển bàn phím sẽ đưa vào máy mã ASCII của kí tự của phím đó.

* Bàn phím gồm 5 khu vực sau:

- Khu phím máy chữ gồm các phím chữ, phím dấu và phím số như các phím trên bàn phím máy chữ trong đó có phím cách (Space bar)

- Khu phím chức năng F1,F2,..F12 có tác dụng thực hiện một chức năng hoàn chỉnh nào đó khi ta gõ phím đó, tác dụng của nó được cài đặt tuỳ theo chương trình. Thí dụ DOS dùng phím F1 để nhắc lại từng kí tự của câu lệnh vừa gõ trước, F3 để nhắc lại toàn bộ câu lệnh vừa gõ.

- Khu vực phím số ở bên phải bàn phím

- Khu phím dịch chuyển con trỏ màn hình gồm các phím dịch lên, dịch xuống, dịch trái, dịch phải, dịch về đầu màn hình (Home), dịch xuống cuối (End), dịch lên một trang (PgUp), dịch xuống một trang (PgDn), Dịch trái và xoá (Back space), phím TAB dịch con trỏ đi một khoảng 8 cột (gọi là một tab)

- Khu phím điều khiển:

ESC để thoát ra khỏi chương trình hoặc lệnh đang thực hiện

CapsLock để chuyển đổi thường xuyên kiểu chữ nhỏ sang to và ngược lại

Shift có tác dụng chuyển tạm thời chữ to thành nhỏ hoặc ngược lại khi đè giữ phím đó và gõ chữ . Với các phím có 2 kí tự thì nếu gõ được kí tự dưới, nếu đè shift và gõ thì cho kí tự trên.

Insert - chèn kí tự vào vị trí con trỏ

Delete - xoá kí tự tại vị trí con trỏ

Print Screen - In màn hình

Scroll Lock - bật hoặc tắt trạng thái cuốn màn hình

Pause/Break - Chờ hoặc ngắt chương trình

Ctrl và Alt dùng để mở rộng chức năng của bàn phím.

Num Lock - để kích hoạt bàn phím số, bàn phím số chỉ có tác dụng khi nào đèn Num Lock sáng.

b) Màn hình (Display*)*

* Màn hình là thiết bị ra thông dụng tối thiểu của may vi tính. Các kí hiệu ta gõ vào từ bàn phím, các kết quả xử lý, các thông báo của máy vi tính đều được hiện ra trên màn hình.

* Màn hình có cấu tạo vật lý giống như một màn hình vô tuyến thông thường cũng gồm có ống phóng tia điện tử, mạch quét ngang và quét dọc, tia điện tử đập vào màn hình tạo thành một điểm sáng, bộ điều khiển CRT tạo tín hiệu bật hoặc tắt tia điện tử theo tín hiệu lấy ra từ bộ nhớ màn hình 1 hay 0, tia điện tử quét từ trái sang phải tạo thành một dòng, từ dòng trên cùng xuống dòng dưới cùng tạo thành một mành. Tần số quét mành bằng 60Hz nghĩa là trong 1 giây đồng hồ màn hình được làm mới lại từ đầu 60 lần, thời gian hồi dọc để tia điện tử chuyển về góc trái trên cùng là 1,25ms

* Có 2 kiểu màn hình: kiểu màn hình văn bản và màn hình đồ hoạ:

- Màn hình văn bản (Text mode) được phân thành 2 chế độ: 80 cột (kí tự) 25 dòng là dạng chuẩn hoặc 40 cột (kí tự) 25 dòng.

- Màn hình Đồ hoạ (Graphic Mode) được thiết kế gồm nhiều điểm sáng (Pixel) theo hai chiều ngang và dọc. Màn hình đồ hoạ được phân thành nhiều chế độ khác nhau tuỳ theo độ phân giải, chẳng hạn chiều ngang 640 pixel chiều dọc 200 pixel (640x200), 640x350, 640x480, 1024x768 . Màn hình càng có độ phân giải cao thì hình ảnh càng min.

- Mầu của màn hình được tạo ra bởi sự pha trộn của 4 yếu tố sau: 3 mầu cơ bản là đỏ (Red), xanh lá cây (Green), xanh da trời (Blue) và độ chói. Với tổ hợp của 3 mầu cơ bản ở độ chói khác nhau của từng mầu mà máy tính có thể tạo ra tới 256 mầu khác nhau.

c) Ổ đĩa và đĩa từ

Ổ đĩa và đĩa từ là thiết bị vào / ra : Có thể đọc thông tin từ đĩa từ và ghi thông tin vào đĩa từ.

* ổ đĩa là khối hình chữ nhật có chứa các thiết bị và linh kiện để làm việc với đĩa từ. Máy vi tính có 2 loại ổ đĩa từ là ổ đĩa cứng (Hard Disk/ Fixed Disk) và ổ đĩa mềm (Soft Disk/Flopy Disk).

- ổ đĩa cứng được lắp trong thân máy, nếu máy có 1 ổ đĩa cứng thì tên ổ đĩa là C: , nếu máy có 2 ổ đĩa cứng thì tên là C: , D: . ổ đĩa cứng chứa các đĩa cứng có dung lượng lưu trữ lớn từ 40 MB đến hàng chục GB.

- ổ đĩa mềm lắp ngay bên ngoài vỏ máy, 2 ổ mềm có tên là A: , B: , ổ đĩa mềm chứa các đĩa mềm có dung lượng lưu trữ nhỏ hơn đĩa cứng, hiện nay thông dụng là 1,44MB.

* Đĩa từ có hình dạng tròn được cấu tạo bằng chất dẻo (đĩa mềm) hoặc nhôm (đĩa cứng) có phủ các hạt sắt từ trên bề mặt để lưu trữ thông tin dựa theo đặc tính nhiễm từ của các hạt sắt từ. Các thông tin lưu trữ trên đĩa từ ở dạng mã nhi phân (bít 0 hoặc 1).

- Mỗi đĩa từ mềm gồm có hai mặt từ hoá được đánh số là mặt 0/1, có 2 đầu từ tiếp xúc trên dưới để đọc và ghi thông tin, mỗi mặt phân thành nhiều rãnh (Track) đánh số từ 0 trở đi ,rãnh ngoài gần mép đĩa là rãnh số 0, mỗi mặt được phân thành nhiều cung (sector) được đánh số từ 1 trở đi, (đĩa 1,2M phân thành 80 rãnh và15 sectơ, đĩa 1,44 M phân thành 80 rãnh và 18 sectơ), mỗi cung chứa 512 byte.

- Đĩa cứng gồm nhiều mặt đĩa cùng trục quay tạo thành khối trụ gọi là cylinder, số mặt tuỳ theo dung lượng của ổ đĩa, chẳng hạn ổ đĩa có 8 đầu từ (Head) đánh số từ 0 đến 7 gồm 4 mặt đĩa, mỗi mặt của đĩa cứng được phân thành 17 sectơ, mỗi mặt gồm nhiều rãnh (Đĩa cứng XT gồm 305 rãnh, đĩa cứng AT gồm 615 rãnh), mỗi rãnh trên một cung chứa 1024 byte.

d) Máy in *(*Printer*)*

* Máy in là thiết bị ra của máy vi tính, dùng để in ra kết quả xử lý, dữ liệu, chương trình, thông báo của máy vi tính.

* Máy in dùng trong máy tính gồm nhiều loại khác nhau hoạt động theo các nguyên lý khác nhau. Ta có thể phân thành các loại cơ bản sau:

- Máy in mầu (Ploter) gồm 8 bút vẽ màu khác nhau hoạt động trên nguyên tắc bút vẽ.

- Máy in Lazer hoạt động trên nguyên tắc dùng tia Laeser ép nóng chảy bột mực khô bám trên lô

- Máy in kim (9 kim hoặc 24 kim) hoạt động trên nguyên tắc kim phun bắn vào băng mực và tạo ra trên giấy các nốt chấm đen.

Hiện nay máy in Laser được sử dụng rộng rãi vì có ưu điểm là in nhanh và chất lượng tốt nhưng giá đắt hơn máy in kim. Máy in kim giá rẻ hơn và bền hơn nhưng chất lượng không cao và in chậm hơn.

e) Thiết bị chuột

Chuột là một thiết bị vào của máy vi tính. Có thể chọn lệnh, chọn các đối tượng, di chuyển và thay đổi kích thước các đối tượng bằng chuột.

Thông thường chuột có 2 nút bấm. Nút trái dùng cho phần lớn các thao tác, nút phải tuỳ theo từng chương trình mà có các chức năng khác nhau, thông thường nhấn nút phải để hiện ra các lệnh có thể thực hiện được tại vị trí vừa chọn của chuột. lệnh này tuỳ thuộc phần mềm đang sử dụng.

f) Các thiết bị khác

Các thiết bị ngoại vi của máy vi tính hiện nay rất phong phú, ngoài các thiết bị cơ bản nêu trên còn các thiết bị khác như : Đĩa CD-ROM, máy chiếu (Projecter), máy quét ảnh

(Scaner), ... Tất cả các thiết bị ngoại vi giúp cho việc đưa thông tin vào máy ở nhiều dạng từ các kí tự đến âm thanh, hình ảnh.

3- Tổ chức một phòng máy tính và cách bảo quản

3.1- Tổ chức một phòng máy

Khi thiết lập một phòng máy tính, cần chọn vị trí thoáng mát, tránh ánh nắng mặt trời, tránh nơi có nam châm và từ trường. Không nên đặt máy gần cửa sổ, nhà bếp. Nên đặt máy vi tính cách xa tường để tránh ẩm và nên để máy sao cho những người ngồi làm máy không quay lưng vào đuôi máy. Nhìn chung, máy tính nên được chạy qua ổn áp để tránh những thay đổi đột ngột điện áp làm ảnh hưởng đến máy tính. Nên lắp điều hoà, máy hút ẩm trong phòng máy để đảm bảo nhiệt độ và độ ẩm trong phòng được chuẩn. Tốt nhất là từ 20^0 C đến 25^0C. Trong phòng máy nên trải thảm cách điện và máy nên nối đất để khử tĩnh điện.

Trong phòng máy phải dùng các biện pháp để ngăn chặn bụi bặm thâm nhập vào máy tính. Vì vậy, không nên đi giày, dép vào phòng máy tính, không hút thuốc trong phòng máy, không để các chất hay gây cháy nổ trong phòng máy. Không để kiến, gián và các loại côn trùng khác vào phòng máy tính.

3.2 Bảo quản đĩa từ, màn hình, bàn phím

*Đối với đĩa từ :

- Đĩa cứng: Tuổi thọ của đĩa cứng thường từ 8.000 đến 20.000 giờ. Đầu máy phải đặt trên bàn bằng phẳng, vững chắc, không lung lay. Khi máy đang vận hành, nếu vô ý đập manh vào đầu máy, hoặc vô ý làm lung lay mạnh thì đầu từ sẽ đập mạnh vào đĩa cứng làm bong lớp từ phủ ngoài hoặc chính đầu từ bị vỡ hoặc lệch dẫn đến tình trạng ổ cứng không sử dụng được.

Khi nhiệt độ tăng cao thì lớp từ tính trên mặt đĩa sẽ bị bong ra hoặc đĩa bị cong. Trong trường hợp này, dữ liệu trên đĩa cứng không thể lấy ra được và bạn cũng không truy nhập thông tin lên đĩa được.

Khi di chuyển đầu máy, để đảm bảo an toàn cho đĩa cứng, bạn cần chạy chương trình "Park" để di chuyển đầu từ ra mé ngoài của đĩa cứng vì ở vị trí này không có dữ liệu.

Nhìn chung, bạn nên thường xuyên kiểm tra tình trạng của đĩa cứng bằng các chương trình tiện ích như Scandisk, NDD…

Khi mở máy, bạn phải tuân thủ các nguyên tắc về vận hành máy như sau: cắm ổn áp vào nguồn lưới, bật ổn áp, bật công tắc ở đầu máy. Khi tắt máy, bạn làm động tác ngược lại: tắt máy tính, tắt ổn áp, rút phích cắm ổn áp ra khỏi điện lưới.

- Đĩa mềm thường có chất lượng không được tốt bằng đĩa cứng, Vì vậy, nếu bạn lưu trữ dữ liệu trên đĩa mềm thì cần phải thường xuyên kiểm tra tình trạng của đĩa và dữ liệu được lưu trữ trên đĩa. Khi dùng đĩa, bạn không được chạm tay vào phần phía trong của đĩa mà chỉ được cầm phần vỏ nhựa bên ngoài. Không để bụi rơi vào phần lõi nhựa bên trong, không để đĩa gần nơi có từ tính, tránh xa các nơi có tia X quang. Không để vật nặng đè lên đĩa, không được để đĩa bị cong, tránh tuyệt đối để đĩa mềm nơi ẩm ướt, nhiệt độ cao. Tránh va chạm mạnh vào ổ đĩa. Khi di chuyển máy, cần có đĩa giả đưa vào ổ đĩa để đầu từ không bị lắc, rung. Khi đĩa bị bẩn do bụi, bạn không nên đưa đĩa vào ổ vì có thể sẽ làm bẩn đầu từ hoặc thậm chí có thể làm hỏng đầu từ.

*Màn hình:Khi không dùng máy, bạn phải phủ kín máy để che bụi, khói và nhất là chất lỏng bám vào. Không được bít các lỗ thông hơi ở hai bên hông và phía trên màn hình. Thường

xuyên lau chùi bụi bặm ở các khe rãnh thông hơi. Bạn cũng cần lưu ý rằng: khi làm việc, màn hình và đầu máy phát nhiệt nhiều nên hút bụi rất mạnh.

Khi làm việc, điện thế trong màn hình có thể lên tới hàng nghìn vôn nên có thể gây chết người trong giây lát. Vì vậy, khi máy đang làm việc, bạn không được mở nắp nhựa, tránh để những nơi ẩm thấp. Bạn không nên để màn hình ở những vị trí có thể bị nước mưa hắt vào. Không được đụng chạm mạnh vào phía trước màn hình. Khi vận chuyển màn hình khỏi khu vực làm việc, bạn bắt buộc phải cho màn hình vào hộp có đệm xốp.

* Bàn phím: ở dưới các phím đều có các lò xo để đàn hồi khi gõ phím. Vì vậy, bạn không nên gõ mạnh lên mặt phím. Phải đậy bàn phím khi không dùng tới. Không được để bàn phím bị bụi bẩn, chất lỏng hoặc nước hắt vào sẽ làm bàn phím bị kẹt, không sử dụng dược. Khi bảo quản bàn phím, bạn nên dùng cồn để lau mặt bàn phím nhưng không được để cồn chảy xuống phía dưới bàn phím.

Câu hỏi ôn tập chương II

1. Nêu các chức năng cơ bản của máy vi tính.
2. Tại sao CPU lại là bộ não của máy vi tính? Các thành phần của CPU.
3. Để chứa các thông tin cần thiết khi máy làm việc thì cần có bộ phận nào? Tại sao phải lưu trữ thông tin ra bộ nhớ ngoài?
4. Tại sao lại gọi là các thiết bị ngoại vi? Trình bày các thiết bị ngoại vi thông dụng của máy vi tính.
5. Cách bảo quản đĩa từ, màn hình, bàn phím.

CHƯƠNG III: HỆ ĐIỀU HÀNH

Để sử dụng máy vi tính trước tiên phải hiểu và biết sử dụng bộ chương trình điều khiển các hoạt động chung của máy vi tính. Bộ chương trình đó chính là hệ điều hành. Chương này tình bày các kiến thức chính về hệ điều hành: Khái niệm hệ điều hành, các hệ điều hành thông dụng hiện nay, hệ điều hành Microsoft Windows 98/2000, khởi động và các thành phần cơ bản của windows 98/2000, thay đổi các biểu tượng hoặc mục chọn trong Windows 98/2000, sử dụng Windows Explorer trong windows 98/2000.

1 - Khái niệm hệ điều hành

1.1 - Khái niệm hệ điều hành

* *Hệ điều hành* là hệ thống các chương trình điều khiển các hành vi cơ bản của dàn máy vi tính. Chỉ khi hệ điều hành được nạp vào trong bộ nhớ thì máy tính mới hoạt động.

Hệ điều hành mục đích giúp người sử dụng máy tính dễ dàng và hiệu quả.

* Chức năng cơ bản của Hệ điều hành:

- Hệ điều hành điều khiển tất cả hoạt động của máy tính và các thiết bị ngoại vi.
- Hệ điều hành là người thông dịch, cầu nối giữa người sử dụng và máy vi tính.

1.2 - Phân loại hệ điều hành

Có thể chia hệ điều hành thành 2 loại: Hệ điều hành máy tính cá nhân, hệ điều hành mạng

* Hệ điều hành máy tính cá nhân là hệ điều hành viết để điều khiển một máy tính riêng lẻ còn gọi là máy tính cá nhân.

Các hệ điều hành máy tính cá nhân thông dụng như: MS-DOS, WINDOWS 95, WINDOWS 98, WINDOWS 2000, WINDOWS XP

* Hệ điều hành mạng là hệ điều hành viết để điều khiển một mạng máy tính bao gồm 1 máy chủ kết nối với các máy trạm, hệ điều hành được cài đặt trong máy chủ.

Theo khoảng cách địa lý có thể phân ra các loại mạng máy tính sau**: Mạng** cục bộ, mạng đô thị, mạng diện rộng, mạng toàn cầu.

- Mạng cục bộ (LAN - Local Area Network): là mạng được cài đặt trong phạm vi tương đối nhỏ hẹp như trong một toà nhà, một xí nghiệp...với khoảng cách lớn nhất giữa các máy tính trên mạng trong vòng vài km trở lại.
- Mạng đô thị (MAN - Metropolitan Area Network): là mạng được cài đặt trong phạm vi một đồ thị, một trung tâm văn hoá xã hội, có bán kính tối đa khoảng 100 km trở lại.
- Mạng diện rộng (WAN - Wide Area Network): là mạng có diện tích bao phủ rộng lớn, phạm vi của mạng có thể vượt biên giới quốc gia thậm chí cả lục địa.
- Mạng toàn cầu (GAN - Global Area Network): là mạng có phạm vi trải rộng toàn cầu đó là Internet

Các hệ điều hành mạng thông dụng hiện nay là: WINDOWS NT, UNIX, WINDOWS 2000 SERVER

1.3 - Một số hệ điều hành thông dụng

*** Hệ điều hành MS-DOS**

Hệ điều hành MS DOS là một hệ điều hành ra đời cách đây khá lâu và rất phổ dụng trước khi có sản phẩm cùng hãng của nó là hệ điều hành WINDOWS ra đời.

DOS quản lý, lưu trữ thông tin dưới dạng các tập tin và thư mục.

Giao diện của DOS với người sử dụng là giao diện dòng lệnh.

*** Hệ điều hành WINDOWS 98/2000**

Hệ điều hành WINDOWS 98/2000 do hãng phần mềm MICROSOFT phát hành. Đây là một hệ điều hành theo phong cách hoàn toàn mới và nó nhanh chóng trở thành một trong những hệ điều hành phổ dụng và được yêu thích nhất hiện nay.

Trước kia để làm việc được với hệ điều hành MS DOS, cần phải nhớ rất nhiều lệnh với cú pháp dài dòng và rắc rối, cùng với việc phải đối diện với một màn hình tối om sẽ làm cho công việc trở nên nhàm chán. Hệ điều hành WINDOWS ra đời, tương thích với hệ điều hành MS DOS, đã mang lại rất nhiều tiện lợi trong việc sử dụng. Hệ điều hành WINDOWS vì sử dụng giao diện đồ họa do đó rất dễ sử dụng. Một số đặc điểm nổi trội của WINDOWS 98/2000

- Cung cấp một giao diện đồ hoạ người-máy thân thiện (GUI- Graphic User Interface).

- Cung cấp một phương pháp điều khiển thống nhất cho mọi ứng dụng trên môi trường WINDOWS 95

- Hoạt động ở chế độ đa nhiệm

- Môi trường Nhúng - Liên kết các đối tượng (OLE - Object Linking and Embeding)

- Tự động nhận dạng và cài đặt trình điều khiển các thiết bị (Plus and Play).

- Hỗ trợ mạng.

*** Hệ điều hành WINDOWS NT**

Windows NT là hệ điều hành mạng được người dùng tin cậy. Qua sử dụng, nó chứng tỏ là hệ điều hành mạng tích hợp nhiều tính năng như các giao thức truyền tin chuẩn, tính năng tìm đường (routing), truy cập từ xa, tốc độ, bảo mật mức C2, giao diện đơn giản, dễ quản trị, đặc biệt đây là hệ điều hành nền tảng cho rất nhiều chương trình và ứng dụng phổ biến hiện nay như MS SQL Server. MS Mail Server, MS Exchange Server 4.0, Internet Infomation Server (bao gồm FTP Server, Gopher Server, W.W.W. Server), MS Proxy Server, ...

Windows NT là hệ điều hành 32 bit, đa nhiệm có ưu tiên nhằm khai thác hết khả năng của các bộ vi xử lý như Intel x86, RISC và các hệ thống đa xử lý đối xứng (symmetric multiproccessing system).

Bên ngoài là một giao diện người sử dụng giống như Windows, Windows NT đã được thiết kế lại phần hạt nhân (kernel) cần thiết với các hệ điều hành đã có Windows NT thực hiện được hầu hết các chương trình đang chạy trên các họ máy tính x86 và RISC dưới MS-DOS, Windows, MS OS/2 version 1.x và các ứng dụng cùng với các chức năng phát triển về bảo mật và quản trị.

Tính mở: Để duy trì tương thích ví dụ với các ứng dụng viết trên UNIX theo chuẩn POSIX. Tuy nhiên để tận dụng hết khả năng của **WINDOWS NT** ta cần thực hiện các ứng dụng viết riêng cho hệ điều hành 32 bit này mà MS SQL Server là một ví dụ.

*** Hệ điều hành UNIX**

UNIX là hệ điều hành mạng đa nhiệm, đa người dùng. UNIX với sức mạnh và tính tin cậy truyền thống, cộng thêm những khả năng mũi nhọn về truyền thông, kết nối mạng qua TCP/IP, thư điện tử, cơ sở dữ liệu, tính an toàn cao. Đặc biệt, các ứng dụng chuyên nghiệp đều được viết rất hoàn thiện trên UNIX.

Hệ điều hành UNIX đã được phát triển tại phòng thí nghiệm AT&T Bell tại Murray bang New Jersey - một trong những phòng nghiên cứu lớn nhất trên thế giới. Từ khi phiên bản đầu tiên của hệ điều hành UNIX được Ken Thompson thiết kế năm 1969, nó đã trải qua

một quá trình phát triển và ngày càng hoàn thiện. Khi hệ điều hành UNIX đã phát triển, rất nhiều máy tính vẫn còn chạy ở chế độ đơn nhiệm, nghĩa là máy tính chỉ phục vụ được một người trong cùng một khoảng thời gian, do đó người sử dụng không khai thác hết được năng lực cũng như tốc độ của máy tính. Hơn thế nữa, môi trường làm việc của các lập trình viên bị cô lập với các lập trình viên khác. Điều đó tạo nên sự khó khăn trong việc chia xẻ dữ liệu và chương trình, điều đó giảm năng suất làm việc của những người làm việc trong cùng một dự án hay cùng một mục đích. Hệ điều hành UNIX ra đời đã cung cấp ba tiến bộ chủ yếu so với hệ thống đơn nhiệm cũ là:

- UNIX cho phép nhiều hơn một người có thể sử dụng máy tính hoặc nhiều chương trình cùng xử lý trong cùng một lúc (đa nhiệm).

-UNIX cho phép từng cá nhân có thể thông tin trực tiếp với các máy tính khác thông qua thiết bị đầu cuối .

- Cuối cùng UNIX làm cho sự chia xẻ dữ liệu và chương trình giữa các cá nhân với nhau dễ dàng hơn.

Hệ điều hành UNIX được xây dựng trên bốn phần chính bao gồm:

- Phần lõi (*kernel)*
- Hệ thống tệp *(file system)*
- Phần vỏ (*shell)*
- Các lệnh (*commands)*

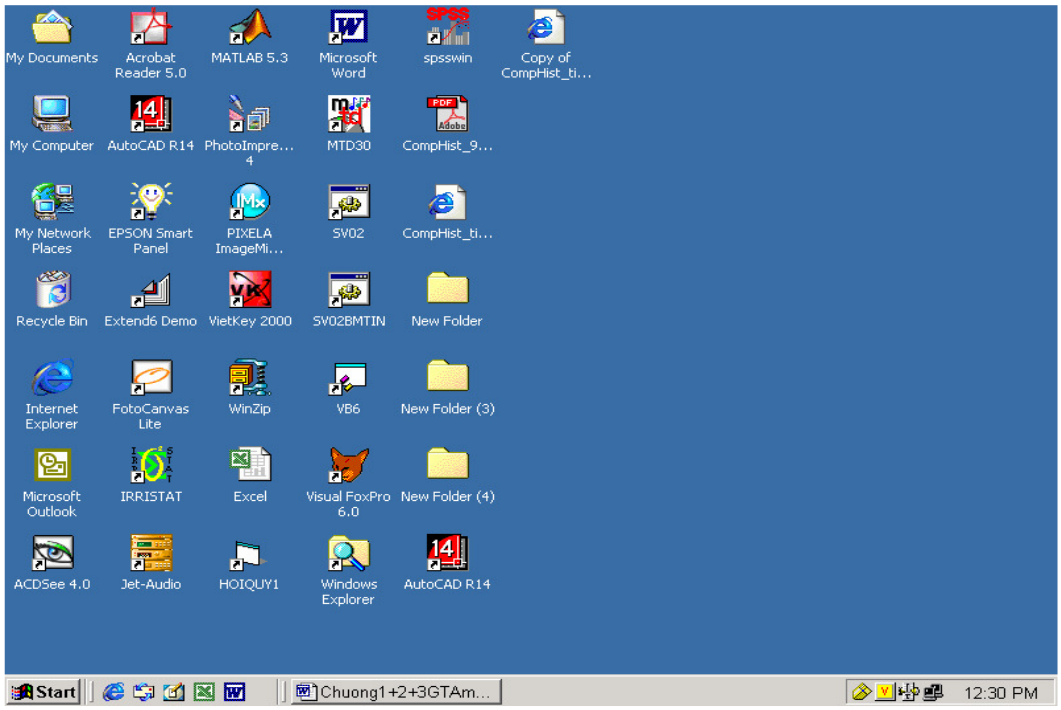
Hiện nay có rất nhiều hệ điều hành UNIX do nhiều hãng phát triển.

2 - Hệ điều hành Microsoft windows 98/2000

2.1 - Khởi động máy và các thành phần cơ bản

a) Khởi động hệ điều hành

Hệ điều hành được cài đặt trong ổ C: của máy vi tính. Khi mở máy (ấn nút Power) thì hệ điều hành được nạp vào trong bộ nhớ của máy vi tính, khi nạp xong sẽ hiện ra màn hình chính của Windows (Desk top) như sau (hình 1.3)

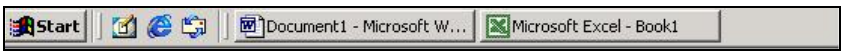


Hình 1.3

- Phía trên là các biểu tượng của các trình ứng dụng. Khi muốn mở 1 trình ứng dụng ta chỉ việc kích đúp chuột vào biểu tượng của nó.
- Dưới cùng là thanh ứng dụng TASKBAR

b) Thanh ứng dụng TASKBAR

Thanh ứng dụng Taskbar như sau:



Thanh ứng dụng này chứa nút Start và danh sách các ứng dụng đã kích hoạt. Tất cả các cửa sổ ứng dụng khi đưa về chế độ cực tiểu đều xuất hiện trên thanh Taskbar. Khi này, các bạn chỉ cần nháy đúp chuột lên biểu tượng của nó trên thanh này.

- *Di chuyển Taskbar*

Thanh Taskbar có thể thay đổi ở các vị trí khác nhau trên màn hình. Muốn thay đổi vị trí, các bạn chỉ cần thao tác nháy và kéo lên nền của Taskbar. Tại nơi xuất hiện mũi tên hai chiều, các bạn nháy và kéo lên các biên phía trong để thay đổi kích thước của Taskbar.

- *Các thao tác chính.*

Khi bạn nháy nút phải chuột lên nền của Taskbar sẽ xuất hiện bảng chọn.

Các mục chọn trong bảng chọn có ý nghĩa như sau:

+ Toolbars: Khi đưa con trỏ đến mục chọn này thì hộp thoại xuất hiện.

Nếu chọn Desktop thì các biểu tượng trên màn hình chính của Windows xuất hiện trên thanh Taskbar.

Nếu chọn New Toolbar thì hộp thoại New Toolbar xuất hiện. Giả sử bạn chọn Printer thì hộp thoại Printer xuất hiện theo dạng rút gọn trên thanh Taskbar.

+ Cascade Windows: Các cửa sổ đang mở xếp lợp lên nhau.

+ Tile Windows Horizontally: Các cửa sổ ứng dụng xếp lợp lên nhau theo chiều ngang.

+ Tile Windows Vertically: Các cửa sổ ứng dụng xếp cạnh nhau theo chiều đứng.

+ Minimize All Windows: Cực tiểu hóa các cửa sổ đang mở và đưa về Taskbar.

+ Properties: Định nghĩa cách thể hiện của Taskbar và thay đổi các thành phần trong Start Menu Programs. Khi chọn mục này, hộp thoại Taskbar Properties xuất hiện. Các mục chọn trong hộp thoại này có các chức năng như sau:

Always on top: Taskbar luôn xuất hiện trên màn hình ở tất cả các ứng dụng.

Auto hide: Taskbar bị che dấu đến khi con trỏ chuột di chuyển đến vị trí của nó.

Show small icon in Start menu: Thay đổi kích thước các biểu tượng trên bảng chọn Start.

Show Clock: Hiện hay ẩn đồng hồ trên Taskbar.

c) Nút Start

Nút Start thực hiện nhiều chức năng quan trọng: Thực hiện các ứng dụng, mở tệp, gọi cửa sổ trợ giúp, tìm tệp, thoát khỏi Windows 98, … và các công cụ thiết lập cấu hình khác.

Khi bạn nháy chuột ở nút Start, trên màn hình xuất hiện bảng chọn Start Menu Programs. Tùy theo việc cài đặt mà trong bảng chọn Start sẽ xuất hiện các mục khác nhau.

Bảng chọn Start có dạng như sau (Hình 2.3)

Hình 2.3

Các mục chọn chính trong bảng chọn này có chức năng như sau:

- *Programs*: Thực hiện các chương trình ứng dụng trên Windows.
- *Documents*: Mở các tài liệu văn bản, đồ họa.
- *Settings*: Thiết lập cấu hình Control Panel, máy in, Taskbar.
- *Find* / Search: Tìm kiếm tệp hoặc thư mục.
- *Help*: Mở cửa sổ trợ giúp.
- *Run*: Chạy các tệp chương trình.
- *Shut Down*: Khi chọn mục này, một mục chọn xuất hiện, bạn có thể chọn một trong các chức năng:

Shut down Tắt máy.
Restart Khởi động lại hệ điều hành Windows.
Stand by Khởi động lại Windows và vào chế độ MS – DOS.

d) Mở trình ứng dụng

Để mở 1 trình ứng dụng ta có thể thực hiện 1 trong 3 cách sau:

Cách 1: Kích đúp chuột vào biểu tượng của trình ứng dụng trên màn hình chính của Windows

Cách 2: Chọn nút START/ chọn mục PROGRAMS/ chọn trình ứng dụng

Cách 3: Chọn nút START/ chọn mục RUN / chọn BROWSE/ chọn thư mục và trình ứng dụng/ chọn OK

e) Tắt máy

Chọn nút **START/ SHUTDOWN / SHUTDOWN/ OK**
Sau đó tắt màn hình.

f) Kkởi động lại

Trong quá trình làm việc máy bị treo (không hoạt động tiếp), để khởi động lại máy ta ấn nút RESET (hoặc ấn đồng thời 3 phím CTRL, ALT, DEL).

2.2 - Tạo, thay đổi các biểu tượng hoặc mục chọn cho một nội dung

a) Tạo biểu tượng hoặc mục chọn cho một chương trình độc lập.

Đối với các chương trình độc lập, chẳng hạn như: Microsoft Excel, Microsoft Word, Foxpro, ta có thể tạo một mục chọn nằm trong một bảng chọn nào đó hoặc tạo một biểu tượng trên màn hình Windows để khi chọn mục chọn hoặc biểu tượng tương ứng thì chương trình thực hiện mà không phải thực hiện các thao tác tìm kiếm phức tạp.

Trình tự thực hiện như sau:

- Nhấn nút Start.
- Chọn mục Settings. Khi này, một bảng chọn xuất hiện.
- Chọn Taskbar. Khi này bảng chọn Taskbar Properties xuất hiện.
- Chọn Start menu Programs, nhấn nút Add… trong khung Customize Start Menu. Khi này hộp Create Shortcut xuất hiện.
- Nhấn nút Browse…
- Chọn thư mục chứa chương trình.
- Chọn tệp chương trình .
- Chọn Next. Khi này, bảng chọn Select Program Folder xuất hiện.
- Trong khung Select folder to place shortcut in, ta phải chọn vị trí để chứa biểu tượng:

+ Nếu ta nhấn chuột trên Desktop, biểu tượng sẽ xuất hiện trên màn hình chính của Windows .

+ Nếu chọn Programs, biểu tượng sẽ xuất hiện trong khung Programs.

+ Nếu chọn Start Menu, mục chọn được đặt trong khung Start.

+ Nếu chọn StartUp, chương trình sẽ được thực hiện ngay sau khi khởi động Windows.

+ Nếu chọn New Folder, Windows sẽ tạo một bảng chọn mới trong bảng chọn Programs để chứa biểu tượng của chương trình.

- Sau khi chọn xong, bảng chọn Select a title for the Program xuất hiện.
- Trong khung Select a name for the Shortcut, ta cần đưa tên cho mục chọn này, nếu ta không đưa tên mới vào khung trên thì Windows sẽ đưa tên ngầm định
- Nhấn nút **Finish** kết thúc.

b) Thay đổi biểu tượng và tên của biểu tượng

* Thay đổi biểu tượng:

Giả sử cần thay đổi biểu tượng cho chương trình đặt trong bảng chọn Programs, các bước tiến hành như sau:

- Đưa con trỏ đến biểu tượng này
- Nhấn chuột phải. Khi này, hộp thoại Properties xuất hiện.
- Nhấn chuột phải vào nút Change Icon để thay đổi biểu tượng. Khi này, hộp thoại Change Icon xuất hiện:

Chọn biểu tượng cần thiết hoặc có thể nhấn nút Browse để tìm biểu tượng thích hợp trong các thư mục khác. Các tệp là biểu tượng có phần mở rộng là *.ICO. Cũng cần lưu ý rằng, trong thư mục System của Windows có tệp Shell32.dll là một tệp chứa rất nhiều biểu tượng. Bạn có thể mở tệp này và chọn biểu tượng cần thiết.

Xin lưu ý: Có thể tự tạo biểu tượng bằng cách tạo hoặc sửa một hình ảnh và đặt tên tệp có phần mở rộng là *.ICO. Khi này, hình ảnh được tạo có thể làm biểu tượng cho một mục chọn nào đó.

* *Thay đổi tên biểu tượng*

Để thay đổi tên biểu tượng tiến hành như sau:

- Nếu biểu tượng nằm trên màn hình của Windows (Desktop) thì có thể nhấn chuột vào tên biểu tượng. Khi này, con trỏ nằm ở tên biểu tượng và sửa lại tên này. Cũng có thể tiến hành như dưới đây.

Vào bảng chọn chứa biểu tượng, chọn biểu tượng cần sửa lại tên, nhấn chuột phải. Khi này, một bảng chọn xuất hiện.

Chọn mục Rename. Khi này, hộp thoại xuất hiện. Sửa lại tên biểu tượng (mục chọn) trong khung New name.

Xin lưu ý: Trong bảng chọn còn có các mục sau:

+ Chọn mục Delete sẽ xóa mục chọn (biểu tượng) đã chọn.

+ Chọn open sẽ thực hiện nội dung của mục chọn.

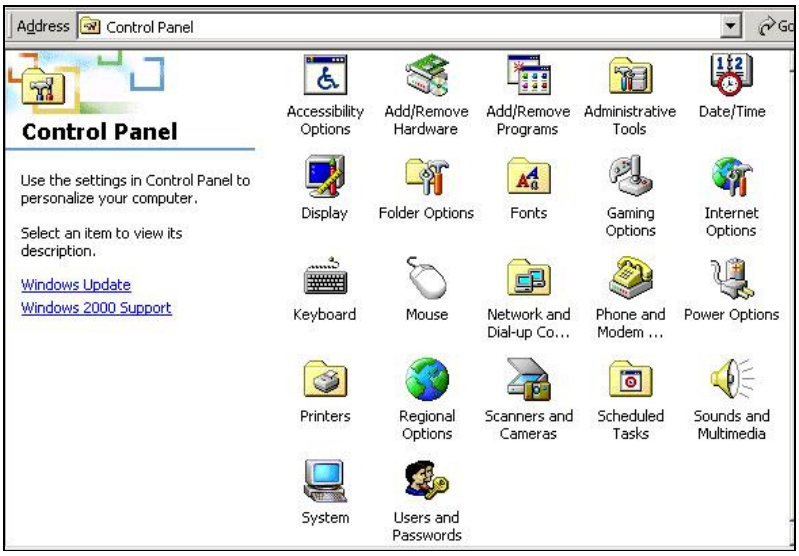
+ Chọn mục Create Shortcut sẽ tạo một biểu tượng cho mục chọn này trên màn hình chính của Windows (Desktop).

2.3 - Sử dụng hộp thoại control panel

Hộp thoại Control Panel là hộp thoại rất quan trọng dùng để cài đặt các tham số liên quan đến giao diện của Windows, phần mềm ứng dụng, phần cứng của máy …

Để mở hộp thoại Control Panel, tiến hành các bước sau:

- Vào bảng chọn Start, chọn mục Settings.
- Chọn Control Panel. Khi này, hộp thoại Control Panel xuất hiện như hình sau (Hình 3.3)



Hình 3.3

a) Thiết lập tốc độ của bàn phím và con trỏ màn hình

Trong hộp thoại Control Panel, chọn biểu tượng:

Hộp thoại xuất hiện:

- Trong khung Repeat rate: Chọn tốc độ của bàn phím bằng cách kéo thanh trượt trong khung này. Tốc độ khi gõ các phím, kéo con trỏ bằng các phím mũi tên và ngay cả khi sử dụng NC, sẽ càng nhanh khi kéo gần đến vị trí Fast và ngược lại, sẽ chậm dần khi kéo thanh trượt đến gần với Slow.
- Trong khung Cursor blink rate: Thay đổi tốc độ nhấp nháy của con trỏ. Khi thanh trượt càng gần đến vị trí cuối (Fast) thì tốc độ nhấp nháy càng nhanh. Ngược lại, khi càng gần đến vị trí đầu (Slow) thì tốc độ càng chậm.
-

b) Thiết lập các tham số cho màn hình

Trong hộp thoại Control Panel, chọn biểu tượng:

Khi này, hộp thoại Display Properties xuất hiện:

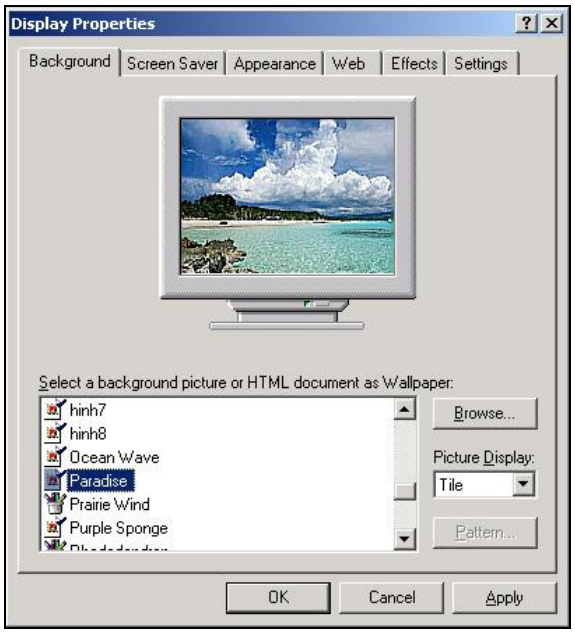
Ta chọn các mục sau:

* *Mục Background*

Mục này chọn ảnh hoặc màu nền cho màn hình.

Khi chọn khung này, hộp thoại Display Properties có dạng như hình sau.

Trong khung Wallpaper, chọn ảnh hoặc màu nền cho màn hình Windows. Có thể sử dụng nút Browse để chọn ảnh cho màn hình trong thư mục hoặc ổ đĩa khác.



Hình 4.3

* *Mục Screen Saver*

Mục này chọn hình ảnh xuất hiện khi không sử dụng chuột hay bàn phím và cài đặt mật khẩu.

Khi chọn mục này thì hộp thoại Display properties xuất hiện.

-Trong khung Screen Saver của hộp thoại này chọn một hình ảnh. Hình ảnh này sẽ xuất hiện sau một khoảng thời gian không sử dụng chuột hay bàn phím. Khoảng thời gian này tính bằng phút và được xác định trong khung Wait.

- Khi chọn nút Password protected, chỉ chọn được nút Password protected khi đã chọn một hình ảnh trong khung Screec Saver (nếu chọn mục None thì không chọn được nút này). Nhấn chuột vào nút này, hộp thoại xuất hiện.Khi đưa mật khẩu vào trong khung New password, cần nhắc lại mật khẩu này một lần nữa trong khung Confirm new password. Khi đã thiết lập chế độ này, sau một thời gian xác định trong khung Wait mà ta không sử dụng đến máy tính bằng cách tác động lên chuột hoặc các phím của bàn phím, hình ảnh đã được chọn trong khung Screen Saver sẽ xuất hiện. Nếu ấn một phím hoặc nhấn chuột thì hộp thoại xuất hiện.Ta cần phải đưa vào mật khẩu đã cài đặt để có thể làm việc với máy tính. Nếu mật khẩu đưa vào không đúng, ta không thể tiếp tục công việc.

- Khi cần xóa mật khẩu hoặc sửa đổi lại mật khẩu, bạn vào hộp thoại Change Password thực hiện tương tự như trên.

- Chọn mục Power sẽ xuất hiện:

Khung Turn off monitor, xác định khoảng thời gian mà sau khoảng thời gian này không sử dụng đến máy tính (không nhấn các phím trên bàn phím hoặc nhấn chuột) thì màn hình sẽ tự động tắt.

Tương tự với các thao tác trên đối với ổ cứng khi chọn thời gian trong khung Turn off hard disks.

* *Mục Appearance*

Khung này để chọn phông chữ, kích thước chữ cho các biểu tượng, các bảng chọn trong các ứng dụng của Windows …

Khi chọn khung này, hộp thoại Display Properties xuất hiện

-Trong khung Scheme, chọn các cặp màu cho màn hình giao diện của Windows cũng như trong các ứng dụng khác: Microsoft Excel, Microsoft Word …

-Trong khung Item, chọn phông chữ, kích thước chữ, màu sắc chữ cho các bảng chọn, các dòng thông báo, biểu tượng, …Khi nhấn chuột lên mũi tên xuống trong khung này, hộp thoại xuất hiện.

+ Chọn đối tượng cần chọn phông chữ, kích thước chữ trong hộp thoại trên. Một số đối tượng như một mục chọn trong bảng chọn, nội dung văn bản trong cửa sổ, … được miêu tả trong hộp thoại Display Properties và bạn có thể nhấn chuột lên các đối tượng này để chọn thay cho việc chọn chúng trong khung Item.

+ Chọn phông chữ trong khung Font (khung này sẽ trở thành sáng nét khi đã chọn một đối tượng).

+ Chọn kích thước chữ trong khung Size và chọn màu trong khung Color cho đối tượng đã lựa chọn.

* *Mục Effects*

Hộp thoại này dùng để thay đổi các biểu tượng (Icon) trên giao diện của Windows (Desktop).

Khi chọn khung này, hộp thoại Display Properties xuất hiện.

- Nhấn chuột vào biểu tượng cần thay đổi.
- Nhấn chuột vào biểu tượng ***Change Icon***

Khi này, một hộp thoại là danh sách các biểu tượng xuất hiện. Chọn biểu tượng cần thiết.

* *Mục* Web

Khung này để cài đặt các tham số cho các trang Web khi máy của bạn được nối với Internet.

* *Mục Settings*

Mục này khai báo số màu của nàm hình và độ phân giải của hình.

Khi chọn khung này, hộp thoại Display Properties xuất hiện.

- Trong khung Color palette, chọn số màu cho màn hình. Số màu có thể có được còn phụ thuộc vào Card màn hình mà bạn đang sử dụng.
- Thanh trượt trong khung Desktop area dùng để chọn độ phân dải cho màn hình.
- Nhấn chuột vào khung Advanced Properties để xem các thông tin về Card màn hình và màn hình. Trong trường hợp cần thiết, bạn có thể thay đổi Card màn hình khi chọn nút này.

Cuối cùng chộn OK kết thúc.

Xin lưu ý: Thay cho việc chọn biểu tượng Display trong hộp Control Panel, có thể thao tác như sau: Trên màn hình của Windows, nhấn chuột phải, chọn Properties.

c) Thay đổi ngày tháng và thời gian cho hệ thống

Trong hộp thoại Control Panel, chọn biểu tượng:

Hoặc nhấn chuột vào vị trí ghi thời gian bên phải dưới đáy màn hình. Khi này, hộp thoại Date/Time Properties xuất hiện.

- Trong khung Date: Chọn ngày, tháng, năm cho hệ thống.

- Trong khung Time: Chọn thời gian. Bạn cần chọn ba lần cho các giá trị: giờ, phút và giây.

- Trong Time zone: chọn múi giờ.

Sau khi đã thiết lập xong các tham số, nhấn nút Apply phía dưới hộp thoại để chấp nhận các giá trị.

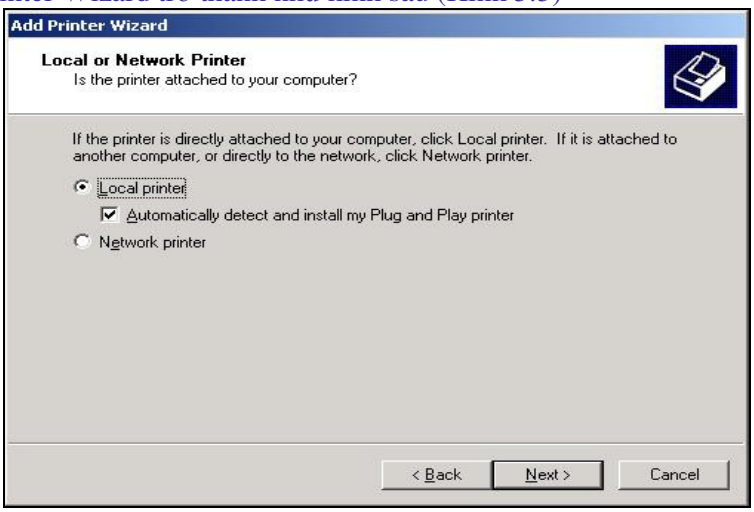
d) Cài đặt máy in

Khi sử dụng một máy in mới, bạn cần cài đặt chương trình điều khiển máy in thì máy in mới sử dụng được. Các bước tiến hành như sau: Trong hộp thoại Control Panel, chọn biểu tượng:

Khi này, hộp thoại Printers xuất hiện. Trong hộp thoại Printers, nhấn chuột lên biểu tượng

Hộp thoại Add Printer Wizard xuất hiện. Nhấn chuột vào nút

hộp thoại Add Printer Wizard trở thành như hình sau (Hình 5.3)

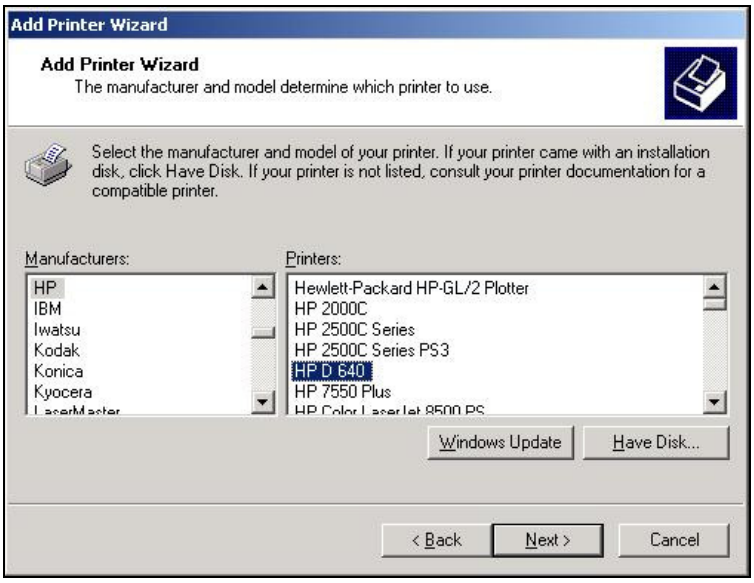


Hình 5.3

Chọn một trong hai phương án:

- Local printer: Cài đặt máy in cục bộ (được nối trực tiếp vào máy tính đang sử dụng).
- Network printer: Máy in cài đặt trên mạng.

Giả sử bạn chọn Local printer. Nhấn chuột vào nút Next. Khi này, hộp thoại Add Printer Wizard trở thành như hình sau (Hình 6.3)



Hình 6.3

Trong khung Manaufacturers, chọn nhóm máy in. Khi này, trên cửa sổ bên trái xuất hiện các máy in của nhóm này như hình trên. Chọn máy in cần cài đặt. Nhấn chuột vào nút

Khi này, hộp thoại Install From Disk xuất hiện.

Nhấn chuột vào nút OK để cài đặt chương trình điều khiển máy in.

Khi mua một máy in, nhà cung cấp bao giờ cũng đưa cho bạn một, thậm chí một vài đĩa là chương trình điều khiển máy in. Nếu là đĩa mềm bạn chỉ cần đưa đĩa vào ổ A và nhấn chuột vào nút

. Nếu chương trình điều khiển được đặt trong thư mục hoặc ổ đĩa khác, bạn nhấn chuột vào nút

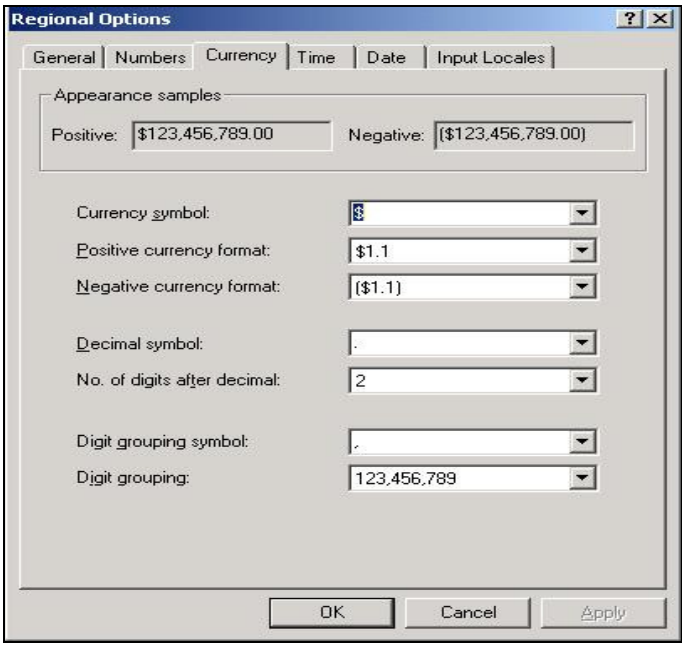
khi chọn thư mục hoặc nhấn chuột vào mũi tên xuống trong khung Copy manufacture's files from để chọn ổ đĩa cần thiết. Có một số máy in, Windows tự cài đặt chương trình điều khiển và bạn chỉ cần chọn thư mục Windows (là thư mục cài đặt chương trình Windows) trong hộp thoại trên. Các thế hệ Windows càng về sau thì khả năng này càng lớn. Khi chọn được vị trí chứa chương trình điều khiển, nhấn chuột vào

e) Thiết lập chế độ hiển thị của dữ liệu

Để thiết lập chế độ hiển thị của dữ liệu (dạng số, tiền tệ, thời gian, …), các bạn tiến hành như sau:

- Trong hộp thoại Control Panel, chọn biểu tượng

Khi này, hộp thoại Reional Settings Properties xuất hiện. Nhấn chuột vào nút Numper để chọn chế độ hiển thị cho các dữ liệu dạng số. Hộp thoại Reional Settings Properties có dạng như hình sau:



Hình 7.3

- Trong khung Decimal symbol, chọn ký hiệu biểu thị phân cách phần thập phân. Bạn có thể chọn dấu phẩy (,) và khi này, các ứng dụng của Windows như Microsoft Excel, Microsoft Word… sẽ sử dụng ký hiệu này trong các phép toán, công thức, trình bày kết quả… của dữ liệu kiểu số.

- Trong khung No.of digits after decimal: Xác định số chữ số thập phân.

- Trong khung Digit grouping symbol: Xác định ký hiệu dùng làm dấu phân cách giữa các số hàng nghìn, triệu, tỷ …

- Trong khung No. of digits in group: Xác định số chữ số tạo thành một nhóm và được phân cách bằng ký hiệu ở mục trên. Thông thường, số lượng này là 3.

Các khung còn lại dùng để thiết lập cách thể hiện số âm. Bạn có thể chọn số âm được viết:

-123.456,78

hoặc

(123.456,78)

bằng cách chọn trong khung Negative number format.

Bạn có thể nhấn chuột lên khung Currency, Time và Date trong hộp thoại Regional Settings Properties để xác lập chế độ hiển thị cho dữ liệu kiểu tiền tệ, thời gian và ngày tháng.

2.4 - Sử dụng chương trình Windows Explorer

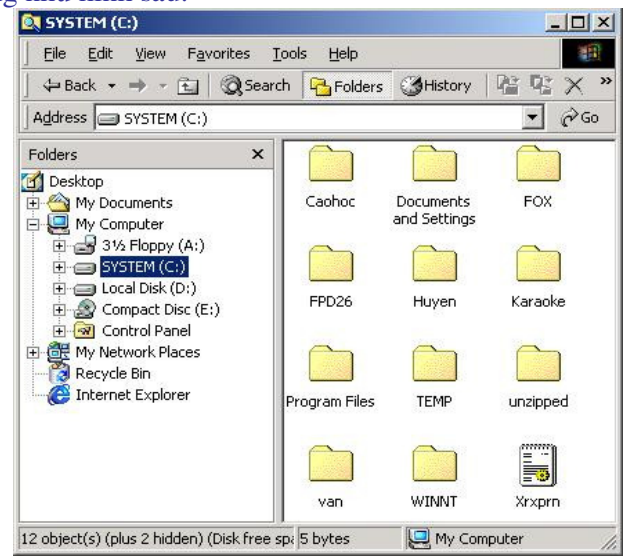
Chương trình Windows Explorer là một chương trình tiện ích nằm trong các bộ Windows 95/98/2000, … dùng để quản lý đĩa. Bạn có thể sử dụng chương trình này để sao chép, xóa, đổi tên các tệp, tạo các thư mục mới …

a) Khởi động chương trình Windows Explorer

Có thể khởi động chương trình Explorer bằng một trong 3 cách sau:

- Cách 1: Kích đúp chuột vào biểu tượng **Windows Explorer** trên Desktop
- Cách 2: Chọn nút **Start** / chọn mục **Program**/ chọn **Windows Explorer**
- Cách 3: Chọn nút **Start** / chọn mục **Run** / chọn nút **Browse** / chọn thư mục và chon chương trình ứng dụng

Sau khi khởi động bằng một trong ba cách trên, cửa sổ làm việc của Windows Explorer có dạng như hình sau:



Hình 8.3

Cửa sổ làm việc của Windows Explorer được chia thành hai cửa sổ con: Khi đưa con trỏ đến một thư mục hay ổ đĩa ở cửa sổ bên trái thì trên cửa sổ bên phải sẽ xuất hiện nội dung của thư mục hay ổ đĩa này.

b) Ra khỏi Windows Explorer

Chọn mục FILE/ chọn mục EXIT

c) Các ứng dụng của Windows Explorer

** Tạo thư mục mới*

Để tạo một thư mục mới, các bạn tiến hành các bước như sau:

- Trên cửa sổ bên trái, đưa con trỏ đến ổ đĩa hoặc thư mục cần tạo thư mục mới. Nếu con trỏ nằm trên ổ đĩa thì thư mục được tạo sẽ nằm trực tiếp trên ổ đĩa này. Nếu con trỏ nằm ở thư mục nào thì thư mục mới sẽ nằm trực tiếp trong thư mục này.
- Vào bảng chọn File, chọn New. Khi này, một hộp thoại xuất hiện.
- Chọn Folder. Khi này, trên cửa sổ bên phải xuất hiện một biểu tượng như hình sau:

Con trỏ đang nằm ở New Folder. Đổi tên thành tên thư mục cần tạo. Ví dụ sửa thành

** Xem thông tin*

- Giả sử, khi bạn đưa con trỏ vào thư mục **Giaotrinh** ở cửa sổ bên trái thì trên cửa sổ bên phải xuất hiện nội dung của thư mục Giaotrinh.
- Khi ấn vào phím + bên trái thư mục, các thư mục trực thuộc thư mục này (nằm trực tiếp trong thư mục) xuất hiện phía dưới để có thể xem tiếp nội dung của các thư mục con.
- Khi muốn xem thư nội dung của thư mục con trong một thư mục thì đưa con trỏ đến thư mục con này và nhấn chuột. Nội dung sẽ được xuất hiện trên cửa sổ bên phải bao gồm cả tệp và các thư mục con trực thuộc. Trên cửa sổ bên trái, dưới thư mục này xuất hiện danh mục các thư mục nhưng chỉ xuất hiện các thư mục trực thuộc.
- Khi cần quay trở về thư mục mẹ thì nhấn chuột lên dấu trừ (-) bên cạnh thư mục, danh mục các thư mục phía dưới của thư mục trên cửa sổ bên trái biến mất.

** Sao chép*

Giả sử cần sao chép thư mục **Excel** trong thư mục **Giaotrinh** sang thư mục **Sach**, có thể tiến hành theo hai cách như sau:

- Cách 1:

- Đưa con trỏ đến thư mục Excel, nhấn chuột phải. Khi này, một hộp hội thoại xuất hiện
- Chọn Copy.
- Nhấn chuột vào thư mục Sach, nhấn chuột phải, chọn Paste.

- Cách 2:

- Trên cửa sổ bên trái, đưa con trỏ đến thư mục Giaotrinh. Khi này, trên cửa sổ bên phải xuất hiện nội dung của thư mục Giaotrinh.
- Nhấn chuột vào thư mục Excel. Khi này, thư mục đổi màu. Chú ý: Nếu bạn nhấn đúp chuột thì sẽ xuất hiện nội dung của thư mục này trên cửa sổ bên phải.
- Vào bảng chọn Edit, chọn Copy, nhấn chuột trên thư mục Giaotrinh, vào bảng chọn Edit, chọn Paste hoặc nhấn Ctrl+V.

Xin các bạn lưu ý

Để chọn một nhóm các thư mục hoặc tệp để sao chép hoặc dịch chuyển, trước khi chọn tệp hoặc thư mục mới, nhấn và giữ phím Ctrl.

** Dịch chuyển*

Dịch chuyển một thư mục, tệp hoặc một nhóm thư mục hoặc tệp được thực hiện tương tự như khi sao chép chỉ khác là khi sao chép thì chọn Copy trong các bảng chọn còn khi dịch chuyển thì thay bằng mục chọn Cut trong các bảng chọn.

Cũng có thể tiến hành bằng cách kéo chuột như sau:

- Chọn thư mục cần dịch chuyển (có thể ở cửa sổ bên phải hoặc bên trái).
- Nhấn và giữ chuột, kéo thư mục đến thư mục cần dịch chuyển và nhả chuột (khi này, thư mục hoặc ổ đĩa cần dịch chuyển đến đổi màu).

** Xóa tệp hoặc thư mục*

Nhấn chuột vào thư mục hoặc tệp cần xóa và thực hiện một trong ba cách sau:

- Nhấn chuột phải, chọn mục Delete hoặc chọn Menu Edit/ Delete .
- Nhấn chuột vào biểu tượng Delete
- Nhấn phím Delete trên bàn phím.

Khi thực hiện một trong ba cách trên, hộp thoại Confirm Folder Delete xuất hiện, chọn Yes nếu muốn xóa, chọn No nếu muốn hủy bỏ lệnh.

** Đổi tên*

Nhấn chuột vào tệp hoặc thư mục cần đổi tên, nhấn chuột phải chọn Rename.

** Mở tệp hoặc thư mục*

- Chọn tệp hoặc thư mục cần mở. Nhấn chuột phải. Khi này, một hộp thoại xuất hiện.
- Chọn Open.
  + Nếu là thư mục thì tạo một cửa sổ để chứa nội dung của thư mục.
  + Nếu là tệp chương trình thì thực hiện chương trình này.
  + Nếu là tệp ứng dụng được tạo ra từ các chương trình hệ thống như Microsoft Excel, Microsoft Word … thì khởi động chương trình này đồng thời mở tệp được chỉ định.

** Xem thông tin của tệp hoặc thư mục*

Chọn tệp hoặc thư mục cần xem thông tin, nhấn chuột phải. Khi này, một hộp thoại xuất hiện. Chọn Properties.

** Tìm kiếm thông tin*

Tìm kiếm thông tin trong Windows 98 và Windows 2000 có khác nhau.

- Tìm kiếm trong Windows 98:

Để tìm kiếm thông tin, tiến hành các bước như sau:

+ Vào bảng chọn Tools, chọn mục Find. Khi này, một bảng chọn phụ xuất hiện.
+ Chọn mục Find Files and Folder. Khi này hộp thoại Find xuất hiện.
+ Chọn mục Name & Location: Trong khung Named, chọn tên tệp cần tìm. Bạn có thể sử dụng nhiều ký hiệu dấu sao (*) cho một nhóm ký tự bất kỳ và dấu hỏi (?) cho một ký tự bất kỳ.
+ Trong khung Containing text, đưa vào đoạn văn bản có chứa trong tệp cần tìm.
+ Trong khung Look in, đưa tên ổ đĩa hoặc thư mục cần tìm kiếm.
+ Nhấn nút Find Now khi đã đưa các thông tin trên. Bạn có thể tìm theo một chỉ tiêu về tên tệp hoặc đoạn văn bản chứa trong tệp (khi này, chỉ tiêu kia để trống) hoặc kết hợp hai chỉ tiêu tìm kiếm nêu trên.
+ Trong hộp hội thoại Find, nhấn chuột vào khung Date để chọn điều kiện tìm kiếm theo ngày tháng. Một hộp thoại xuất hiện.
+ Chọn nút Find all files. Nhấn chuột vào mũi tên xuống trong khung này để chọn chế độ tìm kiếm theo thời gian sửa đổi hay thời gian tạo tệp.

Bạn chỉ có thể chọn một trong ba nút sau:

+ between … and … : Giữa một khoảng thời gian.
+ during the previous … month(s): Trước một số tháng.
+ during the previous … day(s): Trước một số ngày.

Sau khi khai báo xong các tham số trên, nhấn chuột lên nút Find Now.

+ Trong hộp thoại Find, nhấn chuột lên khung Advanced để chọn chế độ tìm kiếm theo kích thước tệp.

- Tìm kiếm trong Windows 2000:

Để tìm kiếm thông tin trên đĩa, các bạn tiến hành các bước sau:

+ Nhấn chuột vào biểu tượng Search trên thanh công cụ. Khi này hộp thoại Search xuất hiện.
+ Đưa tên tệp hoặc thư mục cần tìm kiếm vào khung Search for files or folders named.
+ Đưa đoạn văn bản có trong tệp tìm kiếm
+ Chọn thư mục hoặc ổ đĩa cần tìm trong khung Look in.

+ Khi đã đưa các tham số cần thiết, nhấn chuột vào nút Search now để bắt đầu tìm kiếm.

Trong khung Search Results xuất hiện kết quả các thư mục hoặc các tệp thỏa mãn điều kiện tìm kiếm cùng các tham số của chúng.

** Format đĩa:*

Giả sử cần format đĩa mềm (ổ A), đưa vệt sáng đến ổ A trên cửa sổ bên trái , nhấn nút chuột phải. Khi này, một hộp thoại xuất hiện. Chọn mục Format.

Câu hỏi ôn tập chương III

1. Nêu khái niệm hệ điều hành, các chức năng cơ bản của hệ điều hành.
2. Nêu sự khác nhau giữa hệ điều hành máy tính cá nhân và hệ điều hành mạng. Nêu một số hệ điều hành đang được sử dụng rộng rãi hiên nay ở nước ta.
3. Nêu cách khởi động, mở trình ứng dụng, khởi động lại, tắt máy trong hệ điều hành Windows 2000.
4. Trình bày về màn hình chính của Windows 2000, thanh ứng dụng TASKBAR, nút START.
5. Chương trình Windows Explorer để làm gì ?. Nêu một số ứng dụng của Windows Explorer.

CHƯƠNG IV: HỆ SOẠN THẢO VĂN BẢN MIROSOFT WORD

Một trọng các ứng dụng thông dụng nhất của máy vi tính là để soạn thảo văn bản. Hệ soạn thảo văn bản là bộ chương trình giúp cho người sử dụng soạn thảo văn bản trên máy vi tính. Hệ soạn thảo văn bản được sử dụng rộng rãi nhất hiện nay là hệ soạn thảo văn bản Microsoft Word. Trong chương này giới thiệu về hệ soạn thảo văn bản Microsoft Word. Nội dung chính đề cập tới các vấn đề: Khởi động Word, các thành phần trên màn hình word, soạn thảo văn bản, định dạng dữ liệu, các lệnh về tệp, kẻ bảng biểu, các thao tác thường gặp trong Word.

1 - Giới thiệu chương trình Ms-Word

1.1 - Giới thiệu chung

MS-Word là trình soạn thảo văn bản do hãng MicroSoft thiết kế. MS-Word hoạt động trong môi trường Window. Hệ điều hành WINDOWS có giao diện đồ hoạ giữa người sử dụng và máy vi tính, mọi lệnh của WINDOWS được thể hiện qua các biểu tượng đồ hoạ trên màn hình và việc ra lệnh cho máy tính chỉ là chọn các biểu tượng hoặc thực đơn phù hợp thông qua bàn phím hoặc thiết bị chuột.

Các máy tính ở nước ta hiện nay phổ biến dùng hệ điều hành Window 98 hoặc Window 2000.

Chương trình MS-WORD là hệ soạn thảo văn bản nằm trong bộ chương trình Tin học văn phòng (MicroSoft Office), chúng ta có thể gặp các phiên bản khác nhau của Word như Word6.0 trong Office 4.3 hoặc Word 97; Word 2000. Những phiên bản mới này được bổ xung thêm một số tính năng về đồ hoạ, bảng biểu mà các phiên bản trước không có.

Trong môi trường WINDOWS có thể tìm thấy bảng tính điện tử Excel, hệ quản trị dữ liệu Access, công cụ vẽ Powerpoint và một số trình ứng dụng khác. Trong phần này chúng ta chỉ đề cập đến MS-WORD.

Bộ chương trình Windows được cài đặt trong ổ cứng của máy vi tính. Với phiên bản 3.1 (là phiên bản làm việc trong môi trường DOS) sau khi khởi động từ dấu nhắc hệ thống C:\> ta chỉ cần gõ lệnh WIN và bấm tiếp phím Enter là chương trình được khởi động. Các phiên bản Windows 95 trở đi việc khởi động được hoàn toàn tự động.

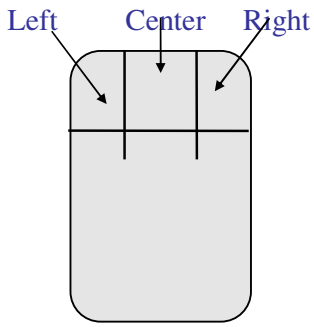
1.2 - Cách sử dụng chuột

Trước khi làm việc với MS-WORD ta cần phải biết cách sử dụng thiết bị chuột (Mouse). Mouse có thể có 2 hoặc 3 phím (Hình 1.4). Khi di chuyển mouse trên bàn làm việc con trỏ mouse trên màn hình sẽ di chuyển theo, tuỳ thuộc vào vị trí của Mouse trên màn hình mà hình dạng của nó sẽ thay đổi. Phần lớn các thao tác của mouse chỉ sử dụng phím trái (Left) do vậy trong tài liệu này nếu nói bấm mouse thì có nghĩa là bấm phím trái, những trường hợp bấm phím phải sẽ có ghi chú riêng.

Có 3 thao tác bấm phím của mouse:

- **Bấm đơn**: đưa mouse đến biểu tượng hoặc vị trí cần thiết rồi bấm phím trái
- **Bấm kép**: đưa mouse đến biểu tượng hoặc vị trí cần thiết rồi bấm phím trái hai lần liên tiếp
- **Bấm rê**: đưa mouse đến vị trí cần thiết bấm và giữ phím trái sau đó di mouse trên bàn, thao tác bấm rê nhằm bôi đen một vùng văn bản hoặc di chuyển một đối tượng từ chỗ này sang chỗ khác.

Hình dạng con trỏ chuột phụ thuộc vào vị trí của nó trên màn hình, mỗi hình dạng có một công dụng khác nhau, chúng ta sẽ thấy trong từng ứng dụng cụ thể.



Hình 1.4

Một số chuột hiện nay phím giữa được thay bằng một bánh xe. Quay bánh xe này màn hình sẽ được cuộn lên hay xuống.

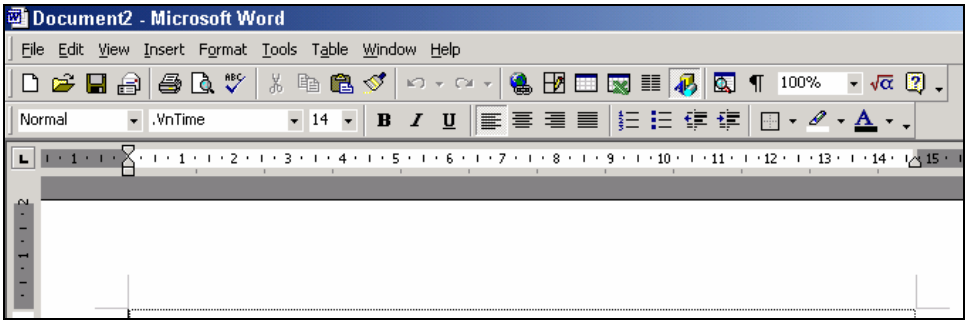
1.3 - Khởi động và ra khỏi Word

a) Khởi động Word

Có thể khởi động Word theo 3 cách sau:

- Cách 1: Kích đúp chuột vào biểu tượng **Word** trên Desktop
- Cách 2: Chọn nút **Start** / chọn mục **Program**/ chọn **Micrsoft Word**
- Cách 3: Chọn nút **Start** / chọn mục **Run** / chọn nút **Browse** / chọn thư mục và chon chương trình ứng dụng.

Sau khi khởi động bằng một trong ba cách trên, cửa sổ làm việc của Micrsoft Word có dạng như hình sau (Hình 2.4)



Hình 2.4

Theo thứ tự từ trên xuống dưới thì màn hình bao gồm các phần sau:

- Thanh tiêu đề: Thanh này cho biết tên cửa sổ đang làm việc (Micrsoft Word) và tên văn bản đang soạn thảo.
- Thanh thực đơn (Menu): Thanh này trình bày các thực đơn ngang, mỗi mục chọn trong thực đơn ngang sẽ cho tiếp một thực đơn dọc.
- Thanh công cụ (Toolbars): Trên thanh công cụ là các nút (Button), các nút này là công cụ giao tiếp thay cho việc chọn các thực đơn con trong thanh Menu.

- Thanh định dạng (Format): Thanh này gồm các nút phục vụ cho việc định dạng văn bản, các chức năng định dạng văn bản thực ra cũng đã có trong thực đơn Format, tuy nhiên định dạng bằng thanh Format sẽ nhanh hơn và tiện dụng hơn.

- Thước (Ruler): Chỉ ra lề trái , lề phải, vị trí các cột trong bảng.

- Phần hiện ra văn bản: Gồm nhiều dòng, để hiện ra văn bản mà ta gõ vào

- Cuối cùng là thanh trạng thái: Chỉ ra các thông tin hiện thời về tệp văn bản đang soạn thảo: Trang hiện thời (Page), đoạn (Sec), dòng (Ln), cột (Col), …

Có thể dùng chức năng Tools - Customize để thay đổi các nút trong các thanh công cụ hoặc dùng View-Toolbars... để thêm bớt các thanh.

b) Ra khỏi Word

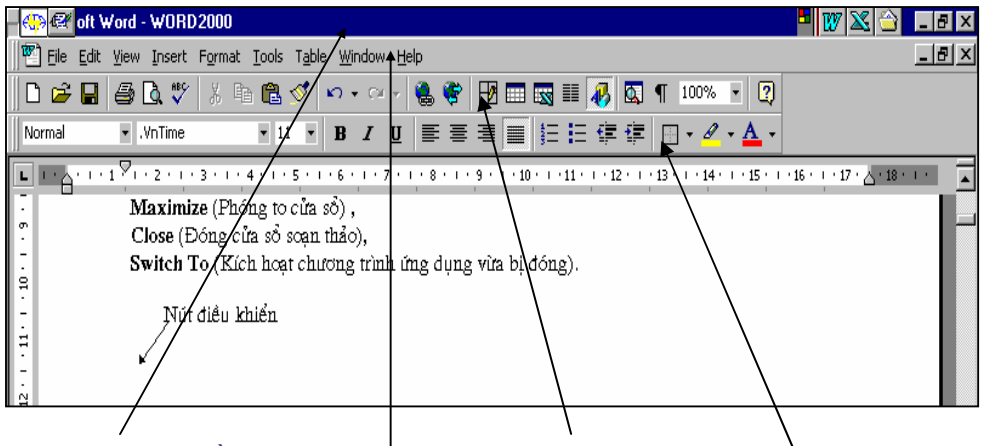
Chọn mục FILE/ chọn mục EXIT hoặc nhấn đồng thời hai phím Alt và F4

1.4 - Mô tả các thành phần trong màn hình Word

a) Thanh tiêu đề (Title Bar)

Phần giữa thanh tiêu đề là tên của cửa sổ đang mở (Microsoft Word) và tên tệp văn bản đang soạn thảo, khi bắt đầu làm việc với MS-WORD 6.0 tên tệp văn bản được đặt ngầm định là Document1, nếu chúng ta ghi văn bản vào đĩa với tên khác thì tên đó sẽ thay thế cho Document1. Bên trái thanh tiêu đề là nút điều khiển (Ctrol menu box). Khi bấm mouse vào nút này ta thấy hiện lên một menu dọc gồm các Menu con:

Restore (Khôi phục cửa sổ về trạng thái trước),
Move (Dịch chuyển cửa sổ đến vị trí mới),
Size (Thay đổi kích thước cửa sổ),
Minimize (Thu cửa sổ thành biểu tượng),
Maximize (Phóng to cửa sổ),
Close (Đóng cửa sổ soạn thảo),
Switch To (Kích hoạt chương trình ứng dụng vừa bị đóng).
Nút điều khiển



Hình 3.4

Phía bên phải thanh tiêu đề có ba nút

Thu nhỏ màn hình làm việc thành một dòng

Thu nhỏ hoặc phóng to cửa sổ

Kết thúc làm việc, đóng cửa sổ lại

b) Thanh thực đơn (Menu Bar)

Thanh thực đơn gồm một danh mục các thực đơn chính xếp theo hàng ngang. Để chọn các ứng dụng trong một Menu ngang ta bấm đơn phím trái mouse vào menu đó hoặc bấm tổ hợp phím nóng phím (đè phím ALT rồi gõ tiếp chữ cái gạch chân của menu). Mỗi menu ngang lại gồm nhiều mục dọc (Menu popup), để chọn một mục dọc ta làm hoàn toàn tương tự như chọn menu ngang. Các menu ngang và mục dọc tương ứng được liệt kê dưới đây:

* **File** (các lệnh về xử lí tệp):

New (mở cửa sổ mới để soạn thảo văn bản)
Open (mở văn bản đang lưu trong đĩa ra màn hình)
Save (cất văn bản đang soạn thảo vào tệp trên đĩa)
Save as (cất văn bản vào tệp với tên khác)
Close (đóng văn bản đang soạn thảo)
Find file (tìm tệp trên đĩa)
Summary Information (tạo thông tin tóm tắt về văn bản)
Templates (Chọn khuôn mẫu trình bầy văn bản)
Page Setup (định dạng trang văn bản , chọn cỡ giấy, lề in ...)
Print Preview (xem toàn cảnh văn bản trước khi in)
Print (in văn bản).

* **Edit** (các lệnh phục vụ việc soạn thảo):

Undo (bỏ thao tác vừa làm)
Repeat (lặp lại thao tác vừa làm)
Cut (xoá đối tượng đã chọn hoặc đoạn văn bản đã bôi đen)
Copy (chép đối tượng đã chọn hoặc đoạn văn bản đã bôi đen vào vùng đệm clipboard)
Paste (dán đối tượng đã có trong vùng đệm vào vị trí mới)
Paste Special (dán đối tượng theo một phương pháp đặc biệt, chuyển cột thành hàng hoặc hàng thành cột, nâng cao mật độ khi in)
Clear (xoá đoạn văn bản đã chọn)
Find (tìm kiếm từ ngữ hoặc đoạn văn trong văn bản)
Replace (tìm và thay thế)
Goto (nhẩy tới trang số..)
AutoText (Tạo một đoạn văn bản mẫu để có thể gọi ra bất kỳ lúc nào)
Bookmark (đánh dấu định vị trong văn bản).

* **View** (chọn chế độ hiển thị văn bản trên màn hình):

Normal (bình thường)
Layout (hiện lề ngoài văn bản hoặc hiện văn bản kèm theo các hình vẽ)
PageLayout (hiện lề ngoài trang văn bản)
Master Document (tạo văn bản chính với các văn bản con)
Fulscreen (mở cửa sổ văn bản rộng kín toàn màn hình)
Toolbars..(Cho hiện hoặc không hiện các thanh công cụ, thanh định dạng, thanh đường viền...)
Ruler (cho hiện hoặc không hiện thước kẻ trên đầu cửa sổ văn bản)
Header and Footer (cho hiện tiêu đề đầu và cuối trang)
Footnote (cho hiện chú giải cuối trang)

Annotation (Cho hiện chú thích trong văn bản)
Zoom (Thay đổi kích thước cửa sổ soạn thảo)

* **Insert** (chèn các đối tượng khác nhau vào văn bản):
Break (chèn dấu ngắt trang)
Page Number (đánh số trang)
Annotation (thêm lời chú thích)
Date and Time (Ngày và giờ)
Field (chèn mã trường có chứa thông tin xác định vào văn bản),
Symbol (chèn các kí tự đặc biệt như α,β... hoặc các dấu hoa văn)
Form Field (biểu mẫu định sẵn)
Foonote (chú giải cuối trang)
Caption (chèn thêm chú giải cho nội dung đã chọn)
Cross reference (chèn các đối tượng qua tham khảo chéo trong hộp liệt kê)
Index and Table (chèn bảng mẫu)
File (chèn tệp)
Frame (chèn khung để đóng gói văn bản hoặc một bức tranh)
Picture (chèn thêm bức tranh đã có sẵn trong thư viện vào văn bản hoặc vào khung)
Objects (chèn các đối tượng khác như bảng tính, các dấu toán học ..)
Database (chèn cơ sở dữ liệu)

* **Format** (chức năng định dạng văn bản):
Font :định dạng chữ bao gồm kiểu chữ: (Font); kích thước: (Font size); màu sắc: (Color) ; dáng chữ: (Style)
Pragraph (định dạng đoạn văn bản: khoảng cách dòng, độ thụt dòng, lề, khoảng cách giữa các đoạn văn bản)
Tabs (định dạng bước nhẩy cột khi gõ phím Tab)
Borders and Shading (định dạng đường bao và bóng của khung, bảng)
Columns (phân chia đoạn văn bản hoặc trang giấy thành các cột ...)
Change-Case (biến chữ to thành nhỏ và ngược lại, định dạng nhóm từ kiểu tên riêng...)
Drop cap ...(định dạng chữ cái đầu tiên của một đoạn văn bản)
Bullets and Numbering (định dạng nét gạch và đánh số đầu mục)
Heading Numbering (đánh số các tiêu đề)
Autoformat (thiết lập chế độ định dạng tự động)
Style gallery (định dạng theo mẫu đã được thiết kế của Windows)
Styles (lựa chọn hoặc tổ chức kiểu trình bầy có sẵn cho đoạn văn hoặc cho kí tự)
Frame (định dạng khung bao)
Picture (định dạng lại các hình ảnh đã chèn vào văn bản)
Drawing object (định dạng các đối tượng vẽ)

* **Tools** (Các công cụ trợ giúp)
Spelling (kiểm tra lỗi chính tả tiếng anh)
Grammar (chỉnh lí văn phạm)
Thesaurus (tìm từ đồng nghiã)
Hyphenation (đặt gạch nối các từ một cách tự động hay bằng tay)
Language (Chọn ngôn ngữ viết văn bản)
Word count (đếm số chữ , số từ , số dòng , số đoạn có trong đoạn văn đã lựa chọn)

Autocorrect (thiết lập tên cho một cụm từ hoặc tiêu đề mà nội dung của nó được tự động chỉnh lỗi chính tả)

Mail Merge (trộn dữ liệu vào văn bản chính, dữ liệu có thể tạo ra trong Word hoặc các phần mềm khác như Foxpro, Lotus, Foxbase..., nhằm mục đích tạo ra một văn bản mới)

Envelopes and labels (tạo bao thư cho văn bản)

Protect Document (bảo vệ tài liệu)

Revions (hiệu đính)

Macro (tạo lập các vĩ lệnh, có thể hiểu Macro là một tập hợp tất cả các lệnh được gõ từ bàn phím trừ các thao tác chuột)

Customize (tạo các nút công cụ hoặc menu riêng)

Options (các phương án lựa chọn tổng hợp trình bầy màn hình hoặc văn bản)

* **Table** (Thực đơn về tạo lập bảng):

Insert table (chèn thêm một bảng vào vị trí con trỏ)

Delete cells (xoá các ô đã bôi đen)

Merge cells (liên thông các ô đứng gần nhau theo hàng ngang thành một ô)

Split cells (phân chia một nhóm ô thành nhiều ô)

Select row (chọn hàng)

Select column (chọn cột)

Select table (chọn bảng)

Table Autoformat (tự động định dạng bảng theo khuôn có sẵn)

Cell Height and Width (định kích thước của ô hoặc hàng, cột)

Headings (nhập tiêu đề cho cột)

Convert Text to Table (chuyển văn bản dưới dạng bảng thành bảng dữ liệu. Điều kiện cần là các dữ liệu phải ngăn cách nhau bằng dấu phẩy .

Sort Text (Xếp bảng dữ liệu theo cột số hoặc cột chữ)

Formula (tính toán với các số liệu có trong bảng)

Split table (phân chia bảng thành 2 bảng)

Gridlines (cho hiện hoặc không cho hiện lưới kẻ bảng)

* **Window** (các lệnh xử lí của sổ văn bản):

New (mở cửa sổ văn bản mới)

Arrange all (cho hiện đồng thời trên màn hình cửa sổ của các văn bản đang mở)

Split chia cửa sổ hiện hành thành hai phần

* **Help** (xem hướng dẫn các lệnh)

c) Thanh công cụ (Tools Bar) (hình 4.4)

Hình 4.4

Các nút trên thanh công cụ tính từ trái qua phải gồm:

New: mở một màn hình mới che lên màn hình đang làm việc

Open: mở một văn bản đang lưu trữ trong đĩa của máy

Save: ghi văn bản hiện có trên màn hình vào đĩa

Print: In văn bản ra giấy

Print Preview: quan sát toàn cảnh văn bản trước khi in
Spenlling and Grammar: dò lỗi chính tả và ngữ pháp tiếng Anh
Cut: xoá phần văn bản đang được bôi đen trên màn hình
Copy: sao chép phần đang bôi đen vào Clipad (vùng đệm bộ nhớ)
Paste: dán những gì đang có trong Clipad vào vị trí hiện thời trên màn hình
Format Painter: sao chép định dạng của một đoạn văn bản
Undo Typing: bỏ thao tác vừa thực hiện (khôi phục lại tình trạng ttrước đó)
Redo cut: khôi phục lại những gì vừa bị xoá
Isert Hyperlink: thêm siêu liên kết
Web Toolbar: hiện thanh công cụ kết nối Internet
Tables and Border: tạo bảng biểu bằng bút vẽ
Insert Table: tạo bảng biểu
Isnert Microsoft Excel Worksheet: chèn vào văn bản một trang tính Excel
Columns: chia văn bản thành nhiều cột
Drawing: hiện thanh công cụ vẽ
Document Map: hiện sơ đồ các đề mục của văn bản
Show/Hide: hiện hoặc tắt ký hiệu xuống dòng
Zoom: Thay đổi tỷ lệ kích thước màn hình
Office Assistant: trợ giúp

Thanh công cụ chứa các biểu tượng dưới dạng các nút giúp lựa chọn nhanh một chức năng nào đó. Thanh công cụ chuẩn của Office 97 hoặc Office 2000 (Standard Tools bar) gồm các nút như trong hình 3.

Các nút trong thanh công cụ được sử dụng bằng cách đưa chuột đến nút rồi bấm đơn, thao tác này cũng tương tự như chọn một chức năng trong thanh thực đơn rồi chọn chức năng con có tên tương ứng. Ví dụ để ghi văn bản đang có trên màn hình vào đĩa ta chỉ việc bấm chuột vào nút Save thay vì phải chọn thực đơn File rồi thực đơn con Save trong Menu File.

d) Thanh định dạng (Formating Bar) (Hình 5.4)

Hình 5.4

Thanh này gồm có các nút dùng để định dạng văn bản, tính từ trái qua phải bao gồm:
Style (Kiểu trình bầy)
Font (chọn kiểu chữ)
Font size (chọn cỡ chữ)
Bold (chữ đậm)
Italic (chữ nghiêng)
Under Line (gạch chân)
Align Left (dóng thẳng lề trái)
Align Right (dóng thẳng lề phải)
Justify (dãn đều 2 lề)
Numbering (đánh số thứ tự các đoạn văn bản)
Bullets (Dùng các biểu tượng đánh dấu đoạn văn bản)
Decrease Indent (dịch chuyển cả đoạn văn bản sang trái)
Increase Indent (dịch chuyển cả đoạn văn bản sang phải)
Border (định dạng đường viền khung)
Highlight (chọn màu nền văn bản)
Font Color (chọn màu chữ văn bản)

2 - Các phím gõ tắt *(ShortCut-Key)*

Các phím gõ tắt thường là một tổ hợp phím ngầm định của Windows hoặc các phím do người sử dụng cài đặt để thực hiện nhanh một công việc nào đó thông qua việc gõ bằng bàn phím . Các phím gõ tắt bao giờ cũng bắt đầu bằng một trong các phím chức năng phụ Ctrl, Shift, Alt kết hợp với một hoặc hai phím khác. Dưới đây là một bảng liệt kê các phím gõ tắt có sẵn của Windows và chức năng tương ứng của nó. Cách bấm phím gõ tắt như sau: Đè phím thứ nhất rồi bấm tiếp phím thứ hai hoặc đè phím thứ nhất, đè tiếp phím thứ hai rồi tiếp phím thứ 3 sau đó buông tất cả ra.

Ví dụ: khi ghi tổ hợp phím Alt + F thì có nghĩa là đè phím Alt sau đó gõ phím F, còn nếu ghi Ctrl+Shift+W thì có nghĩa là đè hai phím Ctrl và Shift sau đó gõ phím W rồi buông tay ra.

2.1 - Các phím gõ tắt bắt đầu bằng phím Alt

Alt+F	Chọn thực đơn File trên thanh Menu
Alt+E	Chọn thực đơn Edit trên thanh Menu
Alt+V	Chọn thực đơn View trên thanh Menu
Alt+I	Chọn thực đơn Insert trên thanh Menu
Alt+O	Chọn thực đơn Format trên thanh Menu
Alt+T	Chọn thực đơn Tools trên thanh Menu
Alt+A	Chọn thực đơn Table trên thanh Menu
Alt+W	Chọn thực đơn Window trên thanh Menu
Alt+H	Chọn thực đơn Help trên thanh Menu
Alt+N	Chọn thực đơn Font trên thanh Menu

2.2 - Các phím gõ tắt bắt đầu bằng phím Ctrl

- Bôi đen đoạn văn bản sau đó bấm:

Ctrl+C	Sao chép đoạn văn bản đã bôi đen vào Cliboard
Ctrl+V	Dán đoạn văn bản đã có trong Cliboard vào vị trí mới
Ctrl+B	In đậm, chữ béo
Ctrl+I	In nghiêng
Ctrl+U	Chữ có gạch chân
Ctrl+Shift+W	Gạch chân từng từ
Ctrl+Shift+D	Gạch chân bằng nét kép
Ctrl+=	Viết chỉ số dưới (x_1, m_2,....)
Ctrl+Shift+=	Viết chỉ số trên, số mũ (a^2, b^{sinx},...)
Ctrl+ Shift+K	Chữ in hoa nhỏ
Ctrl+ Shift+A	Tất cả chữ in hoa
Ctrl+ Shift+H	Cho ẩn văn bản
Ctrl+ Shift+C	Copy định dạng
Ctrl+ Shift+V	Dán kiểu định dạng
Ctrl+Spacebar	Loại bỏ định dạng

2.3 - Các phím gõ tắt bắt đầu bằng phím Shift

Shift+End	Bôi đen đoạn văn bản từ vị trí hiện thời đến cuối dòng
Shift+Home	Bôi đen đoạn văn bản từ vị trí hiện thời đến đầu dòng
Shift+ ↓	Bôi đen một dòng văn bản
Shift+ ← (→)	Bôi đen một ký tự bên trái (hoặc bên phải con trỏ)

3 - Các thao tác định dạng

3.1 - Phương pháp viết tiếng Việt

Hiện có hai phương pháp viết tiếng Việt đang được sử dụng là phương pháp viết kiểu TELEX và phương pháp viết kiểu Đánh máy. Trong phạm vi chương trình chúng ta chỉ học phương pháp TELEX. Bộ phông chữ tiếng Việt sử dụng trong chương trình này là bộ phông VietKey, hiện nay ở nước ta còn tồn tại nhiều phông chữ như ABC, Freecode, Vietwear, Vni,... Với các máy kết nối Internet thì phải dùng bộ phông Unicode.

Cách gõ chữ Việt theo kiểu Telex:

Kiểu Telex là cách gõ tiếng Việt thuận tiện và dễ nhớ, đảm bảo có được tiếng Việt đúng chính tả, đồng thời bỏ đi những cách đặt dấu sai đã tồn tại thành thói quen.

Cách gõ các ký tự đặc biệt của tiếng việt như sau:

Gõ vào	Nhận được	Gõ vào	Nhận được
aw	ă	Aw	Ă
ow	ơ	Ow	Ơ
uw	ư	Uw	Ư
aa	â	AA	Â
oo	ô	OO	Ô
ee	ê	EE	Ê
dd	đ	DD	Đ

Gõ dấu: **f: huyền ; s: sắc ; r: hỏi ; x: ngã ; j: nặng**

Ví dụ : để có dòng chữ " Trường Đại học Nông nghiệp I" ta phải gõ như sau:

Truwowngf DDaij hocj Noong nghieepj I

*** Lưu ý :** - Dấu phải gõ vào cuối chữ , nghĩa là viết xong chữ rồi mới đánh dấu.

Ví dụ: Truwowngf -- Trường, DDieenj -- Điện, baos -- báo

- Phím xoá dấu : khi muốn xoá dấu đã đánh trên một từ ta đưa con trỏ đến sát ký tự cuối cùng của từ và gõ phím z.

- Chữ ư và ơ hay đi liền nhau nên chúng còn được bố trí ở phím] và [, đây là hai phím liền nhau trên bàn phím kiểu QWERTY, vì vậy gõ sẽ nhanh hơn.

] -- ư [-- ơ } -- Ư { -- Ơ

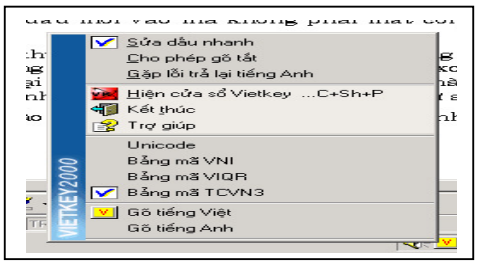
Thí dụ : tr][ngf -- Trường dd][cj -- được

- Gõ ooo nhận được oo như cooongs -- coóng

- Nếu bạn muốn gõ dấu [và] thì bạn chỉ việc gõ 2 lần phím đó.

Khi gõ Telex, Vietkey sẽ tự động bỏ dấu đúng chính tả tiếng Việt. Nếu bạn gõ nhầm dấu, bạn cứ việc gõ dấu mới vào mà không phải mất công xoá dấu cũ đi.

Cũng cần lưu ý thêm rằng khi gõ sai dấu trên một từ và chúng ta đã chuyển sang gõ từ khác thì không quay về sửa dấu được mà phải xoá hết nguyên âm của từ sai rồi mới gõ lại từ đó. Để tránh điều phiền toái này bạn có thể chọn tính năng sửa dấu nhanh của Vietkey. Cách thức chọn như sau: Bấm đơn phím phải chuột vào biểu tượng Vietkey, trên màn hình xuất hiện hộp thoại (hình 6.4). Chọn chức năng "Sửa dấu nhanh" bằng cách bấm đơn vào chức năng này. Kể từ nay khi quay lại từ đánh dấu sai ta chỉ việc gõ lại dấu đúng.



Hình 6.4

3.2 - Định dạng ký tự

a) Sử dụng thanh thực đơn

- Bôi đen đoạn văn bản muốn định dạng
- Chọn chức năng **Format**
- Trong Format chọn **Font**, hộp thoại Font xuất hiện (Hình 7.4). Trong hộp này có hai mục chọn là **Font** (kiểu chữ) và **Character Spacing** (vị trí chữ). Trong mục chọn Font có các nút định dạng như sau:

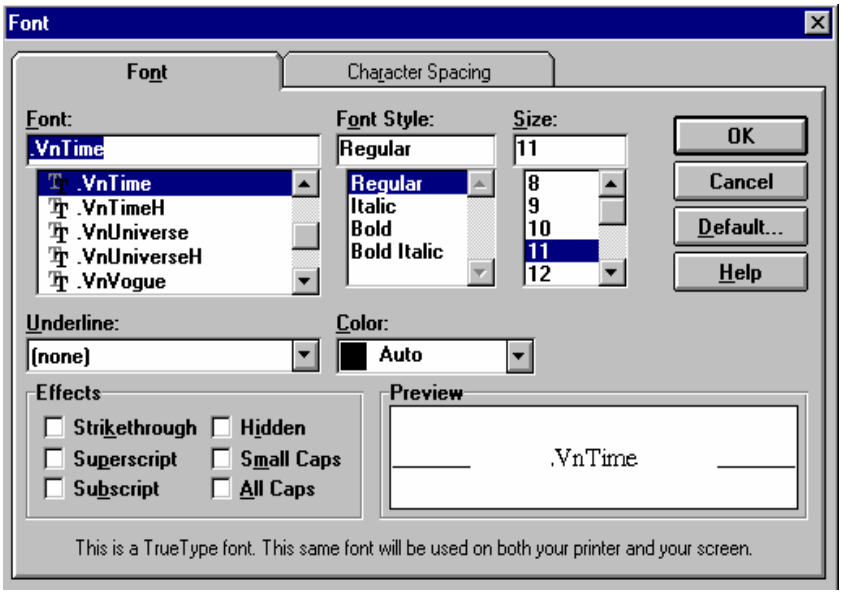
+ **Font**: kiểu chữ, ngay phía dưới là các kiểu chữ có thể lựa chọn. Với bộ phông chữ tiếng Việt chuẩn Vietkey (phông bắt đầu bằng hai ký tự Vn) nếu tận cùng là chữ H thì là phông chữ in.

+ **Font Style**: dáng chữ, có các dáng Regular, Bold, Italic, Bold Italic. Chọn một trong các dáng này và quan sát khung Preview để xem hiệu quả.

+ **Size**: kích thước chữ, có thể chọn kích thước từ 8 đến 72, ngầm định là 12.

+ **Underline**: chữ có gạch chân

+ **Color**: chọn màu cho chữ



Hình 7.4

+ **Effect**: Một số thao tác định dạng khác:

- **Strikethruogh**: tạo nét gạch xuyên qua từ
- **Supescript**: đưa chữ lên cao, tạo số mũ
- **Subscript**: đưa chữ xuống dưới, tạo chỉ số dưới
- **Hidden**: cho ẩn chữ , muốn cho hiện lại bấm tổ hợp phím Ctrl+Z
- **Small Caps**: Biến chữ hoa thành chữ thường
- **All Caps**: Biến tất cả thành chữ hoa

+ **Preview**: quan sát hiệu quả định dạng

- Trong mục chọn Character Spacing có các nút định dạng như sau:

+ **Spacing** : cách thức hiện chữ trên dòng, có 3 khả năng lựa chọn:

- **Normal:** hiện bình thường
- **Expanded:** các ký tự viết dãn cách nhau
- **Condensed:** các ký tự viết sát vào nhau

+ **Position**: vị trí của ký tự trên dòng

- **Normal**: hiện bình thường
- **Raised**: đưa chữ lên cao hơn
- **Lowered**: đưa chữ xuống thấp hơn

Sau khi lựa chọn xong các đặc tính cần thiết cho chữ bấm chuột vào OK hoặc bấm phím Enter để quay về màn hình soạn thảo văn bản.

b) Sử dụng thanh Format

Trong trường hợp chỉ cần định dạng kiểu chữ, kích thước chữ, và các dáng đậm, nghiêng, gạch chân ta có thể dùng các nút trên thanh định dạng. Phương pháp tiến hành như sau: bôi đen đoạn văn bản cần định dạng, sau đó muốn có chữ đậm chỉ việc bấm chuột vào nút , cần chữ nghiêng bấm nút , cần chữ gạch chân bấm nút , muốn đưa đoạn văn bản vào giữa trang giấy bấm tiếp nút . còn muốn văn bản dãn đều hai bên mép giấy thì bấm nút ...

Lưu ý:

** Việc chọn màu cho ký tự trong trường hợp a chỉ có ý nghĩa khi quan sát trên màn hình, chứ không thể in ra giấy (trừ trường hợp chúng ta có máy in màu).*

** Văn bản hiện có trên màn hình và văn bản khi in ra giấy có thể không giống nhau. Muốn chắc chắn văn bản in ra ta phải chọn chức năng quan sát toàn cảnh Print Preview tức là chọn nút* *Những gì quan sát được qua Print Preview thì khi in ra giấy sẽ giữ nguyên 100%.*

c) Sao chép định dạng

Sau khi đã làm các thủ tục định dạng một đoạn văn bản ta có thể sao chép sự định dạng này sang các đoạn văn bản khác, thao tác như sau:

- Bôi đen đoạn văn bản đã định dạng
- Chọn nút Format Painter , di chuột vào phần màn hình chứa văn bản khi đó chuột sẽ có thêm một chiếc chổi bên trái.
- Bấm và rê chuột trên đoạn văn bản cần định dạng sau đó buông tay ra, đoạn văn bản sẽ được định dạng theo cách của đoạn ban đầu.

Nếu muốn sao chép định dạng cho nhiều đoạn văn bản cùng một lúc, ta bấm kép vào nút Format Painter rồi lặp lại thao tác như trong bước trên, khi nào sao chép xong bấm đơn lần nữa vào nút Format Painter .

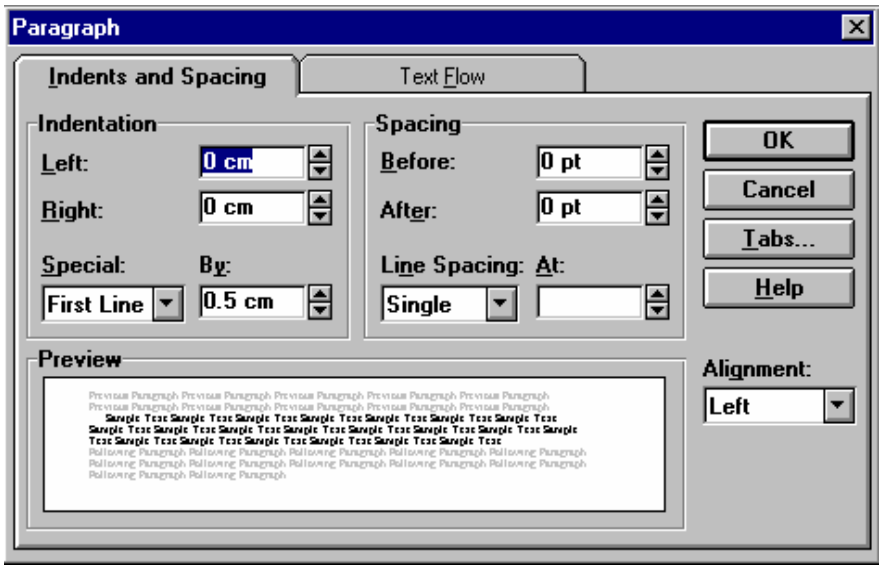
3.3 Định dạng đoạn văn bản

Đoạn văn bản (Paragraph) được hiểu là toàn bộ những từ nằm giữa hai lần bấm phím Enter.

Thao tác định dạng đoạn văn bản có thể làm trước hoặc sau khi viết văn bản. Nếu ta đã viết xong văn bản rồi mới định dạng thì phải bấm Ctrl+A để bôi đen toàn bộ văn bản.

Các bước tiến hành:

* Bôi đen văn bản hoặc một số đoạn muốn định dạng
* Chọn nút Format - chọn tiếp Paragraph, hộp thoại Paragraph xuất hiện (Hình 8.4)



Hình 8.4

+ **Indents and Spacing**: vị trí đoạn văn bản
+ **Text Flow**: sự liên kết các đoạn

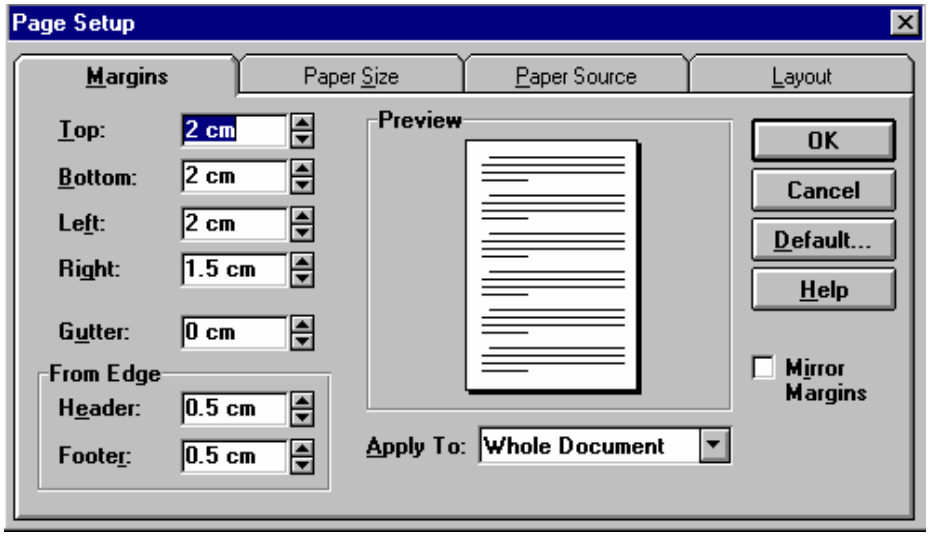
Trong mục Indents and Spacing có các thông số lựa chọn sau:

Left: Khoảng cách từ đoạn văn bản đến lề trái
Right: Khoảng cách từ đoạn văn bản đến lề phải
Befor: Đoạn hiện thời cách đoạn trên bao nhiêu
After: Đoạn hiện thời cách đoạn dưới bao nhiêu
Special: trong Special có hai chức năng
First line: Vị trí của dòng đầu tiên viết lùi vào bao nhiêu
Hanging: Các dòng còn lại viết lùi vào bao nhiêu
Line Spacing: Khoảng cách giữa các dòng, bao gồm các khả năng
Single: cách nhau bình thường theo kích thước chuẩn
1.5 Line: các dòng cách nhau gấp rưỡi
Double: các dòng cách nhau gấp đôi
At least hoặc **Exactly**: các dòng cách nhau theo lựa chọn của người sử dụng
Multiple: Cách nhau gấp 3 (chữ thường bị mất dấu)

3.4 - Định dạng trang giấy

Định dạng trang giấy có thể làm trước hoặc sau khi viết văn bản, nói chung nên làm trước. Thao tác định dạng trang giấy bao gồm:

Chọn chức năng File chọn tiếp Page Setup. Hộp thoại Page Setup hiện lên (Hình 9.4) với các khai báo:



Hình 9.4

Margins: lề trang giấy, trong Margins có:

Top: khoảng cách từ mép giấy phía trên đến vị trí dòng đầu tiên

Bottom: khoảng cách từ mép giấy phía dưới đến vị trí dòng cuối cùng

Left: lề bên trái

Right: lề bên phải

Gutter: khoảng cách giữa các cột nếu trang giấy chia thành nhiều cột

From Edge: khoảng cách tính từ mép giấy, bao gồm:

Header: khoảng cách từ mép giấy phía trên đến vị trí viết tiêu đề hoặc số trang

Footer: khoảng cách từ mép giấy phía dưới đến vị trí viết dòng chú thích ở đáy trang

Paper Size : kích thước trang giấy

Trong mục chọn này có thể chọn các cỡ giấy chuẩn : A4, A3, Letter .. hoặc tự định kích thước trong các mục

Width: chiều rộng

Height: chiều cao trang giấy

Orientation: định hướng in :

Portrait: in văn bản theo chiều dọc trang giấy

Landscape: in văn bản theo chiều ngang trang giấy

Paper Source: Nguồn lắp giấy, thông thường nên chọn giấy rời lắp bằng tay nghĩa là chọn Manual Feed.

4 - Kẻ bảng biểu

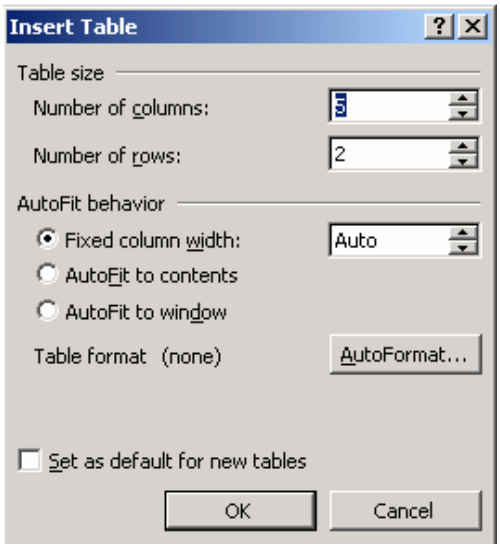
4.1 - Chèn bảng biểu vào văn bản

Có hai phương pháp đưa vào văn bản một bảng biểu, hoặc là dùng chức năng **Table** trên thanh thực đơn hoặc là dùng nút **Insert Table** trên thanh công cụ.

Đưa con trỏ đến vị trí cần chèn bảng biểu, bấm đơn vào chức năng Table rồi chọn tiếp Insert - Table hộp thoại Insert Table hiện lên như hình 10.4

Mục **Number of Columns** cho phép chọn số cột của bảng biểu, bấm vào các mũi tên phía bên phải con số ngầm định (số 2) để tăng hoặc giảm số cột.

Mục **Number of Rows** cho phép chọn số hàng của bảng biểu, bấm vào các mũi tên phía bên phải con số ngầm định (số 2) để tăng hoặc giảm số hàng.



Hình 10.4

Mục **Fixed Columns Withs** cho phép chọn độ rộng của các cột trong bảng biểu tính theo Inch hoặc cm, nếu để ngầm định Auto thì MS-WORD sẽ tự động chọn các cột có độ rộng bằng nhau và bằng chiều rộng trang giấy chia cho số cột đã chọn.

Mục **AutoFormat** dùng để chọn dạng bảng biểu đã thiết kế sẵn trong Word. MS-WORD 2000 đã thiết kế 42 kiểu bảng biểu gọi là 42 style . Các style này chủ yếu khác nhau về các dòng tiêu đề phía đỉnh của bảng biểu. Ví dụ bảng biểu trong hình 11.4 là dạng chuẩn có tên là Grid 5.

Hình 11.4

Bảng biểu đã lựa chọn trong cửa sổ Insert Table sẽ hiện lên tại vị trí con trỏ hiện thời trong văn bản, đó mới là bảng biểu tượng trưng, khi in ra giấy sẽ không có các đường kẻ . Muốn hiện các đường kẻ ta phải bôi đen toàn bộ bảng và chọn chức năng **Format - Borders and Shading.** Hộp thoại Table Borders and shading hiện lên có các phần sau:

Phần **Bordrs** (đường bao) có một số lựa chọn:

None: không tạo các đường kẻ

Box: chỉ tạo đường bao xung quanh bảng

Shadow: tạo bóng đen phía dưới và bên phải bảng

3-D: hiện bảng dưới dạng hộp không gian ba chiều

Custom: tự chọn các nét vẽ bảng

Để chọn một nét vẽ nào đó cho bảng, ta chọn nét trước ở mục Style, màu sắc nét vẽ chọn ở mục Color, độ dày mảnh của nét vẽ chọn ở mục Width, sau khi đã chọn xong thì bấm chuột vào nét vẽ tượng trưng ở mục Preview rồi chọn OK.

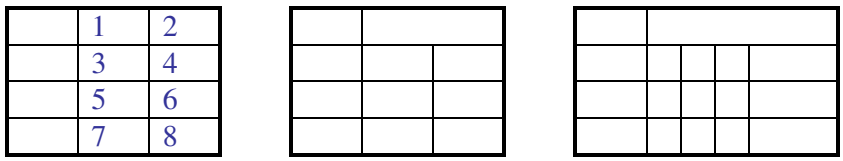
Page Border: Tạo đường viền cho trang văn bản

Shading: chức năng shading dùng để tạo hoa văn nền cho vùng ô đã bôi đen trong bảng biểu, đồng thời với việc chọn hoa văn còn có thể chọn màu cho hoa văn song với các máy in thông dụng hiện nay ta chỉ có thể in đen trắng.Trong cửa sổ Table Borders and shading ta thấy có nut **Show Toolbar**. Nút này dùng để hiện lên thanh công cụ Borders. Sử dụng các nút trên thanh Borders ta cũng có thể tạo nên các đường viền theo ý muốn.

4.2 - Một số thao tác sửa chữa bảng biểu

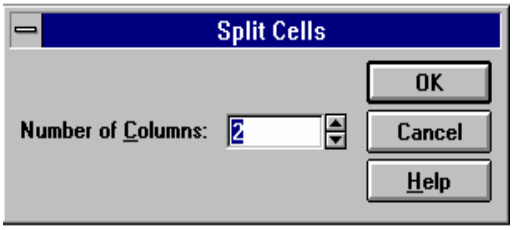
Trong một số trường hợp cần tạo nên các bảng biểu phức tạp theo yêu cầu ví dụ chia một số ô thành các ô nhỏ hơn hoặc nối thông một số ô với nhau. Ta có thể làm việc này thông qua các chức năng có trong thực đơn Table.

Hình 12.4. a là bảng biểu ban đầu. Để có hình 12.4.b ta bôi đen hai ô 1 và 2 sau đó chọn chức năng **Table - Merge cells.** Để có hình 12.4.c ta bôi đen các ô 3,4,5, sau đó chọn **Table - Split cells** hộp thoại split cells xuất hiện (Hình 13.4). Mục chọn **Number of Columns** cho phép chia các ô đã bôi đen thành số ô tuỳ ý. Bấm vào mũi tên lên hoặc xuống để chọn số ô mà ta định chia ra, cụ thể ở đây là chia 3.



a b c

Hình 12.4



Hình 13.4

Với hệ soạn thảo Word 2000 trên thanh công cụ có thể thấy một nút vẽ bảng tên nút này là Tables and Borders

Bấm đơn để chọn nút này trên màn hình xuất hiện một thanh công cụ vẽ bảng và chuột có hình dạng bút chì, dùng bút chì chúng ta có thể vẽ một khung đồng thời vẽ thêm các nét ngang dọc nghĩa là thêm vào trong bảng các cột hoặc hàng.

Trường hợp vẽ sai bấm đơn vào viên tẩy chuột sẽ biến thành viên tẩy. Bấm rê chuột trên một nét vẽ nào đó chúng ta sẽ xoá được nét vẽ đó.

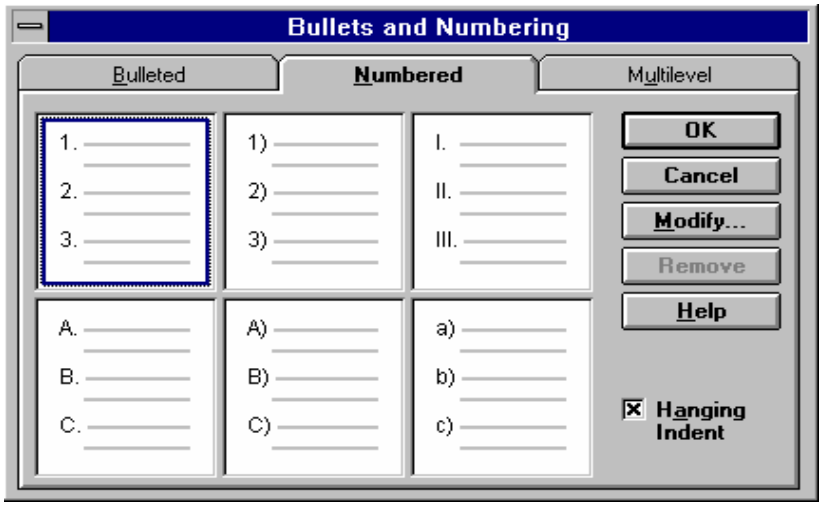
Chú ý: Những nét vẽ tạo nên đường bao của bảng khi xoá sẽ để lại nét mờ, khi in ra sẽ không có.

Hình 14.4

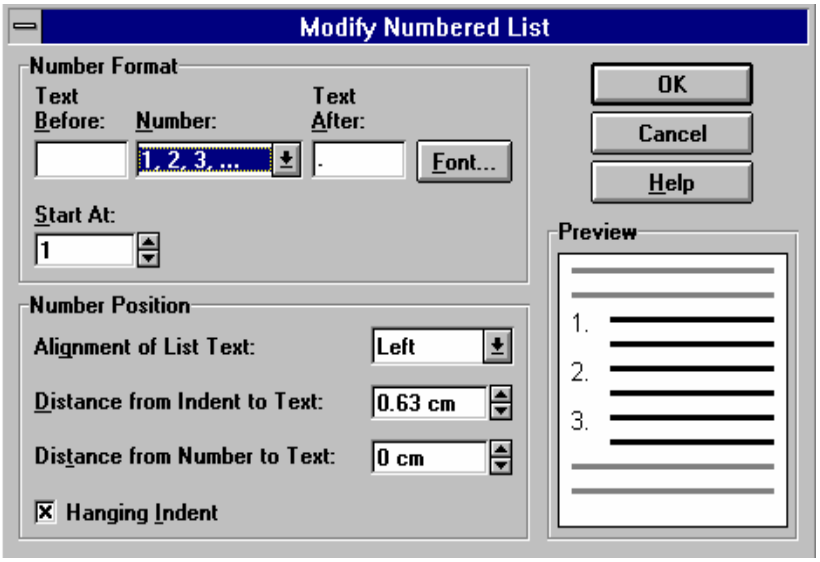
4.3 - Đánh số thứ tự trong một cột

Giả sử cần đánh số thứ tự từ 1 đến 50 trong 50 ô của một cột nào đó

- Bôi đen số ô muốn đánh số thứ tự
- Chọn Format - chọn Bullets and Numering xuất hiện cửa sổ hình 15.4
- Chọn Numbered trong cửa sổ, chọn tiếp một dạng số mà ta muốn điền vào trong cột, ví dụ dạng đầu bên phải các số sẽ có dấu chấm. Để bỏ dấu chấm đó hãy chọn Modify ta có cửa sổ hình 16.4



Hình 15.4



Hình 16.4

Dưới mục Text After máy để ngầm định dấu chấm, hãy xoá dấu chấm đó đi rồi chọn OK.

4.4 - Tạo bảng bằng bút chì

Sử dụng nút Tables and Borders hoặc chọn Table - Draw Table trên thực đơn ta có một thanh công cụ vẽ bảng, hình 14.4.

Lúc này chuột biến thành một chiếc bút chì, bằng cách bấm rê chuột ta có thể vẽ nên một bảng tuỳ ý, nếu nét vẽ sai có thể dùng chiếc tẩy trên thanh công cụ này để tẩy nét vẽ đi, cách thức tiến hành là:

Bấm đơn chuột vào viên tẩy, chuột sẽ có hình dạng tẩy, bấm rê chuột trên nét vẽ sai rồi buông tay ra nét vẽ sẽ biến mất.

5 - các thao tác thường gặp

5.1 - Viết chữ cái đầu tiên

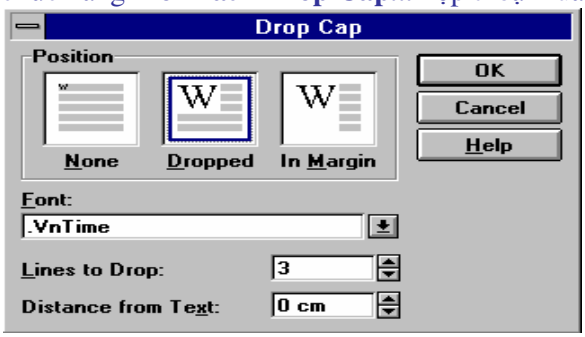
Chữ cái đầu tiên của một đoạn có thể viết to hơn các chữ khác , thậm chí có thể viết trên hai, ba dòng

Ví dụ:

Bông lau trắng giữa rừng xanh
Mong manh trước gió , mong manh trước đời
Mùa lá đổ, mùa mưa rơi
Phất phơ lau trắng ven đồi sương buông

Các thao tác:

* Đặt con trỏ màn hình vào đầu dòng muốn điều chỉnh chữ
* Chon chức năng **Format - Drop Cap...** hộp thoại xuất hiện như hình 17.4



Hình 17.4

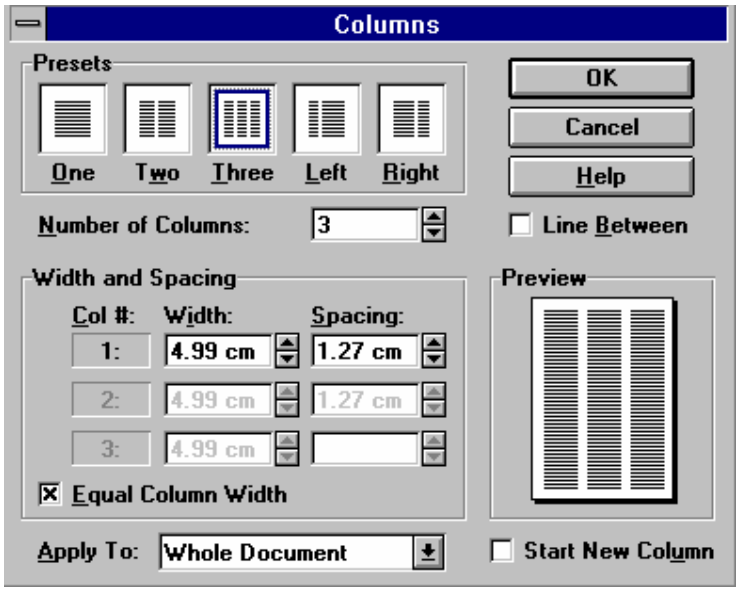
Các khả năng lựa chọn:

None: Chữ viết bình thường , không viết to
Dropped: Viết chữ to nằm gọn trong khuôn khổ văn bản
In Margin: chữ viết to năm ngoài lề văn bản
Font: Chọn phông chữ cho chữ viết to
Lines to Drop: Chữ viết trên bao nhiêu dòng
Distance from Text: khoảng cách giữa chữ và văn bản

5.2 - Chia trang hoặc đoạn văn bản thành nhiều cột

Chọn **Format - Columns**... Hộp thoại xuất hiện như trong hình 18.4 các mục chọn:

One: Trang giấy chỉ có 1 cột
Two: Trang giấy chia thành 2 cột
Three: Trang giấy chia thành 3 cột
Left: Trang giấy chia thành 2 cột , cột bên trái có kích thước nhỏ hơn cột phải



Hình 18.4

Right: Trang giấy chia thành 2 cột , cột bên trái có kích thước lớn hơn cột phải

Numbers of Columns: Số lượng cột cần có

Width and Spacing:

Width: Độ rộng của một cột

Spacing: Khoảng cách giữa các cột

Cần chú ý rằng sau khi chọn kích thước giấy và lề trang giấy thì MS-WORD sẽ tự động xác định độ rộng của các cột tuỳ thuộc vào số lượng cột và khoảng cách giữa các cột mà ta đã chọn.

5.3 - Tạo thuật ngữ viết tắt

Nhiều thuật ngữ đặc biệt là tiếng La tinh dùng trong y học hoặc những tên riêng chúng ta hay gặp khi soạn thảo có thể viết tắt theo phương pháp:

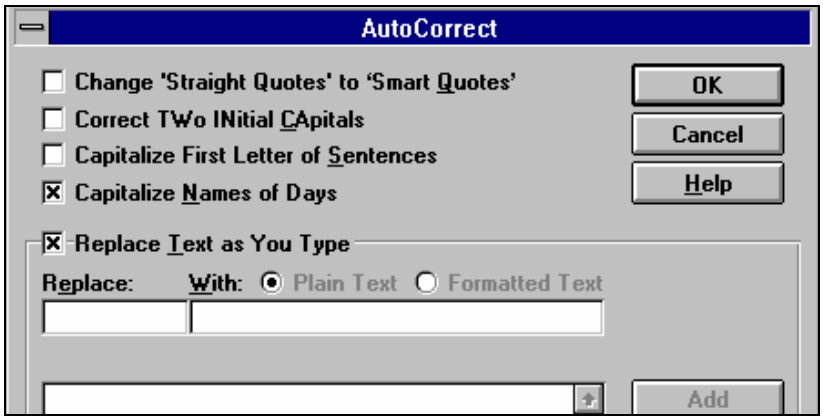
Chọn **Tools - AutoCorrect**. Hộp thoại AutoCorrect xuất hiện như hình 19.4. Giả sử chúng ta muốn viết tắt dòng chữ " Cộng hoà xã hội chủ nghĩa Việt nam " bằng cách bấm hai chữ cái vn

Dưới mục chọn Replace ta viết chữ "vn"

Dưới mục chọn With ta viết dòng chữ Cộng hoà xã hội chủ nghĩa Việt nam sau đó bấm chuột vào nút Add rồi chọn OK.

Trong văn bản từ nay trở đi khi ta gõ vn và bấm phím khoảng cách thì toàn bộ dòng chữ Cộng hoà xã hội chủ nghĩa Việt nam sẽ xuất hiện thay cho chữ vn.

Các từ viết tắt tạo ra như trên chỉ có thể viết được một dòng còn muốn tạo ra nhiều dòng thì phải dùng phương pháp khác.



Hình 19.4

5.4 - Tạo đoạn văn bản mẫu

Nếu hàng này chúng ta phải thương xuyên gửi công văn đi các nơi thì phần tiêu đề của công văn có thể tạo sẵn và lưu trong máy sau đó chỉ việc gọi ra và điền nội dung vào

Giả sử cần tạo ra một tiêu đề mẫu sau đây:

Cộng hoà xã hội chủ nghĩa việt nam
Độc lập - Tự do - Hạnh phúc

Công văn

Hà nội, ngày ..... tháng ..... năm 199...

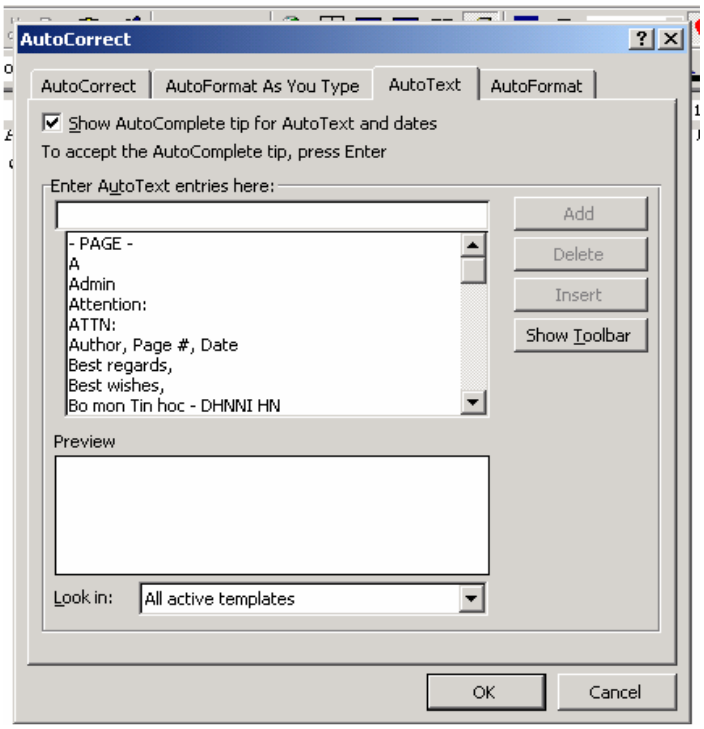
Các bước thực hiện:

* Viết đoạn tiêu đề công văn
* Trình bày kiểu chữ theo ý muốn
* Bôi đen đoạn tiêu đề
* Chon Tools - Autocorect - AutoText ...
* Trong hộp thoại Autocorect (Hình 20.4) dưới mục Enter Autotext entries here ta viết một tên cho đoạn văn bản này. Tên cần chọn sao cho dễ nhớ và chỉ được viết bằng tiếng Việt không dấu. Tiếp đó bấm đơn vào nút Add

Khi cần điền đoạn tiêu đề trên vào trang văn bản ta thực hiện các bước:

Chọn Insert - AutoText, tìm trong khung lựa chọn tên của đoạn văn bản cần thiết, bấm đơn vào tên đó cho nó nằm trong nền xanh tiếp đó chọn tiếp nút Insert.

Do hạn chế thời gian của chương trình học tập trong tài liệu này chỉ giới thiệu những gì mà sinh viên có thể tiếp thu trong khoảng 6 tiết học. MS-WORD 2000 là một hệ soạn thảo đa năng với rất nhiều chức năng phong phú muốn khai thác hết cần có thời gian học lý thuyết dài hơn và đặc biệt là cần được thao tác trên máy nhiều hơn.



Hình 20.4

5.5 - Viết số mũ và chỉ số

* Viết số mũ: ví dụ x^3

Đầu tiên viết chữ x sau đó bấm tổ hợp phím Ctrl+ Shift + =, con trỏ lúc này sẽ chuyển lên vị trí cao để ta gõ số 3, gõ lại một lần nữ tổ hợp phím trên con trỏ sẽ trở lại bình thường

* Viết chỉ số: ví dụ x_1

Gõ chữ x sau đó bấm tổ hợp phím Ctrl + =, con trỏ dịch xuống vị trí thấp ta bấm tiếp số 1 rồi bấm lại tổ hợp phím trên một lần nữa.

5.6 - Chèn vào văn bản một ký tự đặc biệt

Chọn Insert - Symbol ta sẽ có một bảng các ký tự đặc biệt, các ký tự này thường rất nhỏ khó nhận dạng, muốn nhìn rõ bấm đơn vào ký tự đó. Muốn chèn ký tự này vào văn bản bấm kép vào ký tự hoặc bám vào nút Insert ở phía dưới. Xong việc bấm Close.

6 - Tự động hoá quá trình soạn thảo

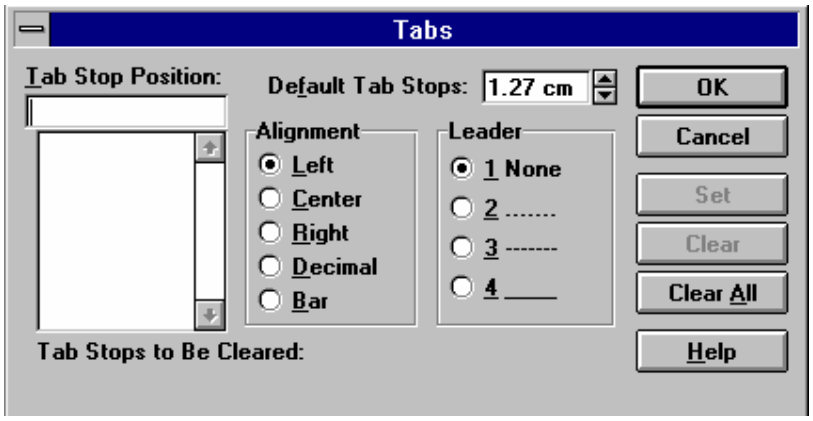
6.1 - Tạo dòng Tab

Khi bấm phím Tab trên bàn phím con trỏ sẽ tự động dịch chuyển một đoạn, Word ngầm định đoạn này là 1,27 Cm (1/2 Inch). Khi dịch chuyển con trỏ có thể để lại trên dòng các dấu chấm, các nét gạch ngang. Giả sử ta cần tạo ra dòng văn bản sau:

Họ và Tên: ..... Ngày sinh: ..... Dân tộc: .....…
8 cm 5 cm 3 cm

Chữ Họ và tên viết cách mép trái 1 cm. Phần dành cho Họ và tên là 8 cm, phần đành cho ngày sinh là 5 cm và phần dành cho dân tộc là 3 cm. Cách thức tạo như sau:

Chọn Format - Tab ta có cửa sổ (Hình 21.4)



Hình 21.4

Tab Stop Psition : gõ 1
Alignment : chọn Left
Leader : chọn None
Chọn Set

Tiếp đó lặp lại các thao tác
Tab Stop Psition : gõ 8
Alignment : chọn Left
Leader : chọn số 2
Chọn Set
Tab Stop Psition : gõ 5
Alignment : chọn Left
Leader : chọn số 2
Chọn Set
Tab Stop Psition : gõ 3
Alignment : chọn Left
Leader : chọn số 2
Chọn Set

Chon OK để trở về của sổ soạn thảo, khi đó trên dòng thước kẻ (Rule) xuất hiện một số ký hiệu thể hiện vị trí Tab.

Dòng Tab vừa tạo ra sẽ có tác dụng cho đến khi ta tạo ra dòng Tab mới

6.2 - Tạo các STYLE

Style được hiểu là tập hợp những thao tác định dạng cho một đoạn văn bản do người sử dụng lựa chọn, chúng được lưu trữ dưới một tên gọi và có thể sử dụng bất kỳ lúc nào. Khi một Style được gọi toàn bộ phần văn bản viết sau đó sẽ được định dạng theo Style này. Phần định dạng đưa vào một Style có thể là kiểu chữ, dạng chữ, kích thước, màu sắc chữ, khoảng cách giữa các dòng, giữa các cột , giữa các đoạn ....

Giả sử cần tạo ra các Style có tên là St1, St2, St3

Trong St1 sẽ đưa vào các định dạng:

Kiểu chữ: VnTimeH , kích thước: 14, màu đỏ, chữ nghiêng

Trong St2 sẽ đưa vào các định dạng:

Kiểu chữ: Vnaritote , kích thước: 16, màu xanh

Trong St1 sẽ đưa vào các định dạng:

Kiểu chữ: VnGothich , kích thước: 16, màu tím, chữ nghiêng và đậm

a) Tạo Style mới

Chọn Format - Style khi đó xuất hiện cửa sổ (Hình 22.4)

Chọn tiếp New nếu muốn tạo Style mới, nếu sửa Style cũ thì chọn Modyfi, giả sử ta chon New, khi đó xuất hiện cửa sổ hình 23.4

Dưới mục **Name** (tên của Style mới) ta gõ St1.

Mục **Style Type** có hai lựa chọn:

Character để định dạng chữ, **Paagraph** để định dạng đoạn văn bản, ngầm định là Paragraph

Giả sử ta chọn Paragraph, sau đó chọn nút Format khi đó sẽ xuất hiện hộp thoại bao gồm:

Font: định dạng chữ

Paragraph: định dạng đoạn

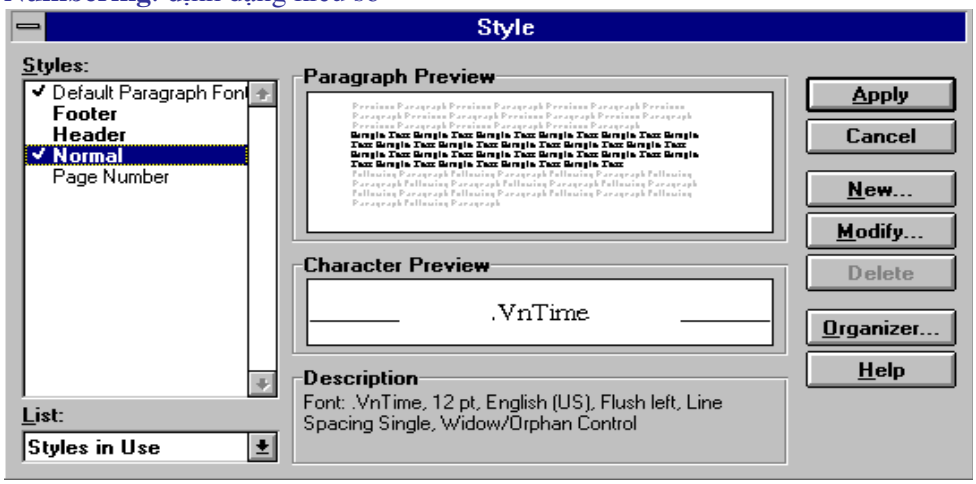
Tabs: định dạng bước nhảy khi bấm phím Tab

Border: định dạng đường bao (bảng biểu)

Language: chọn ngôn ngữ

Frame: định dạng khung

Numbering: định dạng kiểu số



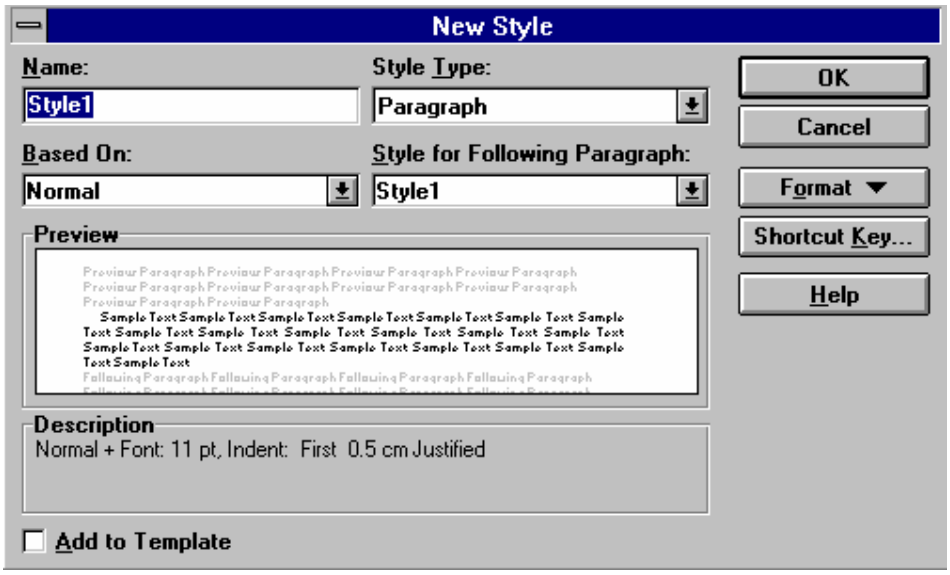
Hình 22.4

Chọn một trong các chức năng của hộp thoại ví dụ chọn Font ta sẽ quay về cửa sổ quen thuộc để làm các thao tác định dạng, cụ thể ta sẽ chọn: Kiểu chữ: VnTimeH, kích thước:14, màu đỏ, chữ nghiêng

Sau khi đã lựa chọn xong chọn OK để quay về các bước trước đó, cuối cùng chọn Close để trở về màn hình văn bản hiện thời. Nếu muốn ứng dụng ngay Style vừa tạo ra thì chọn Apply.

Tên của Style vừa tạo ra được để trong thanh định dạng. Nếu muốn ứng dụng Style nào ta chỉ việc kích chuột vào mũi tên bên phải nút Style tên các Style sẽ hiện lên trong hộp lựa chọn.

Các Style St2, St3 cũng thực hiện tương tự.



Hình 23.4

6.3 - Tạo Macro

Macro tạm hiểu là một vĩ lệnh, nó giống như một máy ghi, ghi lại toàn bộ các thao tác mà người sử dụng đã tiến hành. Khi gọi tên Macro các thao tác này sẽ được lặp lại từ đầu cho đến thao tác cuối cùng.

Macro có thể được lưu trữ trên thanh công cụ, trên bàn phím (với các phiên bản trước window2000 có thể để Macrro trong Menu.)

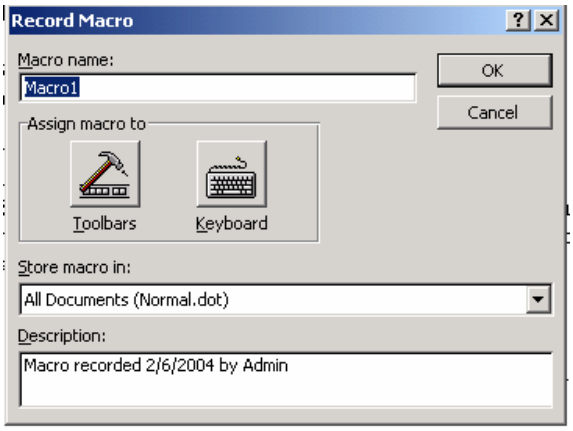
a) Tạo Macro trên ToolsBar

Các bước tiến hành:

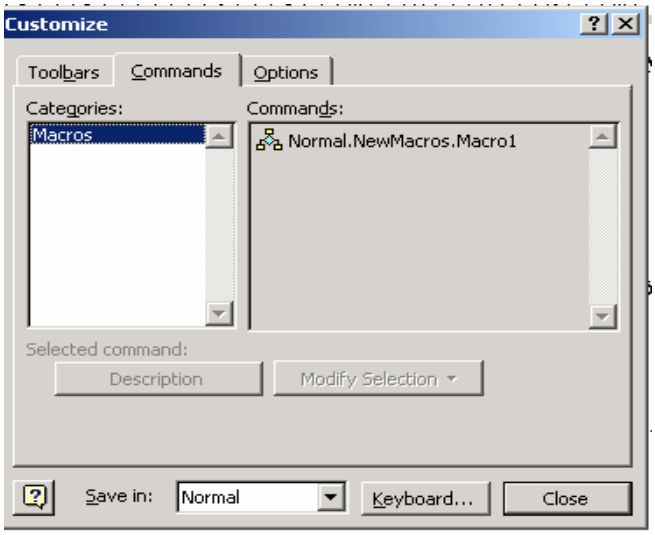
- Chọn Tools
- Chọn Macro - Record new macro, xuất hiện cửa sổ (Hình 24.4)
- Trong mục Macro Name ta ghi một tên nào đó ví dụ M1
- Trong mục Assign Macro to (đặt macro vào) chọn Toolbars có cửa sổ tiếp theo, cách chọn này có nghĩa là Macro sẽ lưu thành một nút trên thanh công cụ, nếu chọn KeyBoard thì Macro được lưu trên bàn phím.
- Dưới mục Commands bấm và di cả dòng chữ Normal.NewMacros.Macro1 lên thanh công cụ sẽ tạo nên một nút mới, bên trong nút này chưa có biểu tượng mà chỉ có dòng chữ đã nói. Để thu bé biểu tượng chọn tiếp Modify Selection sẽ xuất hiện cửa sổ hình 25.4

Bên phải mục Name trong hộp thoại hãy xoá bớt số ký tự đã có và điền vào đó tên mà ta lựa chọn. Cần lưu ý rằng không thể xoá bỏ tất cả mọi ký tự mà phải điền vào ít nhất một ký tự.

Mục Change Button Image cho phép lựa chọn một hình vẽ cho biểu tượng macro , chỉ cần bám đơn vào biểu tượng là nó sẽ tự động được đưa và vị trí nút mà ta đã chọn.



Hình 24.4



Hình 25.4

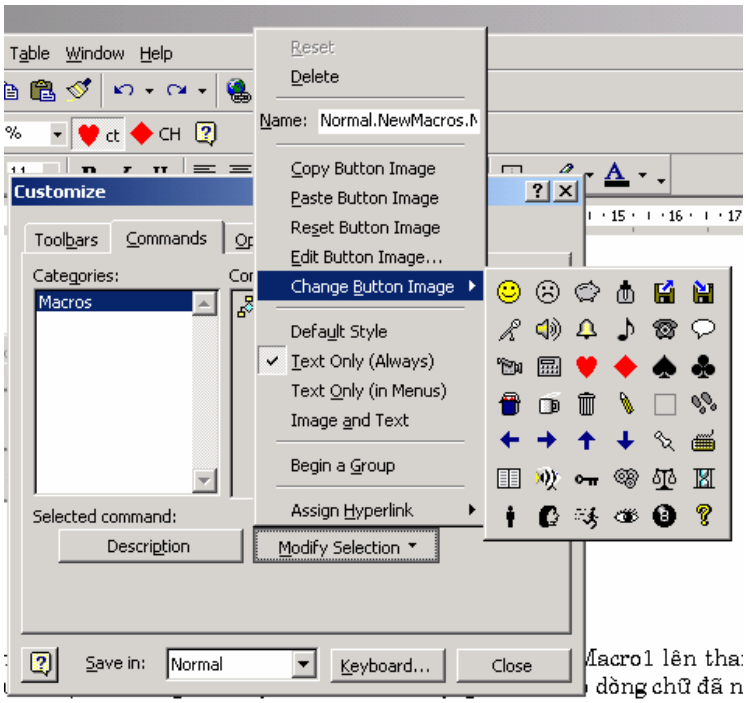
Chọn Close để quay về cửa sổ soạn thảo, lúc này trên màn hình xuất hiện một thanh công cụ gồm hai nút:

Stop: dùng để dừng Macro

Pause: dùng để tạm ngừng ghi Macro

Con trỏ chuột trên màn hình xuất hiện thêm một biểu tượng giống như băng cát xet. Kể từ thời điểm này mọi thao tác trên bàn phím sẽ được ghi lại và được gửi vào biểu tượng vừa tạo ra trên thanh công cụ.

Kết thúc công việc tạo Macro bấm vào nút Stop.



Hình 26.4

***Chú ý* :**

* Khi thực hiện các thao tác tạo Macro, ta không thể bôi đen văn bản bằng thiết bị chuột mà phải dùng phím Shift và các mũi tên dịch chuyển.

* Để gọi Macro ta chỉ việc bấm vào biểu tượng đã tạo ra trên thanh công cụ.

* Để xoá một Macro đã có trên thanh công cụ ta đè phím Alt đồng thời bấm rê kéo biểu tượng Macro vào vùng soạn thảo, biểu tượng sẽ biến mất.

* Để xoá tên Macro chọn: Tools - Macro - chọn tên Macro rồi chọn Delete

b) Tạo Macro trên bàn phím (KeyBoard)

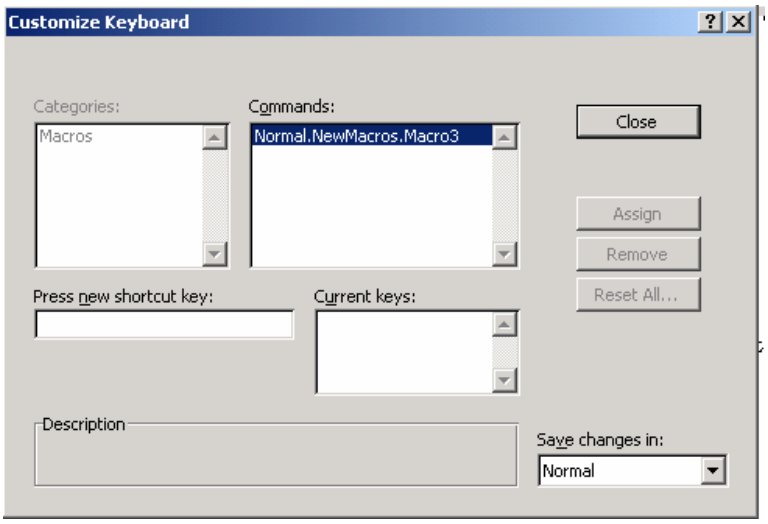
Các bước tiến hành:

- Lặp lại các bước 1a - 1c
- Chọn KeyBoard sẽ xuất hiện cửa sổ mới (Hình 27.4)

Trong mục chọn Press New Shortcut Key ta bấm một tổ hợp phím ví dụ Ctrl - Alt - A,

- Chọn Assign, tổ hợp phím trên được ghi vào vùng Current Key
- Chọn Close để trở về màn hình soạn thảo

Các thao tác tiếp theo sẽ lặp lại như mục 1



Hình 27.4

c) Gọi Macro

Để gọi một Macro tiến hành các thao tác sau:

* Đặt con trỏ ở đầu một dòng trống
* Chọn Tools - Macro - Macro xuất hiện cửa sổ mới
* Chọn tên Macro trong danh mục sau đó chọn Run

7 - Đồ hoạ

7.1 - Các thao tác vẽ cơ bản

Đồ hoạ là phần công cụ phụ trợ trong Winword, ở đây chỉ bao gồm những chức năng cơ bản, nếu muốn sử dụng công cụ đa năng hơn ta có thể dùng PowerPoint là một ứng dụng độc lập với Winword và cũng được cài đặt trong Windows.

Chọn nút Drawing trên thanh công cụ ta sẽ có thêm một thanh công cụ mới, thanh này thường được bố trí dưới đáy màn hình (Hình 28.4). Nếu máy cài đặt bộ Office 97 hoặc office 4.3 thì thanh công cụ vẽ có thể khác đi đôi chút vì đó là các phiên bản cũ.

Hình 28.4

Thanh công cụ đồ hoạ bao gồm các nút có tên như sau (theo thứ tự từ trái sang phải):

Draw	: Kẻ lưới, quay hình, chọn thứ tự hình ….
Select Object	: Chọn đối tượng vẽ
Free Rotace	: Quay hình tự do
AutoShapes	: Chọn các hình mẫu có sẵn
Line	: Vẽ đường thẳng
Arrow	: Vẽ mũi tên
Rectangle	: Vẽ hình vuông hoặc chữ nhật
Oval	: Vẽ hình tròn hoặc Ellipse
Text Box	: Tạo một hộp chứa văn bản
Insert WordArt	: Chèn chữ nghệ thuật
Insert Clip Art	: Chèn ảnh

Fill Color : Chọn màu nền cho những hình khép kín
Line Color : Chọn màu nét vẽ
Line Style : Chọn dạng nét vẽ
Font Color : Chọn màu cho chữ
Dash Style : chọn kiểu nét vẽ
Arow Style : Chọn kiểu mũi tên
Shadow : Chọn hình có bóng
3-D : Chọn hình không gian 3 chiều

Trong mục chọn Draw chúng ta có thể tìm được các chức năng sau (có thể một vài chức năng đã bị xoá):

Bring to Front : Chuyển một hình vẽ ở dưới lên phía trên
Send to Back : Chuyển hình vẽ ở trên xuống dưới
Bring in Front of Text : Đưa hình che lên trên văn bản
Send Behind Text : Cho văn bản hiện lên trên hình
Group : Gộp một số hình thành một nhóm
Ungroup : Tách một nhóm hình thành các phần riêng rẽ
Flip Horiontal : Chuyển hình bên trái sang bên phải
Flip Vertical : Chuyển hình phía trên xuống phía dưới
Rotate Right : Quay hình 90^o theo chiều kim đồng hồ
Reshape : Sửa hoặc bẻ gãy nét vẽ tự do (Freedom)
Snap to Grid : Kẻ lưới trên đối tượng vẽ
Align Drawing Objects : Canh lề các hình vẽ (dóng thẳng lề theo một hình nào đó)
Create Picture : Tạo (sửa chữa một bức vẽ đã có)
Insert Frame : Chèn thêm một khung (chứa văn bản vào trong bức vẽ)

- Chọn một trong năm dạng nét vẽ (Line, Rectangle, Ellipse....) bằng cách bấm đơn vào biểu tượng nét vẽ đó, chuyển chuột vào vùng soạn thảo, chuột sẽ biến thành hình dấu "+".
- Bấm di chuột để tạo hình có kích thước mong muốn.
- Đánh dấu hình: Đưa chuột đến gần hình chuột sẽ xuất hiên thêm biểu tượng hai mũi tên vuông góc. Bấm đơn để đánh dấu hình (xuất hiện các khối vuông đen bao quanh hình)
- Tô màu:
  * Nếu là nét vẽ không khép kín thì chỉ có thể tô màu nét vẽ, chọn Line Color - chọn một màu nào đó
  * Nếu là nét vẽ khép kín (hình vuông , tròn, tam giác....) việc chọn màu nét vẽ như mục 3, để chọn màu nền chọn Fill Color - rồi chọn màu
- Chọn dạng nét vẽ: Đánh dấu hình rồi chọn nút Line Style - sau đó chọn một nét thích hợp.
- Muốn đưa một dòng chú thích và mũi tên chỉ vào một vị trí nào đó:
  + Chọn AutoShapes
  + Chọn Callout
  + Đặt chuột (dấu "+") vào vị trí cần chỉ mũi tên
  + Bấm di chuột sẽ tạo ra một hình để ghi chú thích bên trong

***Một số điều cần lưu ý*:**

*** Chọn nhiều hình cùng một lúc**

Đè phím Shift rồi đánh dấu lần lượt từng hình

*** Chuyển vị trí hình**

Khi vẽ hình, hình vẽ sau sẽ che lên trên hình vẽ trước. Muốn thấy hình vẽ trước ta thực hiện:

- Đánh dấu hình vẽ sau

- Chọn Draw - Order
- Chọn nút Send to Back

Ngược lại hình vẽ sau có thể đưa lên trước bằng cách đánh dấu hình rồi chọn nút Bring to Front

* Nếu có hai hình nằm theo hàng ngang và cả hai hình đang được đánh dấu ta có thể: quay hình đi 90^o (nút Rotate Right), chuyển hình trái sang phải (Flip Horizontal)

* Nếu có hai hình nằm theo hàng dọc và cả hai hình đang được đánh dấu ta có thể: quay hình đi 90^o (nút Rotate Right), chuyển hình trên xuống dưới (Flip Vertical)

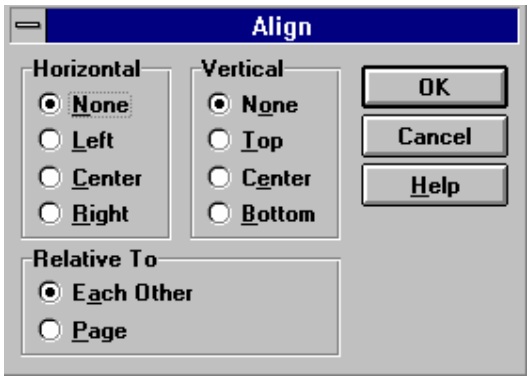
*** Viết chữ trong hình:**

- Chọn nút Text Box rồi vẽ trong hình một khung, trong khung sẽ có con trỏ để viết văn bản
- Nếu không có Text Box ta có thể viết trực tiếp trong hình, chữ viết ra không nhình thấy vì nằm bên dưới hình, muốn chữ hiện lên:
- Đánh dấu hình
- Chọn nút: Send Behind text

Chú ý: *Để có thể đưa con trỏ vào trong hình vẽ, trước khi vẽ cần đưa con trỏ xuống đáy màn hình. Việc dịch chuyển con trỏ sang ngang phải dùng phím Space (phím khoảng cách)*

*** Dóng vị trí (canh lề) các hình:**

-Vẽ hai hình ở vị trí bất kỳ trên màn hình,
- đánh dấu cả hai hình
- chọn Align Drawing Objects, xuất hiện cửa sổ (Hình 29.4)



Hình 29.4

Horizontal: (theo chiều ngang)
None : Không canh lề
Left : Dóng thẳng theo cạnh trái của hình
Center : Đặt hình vào giữa theo chiều ngang
Right : Dóng thẳng theo cạnh phải của hình

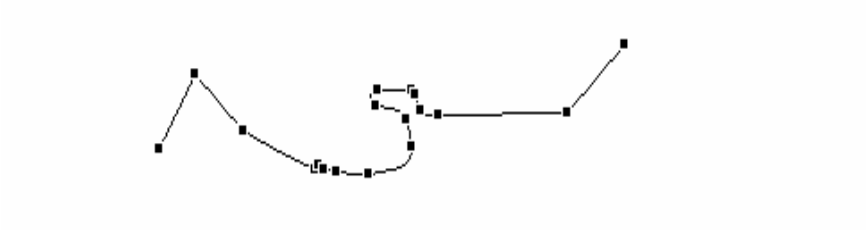
Vertical: (theo chiều dọc)
None : Không canh lề theo chiều dọc
Top : Dóng thẳng theo mép trên của hình
Center : Dóng thẳng theo đường xuyên tâm ngang
Bottom : Dóng thẳng theo mép dưới của hình

Bẻ gẫy nét vẽ:

Chọn AutoShapes - Lines - freedom vẽ 1 nét bất kỳ bao gồm một số đoạn thẳng và đoạn cong (Hình 30.4), đánh dấu nét vẽ

*Chọn Draw - Edit Poitns nét vẽ sẽ được đánh dấu tại các điểm giao nhau (Hình 31.4)

Hình 30.4



Hình 31.4

Trên nét vẽ các đoạn thẳng sẽ được đánh dấu ở hai đầu, còn các đoạn cong sẽ được đánh dấu liên tục bởi các ký hiệu đánh dấu. Đưa chuột vào một trong các ký hiệu đánh dấu bấm rê chuột ta sẽ uốn được nét vẽ theo ý muốn. Nếu muốn bẻ gãy nét vẽ tại một vị trí của đoạn thẳng (trên đó không có ký hiệu đánh dấu) ta có thể làm như sau:

- Đưa chuột đến vị trí cần bẻ gãy
- Bấm rê chuột, nét vẽ sẽ bị bẻ và tạo cho ta vị trí gãy như một cánh sao.

7.2 - Chữ nghệ thuật

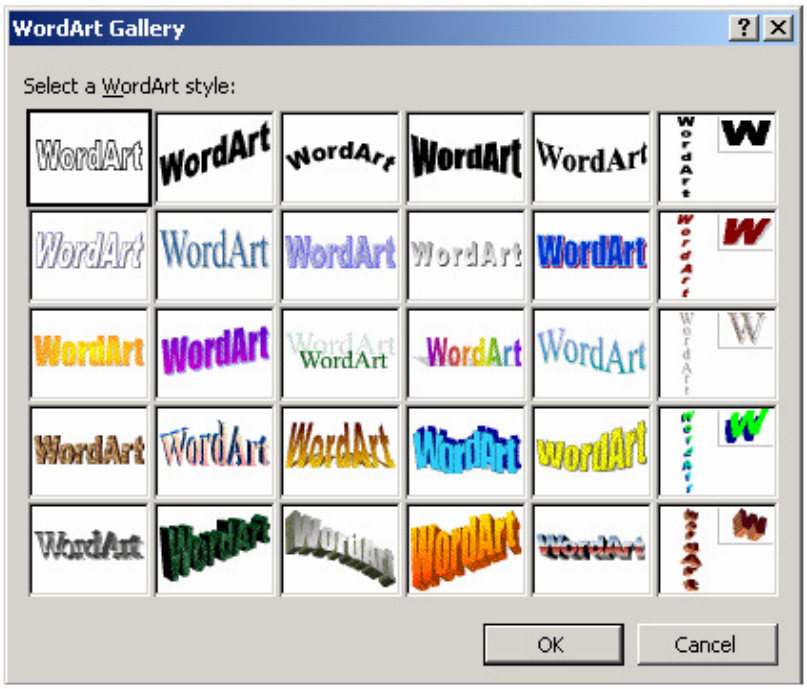
Chữ nghệ thuật được hiểu là các kiểu chữ phi tiêu chuẩn đã được thiết kế sẵn trong Word. Chữ nghệ thuật được sử dụng để viết các khẩu hiệu, trang trí trên các hình vẽ v.v... Thao tác viết chữ nghệ thuật được thực hiện qua các bước sau:

Bấm đơn vào nút đồ hoạ Drawing trên thanh công cụ để xuất hiện thanh công cụ đồ hoạ

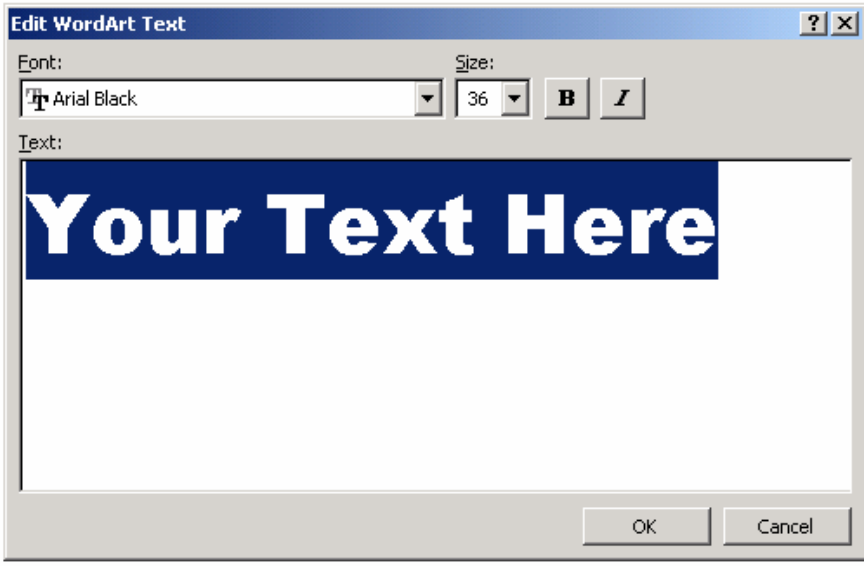
Bấm đơn vào nút Insert WordArt sẽ xuất hiện cửa sổ WordArt Gallery (Hình 32.4)

Chọn một kiểu chữ nghệ thuật mà bạn muốn rồi bấm OK sẽ xuất hiện cửa sổ Edit WordArt Text (Hình 33.4)

Font chữ hiện thời là font tiếng anh. Để chọn Font tiếng Việt trong mục Font kích chuột vào mũi tên bên phải sau đó chọn một Font tiếng Việt, ví dụ chọn VnAristote



Hình 32.4



Hình 33.4

Trong mục Size có thể chọn kích thước chữ, ngoài ra bạn có thể chọn kiểu chữ đậm (B), nghiêng (I).

Xoá dòng chữ Your Text Here và gõ dòng chữ cần trình bày ví dụ :

Chúc mừng sinh nhật

Chọn OK

Dòng chữ sẽ hiên trên màn hình như hình 34.4 đồng thời có một thanh công cụ hiện kèm theo. Sử dụng thanh công cụ này bạn có thể thay đổi dòng chữ theo ý muốn.

Hình 34.4

Các nút trên thanh công cụ này là:

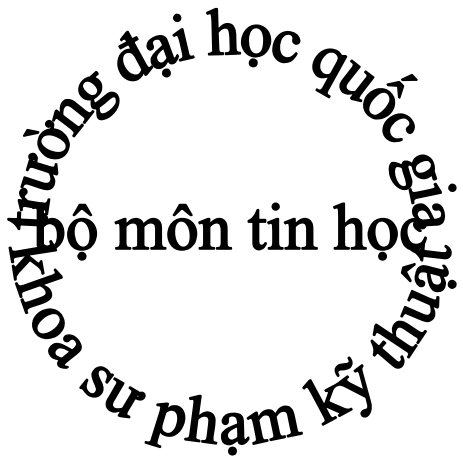
* Insert WordArt: chọn một kiểu chữ mới
* Edit Text: sửa lại chữ cũ
* WordArt Gallery: thư viện các kiểu chữ nghệ thuật
* Format WordArt: Định dạng nét và màu chữ
* WordArt Shape: thay đổi kiểu trình bày
* Free Rotate: quay tự do
* Text Wrapping: cách thức hiện chữ và hình
* WordArt Same letter Hights: biến chữ hoa thành chữ thường và ngược lại
* WordArt Vertical text: chuyển chữ nằm ngang thành thẳng đứng và ngược lại
* WordArt Agliment: canh lề chữ
* WordArt Character Spacing: chọn chữ gầy hay béo

Hình 35.4 cho ta một dạng của chữ nghệ thuật, muốn có chữ kiểu con dấu, ta phải gõ vào vùng your text here ba dòng khác nhau. Ví dụ:

Trên cùng là dòng Trường Đại học Quốc gia

Tiếp đó là dòng Bộ môn Tin học

Dưới cùng là dòng Khoa Sư phạm kỹ thuật



Hình 35.4

Chú ý:

* Việc chọn Font chữ hoặc chọn màu chỉ có thể thực hiện trong cửa sổ của Word Art mà không thể chọn theo cách thông thường.

* Muốn sửa lại một WordArt đã có, bấm kép vào vùng chữ đã được tạo ra ta sẽ trở lại cửa sổ WordArt.

* Với một WordArt đã có ta có thể mở rộng hoặc thu hẹp kích thước bằng cách bấm đơn vào vị trí bất kỳ thuộc Word Art đó, lúc này toàn bộ Word Art sẽ nằm trong một hình chữ nhật. Các cạnh của hình chữ nhật được đánh dấu bởi các hình vuông đen. Đặt chuột vào một trong các hình vuông đen đó, bấm rê chuột sẽ thay đổi được kích thước của WordArt.

8 - Trộn tài liệu tạo thành văn bản mới

Nội dung:

Trộn dữ liệu trong một bảng biểu đã có sẵn (đã lưu trong đĩa cứng với một tên nào đó) với một văn bản mẫu để tạo thành một văn bản mới. Ví dụ đã có sẵn một mẫu giấy mời và một danh sách khách mời ta sẽ trộn chúng với nhau để có một tập giấy mời hoàn chỉnh.

Các bước tiến hành:

* Tạo sẵn mẫu giấy mời và ghi vào đĩa với tên là GM.
* Tạo một bảng biểu chứa danh sách những người sẽ mời
* Trộn lẫn hai văn bản này thành một giấy mời hoàn chỉnh

8.1 - Tạo mẫu giấy mời

cộng hoà xã hội chủ nghĩa việt nam
độc lập tự do hạnh phúc

Giấy mời
Trân trọng kính mời Ông/Bà:
Đến dự :
Thời gian :
Địa điểm :
Rất hân hạnh được đón tiếp

Ngày tháng năm 1996

* Ghi giấy mời vào đĩa với tên GM

8.2 - Tạo danh sách khách mời

* Chọn nút NEW để mở một của sổ mới che lên trên cửa sổ chứa giấy mời
* tạo một bảng chứa danh sách khách mời, ví dụ bảng có bảng 4 cột và 6 dòng theo mẫu dưới đây

Hoten	noidung	thoigian	diadiem
Nguyễn Văn Tâm	Dự họp tổng kết	8h30	Hội trường 204
Trần Thanh Bình	Dự liên hoan chia tay	11h30	Nhà hàng Hương sen
Cao Thị Vân	Học nghị quyết	7h30	Phòng 12 - A1
Đào Hoàng Lâm	Họp thi đua	7h30	Phòng hội thảo
Phạm Minh Hồng	Dự cưới	10h30	Nhà riêng

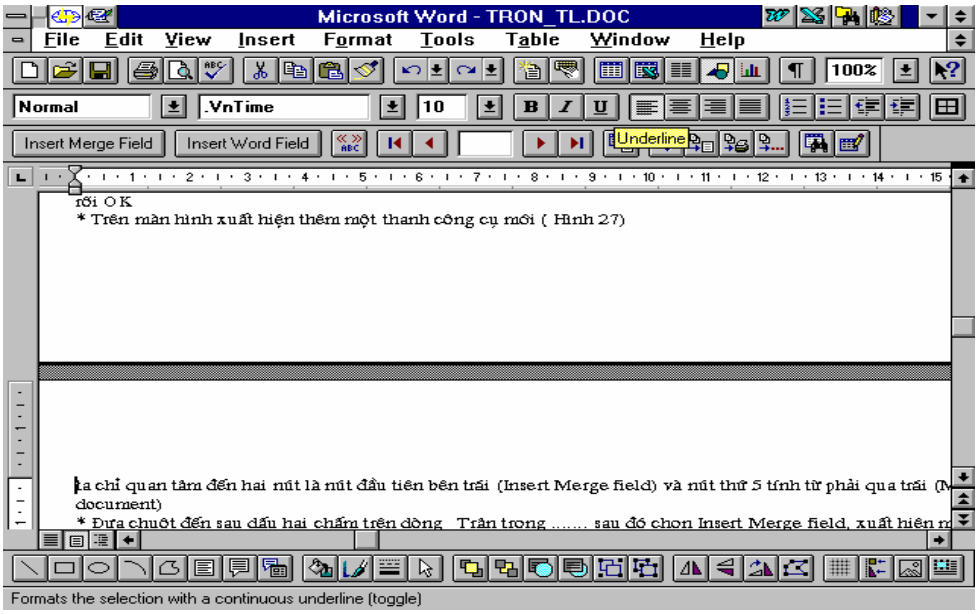
Lưu ý:

Dòng đầu tiên của bảng biểu là Tiêu đề do đó không viết tiếng việt có dấu, không được viết các chữ cách nhau - Không trang trí đường viền của bảng biểu

* Ghi bảng biểu này vào với tên DS (danh sách)
* Chọn FILE - CLOSE đóng cửa sổ bảng biểu lại, lúc này cửa sổ chứa giấy mời lại hiện ra.
* Chọn TOOLS - MAIL MERGE - CREAT - FORM LETTERS...- ACTIVE WINDOS

* Chọn tiếp GET DATA - OPEN DATA SOURSE Khi đó hiện lên tên các tệp lưu trong đĩa cứng của máy, chọn tên tệp DS rồi OK

* Trên màn hình xuất hiện thêm một thanh công cụ mới (Hình 36.4)



Hình 36.4

ta chỉ quan tâm đến hai nút là nút đầu tiên bên trái (Insert Merge field) và nút thứ 5 tính từ phải qua trái (Merge to New document)

* Đưa chuột đến sau dấu hai chấm trên dòng Trân trọng ..... sau đó chọn Insert Merge field, xuất hiện một khung nhỏ chứa tiêu đề các cột bảng biểu - nháy chuột vào Hoten. Lúc này bên cạnh dấu hai chấm sẽ xuất hiện một nhóm ký tự lạ

* Tiếp tục thao tác với các dòng Thời gian, Địa điểm ...

* Cuối cùng nháy chuột vào nút Merge to New document ta sẽ có được số giấy mời cần thiết

* Ghi văn bản này vào đĩa hoặc đóng cửa sổ lại không ghi.

Chú ý:

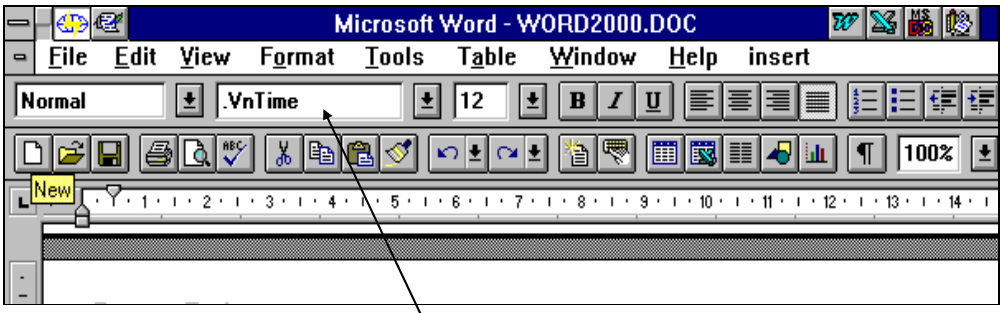
Nếu muốn trình bày kiểu chữ cho văn bản sẽ trộn ta chỉ việc bôi đen cụm ký tự vừa điền vào giấy mời sau đó chọ Format - Font và làm các thao tác như đã trình bày ở phần đầu.

Sau khi trộn có thể mỗi giấy mời sẽ nằm trên một trang giấy. Muốn ghép một số giấy mời vào một trang ta cần để con trỏ ở dưới giấy mời (cách một , hai dòng) sau đó bấm hai lần phím Delete , giấy mời phía dưới sẽ được kéo lên trang hiện thời.

9 - Các thao tác hỗ trợ

9.1 - Khôi phục các thanh công cụ

Với những người mới làm quen với Word rất dễ xảy ra tình trạng làm dịch chuyển hoặc mất thanh công cụ, thanh định dạng. Phổ biến nhất là thanh định dạng nằm phía trên, thanh công cụ nằm dưới (Hình 37.4).



Vị trí bấm và di chuột

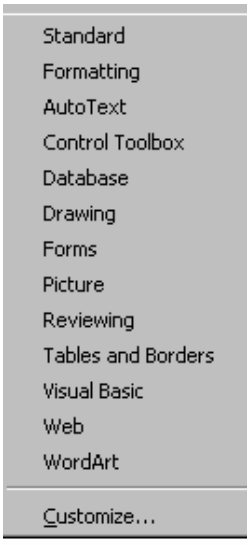
Hình 37.4

Để đưa thanh định dạng xuống dưới ta chỉ việc đưa chuột vào vị trí trống giữa các nút hoặc nhóm nút sau đó đè phím trái chuột, lúc này đường viền thanh sẽ biến thành nét đứt. Bấm và di thanh xuống phía dưới. Khi thanh định dạng nằm trùng lên thanh công cụ thì buông nút trái chuột.

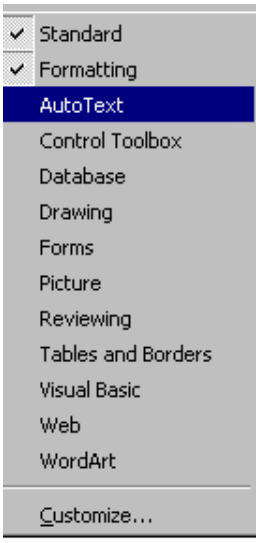
9.2 - Hiện hoặc giấu các thanh công cụ

Ngầm định trên màn hình chỉ có 2 thanh công cụ là thanh Standard và thanh Formating tức là thanh công cụ chuẩn và thanh định dạng. Một số nút trên thanh Standard như nút Drawing (đồ hoạ), nút Table and Border (vẽ bảng) khi chọn lại cho hiện lên thanh công cụ tương ứng. Việc cho hiện hoặc giấu các thanh công cụ có thể thực hiện theo các bước sau đây:

- Khi một nút ứng với một thanh nào đó đang có màu trắng (nghĩa là nó đang được chọn) thì thanh công cụ tương ứng sẽ xuất hiện trên màn hình. Bấm đơn vào nút này thanh công cụ tương ứng sẽ biến mất và nút trở thành màu xám. Bấm tiếp vào nút thanh công cụ sẽ lại hiện ra.
- Nếu trên màn hình không có một thanh công cụ nào, hoặc thiếu thanh công cụ cần thiết, đưa chuột vào một vị trí bất kỳ trên thanh thực đơn, bấm phím phải chuột sẽ xuất hiện một hộp thoại (Hình 38.4)



Hình 38.4



Hình 39.4

Bấm đơn vào tên thanh công cụ muốn chọn thanh này sẽ xuất hiện trên màn hình, thao tác tương tự với các thanh khác.

Chú ý:

- Nếu phía bên trái tên thanh công cụ đã có ký hiệu đánh dấu (Hình 39.4) mà trên màn hình vẫn không thấy thanh công cụ đó thì cần tìm xem nó nằm đâu đó trên màn hình, có khi chỉ nhìn thấy một phần rất nhỏ của thanh công cụ, phần còn lại nằm khuất ngoài phạm vi màn hình. Chỉ cần đưa chuột vào phần nhỏ đó bấm rê nó vào trong màn hình là được.
- Không nên cho hiện lên quá nhiều thanh công cụ trên màn hình vì như vậy sẽ không còn không gian để làm việc. Khi dùng xong một thanh công cụ nào nếu không cần thiết thì nên cất nó đi.

9.3 - Thêm bớt chức năng trên thanh công cụ

Mỗi thanh công cụ của Word có một số biểu tượng (nút) thể hiện một số chức năng, chúng ta có thể tuỳ ý thêm bớt các chức năng vào các thanh này hoặc chuyển các nút từ thanh này sang thanh khác.

a) Xoá hoặc chuyển vị trí một nút

Đè phím Alt trên bàn phím sau đó bấm rê nút muốn xoá vào vùng soạn thảo, buông hai tay ra nút sẽ bị xoá.

Nếu chúng ta lôi nút đến một thanh công cụ khác rồi buông tay ra thì nút sẽ nằm trên thanh công cụ đó.

b) Thêm một nút mới vào thanh công cụ

Giả sử bạn luôn phải định dạng chữ, hãy thêm vào thanh Standard các nút thể hiện Font chữ hay dùng. Cách thức tiến hành như sau:

Chọn Tools - Customize (hoặc đưa chuột lên thanh công cụ rồi bấm phím phải sau đó chọn Customize) xuất hiện cửa sổ.

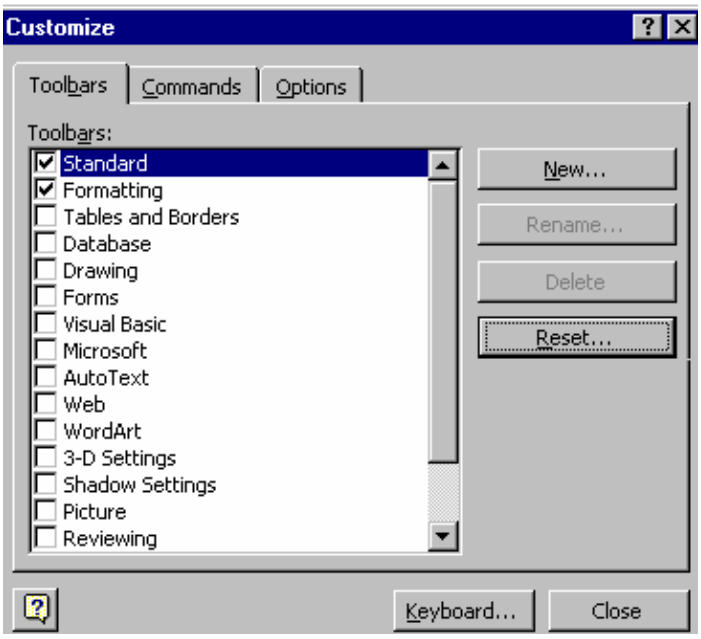
Trong cửa sổ Customize chọn chức năng Command, kéo thanh cuốn phía dưới để tìm chức năng Fonts và bấm đơn vào đó phía bên phải sẽ xuất hiện toàn bộ các font chữ có trong Word. Bấm rê một font nào đó lên thanh công cụ, tiếp đó bấm vào Modify Selection để thay đổi chữ hiện trong nút và chọn biểu tượng cho nút (xem lại phần Macro).

Khi một nút thể hiện một font chữ đã có trên thanh công cụ việc định dạng chữ sẽ trở nên rất đơn giản, chỉ cần bôi đen chữ rồi bấm vào nút font đã có.

c) Hồi phục trạng thái ngầm định của một thanh công cụ

Đôi khi vì tò mò hay vô tình bạn phá hỏng một thanh công cụ, để khôi phục lại dạng ngầm định của thanh đó chúng ta có thể làm như sau:

Chọn Tools - Customize (hoặc đưa chuột lên thanh công cụ rồi bấm phím phải sau đó chọn Customize) xuất hiện cửa sổ hình 40.4



Hình 40.4

Chọn chức năng Toolbars khi đó tên các thanh công cụ xuất hiện phía dưới. Bấm đơn vào dòng chữ tên thanh công cụ để nó nằm trên nền màu xanh. (Nhớ là không bấm vào hình vuông bên trái) sau đó bấm đơn vào nút Reset phía bên phải cửa sổ, lúc này xuất hiện cửa sổ nhỏ. Trong cửa sổ Reset Toolbars chọn OK, thanh công cụ đã chọn sẽ được khôi phục lại nguyên dạng ngầm định mà Word đã thiết kế.

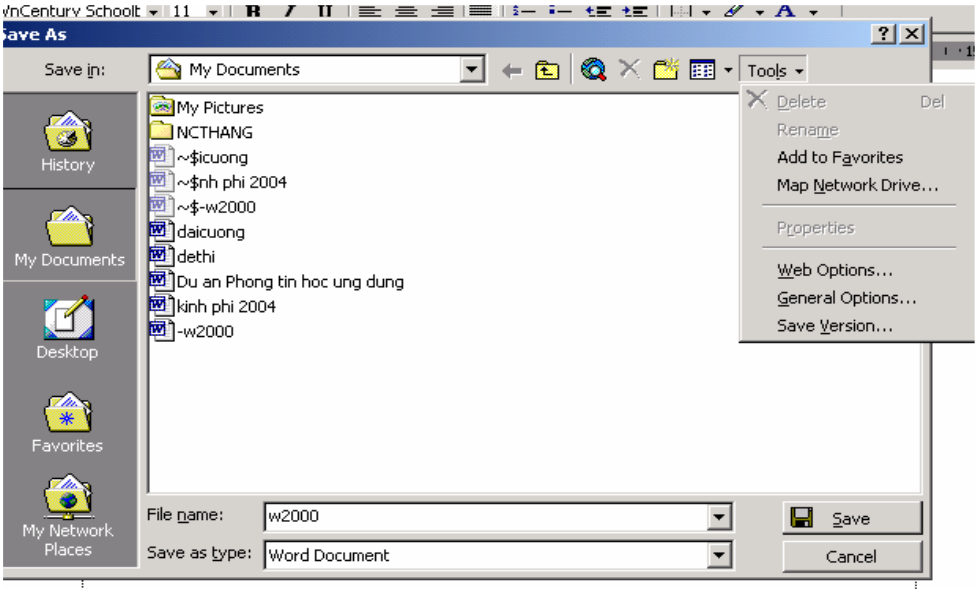
9.4 - Bảo mật văn bản

Những văn bản quan trọng không thể để cho người khác tự động mở ra cần được bảo vệ bằng mật khẩu. Bảo mật văn bản có hai cấp:

* Cho phép đọc nhưng không được sửa chữa
* Không cho phép đọc

Thao tác tiến hành:

- Chọn File - Save As xuất hiện cửa sổ hình 41.4



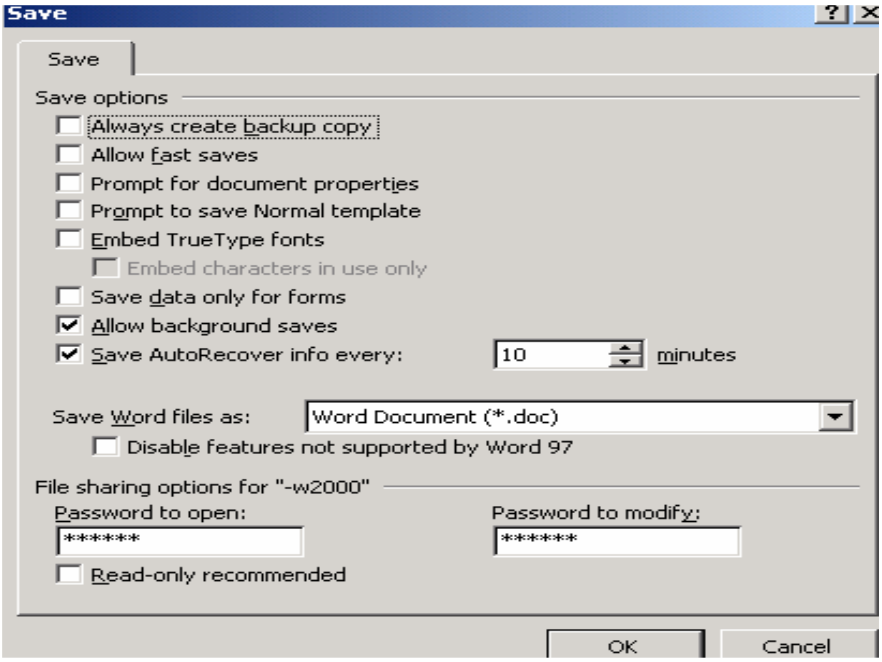
Hình 41.4

- Trong cửa sổ Save As chọn Tools - chọn tiếp General Options, sẽ xuất hiện cửa sổ hình 42.4

Mục Password to Open: Mật khẩu cho phép mở văn bản, chúng ta tuỳ chọn một mật khẩu nào đó. Nên chọn khoảng 5-6 ký tự và cần lưu mật khẩu này vào sổ để đề phòng sau một thời gian có thể bị quên. Với mật khẩu này chúng ta có thể mở văn bản ra xem nhưng không thể thêm bớt gì vào văn bản.

Mục Password to Modify: Mật khẩu cho phép sửa chữa văn bản, có thể chọn mật khẩu giống như Mục Password to Open hoặc chọn khác đi. Chọn xong bấm OK máy sẽ xuất hiện cửa sổ mới yêu cầu nhắc lại mật khẩu một lần nữa. Gõ lại đúng các mật khẩu đã chọn. Bấm OK thoát ra.

Sau khi đã chọn mật khẩu xong chọn Save để ghi lại văn bản kèm theo mật khẩu. Kể từ nay muốn mở văn bản ra xem chúng ta phải khai báo mật khẩu.


Save
Save
Save options
Always create backup copy
Allow fast saves
Prompt for document properties
Prompt to save Normal template
Embed TrueType fonts
Embed characters in use only
Save data only for forms
Allow background saves
Save AutoRecover info every:
10
minutes
Save Word files as:
Word Document (*.doc)
Disable features not supported by Word 97
File sharing options for "-w2000"
Password to open:
******
Password to modify:
******
Read-only recommended
OK
Cancel

Hình 42.4

Bài tập Chương IV

Bài 1: Soạn thảo văn bản và kẻ bảng dưới đây:

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

GIẤY CAM ĐOAN BẢO VỆ MÔI TRƯỜNG ĐIỆN NƯỚC

Kính gửi : .....

Tên tôi là : .....
Công tác tại : .....
Nơi ở : .....

Tôi làm giấy này xin cam đoan trong quá trình cải tạo, nâng cấp căn hộ của gia đình, không làm điều gì ảnh hưởng tới hệ thống điện, nước chung của cả khu vực.

Tôi xin cam đoan sẽ đảm bảo vệ sinh môi trường, không để vật liệu ảnh hưởng đến đường đi và quang cảnh chung của cả khu. Trong quá trình xây dựng không làm ảnh hưởng đến móng, tườmg nhà của 2 hộ bên cạnh, ***nếu xảy ra điều gì tôi hoàn toàn chịu trách nhiệm*** .

Hà nội, ngày ... tháng ... năm .....
Người viết đơn

BẢNG ĐIỂM THI HỌC KỲ II NĂM HỌC 2003 - 2004

Lớp 12 B trường PTTH Trần Hưng Đạo

Stt	Họ và tên	Ngày sinh	Điểm thi			Hạnh kiểm
			Điểm Tin	*Điểm Toán*	*Điểm Lý*	
1	Lê Thu An	20-10-83	7	5	5	**Tốt**
2	Trần Thi Hà	15-01-82	8	6	7	**Tốt**
3	Đỗ Văn Minh	22-11-84	9	7	8	**Tốt**
4	Ngô Thị Thái	18-03-83	5	6	9	**Khá**
5	Lê Văn Thư	01-10-84	6	8	6	**Tốt**

Bài 2: Soạn thảo văn bản và kẻ bảng dưới đây:

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc
--------- o O o ---------

ĐỀ CƯƠNG NGHIÊM CỨU KHOA HỌC CẤP TRƯỜNG

Tên đề tài : **Một số chương trình quản lý hệ thống thông tin đất**
Chủ nhiệm đề tài: .....
Những người tham gia: .....

1- Mục đích:

Xây dựng chương trình để quản lý các thông tin đất một cách tự động ***là nhiệm vụ quan trọng của một kỹ sư ngành Quản lý đất đai trong thời kỳ hiện nay***. Đối với sinh viên ngành Quản lý đất đai nắm được cách tổ chức các cơ sở dữ liệu và cách tổ chức các chương trình về thông tin đất có hệ thống là việc làm vô cùng cần thiết. Điều đó củng cố các kiến thức về Tin học và Chuyên môn, gắn chặt lý thuyết đã học với yêu cầu thực tế trong lĩnh vực chuyên môn của mình.

Xây dựng các chương trình quản lý hệ thống thông tin đất cũng là đáp ứng được yêu cầu đòi hỏi của thực tế trong chuyên ngành Quản lý đất đai hiện nay.

Xuất phát từ các vấn đề trên chúng tôi tiến hành đề tài : " ***Một số chương trình quản lý hệ thống thông tin đất ".***

BẢNG DANH SÁCH LÊN LƯƠNG NĂM 2000

Trường Đại học Nông nghiệp I – Hà nội

STT	HỌ VÀ TÊN	ĐƠN VỊ	BẬC LƯƠNG		GHI CHÚ
			Cũ	*Mới*	
1	Trần Ngọc Anh	Phòng HC	3.37	3.62	
2	Lê Văn Khanh	Phòng ĐT	2.95	3.37	
3	Bùi Ngọc Bình	Phòng ĐT	4.12	4.42	*Phó phòng*
4	Ngô Bình Minh	Phòng QT	5.05	5.37	*Trưởng phòng*
5	Đỗ Minh Thanh	Phòng HC	2.90	3.24	

Bài 3: Soạn thảo văn bản và kẻ bảng dưới đây:

TRƯỜNG ĐẠI HỌC NÔNG NGHIỆP I HÀ NỘI
Phòng HCKH-VT
Số : /CV-HC
V/v làm danh bạ điện thoại

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

Hà nội, ngày 25 tháng 11 năm 2000

Kính gửi : ***Các đơn vị trong toàn trường***

Trong hai năm qua cuốn danh bạ điện thoại của trường đã góp phần không nhỏ vào sự phát triển thông tin liên lạc của Nhà trường. Hiện nay cuốn danh bạ đó không cung cấp đầy đủ số máy của các cá nhân và tập thể trong trường.

Vì vậy, phòng HCKH-VT sẽ phát hành cuốn "***Danh bạ điện thoại của Trường Đại học Nông nghiệp I - năm 2001***". Để công việc hoàn thành trước tết dương lịch, kính đề nghị các cá nhân, tập thể cung cấp đầy đủ và chính xác số điện thoại mới của mình cho phòng HCKH-VT trước ngày 05/12/1999.

Chúng tôi mong nhận được sự cộng tác chặt chẽ của các đơn vị và cá nhân trong toàn trường.

*Xin trân trọng cảm o*n.

Trường phòng
Hành Chính kế hoạch - Vật tư

BẢNG THỐNG KÊ THUÊ NĂM 2000

STT	Tên hợp tác xã	Quí 1			Quí 2		
		Tháng 1	*Tháng 2*	*Tháng 3*	*Tháng 4*	*Tháng 5*	*Tháng 6*
1	Bát tràng	50000000	60000000	50000000	50000000	40000000	50000000
2	Cổ bi	35000000	55000000	35000000	35000000	55000000	75000000
3	Dương xá	22000000	42000000	22000000	22000000	32000000	25000000
4	Đa tốn	20000000	50000000	20000000	20000000	50000000	60000000
5	Trâu quỳ	40000000	80000000	40000000	40000000	20000000	50000000

Bài 4: Soạn thảo văn bản và kẻ bảng dưới đây:

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

ĐƠN XIN HỢP ĐỒNG

Kính gửi **:** .....

Tên tôi là: ..... *Ngày sinh:* .....
Chỗ ở hiện nay: .....
Trình độ văn hoá: .....
Hiện nay Bộ môn do thiếu cán bộ kỹ thuật nên đề nghị khoa và phòng Tổ chức cán bộ cho tôi được ký hợp đồng làm việc tại bộ môn từ 1-1-2000 đến 1-1-2001 .

Nếu được thu nhận ***tôi xin cam đoan tuyệt đối chấp hành nội qui học tập và lao động***. Tuyệt đối phục tùng sự phân công học tập, bố trí của tổ chức đơn vị sử dụng. Nếu vi phạm nội qui, kỷ luật hoặc tự ý thôi, bỏ học, bỏ việc phải bồi thường hoàn toàn học phí, những tổn thất đã gây ra theo quy định của Nhà nước và của nhà trường đơn vị công tác.

Kính mong các cấp giải quyết.

Hà Nội, ngày tháng năm
Người làm đơn

Thời khoá biểu năm học 2000-2001

THỨ 2		THỨ 3		THỨ 4		THỨ 5		THỨ 6	
Sáng	**Chiều**	**Sáng**	**Chiều**	**Sáng**	**Chiều**	**Sáng**	**Chiều**	**Sáng**	**Chiều**
Toán	Anh	Triết	Sinh	Tin	Anh	Triết	Luật	Toán	Anh
Toán	Anh	Triết	Sinh	Tin	Anh	Triết	Luật	Toán	Anh
Toán	Anh	Triết	Sinh	Tin	Anh	Triết	Luật	Toán	Anh
Tin		Luật		Toán		Sinh		Tin	
Tin		Luật		Toán		Sinh		Tin	
Tin		Luật		Toán		Sinh		Tin	

Bài 5: Soạn thảo văn bản và kẻ bảng dưới đây:

ĐỀ THI CHẤT LƯỢNG MÔN TOÁN
(Thời gian 90 phút)

Câu 1: Cho hàm số : $y = x^3 + 3x^2 + (m+1)x + 4m$ (1)

1/ **Khảo sát và vẽ đồ thị của (1) khi m =-1**

2/ Với giá trị nào của m hàm (1) đồng biến trên khoảng (-1, 1)

Câu 2: Trong mặt phẳng toạ độ Oxy cho 2 điểm $F_1(-3,0)$; $F_2(3,0)$

1/ Viết phương trình chính tắc của elíp nhận F_1; F_2 là các tiêu điểm và trục lớn có độ dài bằng 10

2/ Tìm trên elíp nói trên những điểm M(xo,yo) sao cho khoảng cách từ đó tới đường thẳng cho bởi phương trình : **x + y = 6 là nhỏ nhất**.

Hà nội, ngày 20 tháng 4 năm 2000

Chủ tịch hội đồng thi

BẢNG THỐNG KÊ HÀNG NHẬP

Số tt	Tên hàng	Kho A			Kho B		
		Ngày nhập	*Số lượng*	*Đơn giá*	*Ngày nhập*	*Số lượng*	*Đơn giá*
1	Vải hoa	1/1/98	100	10000	12/1/98	200	10000
2	Vải lụa	1/1/98	200	15000	11/1/98	300	14000
3	Khăn	2/1/98	50	7000	21/1/98	150	8000
4	Bột giặt	3/1/98	100	6500	30/1/98	400	6000
5	Bánh	5/1/98	120	6000	11/2/98	220	5000

CHƯƠNG V: INTERNET VÀ CÁCH SỬ DỤNG INTERNET

Internet là mạng thông tin toàn cầu, nó đóng vai trò quan trọng trong mọi lĩnh vực của xã hội. Hiện nay Internet được sử dụng rộng rãi trên thế giới cũng như ở nước ta. Mọi người cần phải hiểu và biết sử dụng Internet để phục vụ cho chuyên môn của mình.

Chương này cung cấp các kiến thức cơ bản về mạng Internet và thư điện tử. Nội dung chính bao gồm các vấn đề : Thế nào là mạng Internet, Internet có các chức năng cơ bản nào, cài đặt Internet, trình duyệt Internet Explorer, E-mail và các mô hình hoạt động của E-mail, cách tạo gửi và nhận thư với Yahoo và với Outlook Express.

1 - Giới thiệu chung

1.1 - Mạng máy tính

* Mạng máy tính: là một nhóm các máy tính được nối kết với nhau theo một cách nào đó. Về nhận thức, nó giống như kiểu mạng truyền hình hoặc radio nối kết một nhóm các trạm truyền hình hoặc radio lại với nhau sao cho chúng có thể cùng chia sẻ chương trình. Các mạng truyền hình gửi thông tin tương tự đến mọi trạm vào cùng lúc, trong mạng máy tính, mỗi thông điệp thường được dẫn đến một máy tính cụ thể nào đó. Khác với các mạng truyền hình, các mạng máy tính luôn hai chiều sao cho khi máy tính A gửi thông điệp tới máy tính B thì B có thể trả lời lại cho A.

Một số mạng máy tính bao gồm một máy tính trung tâm và một nhóm các trạm từ xa có thể báo cáo về máy tính trung tâm, ví dụ một máy tính dịch vụ giữ chỗ hàng không trung tâm có hàng ngàn kênh tại các sân bay và đại lý du lịch.

* Mạng Internet: Là mạng máy tính toàn cầu, hoặc chính xác hơn là mạng của các mạng.

Để các máy tính nối lại và có thể giao tiếp được với nhau, người ta dùng một thiết bị gọi là modem. Modem và chương trình ứng dụng dùng để chuyển đổi từ tín hiệu tương tự (analog) trên đường dây điện thoại máy tính thành tín hiệu số (digital) để đưa vào máy tính. Ngược lại, modem còn chuyển tín hiệu số của máy tính thành tín hiệu tương tự để truyền dẫn trên đường dây điện thoại. Như vậy, một máy tính muốn kết nối và truy nhập thông tin thì phải có một modem nối vào một cổng của máy tính. Về vị trí lắp đặt , có thể chia modem thành hai loại: loại lắp trong máy tính (internal) và lắp ngoài máy tính (external).

* Xuất xứ của Internet

Tổ tiên của Internet là ARPANET, một dự án do bộ Quốc phòng Hoa ki khởi đầu năm 1969 vừa là một thực nghiệm trong việc kết mạng một cách đáng tin cậy, vừa là một kết nối giữa bộ Quốc phòng và các nhà thầu nghiên cứu khoa học và quân sự lại với nhau, bao gồm một số lớn các trường đại học tiến hành các nghiên cứu quân sự được tài trợ. (ARPA là viết tắt của chữ Advanced Research Projects Agency: Cơ quan Các Dự án Nghiên cứu Cao cấp, một bộ phận thuộc bộ Quốc phòng phụ trách việc cấp phát tài trợ.

Kế tiếp là Hội đồng Khoa học Quốc gia Hoa kú (National Science Foundation - NSF) quyết định thiết lập 5 trung tâm siêu máy tính nhằm mục đích nghiên cứu (Một siêu máy tính là một máy tính cực kỳ nhanh với giá rất đắt, khoảng 10 triệu USD mỗi máy). NSF tính toán rằng nên tài trợ cho một ít máy tính, để cho các nhà nghiên cứu trên khắp đất nước sử dụng ARPANET gửi các chương trình của họ đến để được "siêu tính toán" và rồi gửi kết quả trở lại. Qua một thời gian khi các siêu máy tính rõ ràng đã trở nên lỗi thời, *NSF* đã xây dựng *NSFNET*, một mạng riêng và nhanh hơn nhiều của mình để nối với các trung tâm siêu tính toán. Sau đó, NSF dàn xếp để thiết lập một chuỗi các mạng khu vực nhằm liên kết những

người sử dụng trong từng khu vực với NSFNET nối mọi mạng khu vực. NSFNET hoạt động hiệu quả, tõ n¨m 1990 hÇu hÕt c¸c m¹ng ®Òu kÕt nèi vµo NSFNET, NSF đã được xác lập vững chắc trên Internet.

C¸c m¹ng ®éc lËp liên kết với nhau nhờ vào một chương trình kỹ thuật được gọi là IP (Internet Protocol: Giao thức Internet) . NSFNET chỉ cho phép những lưu thông liên quan đến nghiên cứu và giáo dục, do đó những dịch vụ mạng IP độc lập cho phép lưu thông được dẫn từ mạng này sang mạng khác khi cần. Mọi mạng nối bởi IP đều sử dụng IP để giao tiếp nên chúng đều có thể trao đổi các thông điệp với nhau.

Các mạng thương mại nối với các mạng khu vực theo cách tương tự như kết nối của NSFNET và cung ứng kết nối trực tiếp đến khách hàng.

Bên ngoài Hoa Kỳ, các mạng IP xuất hiện tại nhiều nước, do các công ty điện thoại địa phương tài trợ hoặc do một nhà cung ứng quốc gia hoặc khu vực độc lập. Hầu hết trong số họ đều được kết nối trực tiếp hoặc gián tiếp với một mạng nào đó của Hoa Kỳ, có nghĩa là tất cả đều có thể trao đổi lưu thông với nhau.

* M¹ng Internet cã c¸c chøc n¨ng chÝnh sau ®©y:

- Thư điện tử: Đây chắc chắn là dịch vụ được sử dụng rộng rãi nhất - bạn có thể trao đổi thư điện tử với hàng triệu người trên khắp thế giới. Các danh sách thư điện tử cho phép bạn tham gia vào thảo luận nhóm và gặp những người khác trên Net. Các mail server (những chương trình đáp lại các thông điệp thư điện tử) cho phép bạn truy tìm mọi loại thông tin.

- Truy tìm thông tin: Internet giúp bạn tìm kiếm thông tin trên mạng. Nhiều máy tính có các file chứa thông tin cho phép lấy miễn phí. Các file đó bao gồm các file văn bản, các file hình ảnh được số hóa và một số lượng lớn các phần mềm từ trò chơi cho đến các hệ thống điều hành. Hàng núi thông tin sẵn có trên Internet và bạn sẽ mục mục "Navigate" (Tìm đường) chỉ đến những phần giúp bạn lưu thông trên mạng.

- Bảng thông báo điện tử (Bulletin Board): Một hệ thống tên là USENET là một bảng thông báo điện tử vĩ đại với 40 triệu ký tự các thông điệp bao gồm 2.000 nhóm chủ đề khác nhau được tiếp nối hàng ngày. Các chủ đề bao gồm từ các vấn đề máy tính hóc búa cho đến những trò giải trí như đua xe đạp, từ những ý kiến tranh luận chính trị không bao giờ kết thúc cho đến những vấn đề b×nh thường nhất. Nhóm USENET được đọc rộng rãi nhất là các mẩu chuyện vui được chọn lọc, một số trong những chuyện này thực sự buồn cười.

- Trò chơi và tán gẫu: Một trò chơi được gọi là MUD (Multi-User Dungeon) có thể dễ dàng thu hút toàn bộ thời gian của bạn - trong đó bạn có thể thi tài với những người chơi khác ở bất cứ nơi nào trên thế giới.

Internet Relay Chat (IRC) là một đường dây theo nhóm, qua đó bạn có thể có những cuộc đàm thoại thú vị với những người sử dụng khác ở khắp mọi nơi. IRC dường như được sử dụng nhiều nhất bởi những häc sinh, sinh viên, bạn sẽ không bao giờ biết được ai là người đang nói chuyện với bạn.

1.2 - Nhà cung cấp dịch vụ Internet

Để một máy tính cá nhân nối vào được mạng Internet, bạn phải đăng ký thuê bao ở một nhà cung cấp dịch vụ Internet (Internet Service Provider - ISP) và khi này ISP sẽ là vị trí trung gian chuyển tiếp để máy tính của bạn nối với các trang Web hoặc các địa chỉ trên mạng.

Hiện nay có nhiều nhà cung cấp dịch vụ Internet như VNN, FPT, VCD, Phươngnam, Saigonnet, Netnam …. Mỗi một ISP có những khả năng khác nhau. Khi chọn ISP, bạn nên quan tâm đến một số vấn đề sau:

- Khả năng đảm bảo đường truyền

- Giá đăng ký và thuê bao. Hiện nay giá cả của các nhà cung cấp dịch vụ không giống nhau.

- Vị trí của ISP. Đây là một vấn đề cần hết sức quan tâm khi chọn nhà cung cấp dịch vụ Internet. Ví dụ, nếu bạn ở Hà Nội thì nên chọn VNN hoặc FPT vì lúc này khi trả tiền điện thoại, bạn chỉ phải thanh toán theo giá điện thoại nội hạt. Nếu bạn chọn một ISP khác và không có dịch vụ kết nối mạng thì tiền điện thoại sẽ phải thanh toán từ vị trí gọi đến địa điểm đặt máy chủ của mạng. Bạn cũng cần lưu ý rằng, khi sử dụng Internet có ba loại tiền cần thanh toán:

* Tiền điện thoại: Trong thời gian sử dụng Internet, tiền điện thoại được tính như khi gọi điện bình thường. .

* Tiền truy cập mạng: Số tiền này được tính theo thời gian truy cập và thời điểm truy cập. Hầu hết các ISP đều tính giá tiền cho thời gian truy cập vào ban đêm và các ngày nghỉ thấp hơn ban ngày. Số tiền này bạn phải thanh toán cho ISP. Các nhà cung cấp dịch vụ Internet khác nhau số tiền này cũng khác nhau.

* Tiền lấy các thông tin trên mạng: Hầu hết các trang Web trong nước và nhiều trang Web nước ngoài cung cấp các thông tin miễn phí, tức là bạn không phải trả tiền cho các thông tin hoặc các ứng dụng trên mạng. Tuy nhiên, cũng có một số bạn phải trả tiền.

1.3 - Kết nối Internet

* Điều kiện gia nhập mạng Internet: Để gia nhập mạng Internet bạn phải có các các mục sau:

- Máy vi tính
- Đường điện thoại
- Modem
- Tài khoản đăng ký truy nhập với nhà cung cấp (ISP): Để có thể truy cập Internet, bạn cần ACcount hay tài khoản Internet, bao gồm:

+ User name là tên đăng ký với ISP, tên này không thể thay đổi được. Ví dụ User name máy của bộ môn Tin học là: bmthnn1

+ Password là mật khẩu , một lần kết nối ta phải gõ vào mật khẩu, mật khẩu có thể thay đổi được.

* Tạo biểu tượng kết nối Internet:

Từ màn hình chính của Windows chọn các mục sau:

My Computer / Control panel / Network and Dial Up Networking / Make New Connection

Sau đó sẽ xuất hiện hộp thoại ta sẽ khai báo các mục sau:

- Gõ tên của biểu tượng vào hộp Type a name for the computer you are dialing
- Chọn **Next**
- Gõ mã vùng vào hộp **Are code**
- Gõ số điện thoại của nhà cung cấp vào **Telephone Number**
- Chọn tên nước trong hộp **Country code**
- Chọn **Next**
- Chọn **Finish**

* Cách kết nối

Nháy đúp chuột vào biểu tượng kết nối. Khi đó sẽ xuất hiện cửa sổ Connect to, ta khai báo các mục sau:

- **User name**: Gõ tên đăng ký với ISP
- **Paswword**: Gõ mật khẩu
- **Phone number**: Số điện thoại của ISP

- Chọn **connect** thực hiện kết nối.

Sau khi kết nối được, biểu tượng kết nối có 2 máy tính liên kết nằm ở góc phải, phía dưới màn hình, trên thanh Taskbar của Windows.

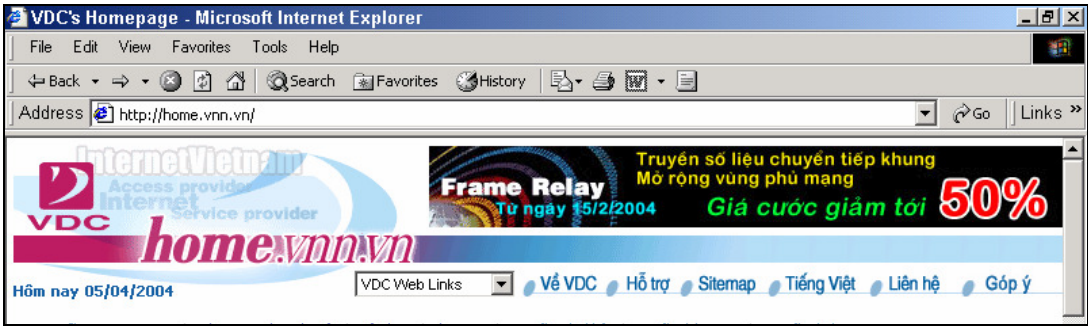
* Huỷ kết nối

- Nháy chuột vào biểu tượng kết nối trên trên thanh **Taskbar**
- Chọn **Disconnect**

2. Sử dụng trình duyệt Internet Explorer (IE)

2.1 - Khởi động

- Kích chuột vào START/ chọn PROGRAM/ chọn INTERNET EXPLORER
- Màn hình IE xuất hiện như sau:



Hình 1.6

Lần lượt từ trên xuống dưới là các phần sau:

+ Dòng đầu tiên là thanh tiêu đề (Title bar): Hiện ra tên trang Web, tên trình duyệt Web.

+ Dòng 2 là thanh Menu (Menu bar: thanh thực đơn): Bao gồm các mục chính của trình duyệt Web.

+ Dòng 3 là thanh công cụ (Tool bar): Bao gồm các biểu tượng của các lệnh hay dùng, muốn chọn lệnh nào ta chỉ việc kích chuột vào biểu tượng của nó.

+ Dòng 4 là thanh địa (Address): Hiện ra địa chỉ của trang Web đang được truy cập. Ta có thể gõ trực tiếp địa chỉ vào đây hoặc nhấn vào mũi tên bên phải mục này để chọn địa chỉ đã lưu.

+ Phần vùng hiển thị: Là phần hiện nội dung trang Web đang truy cập, phần này chiếm nhiều dòng.

+ Thanh trạng thái : Dòng này ở cuối cùng, hiện ra các thông tin hiện thời.

2.2 - Ra khỏi Internet Explorer

- Chọn mục **File**
- Chọn **Close**

2.3 - Tìm kiếm thông tin

* Cách tìm kiếm thông tin nếu có địa chỉ chính xác:

- Gõ địa chỉ cần truy nhập vào thanh địa chỉ (thanh Address)
- Enter (Hoặc kích chuột vào GO)

* Cách tìm kiếm thông tin nếu không có địa chỉ chính xác:

- Sử dụng nút Search trên thanh công cụ

2.4 - Cách lưu trữ các địa chỉ và cách tổ chức lại các địa chỉ

* Cách lưu các địa chỉ ưa thích vào FAVORITES:

Những địa chỉ nào thường xuyên truy cập, mỗi lần truy cập ta phải gõ lại địa chỉ, thay vì việc đó ta lưu lại các địa chỉ để mỗi lần truy cập vào trang web đó chỉ cần kích chuột chọn địa chỉ ở bên trong menu Favorites.

Cách lưu:

- Mở trang WEB cần lưu
- Vào menu Favorites
- Chọn Add to Favorites
- OK

* Tổ chức lại các địa chỉ:

Mỗi lần lưu lại địa chỉ thì menu Favorites sẽ dài thêm. Do vậy để tiện sử dụng ta tổ chức lại menu Favorites như sau:

- Vào menu Favorites/ chọn Organize Favorites/ màn hình xuất hiện:

Trong đó:

+ Create Folder: Tạo một Folder mới
+ Move to Folder: Di chuyển địa chỉ
+ Rename: Đổi tên Folder
+ Delete: Xoá Folder hoặc xoá địa chỉ

2.5 - Một số tuỳ chọn của Internet:

* Thay đổi địa chỉ của trang khởi tạo:

- Vào menu Tool chọn Internet Option
- Chọn General
- Gõ địa chỉ cần đặt vào hộp Address hoặc có thể chọn một trong các nút trong mục đó là: Lấy địa chỉ của trang hiện tại, lấy địa chỉ mặc định, lấy trang trắng
- Chọn Apply
- Chọn OK

* Thay đổi số ngày lưu trong trang web:

- Vào menu Tool chọn Internet Option
- Gõ vào số ngày trong hộp Days to keep page in history
- OK

* Xoá các địa chỉ lưu trong hộp History:

- Xoá toàn bộ các trang:
  + Vào menu Tool chọn Internet Option
  + Chọn Clear History
  + OK
- Xoá từng trang:
  + Mở History
  + Nháy chuột phải vào trang cần xoá
  + Chọn Delete

* Thay đổi cỡ chữ:

- Vào menu View

- Chọn Text size: Largest: Cỡ chữ lớn nhất, Larger: Cữ chữ lớn, Medium: Cỡ chữ trung bình, Small: Cỡ chữ nhỏ, Smallest: Cỡ chữ nhỏ nhất

2.6 - Lưu trang web vào máy tính

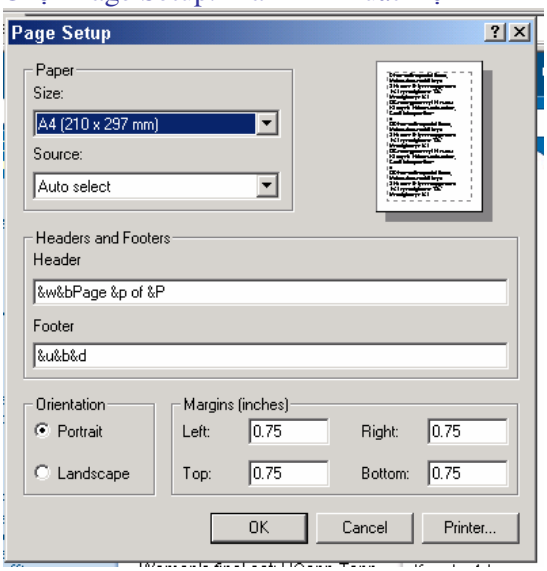
- Vào menu File
- Chọn Save as
- Chọn vị trí cần lưu
- Chọn Save

2.7 - Lưu ảnh

- Nháy chuột phải vào ảnh
- Chọn Save Picture As
- Chọn vị trí cần lưu
- Chọn Save

2.8 - In trang web

* Thiết kế một trang in
- Chọn File
- Chọn Page Setup. Màn hình xuất hiện



Hình 2.6

- Size : khổ giấy
- Source: khay giấy
- Header: Tiêu đề đầu trang
- Footer: Tiêu đề cuối trang
- Orientation : Hướng in, Portrait là theo chiều dọc, Landscap là theo chiều ngang.
- Margins : Lề trang in, Left lề trái, Right lề phải, Top là phía trên, Bottom là phía dưới.

* Xác định Header and footer

Internet Explore có thể in các tiêu đề đầu trang và tiêu đề chân trang theo các cách sau:

- Left header - Tiêu đề nằm ở đầu trang. Theo định sẵn vùng này chứa Title của tài liệu

- Center header - Tiêu đề nằm ở giữa và trên đầu trang. Theo định sẵn thì vùng này trắng.

- Right header - Tiêu đề nằm ở bên phải ở trên đầu trang. Theo định sẵn thì vùng này chứa số trang của tài liệu

- Left footer - Tiêu đề sẽ nằm về bên trái ở phía dưới của trang. Theo định sẵn thì vùng này chứa ngày tháng của tài liệu.

- Center footer - Tiêu đề sẽ nằm ở giữa và phía dưới của trang. Theo định sẵn thì vùng này trắng.

- Right footer - Tiêu đề sẽ nằm ở chính giữa ở phía dưới trang. Theo định sẵn thì vùng này chứa thời gian in tài liệu.

Bạn có thể sử dụng các mã đặc biệt để lấy thông tin hiện nay từ tài liệu (chẳng hạn như số trang) hay từ hệ thống (chẳng hạn như ngày, giờ). Các mã sau đây sẽ bắt đầu bằng dấu & theo sau là một ký tự. Chú ý rằng chữ hoa hay chữ thường rất quan trọng. Bằng cách sử dụng các mã này thì bạn có thể in các tiêu đề theo ý mình:

Nhập vào	Sẽ in ra
&w	Tựa đề của tài liệu
&u	Địa chỉ của trang
&d	Ngày hiện nay ở dạng tắt xác định trong Regional Setting/ Control Panel
&D	Ngày hiện nay ở dạng đầy đủ xác định trong Regional Setting/ Control Panel
&t	Giờ hiện nay ở dạng xác định trong Regional Setting/ Control Panel
&T	Giờ hiện nay theo dạng 24 giờ
&p	Số trang
&P	Tổng số trang
&&	Một dấu &
&b	Vị trí xác định

Dấu cuối cùng (&b) cho phép bạn xác định mục đã cho sẽ in ở header hay footer. Sau đây là cách thức làm việc của nó:

- Nếu bạn sử dụng mã này chỉ một lần thì bên trái của &b sẽ in ở bên trái còn phần bên phải sẽ in ở bên phải.

Chẳng hạn, &w&b&p sẽ in tựa đề của tài liệu ở bên trái còn số trang sẽ in ở bên phải.

- Nếu bạn sử dụng mã này hai lần thì phần bên trái của &b thứ nhất sẽ in ở bên trái, phần ở giữa hai &b sẽ in ở chính giữa và phần bên phải của &b sẽ in ở bên phải.

Chẳng hạn, &d&b&p&b&t sẽ in ngày hiện nay ở bên trái, trang chính giữa và in thời gian hiện nay ở bên phải.

- Bạn cũng có thể kết hợp văn bản và mã. Chẳng hạn, nếu bạn gõ Page %p of &p thì Internet Explore sẽ hiện trang hiện nay trên tổng số trang: Page 7 of 10.

* In tài liệu

- Chọn File, chọn Print hoặc ấn Ctrl + P.

3 - Thư điện tử (E- mail)

3.1 - Giới thiệu E-mail

E-mail - viết tắt của Electronic Mail (thư điện tử) là một dịch vụ được triển khai trên các mạng máy tính cho phép người sử dụng có thể gửi thư cho nhau.

ở đây "thư" là một tệp dạng text, hành động chuyển thư được thực hiện theo phương thức truyền gửi các tệp này trên các đường truyền dẫn của mạng. Trong nhiều trường hợp thư còn có từ đồng nghĩa là thông điệp (message), đúng ra thông điệp cần được hiểu là một thư ngắn không vượt quá một câu.

E- mail là dịch vụ cơ bản nhất và phổ thông nhất trên mạng, là nền tảng để xây dựng một văn phòng không giấy (paperless office).

3.2 - Các mô hình hoạt động của dịch vụ E-mail:

- Mô hình thông điệp trực tiếp: Theo mô hình này, các thông điệp được gửi trực tiếp, ngay lập tức tới các máy đang hoạt động trên mạng LAN.
- Mô hình hộp thư lưu: Là mô hình khách – chủ của dịch vụ thư điện tử, theo đó chương trình thực hiện dịch vụ này được viết thành hai phần chương trình chạy độc lập tương đối:

+ Chương trình phía Client (khách) chạy trên các máy trạm làm việc của mạng và phần phía Server (chủ) chạy trên máy chủ của mạng. Người dùng soạn thảo thư, ấn định địa chỉ nhận và "ra lệnh" gửi thông qua chương trình client về máy chủ của mạng.

+ Chương trình Server đón nhận các thư gửi từ Client, lưu trữ vào các hộp thư lưu.

+ Mỗi người dùng cũng nhận và đọc thư của mình bằng cách dùng chương trình phía client duyệt danh sách các thư gửi cho mình và đọc thư quan tâm.

+ Chương trình Server cũng có thể phát tín hiệu báo có thư cho những người đang làm việc trên mạng.

- Mô hình Internet Mail: Là mô hình liên kết các hộp thư lưu trên Internet, mỗi hộp thư lưu cho một vùng. Nếu tất cả các liên kết đang được thực hiện, thư sẽ được chuyển đến hộp thư lưu đích trong vài giây, trường hợp ngược lại nó sẽ được chuyển bước qua từng hộp thư lưu và các đường liên kết đang thực hiện để đến đích cuối cùng.

3.3 - Tạo hòm thư miễn phí với Yahoo

Có rất nhiều địa chỉ trang web ta có thể tạo được một địa chỉ thư miễn phí như Yahoo.com, hotmail.com, vietfun.com, …

*** Cách tạo hòm thư miễn phí với Yahoo:**

- Gõ địa chỉ www.yahoo.com (hoặc gõ địa chỉ mail.yahoo.com)
- Chọn Email hoặc Chekmail
- Chọn Sign Up Now

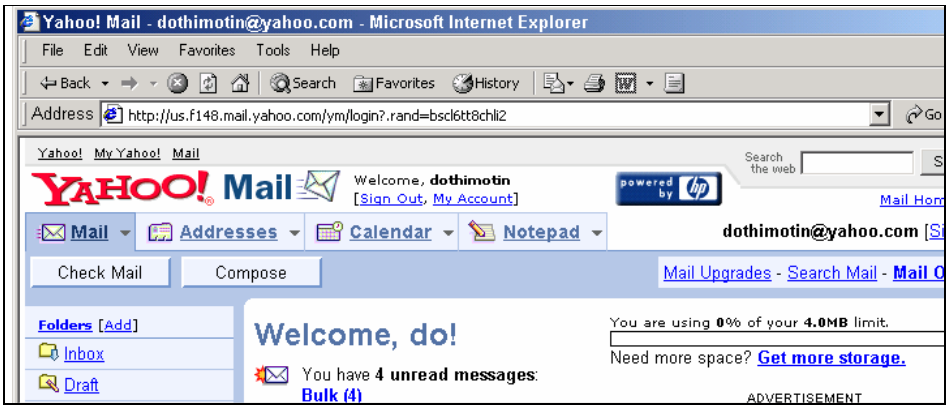
Kết quả xuất hiên cửa số ta phải khai báo các mục sau:

+YahooID:	Gõ tên hòm thư cần tạo (Không dấu, không cách)
+ Password:	Gõ mật khẩu (ít nhất 6 ký tự, phải tắt chế độ gõ tiếng việt)
- Retype password:	Gõ lại mật khẩu
- Sercurity Question:	Chọn câu hỏi tuỳ chọn (chọn bất kỳ)
- Your answer:	Gõ vào câu trả lời (ít nhất có 4 ký tự)
- Birthday:	Gõ ngày tháng năm sinh
- Gender:	Chọn giới tính
- Fist/Last name:	Gõ họ và tên đệm
- Langgue:	Chọn ngôn ngữ
- Zip Code:	Gõ mã nước
- Industry:	Chọn nghề nghiệp (chọn bất kỳ)
- Title:	Chọn chức vụ (chọn bất kỳ)
- Enter the word:	Gõ từ hiển thị ở hộp dưới
Chọn Submit this form	Đệ trình Form

3.4 - Gửi và nhận thư với Yahoo

- Gõ địa chỉ www.yahoo.com (hoặc gõ địa chỉ mail.yahoo.com)
- Chọn Email hoặc Mail
- Gõ địa chỉ thư vào hộp Yahoo ID
- Gõ mật khẩu vào Password
- Chọn Sign In

Kết quả xuất hiện cửa sổ:



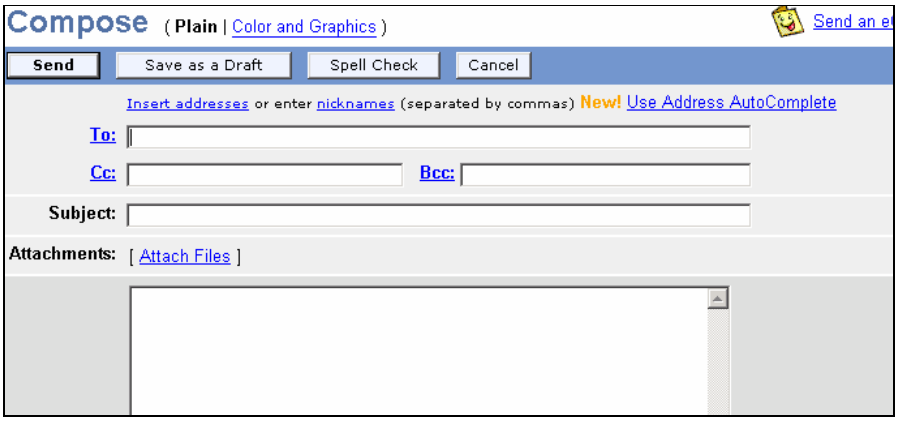
Hình 3.6

* Chọn Chek mail: Để nhận thư. Kết qủa có cửa sổ sau:

Hình 4.6

- Sender : Người gửi
- Subject : Tiêu đề thư
- Date : Ngày gửi
- Size : Kích cỡ file
- Muốn đọc thư nào thì nháy chuột vào thư đó.
- Muốn xoá thư thì chọn vào ô vuông bên trái thư, chọn nút Delete

*** Chọn Compose: Để gửi thư.** Kết qủa có cửa sổ sau:



Hình 5.6

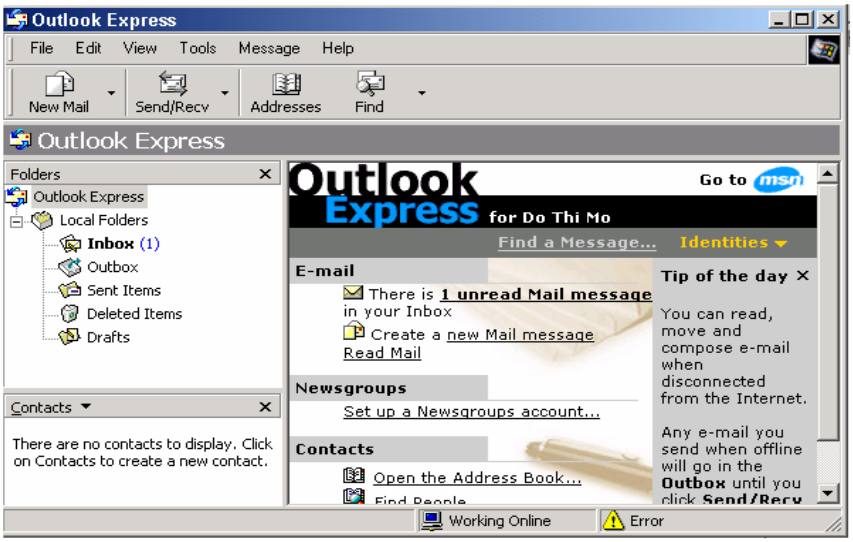
- To : Địa chỉ người nhận
- Cc : Địa chỉ đồng gửi tới
- Bcc : Địa chỉ đồng gửi tới
- Subject : Tiêu đề thư
- Attachmets [Attech Files] : Gửi kèm file
- Phần cửa sổ trắng phía chứa nội dung thư ta gõ vào.
- Chọn nút **Send** để gửi thư đi

3.5 - Gửi nhận thư bằng Outlook Express

* **Khởi động Outlook Express**

- Kích chuột vào Start
- Chọn Program
- Chọn Outlook Express

Kết qủa có cửa sổ sau:



Hình 6.6

- Inbox : Nơi lưu các thư được tải xuống và nó sẽ ở đó đến khi bạn chuyển nó đi nơi khác.
- Outbox : Nơi lưu tạm thời các thư chưa được chuyển
- Delete Items : Lưu các thư được xoá ở Inbox (Tương tự như Recycle Bin, nếu muốn xoá hẳn các thư nằm trong thư mục này thì bạn phải: Nháy chuột phải trên biểu tượng của thư mục và chọn Empty Forder)
- Sent Items : Lưu các bản sao của các thư đã gửi đi
- Drafts : Nếu bạn soạn một thư và cần phải thực hiện một số việc khác trước khi hoàn tất bức thư thì chỉ cần đóng bức thư và lưu trữ nó lại. Outlook Express sẽ lưu bức thư này trong thư mục Drafts (bản nháp).

* **Khai báo Accounts**

- Vào menu Tool/ Chọn Accounts/ Chọn Mail/ Chọn Add/ Chọn Mail
- Gõ tên vào hộp Display name
- Chọn Next
- Khai báo máy chủ nhận thư vào hộp Incoming mail :

+ Thuê bao VNN: mail.hn.vnn.vn

+ Thuê bao FPT : imail.fpt.vn
+ Thẻ VNN : fmail.hn.vnn
+ Thẻ FPT : pop3.cardvn.net

- Khai báo máy chủ gửi thư vào hộp Outgoing mail :
+ Thuê bao VNN: mail.hn.vnn.vn
+ Thuê bao FPT : omail.fpt.vn
+ Thẻ VNN : fmail.hn.vnn
+ Thẻ FPT : smtp.cardvn.net
- Chọn Next
- Gõ tên của Account vào hộp Account name
- Gõ mật khẩu vào Password
- Chọn Next
- Chọn Finish
- Chọn Close

*** Gửi thư:**

Bạn có thể gửi thư theo ba cách:
- Soạn thư mới
- Hồi đáp thư cũ
- Chuyển một bức thư đã nhận

*** Soạn một thư mới:**

- Chọn New Mail (hoặc chọn menu Message/ New Message). Cửa sổ New Message xuất hiện. Các thao tác tương tự như gửi thư tại yahoo.

*** Trả lời một bức thư** (Hồi đáp thư cũ):
- Chọn Inbox
- Chọn thư đọc
- Có hai cách trả lời như sau:

+ Reply : Nếu bạn chọn lựa chọn này thì thư hồi đáp của bạn chỉ tới có một địa chỉ e – mail trong dòng From của phần tiêu đề. Không có bản sao nào được gửi tới một người nào trong phần danh sách được xác định ở mục Cc: Của bức thư ban đầu

+ Reply All : Nếu bạn chọn lựa chọn này thì bức thư của bạn tới không chỉ tác giả mà còn tới tất cả các mọi người trong danh sách Cc.

*** Gửi chuyển tiếp thư đã nhận:**
- Chọn thư cần chuyển
- Chọn Forward Message {Hoặc chọn menu Message (hoặc Compose), chọn Forward hay ấn Ctrl + F}
- Gõ vào địa chỉ cần gửi đi
- Chọn Send

Hoặc có thể chuyển thư dưới hình đính kèm

*** Nhận thư đến:**

Chọn Inbox/ Chọn Send and Receive/ Nhập mật khẩu (Password) vào , rồi ấn OK

*** Đọc thư đến:**

Chọn Inbox: Nếu có thư đến, sẽ thấy hiển thị tên người gửi. Bạn chỉ việc nhấn chuột vào tên thư muốn đọc nội dung của nó ở phía dưới.

*** Tải thư về (Download) máy tính:**
- Chọn thư muốn tải về
- Chọn File/ Save As
- Chọn kiểu file: Mail (.eml), Text (.txt), web (.html).

- Chọn vị trí muốn lưu file đó
- Đặt tên file rồi bấm OK

3.6. Download và một số địa chỉ Internet

*** Download :**

- Là thực hiện tải một phần mềm hoặc một chương trình trên mạng máy tính về máy tính của mình.
- Có thể Download từ các trang cụ thể (nếu có mục Download) hoặc từ các địa chỉ sau:

+ Microsoft Free Product Downloads (): Đối với những người sử dụng Win dows thì đây là nơi bạn có thể tìm thấy các fonts, trò chơi, các chương trình hỗ trợ cho Internet Explore và Microsoft Office….

+ Download.com và Shareware.com (www.shareware.com): Đây là những nơi cho phép bạn tìm kiếm trên một cơ sở dữ liệu của hơn 190.000 chương trình miễn phí và chia xẻ.

*** Một số địa chỉ Internet:**

www.hau1.edu.vn	Trang chủ của Trường ĐH Nông nghiệp I
www.express.net	Trang web của Việt nam Thông tấn xã
www.ptcenter.net	Trang web của trung tâm tin học PT
www.home.vnn.vn	Trang web của VDC
www.msn.com	Trang web của hãng Microsoft
www.fpt.vn	Trang của FPT
www.vnn.vn	Trang của VASC
www.quochoi.gov.vn	Trang của Quốc hội
www.vtv.org.vn	Trang của Truyền hình Việt Nam
www.nxbgd.com.vn	Trang của Nhà xuất bản giáo dục
www.ykhoa.com.vn	Trang Y khoa bằng Tiếng Việt
www.laodong.com.vn	Trang báo Lao động
www.nhandan.org.vn	Trang báo Nhân dân
www.vietnamtourism.com.vn	Trang của Du lịch Việt
www.hut.edu.vn	Trang chủ của Trường ĐH Bách Khoa HN
www.dec.edu.vn	Trang web của Trường ĐH Cần Thơ
www.fao.org	Trang web của tổ chức FAO
www.manguon.com	Trang web của công ty Mắt bão về CNTT
www.real-english.com	Trang web học tiếng Anh trực tuyến
www.caycanhvietnam.com	Trang web giới thiệu về cây cảnh Việt nam
www.bkav.com.vn	Trang web của Trung tâm Phần mềm và Giải pháp An ninh mạng ĐH Bách khoa HN
www.hoahoctro.com	Trang báo Hoa học trò
www.loveclub.saigonnet.vn	Câu lạc bộ tình yêu, hôn nhân và gia đình
www.yahoo.com	Trang chủ của Yahoo
www.google.com	Trang để tìm kiếm
www.download.com	Trang chủ để tải các phần mềm
www.cnn.com	Trang chủ của hãng CNN
www.cocacola.com	Trang chủ của hãng Cocacola
www.healthfinder.gov	Trang cung cấp các thông tin về y tế
www.asd.com	Trang thông tin về các trường học ở Mỹ
www.amazon.com	Cửa hàng sách lớn nhất thế giới.

Câu hỏi ôn tập chương V

1. Mạng Internet là gì? Các chức năng chính của mạng Internet.

2. Để một máy tính các nhân kết nối được vào mạng Internet cần có những gì ?

3. Trình duyệt Internet Explorer có những chức năng gì?

4. E-mail là gì? Các mô hình hoạt động của dịch vụ E-mail.

5. Cách tạo hòm thư miễn phí với Yahoo và với Outlook Express.

CHƯƠNG VI: GIẢI THUẬT

1. Khái niệm giải thuật (Algorithms)

- Khi cần giải quyết một bài toán trong thực tế với sự trợ giúp của máy tính điện tử ta thường phải biết dữ liệu vào của bài toán (Input) là gì? và bài toán yêu cầu dữ liệu ra (Output) là gì?. Bước tiếp theo ta 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 giải thuật.

- Giải thuật của 1 bài toán là một dãy các câu lệnh (Statements) chặt chẽ và rõ ràng xác định một trình tự các thao tác trên một số đối tượng nào đó sao cho sau một số bước hữu hạn thực hiện ta thu được kết quả mong muốn.

- Với định nghĩa như vậy ta thấy rằng đối với một bài toán cụ thể có thể có nhiều giải thuật khác nhau nhưng tất nhiên là các giải thuật đó phải cho cùng một kết quả theo đúng yêu cầu của bài toán.

- Khi nghiên cứu về giải thuật thường ta phải biết được giải thuật đó tác động lên dữ liệu nào. Việc lựa chọn cấu trúc dữ liệu (Data structures) phù hợp và việc thiết lập được các giải thuật đúng đắn có cấu trúc tốt và hiệu quả là những vấn đề mấu chốt của công việc thiết lập phần mềm. Chính vì vậy mà Niklaus Wirth người sáng lập ra ngôn ngữ lập trình Pascal đã tổng kết: Giải thuật+Cấu trúc dữ liệu= Chương trình

Ví dụ: Xây dựng giải thuật tìm UCLN của 2 số nguyên dương a và b, ký hiệu (a,b)

+ Đối với bài toán này ta có

Input: 2 số nguyên dương a, b

Output: (a,b)

+ Giải thuật được xây dựng dựa trên tính chất

Nếu a=b thì (a,b)=a

Nếu a>b thì (a,b)=(a-b,b)

Nếu a<b thì (a,b)=(a,b-a)

+ Cụ thể giải thuật của bài toán như sau:

Bước 1: So sánh a và b, Nếu a=b thì dừng giải thuật và thông báo (a,b)=a. Nếu a≠b thì chuyển sang bước 2

Bước 2: Nếu a>b thì thay thế a bởi a-b, nếu a<b thì thay thế b bởi b-a. Quay lại thực hiện bước 1

Minh hoạ: a=20, b=32

Bước thực hiện	a	b	Kiểm tra điều kiện a=b
Bước 1	20	32	Sai
Bước 2	20	12	
Bước 1	20	12	Sai
Bước 2	8	12	
Bước 1	8	12	Sai
Bước 2	8	4	
Bước 1	8	4	Sai
Bước 2	4	4	
Bước 1	4	4	Đúng

Kết quả là: (20,32)=4

2. Các yêu cầu với giải thuật

Giải thuật của bài toán phải thoả mãn 3 yêu cầu sau:

***Yêu cầu 1*:** Tính dừng

Giải thuật phải dừng sau một số hữu hạn các thao tác, đây là yêu cầu hết sức quan trọng với một giải thuật

Yêu cầu 2: Tính đúng đắn

Ta phải đặt câu hỏi "Liệu giải thuật có thể hiện đúng lời giải của bài toán không?" . Thông thường chúng ta cài đặt giải thuật dưới dạng chương trình và cho thực hiện trên máy tính với một số bộ dữ liệu nào đó, sau đó so sánh với những kết quả mà ta đã biết. Nhưng cách thử này chỉ khẳng định được tính sai chứ chưa thể khẳng định được tính đúng đắn của giải thuật. Bằng cách sử dụng các công cụ toán học ta có thể khẳng định được tính đúng đắn của 1 giải thuật nhưng thường thì đây là một công việc phức tạp.

Yêu cầu 3: Tính đơn giản và hiệu quả

Ta thường mong muốn xây dựng được một giải thuật đơn giản, dễ hiểu, dễ lập trình. Nhưng đôi khi thuật giải đơn giản lại gây ra sự lãng phí thời gian và bộ nhớ. Do đó mục tiêu là phải xây dựng được các giải thuật có thời gian thực hiện nhanh, hạn chế tối đa dung lượng bộ nhớ dành cho việc lưu trữ những kết quả trung gian.

3. Các cách diễn tả giải thuật

3.1. Cách 1: Liệt kê từng bước

Ví dụ: Có 31 que diêm, người và máy thay nhau bốc. Mỗi lần bốc từ 1 đến 4 que. Ai phải bốc sau cùng là thua. Hãy xây dựng thuật giải sao cho máy bốc trước bao giờ cũng thua.

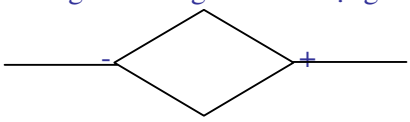
Bước 1: Máy bốc ngẫu nhiên x que diêm (1≤x≤4)

Bước 2: Người bốc (5- x) que, tổng số que diêm giảm đi 5 que. Nếu số que diêm còn lại là 1 que thì chuyển sang bước 3, nếu không thì quay lại thực hiện bước 1

Bước 3: Tuyên bố người thắng cuộc

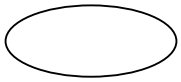
3.2. Cách 2: Sử dụng lưu đồ

Sử dụng phương tiện hình học cũng là một cách tốt để minh hoạ giải thuật của 1 bài toán. Trong lưu đồ người ta sử dụng các hình sau, với kí hiệu + là đúng, - là sai.

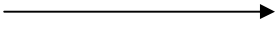


Khối điều kiện

Khối lệnh



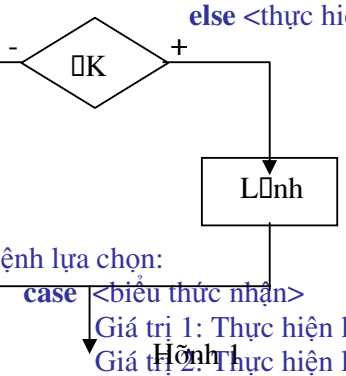
Khối bắt đầu và kết thúc



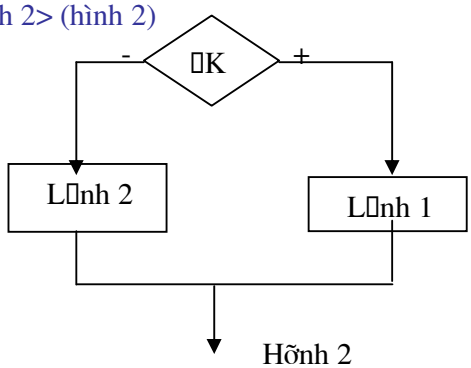
Cung

a. Cấu trúc rẽ nhánh

- Rẽ nhánh dạng khuyết: **if** <đk thoả mãn> **then** <thực hiện lệnh> (hình 1)
- Rẽ nhánh dạng đủ: **if** <đk thoả mãn> **then** <thực hiện lệnh 1>
  else <thực hiện lệnh 2> (hình 2)



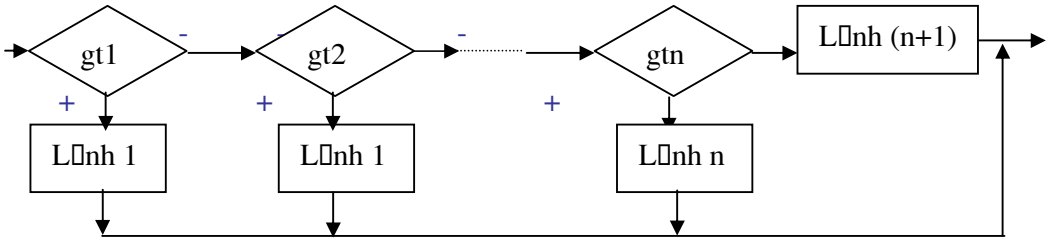
Hình 1



Hình 2

- Cấu lệnh lựa chọn:

case <biểu thức nhận>
Giá trị 1: Thực hiện lệnh 1
Giá trị 2: Thực hiện lệnh 2
.....
Giá trị n: Thực hiện lệnh n
else Thực hiện lệnh (n+1)
end



Hình 3

b. Cấu thúc lặp

+ Lặp một số lần định trước:

Dạng 1: **for** i:=m **to** n **do** <lệnh>

Thực hiện lệnh với i nhận các giá trị nguyên tăng từ m tới n với bước nhảy bằng 1

Dạng 2: **for** i:=n **downto** m **do** <lệnh>

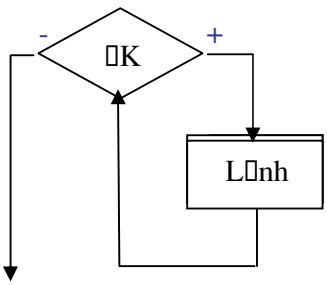
Tương tự như dạng 1 nhưng bước nhảy giảm bằng 1

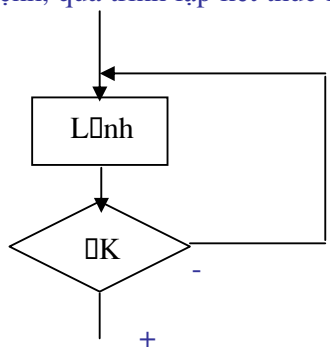
+ Lặp với điều kiện trước: **while** <điều kiện **> do** < lệnh>

Khi điều kiện còn đúng thì còn thực hiện lệnh, quá trình lặp kết thúc khi điều kiện sai (hình 1)

+ Lặp với điều kiện sau: **repeat** <lệnh> **until** <điều kiện>

Khi điều kiện còn sai thì còn thực hiện lệnh, quá trình lặp kết thúc khi điều kiện đúng (hình 2)





3.3 Cách 3: Sử dụng giả ngôn ngữ có cấu trúc tựa ngôn ngữ lập trình bậc cao

Là phương pháp diễn tả giải thuật dựa vào các cấu trúc điều khiển, cùng với các từ khoá của một ngôn ngữ lập trình bậc cao nào đó. Trong giáo trình này ta sẽ sử dụng ngôn ngữ tựa Pascal để diễn tả giải thuật. Cách diễn đạt này đã tiếp cận gần hơn với ngôn ngữ lập trình.

Ví dụ: Với thuật giải tìm UCLN ở trên ta có thể diễn đạt như sau

```
while a≠b
        begin
                if a>b then  thay a bởi a-b
                        else thay b bởi b-a
        end
write ước chung lớn nhất là a
```

4. Thiết kế giải thuật

4.1. Mô-đun hoá và việc giải quyết bài toán

Những bài toán ta gặp trong thực tế thường là phức tạp, trong trường hợp đó người ta thường chia bài toán thành những bài toán nhỏ và dễ giải quyết hơn. Nghĩa là coi bài toán ban đầu là Mô- đun chính, ta chia nó thành các Mô- đun con, và mỗi Mô- đun con này có thể lại được chia thành các Mô- đun nhỏ hơn... Cách giải quyết bài toán như vậy người ta thường gọi là chiến thuật "Chia để trị"(divide and conquer).

Trong khi lập trình việc chia chương trình chính thành các chương trình con thể hiện tính có cấu trúc của ngôn ngữ lập trình về mặt chương trình

4.2. Tinh chỉnh từng bước giải thuật

Tinh chỉnh từng bước là phương pháp thiết kế giải thuật gắn liền với lập trình. Bước đầu giải thuật được minh hoạ bằng ngôn ngữ tự nhiên, càng ở các bước sau ngôn ngữ tự nhiên được thay thế bởi ngôn ngữ tự nhiên pha lẫn ngôn ngữ lập trình mà ta gọi là giả ngôn ngữ.

Ta có sơ đồ sau: Ngôn ngữ tự nhiên→ Giả ngôn ngữ→ Ngôn ngữ lập trình

4.3. Phân tích thuật giải

Phân tích giải thuật phải căn cứ vào 3 tiêu chuẩn đối với một giải thuật: Tính dừng, tính đúng đắn, tính đơn giản và hiệu quả.

Việc kiểm tra giải thuật là một phần hết sức quan trọng, lý tưởng là khi có thể khẳng định một cách hình thức tính đúng đắn của giải thuật. Tuy nhiên trong thực tế thời gian và công sức để viết ra một cách cẩn thận và đầy đủ tất cả những chi tiết chứng minh tính đúng đắn của một giải thuật phức tạp thường là không cho phép. Người lập trình thường áp dụng các biện pháp sau:

- Chứng minh một cách suy diễn rằng những bước trong giải thuật là đúng đắn. Nghĩa là giải thuật bắt đầu bằng một khẳng định (giả thiết) về dữ liệu vào và dùng phương pháp suy luận lôgic để chỉ ra rằng việc thực hiện giải thuật sẽ cho một khẳng định (kết luận) về đầu ra.
- Thể hiện giải thuật bằng một ngôn ngữ lập trình và thử thực hiện chương trình với các bộ dữ liệu vào mà kết quả ta đã biết trước. Thường thì các lỗi về cú pháp và lỗi lúc thực hiện chương trình thường dễ tìm và sửa chữa còn các lỗi lôgic thường khó phát hiện hơn nhiều.

- Có đôi lúc viêc kiểm tra chương trình phải thực hiện thủ công, kiểm tra từng bước, từng thủ tục của chương trình chính. Kỹ thuật này được gọi là "đi bộ qua chương trình"(Walking through the program)

- Quan tâm đặc biệt tới thời gian thực hiện chương trình, thời gian thực hiện phụ thuộc rất nhiều vào việc tổ chức dữ liệu đưa vào (kích thước dữ liệu).

5. Giải thuật sắp xếp (Sorting)

Sắp xếp là một trong số yêu cầu thường xuyên xuất hiện trong quá trình xử lý số liệu. Bản chất của thuật giải sắp xếp là bố trí lại vị trí của số liệu theo thứ tự tăng dần hoặc giảm dần. Có nhiều giải thuật sắp xếp trong tin học, trong giáo trình này chúng ta sẽ đề cập đến một số thuật giải đơn giản đó là sắp xếp lựa chọn (selection sort), sắp xếp chèn (insertion sort) và sắp xếp nổi bọt (bubble sort). Để đơn giản giả sử yêu cầu của bài toán là: Sắp xếp một dãy số cho trước a_1, a_2, ..., a_n theo thứ tự tăng dần.

5.1 Sắp xếp lựa chọn (selection sort)

Thuật giải chọn được diễn tả như sau: Tìm phần tử nhỏ nhất trong dãy số và hoán vị nó với phần tử đầu tiên, tìm phần tử nhỏ nhất kế tiếp và hoán vị nó với phần tử thứ hai. Tiếp tục quá trình này đến khi toàn bộ dãy số được sắp xếp.

```
procedure Selection_Sort;
begin
  for i:=1 to n-1 do
  begin
        m:=i
        for j:=i+1 to  n do if aj<am then m:=j
        if m≠i then đổi chỗ ai và am
  end
end
```

Ví dụ

Dãy số ban đầu :	3	6	-2	7	5
i=1	-2	6	3	7	5
i=2	-2	3	6	7	5
i=3	-2	3	5	7	6
i=4	-2	3	5	6	7

5.2 Sắp xếp chèn (insertion sort)

Thuật giải chèn được diễn tả như sau: Xét lần lượt từng phần tử và chèn vào vị trí thích hợp của phần tử đó trong số các phần tử đã xét trước đó. Cụ thể giả sử đã có (i-1) phần tử được sắp xếp đúng vị trí, để chèn phần tử thư i vào đúng vị trí ta so sánh lần lượt với các phần tử thứ (i-1), (i-2),... khi tìm được vị trí đúng thì chèn phần tử thứ i đó vào.

```
procedure Insertion_Sort
begin
  for i:=2 to n do
  begin
        k:=a_i
        j:=i
        while a_{j-1}>k do begin a_j:=a_{j-1}, j:=j-1 end
        a_j:=k
  end
end
```

Ví dụ

Dãy số ban đầu :	3	6	-2	7	5
i=2	3	6	-2	7	5
i=3	-2	3	6	7	5
i=4	-2	3	6	7	5
i=5	-2	3	5	6	7

5.3 Sắp xếp nổi bọt (bubble sort)

Thuật giải này còn có tên gọi khác là sắp xếp bằng cách đổi chỗ trực tiếp (exchange sort), thuật giải nổi bọt được diễn tả như sau: Duyệt dãy số theo thứ tự từ phải sang trái nếu hai phần tử kề cận ngược thứ tự thì đổi chỗ cho nhau. Như vậy sau lượt duyệt đầu tiên phần tử đầu tiên sẽ là phần tử nhỏ nhất, sau lượt thứ hai phần tử nhỏ thứ hai được chuyển lên vị trí thứ hai...cứ như vậy dãy số sẽ được sắp xếp tăng dần.

```
procedure Bubble_Sort
begin
  for i:=1 to n-1 do
        for j:=n downto i+1 do
              if a_j<a_{j-1} then đổi chỗ a_j và a_{j-1}
end
```

Ví dụ

Dãy số ban đầu :	3	6	-2	7	5
i=1	-2	3	6	5	7
i=2	-2	3	5	6	7
i=3	-2	3	5	6	7
i=4	-2	3	5	6	7

6. Giải thuật tìm kiếm (Searching)

Cùng với các thuật giải sắp xếp, các thuật giải tìm kiếm cũng đóng một vai trò quan trọng trong khi xủ lí số liệu. Bài toán tìm kiếm đặt ra như sau: Giả sủ ta có một dãy số a1, a2,...., an, ta phải tìm vị trí của phần tử có giá trị bằng giá trị X cho trước. Chúng ta sẽ xét hai thuật giải tìm kiếm đó là tìm kiếm tuần tự (sequential searching) và tìm kiếm nhị phân (binary searching).

6.1 Tìm kiếm tuần tự (sequential searching)

Đây là thuật giải tìm kiếm đơn giản nhất, ta sẽ duyệt tuần tự dãy số, thuật giải sẽ kết thúc khi tìm thấy phần tử bằng giá trị X hoặc khi duyệt hết dãy số nhưng không có phần tử nào có giá trị là X

```
procedure Sequential_Searching
begin
i:=1
  while a_i≠ X do i:=i+1
  if i=n+1 then không có phần tử cần tìm
           else vị trí phần tử cần tìm là i
end
```

6.2 Tìm kiếm nhị phân (binary searching)

Giả sử dãy số đã được sắp xếp tăng dần $a_1 \le a_2 \le ..... \le a_n$ (trường hợp sắp xếp giảm dần thì tương tự), thuật giải nhị phân gần giống như khi ta tìm một từ trong từ điển. Để tìm phần tử bằng X trước tiên ta so sánh nó với phần tử ở vị trí giữa của dãy số nếu X nhỏ hơn thì X chỉ có thể ở trong một nửa trước của dãy nếu ngược lại thì X chỉ có thể ở trong nủa sau của dãy. Lặp lại quá trình tìm kiếm đó đến khi tìm thấy hoặc dãy số trở nên rỗng (không tìm thấy).

```
procedure Binary_Searching
begin
  left:=1
  right:=n
  repeat
          mid:=[(left+right)/2]   (*Kí hiệu [a] nghĩa là lấy phần nguyên của số thực a*)
          if X<a_mid then right:=mid-1
                  else left:=mid+1
  until (X=a_mid) or(left>right)
  if X=a_mid then vị trí cần tìm là mid
          else không có phần tử cần tìm
end
```

Ví dụ: Tìm phần tử 28 trong dãy số sau

	[4	15	28	33	67	99	103]
Lặp lần 1	[4	15	28]				
Lặp lần 2			[28]				

7. Giải thuật đệ quy

7.1. Khái niệm đệ qui

Một đối tượng được gọi là đệ qui nếu nó bao gồm một phần của chính nó hay được định nghĩa bởi chính nó.

Trong khi thiết kế giải thuật ta thường thiết kế dưới dạng các mô- đun. Khi giải thuật được cài đặt thành chương trình thí các mô- đun sẽ tương ứng với các chương trình con (hàm- function và thủ tục- procedure),

Chương trình con được gọi là đệ qui nếu trong thân của nó có lời gọi trực tiếp hoặc gián tiếp đến chính bản thân nó.

Đệ qui có ý nghĩa đặc biệt quan trọng trong các định nghĩa toán học

Ví dụ 1: Định nghĩa số tự nhiên

+ 0 là số tự nhiên

+ Số tiếp theo của một số tự nhiên là một số tự nhiên

Ví dụ 2: Định nghĩa n!

+ 0!=1

+ n!=n*(n-1)! nếu n>0

- Định nghĩa một phép đệ qui gồm có 2 phần

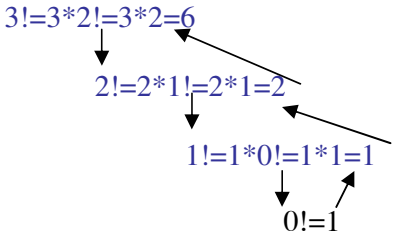
+ Trường hợp suy biến: Giúp cho quá trình đệ qui kết thúc

+ Phần đệ qui (hay phần qui nạp): Trong đó tác động cần được thực hiện cho giá trị hiện thời của các tham số được định nghĩa bằng các tác động hay giá trị được định nghĩa trước đây

Trong ví dụ định nghĩa n! thì trường hợp suy biến định nghĩa 0!, phần qui nạp định nghĩa n! qua các giá trị của n và giá trị của (n-1)!

Dễ nhận xét, nếu (n-1)! đã tính được thì n! sẽ dễ dàng tính được. Với cách suy diễn tương tự, (n-1)! sẽ tính được nếu như (n-2)! đã tính được... cuối cùng 1! sẽ tính được nếu 0! đã tính được. Ta thấy rằng 0! đã cho trong định nghĩa. Do vậy đi ngược từ cuối, vì 0! đã tính được nên 1! cũng tính được,....,sau khi (n-1)! đã có ta sẽ nhận được n!

Minh hoạ: Tính 3!



Giải thuật được viết dưới dạng thủ tục hàm (tựa Pascal) như sau:

```
function giaithua(n)
begin
        if n=0 then giaithua:=1 (* trường hợp suy biến*)
                else giaithua:=n*giaithua(n-1) (* phần đệ qui*)
end
```

Chú ý: Không phải lúc nào tính đệ qui trong cách giải bài toán cũng thể hiện rõ nét và dễ phát hiện như ví dụ trên. Do đó muốn biết giải thuật của một bài toán có thể thiết kế dưới dạng giải thuật đệ qui được hay không? Có thể thấy câu trả lời qua việc trả lời các câu hỏi sau :

+ Có thể định nghĩa được bài toán dưới dạng một bài toán cùng loại nhưng "nhỏ" hơn không?

+ Kích thước của bài toán sẽ giảm đi ở mỗi bước gọi đệ qui như thế nào?

+ Trường hợp nào của bài toán được coi là trường hợp suy biến?

7. 2. Ví dụ về giải thuật đệ qui : Bài toán tháp Hà Nội

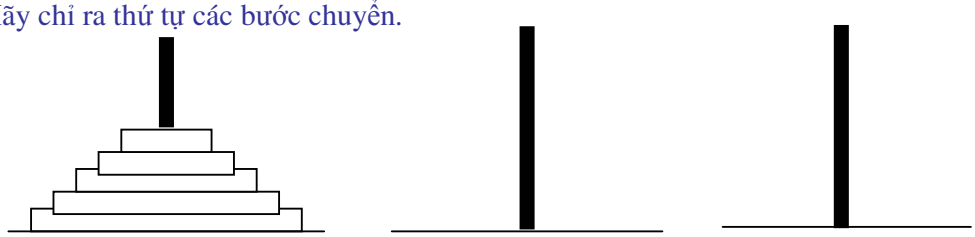
Bài toán Tháp Hà Nội là một ví dụ cổ điển cho thấy thuật toán đệ qui là đặc biệt thích hợp. Có thể giải quyết bài toán một cách dễ dàng nếu dùng đệ qui, nhưng cách giải không đệ qui là tương đối khó.

Nội dung bài toán: Tại cọc A có n chiếc đĩa, đĩa to ở dưới, đĩa nhỏ ở trên. Chuyển các đĩa từ cọc A sang cọc C có thể nhờ cọc B làm vị trí trung chuyển theo các quy tắc sau:

- Mỗi lần chỉ được chuyển một đĩa và phải là đĩa ở trên cùng
- Đĩa lớn không bao giờ được phép nằm trên đĩa nhỏ

- Khi chuyển một đĩa, nó phải được đặt vào một trong 3 cọc ở trên

Hãy chỉ ra thứ tự các bước chuyển.



Một truyền thuyết cho rằng các thầy tu ở Điện Bramah được cho một bài toán đố với một nền vàng có 3 kim vàng trên 1 cọc có 64 đĩa vàng. Khi họ chuyển các đĩa vàng theo các luật của bài toán trên, nếu mỗi giây chuyển được một đĩa và họ bắt đầu công việc từ năm 0, đến khi hoàn thành công việc thì sẽ là ngày tận thế.

Những người mới bắt đầu có thể giải bài toán một cách dễ dàng với số các đĩa là bé, nhưng họ sẽ rất khó khăn khi số đĩa tăng lên 7,8 và lớn hơn. Tuy nhiên với một nhà lập trình thì có thể giải bài toán một cách không mấy khó khăn.

Cách giải: Nếu có một đĩa, chuyển nó từ cọc A sang cọc C. Bài toán giải được với n=1

Giả sử rằng bài toán có nghiệm với n-1 đĩa , nghiệm với n đĩa có thể nhận được một cách dễ dàng nhờ dùng phép đệ quy:

+ Chuyển n-1 đĩa trên cùng ở cọc A sang cọc B, dùng cọc C làm trung chuyển
+ Chuyển đĩa còn lại ở cọc A sang cọc C
+ Chuyển n-1 từ cọc B sang cọc C, dùng cọc A làm trung chuyển

Ta có thể viết giải thuật của bài toán Tháp Hà Nội như sau:

```
procedure Move(n,A,B,C)
(* Chuyển n đĩa từ cọc A sang cọc C, dùng cọc B làm trung chuyển*)

begin
        if n=1 then chuyển đĩa từ A sang C
                else
                 begin
                 Move(n-1,A,C,B)
                 (* Chuyển n-1 đĩa từ A sang B, dùng C làm trung chuyển*)
                 Move(1,A,'  ',C)
                 (* Chuyển 1 đĩa từ cọc A sang cọc C*)
                 Move(n-1,B,A,C)
                 (* Chuyển n-1 đĩa từ B sang C, dùng A làm trung chuyển*)
                 end
  end
```

Bài tập chương VI: Thuật giải

Viết giải thuật cho các bài toán sau:

1. Tính n giai thừa: n! =1.2...n với n>1
2. Tính các tổng: S=1/2 + 1/4 +...+ 1/(2k)
   Q=1.1!+2.2!+...+n.n!
3. Tìm và in ra tất cả các số chính phương nhỏ hơn một số cho trước, cho biết có bao nhiêu số chính phương như vậy.
4. Viết chương trình giải bài toán cổ: " Vừa gà vừa chó, bó lại cho tròn, ba mươi sáu con, một trăm chân chẵn. Hỏi có bao nhiêu gà, bao nhiêu chó?"
5. Viết chương trình tìm ước số chung lớn nhất của 2 số nguyên dương cho trước.
6. Tính $Ex=1+\frac{x}{1!}+\frac{x^2}{2!}+\ldots+\frac{x^n}{n!}+\ldots$ với độ chính xác $\varepsilon=10^{-4}$ ($ABS(x^n/n!) < \varepsilon$), giá trị x được nhập vào từ bàn phím khi chạy chương trình.

7. Cần có 50000 đ từ các loại giấy bạc 1000đ, 2000đ và 5000đ. Tìm tất cả các phương án có thể.
8. Chuyển một số thập phân nguyên dương thành một số nhị phân, in ra màn hình dạng
   $X_{10} = Y_2$
9. Tính tích phân xác định của một hàm số trên một đoạn cho trước
10. Viết chương trình tìm và in ra màn hình các số nguyên tố nhỏ hơn một số cho trước.
11. Cho dãy số sau: $a_1,a_2,....,a_n$. Viết chương trình tìm phần tử lớn nhất, phần tử nhỏ nhất của dãy số đó.
12. Cho dãy số sau: $a_1,a_2,....,a_n$. Viết chương trình sắp xếp dãy theo thứ tự tăng dần .
13. Cho dãy số sau: $a_1,a_2,....,a_n$. Viết chương trình đếm số phần tử dương và xoá đi phần tử thứ m trong dãy (m<=n) .
14. Cho dãy số sau: $a_1,a_2,....,a_n$. Viết chương trình tìm các phần tử có giá trị là x nhập vào từ bàn phím.
15. Cho dãy số sau: $a_1,a_2,....,a_n$. Viết chương trình thêm phần tử có giá trị là x, vào vị trí m trong dãy. Sau đó tính tổng các phần tử của dãy mới.
16. Viết chương trình in ra các số nguyên tố trong khoảng từ $1 \rightarrow n$
17. Cho ma trận có m dòng và n cột, các phần tử là nguyên. Tìm phần tử nhỏ nhất của ma trận.
18. Cho ma trận có m dòng và n cột, các phần tử là nguyên. Tính tổng và trung bình cộng các phần tử của ma trận.

PHẦN II
NGÔN NGỮ LẬP TRÌNH TURBO PASCAL

CHƯƠNG I: CẤU TRÚC CỦA CHƯƠNG TRÌNH PASCAL

1. Giới thiệu về ngôn ngữ lập trình

1.1. Ngôn ngữ lập trình là gì ?

Ngôn ngữ lập trình là một hệ thống các kí hiệu, hệ thống các quy tắc cú pháp và một bộ các chỉ thị (hay còn gọi là câu lệnh) dùng để viết các thuật toán giải các bài toán thành các chương trình máy tính. Những chương trình này sẽ được đưa vào máy "chạy" để tính toán, xử lí bài toán.

Ngôn ngữ lập trình ra đời cùng với sự ra đời của MTĐT vì khi MTĐT ra đời đương nhiên người ta phải nghĩ đến việc xây dựng ngôn ngữ cho chúng. Các thế hệ MTĐT liên tục phát triển, do đó ngôn ngữ lập trình cũng phát triển theo. Có nhiều loại ngôn ngữ lập trình khác nhau, từ ngôn ngữ bậc thấp chuyển lên ngôn ngữ bậc cao.

+ Ngôn ngữ bậc thấp (ngôn ngữ máy): Chỉ sử dụng 2 kí hiệu 0 và 1 để mã hoá mọi đại lượng và phép toán. Ngôn ngữ này có ưu điểm là chương trình chạy nhanh, không phải qua khâu dịch. Tuy nhiên những chương trình viết bằng ngôn ngữ này rất cồng kềnh, viết mất rất nhiều thời gian, dễ sai sót, khó kiểm tra; mặt khác mỗi loại máy có một ngôn ngữ máy riêng.

+ Ngôn ngữ tập hợp: Khắc phục một số nhược điểm trên người ta nghĩ cách thay thế một số mã nhị phân bằng các chữ cái và cho ra đời ngôn ngữ tập hợp (còn gọi là ngôn ngữ kí hiệu). Ưu điểm là việc lập trình đã dễ dàng hơn, tuy nhiên cần phải có chương trình dịch từ ngôn ngữ tập hợp ra ngôn ngữ máy, và vẫn còn phụ thuộc vào từ máy.

+ Ngôn ngữ bậc cao: Chương trình được viết gần với ngôn ngữ tự nhiên của con người, do đó dễ dàng cho người lập trình, tuy nhiên chương trình sẽ chạy chậm hơn. Năm 1958, ALGOL - ngôn ngữ lập trình bậc cao đầu tiên ra đời. Năm 1960 nó được bổ sung, chỉnh lí và có ảnh hưởng sâu sắc đến các ngôn ngữ lập trình bậc cao sau này như FORTRAN, C, BASIC, PASCAL,...

1.2. Turbo Pascal

PASCAL là ngôn ngữ lập trình bậc cao của tác giả Niklaus Wirth (giáo sư người Thuỵ sĩ), được công bố vào đầu những năm 1970. Tên PASCAL là để kỉ niệm nhà Toán học người Pháp B. Pascal.

PASCAL là ngôn ngữ lập trình có **tính cấu trúc và tính hệ thống**: các kiểu dữ liệu đa dạng, các cấu trúc điều khiển chặt chẽ, các cấu trúc khối trong chương trình rõ ràng...

PASCAL là ngôn ngữ lập trình có **định kiểu rõ ràng**: các đại lượng (biến và hằng) đã được khai báo để sử dụng với kiểu dữ liệu này thì không thể đem dùng lẫn với kiểu khác.

PASCAL ban đầu được sáng tác để làm ngôn ngữ dạy học cho những người mới học lập trình. Đặc tính sáng sủa, dễ hiểu, dễ đọc của nó giúp người mới học có thể viết một chương trình máy tính một cách dễ dàng. Sau khi ra đời, do có nhiều ưu điểm,PASCAL đã được nhiều hãng máy tính phát triển và cài đặt cho nhiều hệ thống máy tính, như: ISO PASCAL (PASCAL chuẩn), ANSI PASCAL v.v. Phổ biến nhất hiện nay ở nước ta cũng như trên thế giới là việc cài đặt TURBO PASCAL cho các hệ thống máy tính. TURBO PASCAL được hãng BORLAND INTERNATIONAL hoàn thiện với các ưu điểm là chương trình viết gọn,

dịch nhanh, không ngừng được cải tiến đáp ứng yêu cầu của người sử dụng. TURBO PASCAL đã được phát triển qua các phiên bản ngày càng mở rộng và phức tạp. Việc khai thác sâu các tính năng của từng phiên bản là do khả năng và nhu cầu của người lập trình. Giáo trình này chọn version 5.0 làm chuẩn để minh hoạ, tuy nhiên người đọc có thể áp dụng cho các phiên bản sau này như 6.0, 7.0.

2. Yêu cầu phần mềm và cách khởi động

2.1. Yêu cầu phần mềm

Phần mềm TURBO PASCAL có khá nhiều File, tuy nhiên để có thể bước đầu học và sử dụng được, trên đĩa cần có tối thiểu các file sau:

TURBO.EXE file chính, chứa cả chương trình dịch, hệ soạn thảo và các dịch vụ gỡ rối...

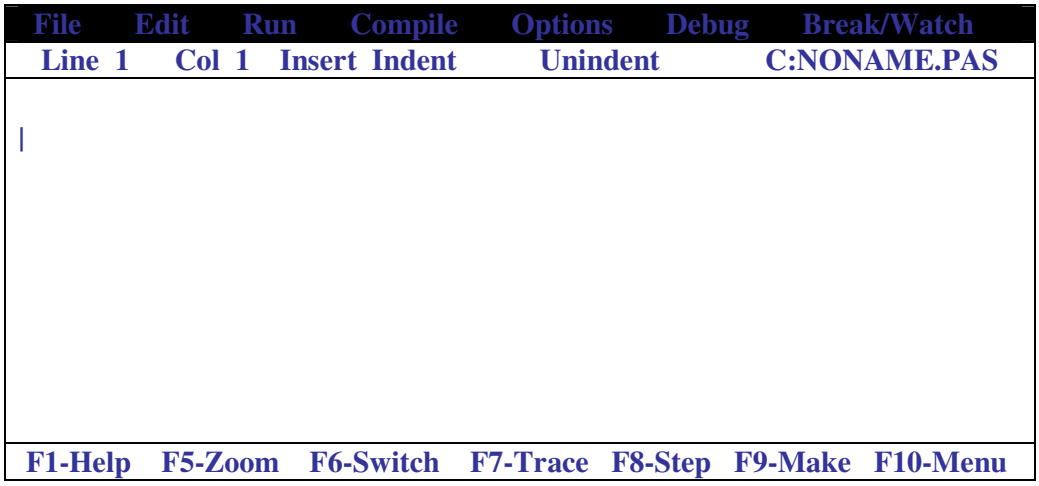
TURBO.TPL là thư viện các chương trình mẫu có sẵn của TURBO PASCAL

Nếu có sử dụng đồ hoạ thì cần chép thêm đơn vị GRAPH.TPU và một số file phụ trợ khác tuỳ theo kiểu màn hình và kiểu chữ sử dụng.

2.2. Khởi động và thoát TURBO PASCAL

Thông thường các file nói trên nằm trong thư mục TP (hoặc TURBO hay TP5, TP7 tuỳ theo phiên bản hay người sử dụng)

Ta chuyển vào thư mục này bằng lệnh CD TP↵, sau đó gõ TURBO ↵ . Màn hình sẽ hiện ra như sau:



Dòng 1: là dòng menu chính, trên đó có ghi các mục cơ bản nhất để người sử dụng có thể chọn bằng cách ấn đồng thời Alt+ chữ cái đầu của mục. Ví dụ: Alt+F để mở mục File.

Dòng 2: là dòng trạng thái, cho người dùng biết một số thông số soạn thảo như toạ độ con trỏ, chế độ soạn thảo là viết chèn hay viết đè, tên tệp đang làm việc...

Dòng cuối cùng là dòng hướng dẫn, nhắc người sử dụng một số phím chức năng chính.

Phần màn hình rộng ở giữa là phần soạn thảo để ta nhập và chạy chương trình.

Người mới học lập trình nên ghi nhớ ngay các phím chức năng:

F2 ghi tệp đang soạn thảo vào đĩa. Nếu chưa đặt tên sẽ có thông báo để ta gõ tên vào (tên được đặt theo quy định của MS-DOS)

F3 mở tệp mới
Alt+F9 dịch chương trình
Ctrl+F9 dịch và cho chạy chương trình
Alt+X Kết thúc PASCAL trở về hệ điều hành

3. Những quy định về cách viết biểu thức và chương trình

3.1. Bộ kí tự

TURBO PASCAL sử dụng bộ chữ viết như sau:
- Bộ 26 chữ cái Latinh: A, B, ..., Z ; a, b, ..., z
- Bộ các chữ số: 0, 1, ..., 9
- Các kí hiệu Toán học: +, -, * , /, =, <, >, ...
- Dấu gạch nối, dấu chấm, dấu ngoặc và các kí hiệu đặc biệt khác: _ . , ; ? [ ] ( ) $ % ...

3.2 Từ khóa

Từ khoá là một số từ dành riêng của TURBO PASCAL, những từ này đã được gán sẵn những chức năng đặc biệt, ta không được dùng vào việc đặt tên hay các công việc khác.
Sau đây là một số từ quan trọng, bạn đọc sẽ lần lượt biết các từ khoá và chức năng của từng từ qua các bài học.
- Các từ khoá chung: Program, begin, end
- Các từ khoá để khai báo: uses, const, type, label, var
  procedure, function
  array, string, record, **...**
- Các từ khoá lệnh:
  if... then... else...
  case ...of ....
  for .... to ... do...
  while... do...
  repeat... until...
- Một số tên chuẩn:
  Boolean, char, integer, real,
  read, readln, write, writeln,
  abs, cos, sin, sqrt, exp...

3.3 Tên

Tên là một dãy kí tự dùng để chỉ tên các đại lượng như tên hằng, tên biến, tên kiểu hay tên chương trình con.

Tên được đặt theo quy tắc sau: bắt đầu bởi một chữ cái, rồi đến chữ số hay dấu gạch nối và không dài quá 127 kí tự (!). Tên thường được đặt mang tính gợi nhớ. TURBO PASCAL không phân biệt chữ hoa và chữ thường khi viết từ khoá hay tên.
Ví dụ : khi viết chương trình giải phương trình bậc 2, ta có thể đặt tên chương trình là GIAI_PHUONG_TRINH_BAC2, đặt tên biến chứa biệt thức Δ là delta (không được dùng kí hiệu Δ), đặt tên các nghiệm là X1 và X2,...
- Dấu chấm phảy (;) dùng để ngăn cách các câu lệnh trong một chương trình
- Lời giải thích được đặt trong cặp dấu (* ... *) hoặc { ... }, các nội dung được chứa trong cặp dấu này sẽ bị chương trình bỏ qua trong khi dịch

4. Cấu trúc chung của một chương trình PASCAL

Ta hiểu một chương trình máy tính là một dãy các lệnh, các chỉ thị hướng dẫn máy thực hiện một nhiệm vụ, một xử lí nào đó trên tập các dữ kiện vào và cho ra kết quả.
Một chương trình PASCAL có cấu trúc chung như sau:

```
PROGRAM     Tên_ chương_trình;
USES        Danh sách đơn vi.;
LABEL       Danh sách  nhãn;
CONST       Danh sách hằng;
TYPE        Danh sách kiểu;
VAR         Danh sách biến;
PROCEDURE   Khai báo thủ tục;
FUNCTION    Khai báo hàm;
BEGIN
        { Các câu lệnh; }
END.
```

Một chương trình PASCAL điển hình gồm 3 phần: Phần tiêu đề, phần khai báo, phần thân chương trình.

4.1. Phần tiêu đề

Phần này có dạng

```
PROGRAM     Tên_ chương_trình;
```

bắt đầu bằng từ khoá PROGRAM, ít nhất một dấu cách rồi đến tên chương trình, kết thúc bởi dấu chấm phảy (;). Tên_ chương_trình do ta đặt ra, mang ý nghĩa nội dung của chương trình, dấu ; là bắt buộc phải có.
Ví dụ : Program Giai_PT_bac2;
Phần tiêu đề là tuỳ chọn, có thể có hay không cũng được, nhưng nếu đã có thì phải đủ các thành phần như trên.

4.2. Phần khai báo

Phần này mô tả các đối tượng của bài toán mà chương trình sẽ xử lí. Các khai báo khác nhau được viết cách nhau bởi dấu ;. Gồm:

Khai báo đơn vị	uses ...
Khai báo nhãn	label
Khai báo hằng:	const
Khai báo kiểu:	Type
Khai báo biến:	Var
Khai báo chương trình con:	Procedure
	Function

Các khai báo của phần này tuỳ thuộc từng bài toán cụ thể mà có thể có hay không, hoặc có một hoặc một vài khai báo. Cách khai báo và sử dụng cụ thể sẽ được giới thiệu ở trong mỗi phần sau.

4.3. Phần thân chương trình

Phần này được bắt đầu bởi từ khoá BEGIN , kết thúc bởi từ khoá END và dấu chấm (.) hết. Đặt giữa BEGIN và END là các lệnh của chương trình , các lệnh được viết cách nhau bởi dấu ; như sau:

```
BEGIN
    Lệnh 1;
    Lệnh 2;
    ...
    Lệnh n;
END.
```

Phần này bắt buộc phải có đối với mọi chương trình, nó chứa các lệnh xử lí các dữ liệu đã được mô tả ở phần khai báo. Đây là phần chính của chương trình để giải quyết bài toán. Người lập trình phải bỏ nhiều công sức để thiết kế thuật giải và viết chương trình cho phần này.

4.4. Các bước cơ bản khi lập trình

Trước khi viết một chương trình giải quyết một bài toán nào đó, người lập trình phải biết cách giải bài toán đó hay chính xác hơn phải biết thuật giải để giải bài toán và trình bày thuật giải bằng ngôn ngữ lập trình .

Bước 1: Soạn thảo chương trình

Đây là bước viết mới một chương trình, có thể viết trên giấy trước rồi nhập vào máy, sửa lỗi, cập nhật. Các chương trình này được gọi là chương trình nguồn, tên chúng được tự động gán đuôi PAS. Ta có thể dùng một hệ soạn thảo văn bản nào đó, thường là hệ soạn thảo văn bản của chính TURBO PASCAL.

Bước 2: Dịch chương trình (Alt+F9)

Gọi chương trình dịch (compiler) dịch chương trình nguồn đã viết ở bước 1 sang dạng mã máy, kết quả thường tạo ra các tệp dạng *.EXE, *.OBJ. Bước này sẽ cho phép ta phát hiện các lỗi để sửa. Thường các lỗi cú pháp như thiếu dấu ; hoặc viết sai từ khoá, sai tên sẽ được thông báo. Ta phải sửa hết các lỗi rồi chuyển sang bước 3.

Bước 3: Chạy chương trình và thử (Ctrl+F9)

Nếu các dữ kiện được cung cấp chính xác mà chương trình cho kết quả sai thì ta phải xem lại thuật giải. Đây là lỗi đặc biệt nghiêm trọng vì nó không thể hiện ra ngoài qua các thông báo lỗi và có thể làm sai toàn bộ bài toán. Ta cũng phải quay lại bước 1 để sửa và chạy lại.

Câu hỏi ôn tập chương I

1- Nêu quy tắc đặt tên trong Turbo Pascal. Cho 3 ví dụ tên đặt đúng, 3 ví dụ tên đặt sai.

2- Nêu cấu trúc chung của một chương trình Turbo Passcal.

CHƯƠNG II
CÁC KIỂU DỮ LIỆU CƠ SỞ VÀ CÁCH KHAI BÁO

1. Khái niệm dữ liệu, kiểu dữ liệu

1.1. Khái niệm dữ liệu

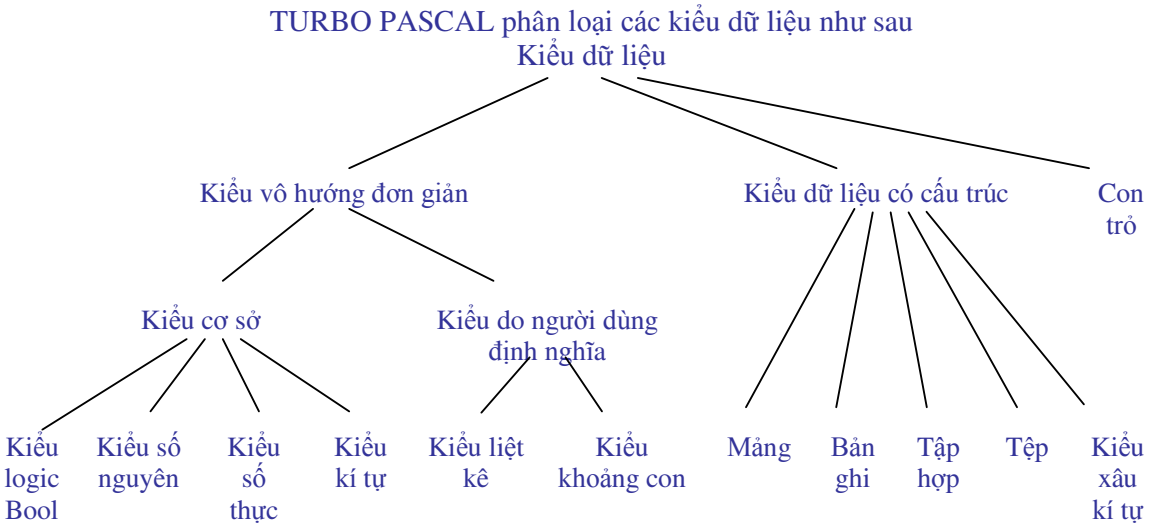
Dữ liệu là tất cả các thông tin có thật được máy tính xử lí. Chúng có nhiều dạng khác nhau, thể hiện qua các đối tượng cần xử lí như văn bản, số liệu, âm thanh, hình ảnh,...

1.2. Kiểu dữ liệu

Kiểu dữ liệu là tập hợp tất cả các giá trị mà một đại lượng thuộc kiểu đó có thể nhận được, trên đó xác định một số phép toán.

Điều đó có nghĩa là một đại lượng (biến, hằng,...) chỉ có thể nhận một tập hợp các giá trị nhất định (chứ không phải mọi giá trị có thể có) và trên tập hợp các giá trị đó có xác định một số phép toán. Chẳng hạn kiểu số nguyên thông thường chỉ nhận các giá trị trong phạm vi -2^{15} đến $2^{15}-1$, trên đó có các phép toán số học +, -, *, /, div, mod và các phép so sánh =,<,>.

1.3. Phân loại các kiểu dữ liệu



2. Các kiểu dữ liệu đơn giản

2.1. Kiểu số nguyên

Kiểu số nguyên bao gồm tất cả các số nguyên có thể biểu diễn được trên máy tính.

Từ khoá : Integer

Phạm vi biểu diễn: từ -32768 đến 32767 , tức -2^{15} đến $2^{15}-1$ (dùng 2 byte=16 bit)

Như vậy kiểu integer chỉ là một đoạn của tập số nguyên

Các phép toán: Kiểu integer được thực hiện với tất cả các phép toán dành cho số nguyên thông thường

+ Các phép toán số học: + (cộng) , - (trừ), * (nhân) , / (chia) , DIV (phép chia lấy phần nguyên, ví dụ: 7 div 3=2), MOD (phép chia lấy phần dư, ví dụ: 7 mod 3=1)
+ Các phép so sánh: = (bằng) , < (nhỏ hơn) , <= (nhỏ hơn hoặc bằng), > (lớn hơn) , >= (lớn hơn hoặc bằng), <> (khác)
Trong TURBO PASCAL, ngoài kiểu số nguyên được khai báo với từ khoá integer nói trên còn có các kiểu số nguyên khác (xem bảng) phục vụ cho các mục đích tính toán khác nhau

Kiểu (từ khoá)	**Phạm vi biểu diễn**	**Kích thước (byte)**
Byte	0..255	1
Shortint	-128..127	1
Integer	-32768..32767	2
Word	0..65535	2
Longint	-2147483648.. 2147483647	4

2.2. Kiểu số thực

Kiểu số thực là tập hợp tất cả các số thực có thể biểu diễn được trên máy tính. Không phải tất cả các số thực đều có thể biểu diễn được trên máy tính mà độ lớn và độ tinh xác của nó phụ thuộc vào máy và chương trình dịch.

- Kiểu số thực được khai báo với từ khoá real
- Phạm vi biểu biễn: từ $1.9x10^{-39}$ đến $1.7x10^{38}$
- Các phép toán: Kiểu real có thể thực hiện với tất cả các phép toán của số thực thông thường: +, -, *, / và các phép so sánh (không có phép div và mod dành cho số thực).
- Cách viết số thực dạng dấu phảy động:

Trong máy tính số thực được viết dùng dấu chấm để ngăn giữa phần nguyên và phần lẻ thập phân. Trong dạng dấu phảy động số thực được viết theo dạng: AE+b, trong đó A gọi là phần định trị, E+b gọi là phần mũ.

Ví dụ: Ta xem xét các cách viết khác nhau của cùng 1 số

$824.12345=8.2412345x10^{2}$ hay 8.2412345E+2

$=824123.45x10^{-3}$ hay 824123.45E-3

Trong TURBO PASCAL, ngoài kiểu số thực được khai báo với từ khoá real nói trên còn có các kiểu số thực khác (xem bảng) phục vụ cho các mục đích tính toán với độ lớn và độ chính xác khác nhau

Kiểu (từ khoá)	**Phạm vi biểu diễn**	**Chữ số có nghĩa**	**Kích thước (byte)**
Single	1.5E-45..3.4E+38	7-8	4
Real	2.9E-39..1.7E+38	11-12	6
Double	5.0E-324..1.7E+308	15-16	8
Extended	3.4E-4932..1.1E+4932	19-20	10

* Các hàm số học chuẩn dùng cho kiểu số nguyên và số thực: Đây là những hàm được định nghĩa sẵn trong TURBO PASCAL, ta có thể sử dụng như các công cụ để tính toán

Tên hàm (trong TP)	Cho kết quả (hàm)
ABS(X)	| X (giá trị tuyệt đối của X)
SQR(X)	X^2 (X bình phương)
SQRT(X)	$\sqrt{X}$ (căn bậc 2 của X)
LN(X)	lnX (Logarit Nepe)
EXP(X)	e^X
SIN(X)	sinX ; X cho bằng radian
COS(X)	cosX; X cho bằng radian
ARCTAN(X)	arctangX
SUCC(n)	n+1 (số tiếp theo của n) , n nguyên
PRED(n)	n-1 (số kề trước của n) , n nguyên
TRUNC(X)	cắt, cho phần nguyên của X
ROUND(X)	Làm tròn phần lẻ của X

2.3. Kiểu kí tự

Kiểu kí tự là tập hợp tất cả các kí tự của bảng mã máy tính (ASCII)
Từ khoá Char
Có thể sử dụng các phép toán so sánh đối với kiểu kí tự, kí tự nào đứng trước theo thứ tự của bảng mã máy tính sẽ được coi là nhỏ hơn, kí tự đứng sau được xem là lớn hơn
Ví dụ: 'a' < 'b'
Hằng kí tự được viết trong cặp dấu nháy ' '
Các hàm dùng cho kiểu kí tự:

Tên hàm (trong TP)	Cho kết quả (hàm)
ORD(C)	cho số thứ tự của kí tự C trong bảng mã
CHR(n)	cho kí tự có số thứ tự là n
SUCC(C)	cho kí tự đứng kề sau
PRED(C)	cho kí tự đứng kề trước

2.4. Kiểu logic (Boolean)

Từ khoá Boolean
Các phép toán logic: NOT , AND, OR, XOR
Các phép toán so sánh: =, <, >, ...
Kiểu logic boolean chỉ có 2 giá trị : True và False
Người ta định nghĩa: False < True

2.5. Kiểu liệt kê (Enumerated scalar type)

Là kiểu cho phépngười lập trình có thể tự định nghĩa ra các kiểu vô hướng bằng cách liệt kê các giá trị của kiểu vô hướng ra thông qua các tên do người lập trình tạo ra. Danh sách các giá trị này được đặt trong ngoặc đơn và được mô tả bằng một tên kiểu trong phần mô tả (Phần TYPE).

Từ khoá: **TYPE**

Ví dụ:

```
TYPE
  Boolean=(False, True);
  Color=(Red, Blue, Green, While, Black);
```

Một biến vô hướng có thể định nghĩa thông qua các kiểu đã được mô tả trong phần **TYPE** như sau:

```
VAR
    Ketqua: Boolean;
    Mau1, Mau2: Color;
```

Hoặc khai bái trực tiếp với mô tả kiểu dữ liệu:

```
VAR
    Gioitinh: (Nam, nu);
    Ngay: (Chunhat, Hai, Ba, Tu, Nam, Sau, Bay);
```

Ví dụ:

```
Ketqua:= True;
Mau1:=Blue;
Gioitinh:=Nam;
Ngay:=Chunhat;
```

2.6. Kiểu khoảng con (Sub- range Type)

Là kiểu vô hướng được dụng khi một biến chỉ được láy giá trị trong một khoảng (xác định bởi cận trên và cận dưới).

- Quy tắc định nghĩa:

  Hang_can_duoi..Hang_can_tren;

Trong đó: **Hang_can_duoi < Hang_can_tren,** và cùng kiểu.

- Tác dụng: - Tiết kiệm ô nhớ.
  - Có thể kiểm tra giá trị của biến khi chạy chương trình không được vượt ra khỏi giới hạn của khoảng con.

  Ví dụ:

```
TYPE
    Ngay= (Chunhat, Hai, Ba, Tu, Nam, Sau, Bay);
    Chu_cai_hoa='A'..'Z';
VAR
    ch: Chu_cai_hoa;
    Ngay_lam_viec: Hai..Bay;
```

3. Các khai báo

3.1. Khai báo đơn vị

Đơn vị là một kiểu tổ chức của chương trinh Pascal, trong đó có các chương trình con, nhằm thực hiện một một chức năng nào đó. Ta khai báo các đơn vị sử dụng trong chương trình theo cú pháp sau:

```
Uses    Tên_đơn_vi;
```

CRT là đơn vị chuẩn quản lý màn hình, bàn phím; GRAPH là đơn vị đồ họa. PRINTER là đơn vị khai báo máy in, máy in được cài đặt tên là LST.

```
Ví dụ 1:      Uses   crt;
Ví dụ 2:      Uses   crt, graph, printer;
```

3.2. Khai báo nhãn

Nhãn là một tên trong Pascal. Nhãn dùng đánh dấu một câu lệnh trong chương trình để có thể chuyển điều khiển tới đó. Khai báo nhãn trong chương trình theo cú pháp sau:

```
Label    Tên_nhãn;
```

Cách đánh dấu nhãn cho câu lệnh như sau:

```
Tên_nhãn : Câu lệnh;
```

Ví dụ:

```
    Label    Tiep;
    . . .
    Tiep : Readln(x);
```

3.3. Khai báo hằng

Hằng là đại lượng không thay đổi giá trị trong quá trình xử lí. Ta khai báo hằng theo cú pháp:

```
    const     Tên_hằng=Giátrị ;
```

ở đó: Tên_hằng do ta đặt ra, Giátrị là giá trị mà tên hằng sẽ nhận và giữ không đổi suốt quá trình xử lí.

```
    Ví dụ:      const    n=10;   { khai báo hằng số nguyên n=10}
                pi=3.1416;       { khai báo hằng số pi }
                ki_tu='Y';       { khai báo hằng kí tự ki_tu nhận kí tự Y}
```

3.4. Khai báo kiểu

Khai báo kiểu là định nghĩa một kiểu dữ liệu do người sử dụng tạo ra. Khai báo kiểu có cú pháp sau:

```
Type  Tên_kiểu = Mô_ tả_kiểu;
```

```
Ví dụ:     Type Mau = (do, xanh, vang, tim, nau);
           Type Phuong_tien_GT = (xe_dap, xe_may, o_to, tau_hoa);
           Type  Tuoi_nguoi = 1..300;
```

3.5. Khai báo biến

Biến là đại lượng có thể thay đổi giá trị trong quá trình xử lí. Biến của chương trình là tên của ô nhớ cất giữ dữ liệu. Muốn sử dụng biến nào ta phải khai báo trước biến đó bằng cách viết tên biến trong phần khai báo ở đầu chương trình sau từ khoá Var.
Cú pháp:

Var Tên_biến:Kiểu_dữ_liệu;

ở đó: Tên_biến do ta đặt ra, Kiểu_dữ_liệu là tên của một trong các kiểu dữ liệu của TURBO PASCAL, dấu hai chấm bắt buộc phải có để ngăn giữa 2 phần của khai báo, dấu chấm phảy kết thúc dòng khai báo. Các biến có cùng kiểu có thể được khai báo cùng nhau bằng cách viết các tên biến cách nhau bởi dâú phảy (,), nhóm các biến khác kiểu nhau được viết cách nhau bởi dấu ;

```
Ví dụ:  Var     i:integer;          { khai báo một biến có tên là I, biến này có kiểu nguyên}
                x,y,z:real;         { khai báo 3 biến x,y,z nhận các giá trị có kiểu thực }
                t: char;            { khai báo biến t có kiểu kí tự }
                kiem_tra:boolean;       { khai báo biến logic }
                Tuoi: 1..300;
```

Đối với mỗi bài toán, khi phân tích thuật toán, ta phải xác định rõ ngay chương trình cần phải nhập vào các dữ kiện gì ? các kết quả trung gian nào cần phải được chứa, dữ liệu ra là những gì?... để từ đó có phần khai báo chính xác, không bị thiếu mà cũng không thừa gây lãng phí ô nhớ (!)

4. Biểu thức và câu lệnh

4.1. Biểu thức

Biểu thức là một công thức tính toán để có một giá trị theo một công thức toán học nào đó. Một biểu thức bao gồm các toán tử kết hợp với các toán hạng.
Các toán tử trong TURBO PASCAL gồm: các phép toán số học (+, -, *, /), các phép so sánh (=, <, >, ...), các phép toán logic (NOT, AND, OR);
Các toán hạng: gồm các hằng, các biến, các hàm được dịnh nghĩa trong TURBO PASCAL
Thứ tự ưu tiên các phép toán như sau:

1. dấu ngoặc ( ) nội dung viết trong dấu ngoặc được ưu tiên thực hiện trước nhất
2. phép lấy đối (-), phép NOT
3. các phép tính loại nhân, chia: *, /, DIV, MOD, AND
4. các phép tính loại cộng, trừ: +, -, OR
5. các phép so sánh: +, <>, <, >, ...

Chú ý: Biểu thức trong Pascal chỉ được viết trên 1 dong do đó phải chú ý dùng các dấu ( ) để viết cho chính xác.

Ví dụ: Biểu thức toán học viết là: $\dfrac{b^2-4ac}{1+\sqrt{2ab}}$

Trong Pascal viết là: (b*b- 4*a*c)/(1+sqrt(2*a*b))

Biểu thức toán học viết là: $\dfrac{2+\sqrt{\sin x+\cos x}}{2\sin x+1}$

Trong Pascal viết là: (2+ sqrt (sin(x)+cos(x)))/(2* sin(x)+1)

4.2. Câu lệnh

Câu lệnh xác định công việc, thao tác mà chương trình phải thực hiện để xử lí các dữ liệu đã mô tả, khai báo. Các lệnh viết cách nhau bởi dấu (;), TURBO PASCAL phân loại các lệnh như sau:

- Lệnh đơn: là các lệnh không chứa các lệnh khác
- Lệnh có cấu trúc: thường chứa nhiều hơn một lệnh đơn, bao gồm khối lệnh, các lệnh thử và rẽ nhánh, các lệnh lặp.
- Lệnh hợp thành (còn gọi là lệnh phức hay lệnh ghép): gồm một nhóm các lệnh được đặt giữa cặp từ khoá begin ... end; (dấu chấm phẩy (;) kết thúc chứ không phải dấu chấm) như sau:

```
begin
        lệnh1;
        lệnh2;
        ...
        lệnhn;
end;
```

Dạng lệnh này cho ta thấy tính có cấu trúc của TURBO PASCAL. Một nhóm các lệnh giải quyết trọn vẹn một công việc nào đó sẽ được tổ chức trong một khối lệnh dạng này, và từng công việc của bài toán lớn sẽ được giao cho từng người làm trước khi ghép lại để giải bài toán tổng thể. Chúng ta sẽ đề cập kĩ hơn về vấn đề này ở phần chương trình con (thủ tục và hàm).

Bài tập chương 2

Hãy viết các khai báo biến cho các bài toán sau:

1. Tính n giai thừa: n! =1.2...n với n>1
2. Tính các tổng: S=1/2 + 1/4 +...+ 1/(2k)

$$Q=1.1!+2.2!+...+n.n!$$

3. Tìm và in ra tất cả các số chính phương nhỏ hơn một số cho trước, cho biết có bao nhiêu số chính phương như vậy.
4. Viết chương trình giải bài toán cổ: " Vừa gà vừa chó, bó lại cho tròn, ba mươi sáu con, một trăm chân chẵn. Hỏi có bao nhiêu gà, bao nhiêu chó?"
5. Viết chương trình tìm ước số chung lớn nhất của 2 số nguyên dương cho trước.
6. Tính Ex= $1+\frac{x}{1!}+\frac{x^2}{2!}+\ldots+\frac{x^n}{n!}+\ldots$ với độ chính xác $\varepsilon=10^{-4}$ ($ABS(x^n/n!) < \varepsilon$), giá trị x được nhập vào từ bàn phím khi chạy chương trình.
7. Cần có 50000 đ từ các loại giấy bạc 1000đ, 2000đ và 5000đ. Tìm tất cả các phương án có thể.
8. Chuyển một số thập phân nguyên dương thành một số nhị phân, in ra màn hình dạng

$X_{10} = Y_2$

CHƯƠNG III

CÁC THỦ TỤC VÀO RA DỮ LIỆU VÀ CÁC CẤU TRÚC ĐIỀU KHIỂN

1. Phép gán

Phép gán dùng để gán giá trị của một biểu thức cho một biến. Kí hiệu phép gán là **:=**

Cú pháp

Tên_biến:=Biểu thức;

Ý nghĩa: Khi gặp lệnh này, trước hết máy sẽ tính giá trị của biểu thức ở vế phải rồi gán giá trị tính được cho biến ở vế trái.

Chú ý: kiểu dữ liệu của biểu thức phải phù hợp với kiểu dữ liệu của biến ở vế trái.

Ví dụ: Với các biến đã được khai báo thích hợp ở đầu chương trình, trong thân chương trình ta có thể viết

```
x:=5;          nghĩ là biến x nhận giá trị 5
x:=x+1;    giá trị của biến x được thay thế bởi giá trị của chính nó tăng thêm một
delta:=b*b-4*a*c;  tính giá trị của delta trong chương trình giải phương trình bậc 2
doan:= true;
chu_cai:= 'A';
```

2. Các thủ tục Vào- Ra dữ liệu

2.1. Các thủ tục vào dữ liệu

Các biến của chương trình có thể nhận giá trị thông qua phép gán ở trong chương trình, tuy nhiên, để gán giá trị cho biến thông qua các thiết bị nhập chuẩn như bàn phím ta dùng một trong 2 thủ tục sau

read(danh sách các tên biến);

readln(danh sách các tên biến);

ý nghĩa: nhập giá trị cho từng biến trong danh sách tên biến bằng cách gõ vào từ bàn phím. Các biến trong danh sách các tên biến viết cách nhau bởi dấu phảy (,). Khi nhập các giá trị cụ thể, các giá trị phải phù hợp với các biến trong danh sách về số lượng biến, kiểu của biến và vị trí các biến trong danh sách. Các giá trị số được viết cách nhau bởi ít nhất một dấu khoảng trắng (dấu cách).

Các biến trong thủ tục này cho phép là các kiểu sau: Nguyên, thực, kí tự, xâu kí tự, khoảng con,... Không được là kiểu logic.

Ví dụ: với các khai báo ở trên (bài 2.III), ta có thể viết

```
read(x,y,z);
readln(t);
...
```

khi chạy chương trình ta có thể gõ: 3 6 4 ↵

N↵

nghĩa là biến x nhận giá trị 3, biến y nhận giá trị 6, biến z nhận giá trị 4 còn biến t nhận giá trị N (kiểu char).

Hai cách viết read và readln có ý nghĩa như nhau về tác dụng nhập dữ liệu, chỉ khác là sau khi thực hiện xong chức năng này thì lệnh readln sẽ đưa con trỏ về đầu dòng tiếp theo, còn lệnh read thì không.

Ngoài 2 cách viết trên còn có thủ tục readln; không có tham số cho phép dừng chương trình chờ gõ một phím bất kì trước khi tiếp tục.

Chú ý: Khi vào dữ liệu cho biến kí tự hoặc biến xâu kí tự ta dùng thủ tục Readln(..) và mỗi biến trong một thủ tục.

Ví dụ: Đầu chương trình khai báo các biến

```
Var  t: string; p: char; i,j,k : integer;
```

Trong thân chương trình vào dữ liệu cho các biến trên có thể viết như sau:

```
Readln(t); readln(p); readln(i,j,k);
```

2.2. Các thủ tục ra dữ liệu

Có 2 cách viết dữ liệu ra màn hình:

```
write( các mục cần ghi ra );
writeln( các mục cần ghi ra );
```

- ý nghĩa: đưa ra màn hình các kết quả tính toán trong chương trình, giá trị các biến, hay tất cả những gì nằm giữa cặp dấu nháy ' ... '

- Ví dụ: `write(x,y);` In ra màn hình các giá trị x và y, kết quả trên màn hình là 36

Có thể viết thành

```
write(x); write(y);
```

hay rõ hơn

```
write('x=',x);write('y=',y); thì kết quả là  x=3 y=6
```

các kí tự 'x=' và 'y=' nằm trong cặp dấu ngoặc chỉ có tác dụng trình bày, làm phân biệt rõ các giá trị được viết ra.

Hai cách viết trên có ý nghĩa như nhau về việc thể hiện dữ liệu ra màn hình, tuy nhiên thủ tục writeln sẽ chuyển con trỏ xuống đầu dòng tiếp theo sau khi thực hiện, con thủ tuc write giữ nguyên vị trí con trỏ.

Ngoài 2 dạng trên còn có thủ tục writeln; không có tham số chỉ có tác dụng đưa ra một dòng trắng không chứa gì.

2. 3. Kết hợp read, readln và write, writeln để nhập dữ liệu

Ta có thể kết hợp các thủ tục trên để tạo ra dạng nhập dữ liệu sáng sủa và đẹp mắt theo mẫu:

```
write(' câu nhắc nhập dữ liệu ');  readln(biến);
```

Ví dụ:

```
write( ' Hãy nhập dữ liệu cho biến x:'); hay   write('x=');
readln(x);                                     readln(x);
```

kết quả sẽ là:

```
Hãy nhập dữ liệu cho biến x:( ta gõ ) 3        hay x= ( ta gõ ) 3
```

nên viết câu dẫn nhập gợi ý biến sẽ được nhập vào, không nên bỏ qua hoặc viết

```
writeln( ' Hãy nhập dữ liệu cho biến x:'); hay        writeln('x=');
read(x);                                              read(x);
```

vì như thế khi chạy chương trình con trỏ nhập sẽ nhảy xuống dòng dưới dòng hướng dẫn, nhìn không đẹp mắt thế này

```
Hãy nhập dữ liệu cho biến x:          hay x=
( ta gõ ) 3                           ( ta gõ ) 3
```

2.4. Viết ra dữ liệu có quy cách

Một trong những yêu cầu khi trình bày dữ liệu ra màn hình là phải sáng sủa, đúng quy cách, dễ đọc, dễ hiểu. TURBO PASCAL có các quy định cho các kiểu dữ liệu như sau

. Viết ra kiểu số nguyên

write(i:n); hoặc writeln(i:n);

trong đó I là số nguyên cần ghi ra, n là số chỗ dành để viết ra số nguyên đó trên màn hình, máy sẽ bố trí số nguyên từ phải sang trái, nếu thừa sẽ bỏ trống bên trái.

Ví dụ: với I=23, j=234 thì writeln(i);writeln(i:5); writeln(j:5); sẽ cho

```
23
 _ _ 23
 _  234
```

Như vậy viết không có quy cách sẽ căn lề trái, trong khi viết có quy cách sẽ căn phải.

- Viết ra kiểu số thực

mẫu 1:

write(r:m:n); hoặc writeln(r:m:n);

trong đó r là số thực cần ghi ra, m là số chỗ dành cho cả số thực (kể cả dấu chấm ngăn cách phần nguyên và phần lẻ, nếu có) n là số chỗ dành cho phần lẻ thập phân.

mẫu 2:

write(r:n);

trong đó n là số chỗ cho cả số thực viết dưới dạng e-mũ

Ví dụ: với r=123.45

thì writeln(r); writeln(r:8:3); writeln(r:7); sẽ cho kết quả

```
1.234500000E+02      ( 10 chữ số dành cho phần lẻ )
123.450
1.2E+00         ( có làm tròn )
```

- Viết ra kiểu kí tự

Với kiểu kí tự, viết không quy cách sẽ cho ra kí tự bình thường, mỗi kí tự chiếm một chỗ, còn viết có quy cách thì các kí tự sẽ được bố trí từ phải sang trái, thêm các dấu khoảng cách vào bên trái nếu thừa chỗ.

Ví dụ: với t='Y' thì

writeln(t); writeln(t:3); writeln('PASCAL'); writeln('PASCAL':8);

sẽ cho

```
Y
  Y
PASCAL
   PASCAL
```

- Viết ra kiểu boolean

kiểu boolean với 2 giá trị là TRUE và FALSE cũng được viết ra theo dạng

write(ok); hoặc writeln(ok:n);

trong đó ok là biến kiểu boolean, n là số chỗ để viết ra biến ok.

e- In ra máy in

Các thủ tục write và writeln cũng dùng để đưa dữ liệu ra máy in. Muốn vậy, ở đầu chương trình phải có lời gọi chương trình chuẩn USES PRINTER; ở đầu chương trình và phải có thành phần Lst và dấu phảy (,) đứng trước nội dung cần in ra.

Ví dụ: writeln(Lst, 'kết qủa là', S:8:2);

- Các thủ tục trình bày màn hình của TURBO PASCAL

Đây là những thủ tục có sẵn của TURBO PASCAL phục vụ cho việc trình bày màn hình. Muốn dùng các thủ tục này phải có lời gọi chương trình chuẩn USES CRT; ở đầu chương trình. Các thủ tục đó là:

CLRSCR; xoá toàn bộ màn hình, chuyển con trỏ về góc trên bên trái của màn hình.

CLREOF; xoá các kí tự bên phải vị trí con trỏ hiện thời

GOTOXY(x,y); chuyển con trỏ màn hình đến điểm có toạ độ (x,y) (màn hình văn bản có 25 dòng, 80 cột, trong thủ tục này x là toạ độ cột, y là toạ độ dòng)

TEXTCOLOR(Mau); thiết lập màu cho văn bản, Mau là số nguyên có thể nhận giá trị từ 0 trở lên, giá trị lớn nhất của Mau phụ thuộc loại màn hình của máy. Ta có thể chỉ định Mau bằng một chữ cái tiếng Anh chỉ màu (xem bảng)

Số nguyên chỉ màu	Tên màu
0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	LightGrey
8	DarkGrey
9	LightBlue
10	LightGreen
11	LightCyan
12	LightRed
13	LightMagenta
14	Yellow
15	White

TEXTBACKGROUND(Mau); xác lập màu nền cho văn bản.

Ví dụ: đoạn lệnh chương trình in dòng chữ THU DO HA NOI màu xanh trên nền vàng lên màn hình

```
uses  CRT;
textbackground(14);
textcolor(blue);
write(' THU DO HA NOI');
```

2.5. Các chương trình ứng dụng những lệnh đã học

*** Bài toán 1:** Tính diện tích và chu vi của hình tròn.

Phân tích bài toán :Trong bài toán này biến vào là R; biến ra là: s (diện tích), cv (chu vi); các biến đều có kiểu số thực.

Tính toán theo công thức sau:

$s = 3.14 * R^2$;

cv = 2* 3.14 * R.

Chương trình được viết như sau:

```
Program  Dien_tich_chu_vi_hinh_tron;
Uses  crt;
Var cv, s, r : real;
Begin
  Clrscr;
  Write(' hay nhap vao ban kinh r: ');
  Readln(r);
  S:= 3.14 * r*r;
  Cv:=2*3.14*r;
  Writeln(' Dien tich = ', s:8:2);
  Writeln(' Chu vi = ', cv:8:2);
  Readln;
End.
```

*** Bài toán 2:** Bài toán tính lương, bảo hiểm xã hội, bảo hiểm y tế và tiền lĩnh của cán bộ công chức nhà nước. Theo quy định hiện nay thì lương = hệ số lương * 350000;

bảo hiểm xã hội = 5% * lương ; bảo hiểm y tế = 1% * lương;

tiền lĩnh = lương - bảo hiểm xã hội - bảo hiểm y tế.

Phân tích bài toán: Trong bài toán này các biến vào là Ht (họ và ten) kiểu xâu kí tự và biến HS (hệ số lương) kiểu số thực. Các biến ra là Ht, HS, luong, BHXH (bảo hiểm xá hội) kiểu thực, BHYT (bảo hiểm y tế) kiểu thực, TL (tiền lĩnh) kiểu thực.

Tính toán theo các công thức đã cho ở trên.

Trong chương trình có sử dung các câu lệnh trình bày màn hình.

Chương trình như sau:

```
Program  Tinh_luong;
Uses        Crt;
Var         HT:string[25]; LUONG, HS, BHXH, BHYT, TL:Real;
Begin
  Clrscr;
  Textcolor(red);
  Textbackground(Blue);
  Gotoxy(10,5);
  Write(' Nhap ho va ten: '); Readln(HT);
  Write(' Nhap he so luong '); Readln(HS);
  LUONG:= HS * 350000;
  HBXH:= LUONG * 0.05;
  BHYT:= LUONG * 0.01;
  TL:= LUONG-BHXH-BHYT;
  WRITELN(HT:25,HS:6:2,LUONG:10:1,BHXH:10:1,BHYT:10:1,TL:10:1);
  READLN;
END.
```

3. Các lệnh điều kiện

Phần này ta sẽ nghiên cứu các lệnh cho phép chương trình rẽ nhánh thực hiện một công việc dựa trên giá trị một điều kiện nào đó. Có 2 dạng lệnh điều kiện là câu lệnh điều kiện (cho phép rẽ tối đa 2 nhánh) và câu lệnh lựa chọn (cho phép rẽ nhiều nhánh).

3.1. Câu lệnh điều kiện

a. Dạng lệnh

- Dạng đơn giản

IF <điều kiện> THEN < lệnh >;

trong đó

<điều kiện> là một biểu thức điều kiện liên hệ bởi các toán tử: =, < , > , <= , >= và (hoặc) các phép toán logic NOT, AND, OR

< lệnh > có thể là một câu lệnh đơn giản hoặc phức hợp.

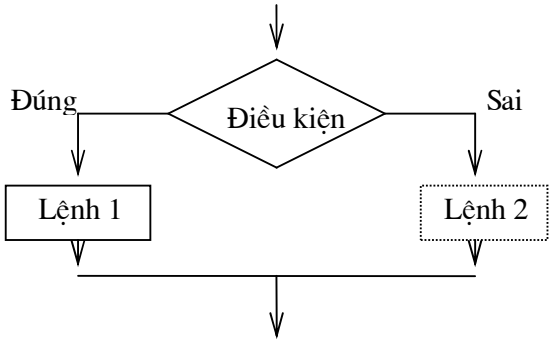
- ý nghĩa : khi gặp lệnh này, trước tiên máy sẽ kiểm tra <điều kiện>, nếu <điều kiện> là đúng thì sẽ cho thực hiện < lệnh >, còn nếu <điều kiện> sai thì máy sẽ bỏ qua < lệnh > và chuyển sang công việc tiếp theo.

- Dạng tổng quát

IF < điều kiện > THEN < lệnh1 > ELSE < lệnh2 >;

- ý nghĩa : khi gặp lệnh này, trước tiên máy cũng sẽ kiểm tra <điều kiện>, nếu <điều kiện> là đúng thì sẽ cho thực hiện < lệnh1 >, còn trái lại là <điều kiện> sai thì máy sẽ thực hiện <lệnh2 >.

Sơ đồ khối của câu lệnh điều kiện như sau



b. Ví dụ áp dụng

VD1: Viết chương trình nhập vào một số a tuỳ ý, rồi kiểm tra nếu a không âm thì in ra căn bậc 2 của a, trái lại in ra thông báo 'số âm không có căn bậc 2'

Chương trình được viết như sau:

```
Program        canbac2;
Uses   crt;
Var    a: real;
Begin
        write(' Hay nhap vao so a:');
        readln(a);
        if  a >= 0 then writeln(' Can bac hai cua a la:', sqrt(a))
                        else writeln(' So am khong co can bac hai');
        Readln;
End.
```

VD2: Viết chương trình giải phương trình bậc hai $ax^2+bx+c=0$

```
Program        GPTB2;
Uses crt;
  var
   a,b,c,x1,x2,dta:real;
  Begin
    clrscr;
    gotoxy(5,5);
    writeln('CHUONG TRINH GIAI PHUONG TRINH BAC HAI');
    writeln;
    write('Vao cac he so:');
    readln(a,b,c);
    dta:=b*b-4*a*c;
    if dta>0 then
     begin
       writeln('Phuong trinh co 2 nghiem thuc:');
        X1:=(-b+sqrt(dta))/(2*a);
        X2:=(-b-sqrt(dta))/(2*a);
       writeln('X1=',X1:6:2);
       writeln('X2=',X2:6:2);
     end;
    if dta=0 then
       writeln('Phuong trinh co nghiem kep X=',-b/(2*a):6:2);
    if dta<0 then
     begin
       writeln('Phuong trinh co 2 nghiem phuc:');
       writeln('X1=',-b/(2*a):6:2,'-',sqrt(-dta):6:2,'i');
       writeln('X2=',-b/(2*a):6:2,'+',sqrt(-dta):6:2,'i');
     end;
    readln;
 End.
```

Lưu ý: - Câu lệnh phức hợp: trong chương trình có những chỗ TURBO PASCAL chỉ cho phép viết một lệnh, nhưng ta lại muốn viết nhiều hơn một lệnh thành phần thì các lệnh thành phần đó phải được đặt trong cặp từ khoá Begin ... end; (dấu chấm phảy kết thúc). Chẳng hạn trong ví dụ trên, ứng với trường hợp dta>0 máy phải thực hiện cả 5 lệnh trong cặp từ khoá begin...end; sau từ khoá then, còn ứng với trường hợp dta=0 thì máy chỉ phải thực hiện đúng một lệnh, do đó không cần dùng câu lệnh phức hợp.

- Các câu lệnh điều kiện có thể viết lồng nhau. Tức là các <lệnh1> và <lệnh2> lại có thể là câu lệnh điều kiện dạng IF...THEN IF...THEN...ELSE... khi đó cần lưu ý IF nào đi với THEN nào. Chẳng hạn xem đoạn chương trình phân loại kết quả học tập sau:

```
IF diem>=5 THEN
                IF diem>=7 THEN loai:='Kha gioi'  ELSE loai:='TB'
          ELSE
                IF diem>=3 THEN loai:='Yeu' ELSE loai:='Kem';
```

3.2. Câu lệnh lựa chọn: CASE ... OF...

Câu lệnh điều kiện chỉ cho phép ta thực hiện rẽ 2 nhánh ứng với hai giá trị đúng hay sai của biểu thức điều kiện. Để có thể thực hiện rẽ nhiều nhánh ứng với nhiều giá trị khác nhau của một biểu thức, ta phải sử dụng câu lệnh lựa chọn.

a. Dạng lệnh

Dạng đơn giản:

```
CASE  <biểu thức> OF
        hằng1: <lệnh1>;
        hằng2: <lệnh2>;
        ...
        hằngn: <lệnhn>;
END;
```

Dạng tổng quát:

```
CASE  <biểu thức> OF
        hằng1: <lệnh1>;
        hằng2: <lệnh2>;
        ...
        hằngn: <lệnhn>
        ELSE  <lệnh n+1>;
END;
```

Trong đó <biểu thức> và các hằng phải có *cùng kiểu* và phải là các kiểu vô hướng đếm được (các kiểu số nguyên kiểu kí tự, kiểu Boolean đã biết).

ý nghĩa:

Khi gặp câu lệnh rẽ nhánh, <biểu thức> đã được nhận một giá trị nào đó, giá trị này là một hằng (số hoặc kí tự). Nếu <biểu thức> nhận giá trị là hằng nào thì <lệnh> tương ứng với nó sẽ được thực hiện. Còn nếu <biểu thức> nhận giá trị không rơi vào hằng nào thì máy sẽ bỏ qua lệnh đối với dạng đơn giản, máy sẽ thực hiện <lệnh n+1> sau từ khoá ELSE đối với dạng tổng quát.

Lưu ý: Từ khoá END với dấu chấm phảy (;) ở trong câu lệnh này để chỉ rằng kết thúc câu lệnh lựa chọn chứ không phải kết thúc chương trình con.

b. Ví dụ áp dụng

VD1: Viết chương trình xem thời khoá biểu của một ngày trong tuần. Yêu cầu: máy in ra câu hỏi 'Bạn muốn xem thời khoá biểu của thứ mấy?' ta gõ vào ngày thứ (của tuần) muốn xem và máy sẽ in ra thời khoá biểu của ngày hôm đó.

Chương trình được viết như sau: (giả sử xem thời khoá biểu của một lớp phổ thông, bạn đọc có thể hiệu chỉnh theo ý mình)

```
Program       XEM_TKB;
 Uses  crt;
 Var   thu: 2..7;       {biến thu để chứa các thứ ngày trong tuần}
Begin
        write(' Ban muon xem thoi khoa bieu cua thu may?');
        readln(thu);
        CASE  thu  OF
                2:writeln('Toan      Ly     Van');
                3:writeln('Sinh      Ki     Hoa');
                4: writeln('Toan     Hoa    Ly');
                5: writeln('Van      Sinh   The duc');
                6: writeln('Su       Dia    Chinh tri');
                7: writeln('Van      Toan   Sinh hoat');
        END;
        Readln;
End.
```

Bạn đọc có thể áp dụng câu lệnh dạng tổng quát để mở rộng chương trình xem thời khoá biểu này sao cho khi gõ vào một số nào đó không phải là thứ trong tuần (2..7) máy sẽ in ra thông báo rằng đó là ngày nghỉ.

Lưu ý:

- Sau mỗi hằng máy chỉ thực hiện đúng một lệnh. Do đó nếu muốn dùng nhiều hơn một lệnh thì ta phải sử dụng câu lệnh phức hợp.
- Các hằng có thể viết gộp lại với nhau. Chẳng hạn nếu thứ 3 và thứ 5 có cùng thời khoá biểu thì thay vì phải viết 2 dòng lệnh, ta có thể viết:

```
3,5: writeln('Sinh        Ki        Hoa');
```

- Các câu lệnh lựa chọn có thể lồng nhau. Tức là trong câu lệnh lựa chọn lại có thể chứa câu lệnh lựa chọn khác.

4. Các lệnh lặp

Trong lập trình giải quyết các bài toán, ta có thể gặp trường hợp chương trình phải thực hiện lặp đi lặp lại một công việc nào đó. Việc thực hiện lặp phải theo một quy luật nào đó. Số lần lặp có thể được xác định trước hoặc không xác định. Sau đây ta sẽ lần lượt xét các lệnh lặp dạng này.

4.1- Lệnh lặp có số lần lặp xác định

Đây là một câu lệnh có cấu trúc cho phép thực hiện lặp đi lặp lại một công việc (đoạn lệnh) nào đó với số lần thực hiện được xác định trước. Để dễ hiểu trước hết ta xét một ví dụ đơn giản sau:

Giả sử ta phải viết ra các số từ 1 đến 10, mỗi số chiếm một dòng trên màn hình. Ta có thể thực hiện công việc này bởi 10 lệnh writeln như sau:

```
writeln(1);
writeln(2);
...
writeln(10);
```

Đây là cách viết dài dòng và đơn điệu trong khi ta có thể thực hiện công việc trên bởi một dòng lệnh ngắn gọn:

```
FOR  I:=1  TO  10  DO  writeln(I);
```

ý nghĩa của câu lệnh này là: cho một biến nhận giá trị nguyên I chạy từ 1 đến n (biến I lần lượt nhận các giá trị số nguyên từ 1 đến n), với mỗi giá trị của I máy sẽ thực hiện công việc sau từ khoá DO là in ra giá trị của I.

Chi tiết các bước thực hiện của vòng lặp FOR này như sau:

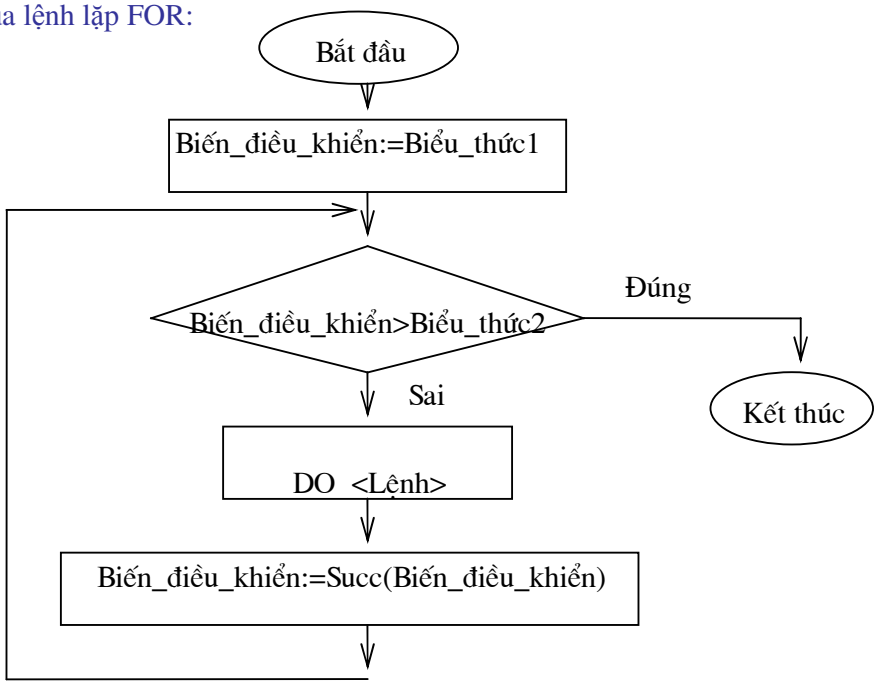
Đầu tiên biến I nhận giá trị khởi đầu là 1 (do lệnh gán I:=1), máy kiểm tra giá trị này không vượt quá giá trị cuối là 10 nên cho thực hiện lệnh writeln(I) viết ra giá trị 1. Sau đó biến I được tăng thêm một đơn vị, tức là I:=I+1. Bây giờ I=2, chưa vượt qua giá trị cuối là 10 nên lệnh writeln(I) lại được thực hiện để in ra giá trị của I là 2 lên màn hình. Rồi I lại được tăng lên... giá trị cuối cùng của I để lệnh writeln(I) được thực hiện là I:=10. Quá trình kết thúc khi I nhận giá trị 11 vượt quá giá trị cuối 10.

Câu lệnh trên là một ví dụ đơn giản của lệnh tổng quát sau:

a. Dạng lệnh lặp tăng (dạng tiến)

FOR Biến_điều_khiển:=Biểu_thức1 TO Biểu_thức2 DO <Lệnh>;

Trong đó Biến_điều_khiển, Biểu_thức1 và Biểu_thức2 phải cùng kiểu dữ liệu và phải là các kiểu vô hướng đếm được (như các kiểu số nguyên, kiểu kí tự...)
ý nghĩa của câu lệnh này như sau: Trước tiên Biến_điều_khiển nhận giá trị khởi tạo là Biểu_thức1, máy kiểm tra nếu giá trị đó không vượt quá Biểu_thức2 thì cho thực hiện <Lệnh>. Thực hiện xong lệnh này, Biến_điều_khiển được tăng thêm một đơn vị, máy lại kiểm tra với Biểu_thức2 để thực hiện <Lệnh>... Qúa trình tiếp diễn đến khi Biến_điều_khiển nhận giá trị vượt quá Biểu_thức2 thì dừng.
Số lần thực hiện lặp đi lặp lại <Lệnh> được xác định trước bởi giá trị của Biểu_thức1 và Biểu_thức2 nên câu lệnh trên còn được gọi là lệnh lặp có số lần lặp xác định.
Lưu đồ của lệnh lặp FOR:



* ***Ví dụ***

VD1: Viết chương trình tính tổng của n số tự nhiên đầu tiên S=1+2+...+n theo phương pháp cộng dồn.

Phương pháp cộng dồn được thực hiện như sau: ban đầu ta khởi tạo S:=0, rồi lấy S cộng với 1, rồi cộng với 2,... cộng đến n. Như vậy ta sẽ phải sử dụng một vòng lặp FOR với một biến điều khiển I chạy từ Biểu_thức1 là 1 đến Biểu_thức2 là n. Ta sẽ dùng chính giá trị của biến điều khiển này để tính tổng S.
Chương trình được viết như sau:

```
Program      Tinh_tong;
Uses     crt;
Var      i,n:integer;
         S:real;
Begin
         Write('Cho biet gia tri cua n:');
         Readln(n);
         S:=0;
         For I:=1 to n do S:=S+i;
         Writeln(' Tong tinh duoc la S:',S:10:2);
```

```
        Readln;
    End.
```

VD2: Một người gửi tiền tiết kiệm với số tiền ban đầu là A đồng, thời gian gửi t tháng, lãi suất tháng là p. Hỏi sau t tháng gửi người đó thu được số tiền lãi là bao nhiêu đồng?

Phân tích bài toán: Ta biết rằng gửi tiết kiệm thì số tiền thu được sau mỗi tháng sẽ bằng số tiền tháng trước cộng với lãi của tháng đó. Nếu gọi S là số tiền thu được sau mỗi tháng, t là số tháng gửi thì ta có:

Ban đầu $S_0=A$

t=1 --> $S_1=S_0 +S_0*p$ (gửi 1 tháng)

t=2 --> $S_2= S_1 +S_1*p$ (gửi 2 tháng)

...

t=n --> $S_n= S_{n-1} +S_{n-1}*p$ (gửi n tháng)

Đây chính là quy luật để thực hiện vòng lặp tính số tiền thu được. Chương trình được viết như sau:

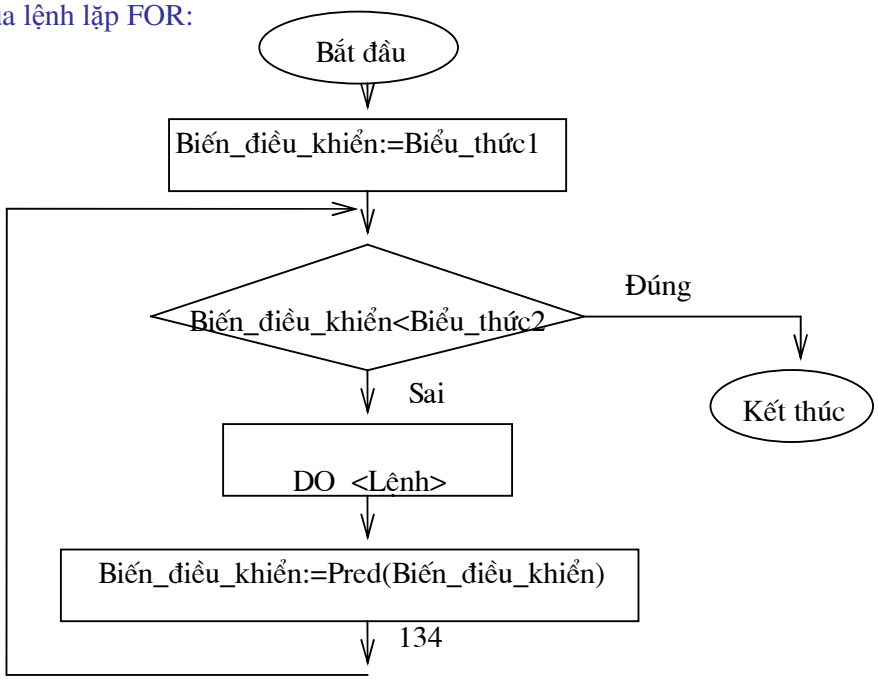
```
Program      Tinh_tien_lai;
Uses   crt;
Var    i,t:integer; A,P,S:real;
Begin
       Write('Cho biet so tien gui ban dau: '); Readln(A);
       Write('Cho biet so thang gui: '); Readln(t);
       Write('Cho biet lai suat theo thang: '); Readln(P);
       S:=A;
       For  I:=1  to  t  do  S:=S+ S*P;
       Writeln(' Lai thu duoc la : ',S-A:10:2);
Readln;
End.
```

b. Dạng lệnh lặp giảm (dạng lùi)

FOR Biến_điều_khiển:=Biểu_thức1 DOWNTO Biểu_thức2 DO <Lệnh>;

ý nghĩa hoàn toàn tương tự như dạng trên, chỉ khác là Biến_điều_khiển nhận giá trị giảm dần chứ không phải tăng dần, và điều kiện kiểm tra là Biến_điều_khiển không nhỏ quá Biểu_thức2.

Lưu đồ của lệnh lặp FOR:



Với dạng này thì câu lệnh FOR trong ví dụ 1 ở trên được viết là:

```
For I:=n downto 1 do S:=S+i;
```

Tổng S sẽ được cộng dồn theo thứ tự từ n đến 1, tức là S=n+(n-1)+...+2+1.

Ví dụ: Viết chương trình in ra các kí tự trong bảng mã ASCII theo thứ tự giảm dần.

Các kí tự trong bảng mã ASCII có mã giảm dần từ 255 đến 0. Ta dùng hàm CHR(n) để nhận được kí tự.

Chưng trình được viết như sau:

```
Program In_cac_ki_tu;
Uses   crt;
Var    I:integer;
Begin
       Clrscr;
       For i:=255 downto 0 do write(chr(i):4);
       Readln;
End.
```

c. Các lưu ý

1) Sau từ khoá DO chỉ được viết một lệnh, do đó nếu cần thực hiện nhiều hơn một lệnh đơn thì phải sử dụng câu lệnh phức hợp

2) Các lệnh lặp có thể lồng nhau, chẳng hạn dạng:

```
FOR   I:=1   TO  n  DO
  FOR   J:=1   TO  m  DO
        <Lệnh>
```

Khi đó với mỗi giá trị của biến điều khiển của vòng lặp ngoài thì biến điều khiển của vòng lặp trong sẽ chạy hết các giá trị của nó. Tức là với mỗi giá trị của I thì J sẽ chạy từ 1 tới m. Xét ví dụ đoạn chương trình sau:

```
For  I:= 1  to 2  do
For  J:=1   to  3  do
       begin
              k:=i+j;
              writeln(k);
       end;
```

sẽ cho kết quả là:

```
2
3
4
3
4
5
```

3) <Lệnh> sau từ khoá DO không được tuỳ tiện thay đổi giá trị của Biến_điều_khiển, làm như vậy ta có thể không kiểm soát được giá trị của Biến_điều_khiển và có thể làm rối vòng lặp.

Ví dụ: In ra tất cả các chữ số có 3 chữ số mà tổng các chữ số chia hết cho 3.

Trong bài toán này mỗi số đều có 3 chữ số, chữ số hàng trăn có thể nhận giá trị từ 1 đến 9, chữ số hàng chục, chữ số hàng đơn vị đều có thể nhận giá trị từ 0 đến 9. Để quét hết các chữ số có 3 chữ số ta dùng 3 vòng lặp lồng nhau, sau đó ta kiểm tra điều kiện nếu tổng các chữ số chia hết cho 3 thì in ra số đó.

Chương trình được viết như sau:

```
Program In_cac_so;
Uses   crt;
Var    t,c,v:integer;
Begin
       Clrscr;
       For t:=1 to 9 do
         For c:=0 to 9 do
           For v:=0 to 9 do
             If  (t+c+v) mod 3 = 0 then  writeln(t,c,v);
       Readln;
End.
```

4.2- Lệnh lặp có số bước lặp không xác định

ở trên ta đã xét lệnh lặp có số bước lặp xác định, tức là số lần thực hiện công việc lặp đi lặp lại là được định trước. Nhưng trong khi lập trình có những bài toán yêu cầu thực hiện một công việc nào đó mà số lần lặp là không thể xác định trước được, mà phụ thuộc vào một biểu thức điều kiện nào đó. Có hai dạng lệnh lặp với số bước lặp không xác định.

a. Lệnh lặp với điều kiện trước

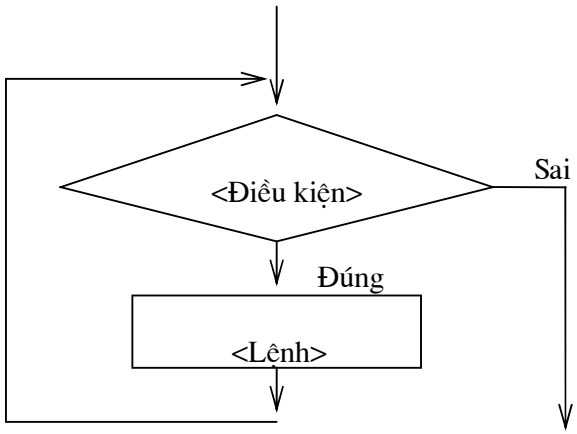
- Dạng lệnh:

WHILE <điều kiện> DO <Lệnh>;

ý nghĩa:

Khi gặp câu lệnh này, trước tiên máy sẽ kiểm tra <điều kiện>. Nếu <điều kiện> có giá trị đúng (TRUE) thì <Lệnh> được thực hiện. Thực hiện xong lệnh này máy sẽ quay lại kiểm tra <điều kiện> để thực hiện <Lệnh>... quá trình tiếp diễn đến khi <điều kiện> sai thì dừng.

Sơ đồ



- Ví dụ:

VD1: Một người gửi tiết kiệm không kì hạn với số tiền ban đầu là A đồng Hỏi sau bao nhiêu tháng người đó thu được số tiền là B đồng, biết rằng lãi suất là 1.8%?

Phân tích bài toán: Ta biết rằng gửi tiết kiệm không kì hạn thì số tiền thu được sau mỗi tháng sẽ bằng số tiền tháng trước cộng với lãi của tháng đó. Nếu gọi S là số tiền thu được sau mỗi tháng, t là số tháng gửi, ls là lãi suất thì ta có:

nếu t=0 --> S_0=A (gửi 0 tháng)
nếu t=1 --> S_1=S_0 +S_0*ls (gửi 1 tháng)
nếu t=2 --> S_2= S_1 +S_1*ls (gửi 2 tháng)
...

Đây chính là quy luật để thực hiện vòng lặp tính số tiền thu được. Như vậy ta phải so sánh số tiền thu được S sau mỗi tháng với số tiền cần đạt B: nếu S đạt hoặc vượt B thì ta có kết luận của bài toán. Đây cũng là điều kiện dừng vòng lặp. Ta sẽ sử dụng một biến đếm t để đếm số lần thực hiện lặp, cũng chính là số tháng gửi.

Chương trình được viết như sau:

```
Program     Gui_tiet_kiem;
Const  ls=0.018;
Var    A,B,S:real;
       t:integer;
Begin
       Clrscr;
       Write('Cho biet so tien gui ban dau:');
       Readln(A);
       Write('Cho biet so tien can dat :');
       Readln(B);
       S:=A;
       t:=0;
       WHILE  S<B  DO
              begin
                     S:=S+S*ls;
                     t:=t+1;
              end;
       Writeln(' Ban can gui it nhat la ', t, ' thang');
       Readln;
End.
```

VD2: Ta có thể viết lại chương trình tính tổng S=1+2+...+n sử dụng lệnh lặp WHILE... DO... như sau:

```
Program     Tinh_tong1;
Var    i,n:integer;
       S:real;
Begin
       Write('Cho biet gia tri cua n:');
       Readln(n);
       S:=0;
       i:=1;
       While  i<= n  do
              begin
                     S:=S+i;
                     i:=i+1;
              end;
```

```
        Writeln(' Tong tinh duoc la S:',S:10:2);
    End.
```

Như vậy ta có thể dùng lệnh lặp có số bước lặp không xác định để thay cho lệnh lặp có số bước lặp xác định. Tuy nhiên khi đó biến điều khiển vòng lặp không tự động thay đổi mà ta phải đưa nó vào thân vòng lặp (biến i trong VD trên).

b. Lệnh lặp với điều kiện sau

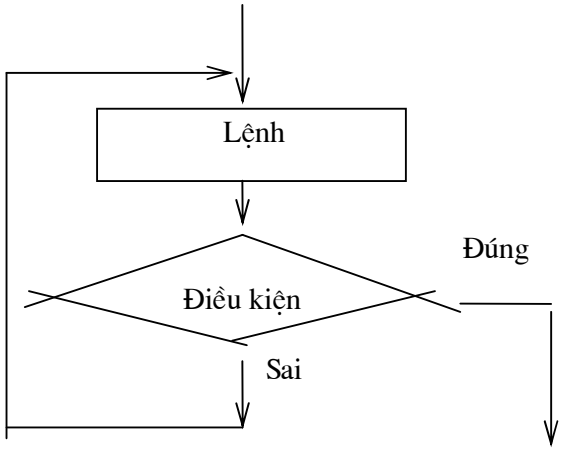
- Dạng lệnh:

    REPEAT <Lệnh> UNTIL <điều kiện>;

ý nghĩa:

Khi gặp câu lệnh này, trước tiên máy sẽ cho thực hiện <Lệnh>, sau đó kiểm tra <điều kiện>. Nếu <điều kiện> có giá trị sai (FALSE) thì quay lại thực hiện <Lệnh>. Thực hiện xong lệnh này máy sẽ lại kiểm tra <điều kiện> để thực hiện <Lệnh>... quá trình tiếp diễn đến khi <điều kiện> đúng thì dừng.

Sơ đồ



- Ví dụ: Lấy lại VD gửi tiết kiệm ở trên. Bây giờ ta sẽ viết lại chương trình sử dụng dạng lệnh lặp với điều kiện sau. Chương trình chỉ cần thay đổi một chút ở lệnh lặp.

```
    Program     Gui_tiet_kiem;
    Const  ls=0.018;
    Var    A,B,S:real;
           t:integer;
    Begin
           Write('Cho biet so tien gui ban dau:');
           Readln(A);
           Write('Cho biet so tien can dat :');
           Readln(B);
           S:=A;
           t:=0;
           REPEAT
                      S:=S+S*ls;
                      t:=t+1;
           UNTILS>=B;
           Writeln(' Ban can gui it nhat la ', t, ' thang');
           Readln;
```

```
End.
```

Ta thấy rằng trong phần thân của vòng lặp các lệnh đặt giữa REPEAT và UNTIL không cần sử dụng câu lệnh phức hợp begin...end;

Bạn đọc có thể sử dụng lệnh này để viết lại chương trình tính tổng ở trên.

c. Các lưu ý khi sử dụng các lệnh lặp WHILE và REPEAT

1) Trong thân vòng lặp phải có ít nhất một lệnh làm thay đổi giá trị của biểu thức điều kiện nhằm dừng vòng lặp. Nếu điều kiện luôn được thoả mãn thì lệnh có thể rơi vào vòng lập vô tận. Khi đó chỉ còn cách là tắt máy hoặc dừng chương trình.

Xem ví dụ sau:

```
I:=1;
WHILE  I<2  DO  write('Stop!');
```

Vòng lặp này sẽ chạy vô tận vì trong thân vòng lặp sau từ khoá DO chỉ có một lệnh in ra xâu chữ 'Stop!', không có lệnh nào làm thay đổi giá trị của I. Tức là với I được khởi tạo ban đầu là 1 luôn nhỏ hơn 2, điều kiện luôn thoả mãn.

2) Trong lệnh lặp với điều kiện trước WHILE thì thân vòng lặp (sau từ khoá DO) phải sử dụng câu lệnh phức hợp begin... end; còn trong lệnh lặp với điều kiện sau thì không cần.

3) Trong lệnh lặp với điều kiện trước, máy luôn kiểm tra điều kiện trước khi thực hiện lệnh, do đó nếu điều kiện sai ngay từ đầu thì lệnh không được thực hiện lần nào. Còn trong lệnh lặp với điều kiện sau, máy luôn thực hiện lệnh một lần rồi mới kiểm tra điều kiện, dù ban đầu điều kiện có thể đúng. Nhớ rằng trong lệnh lặp với điều kiện sau lệnh được quay lại thực hiện chỉ khi điều kiện sai.

4) Các lệnh lặp có thể viết lồng nhau

5) Người ta thường dùng các lệnh lặp có số bước lặp không xác định để quay vòng thực hiện nhiều lần cả một công việc nào đó hoặc cả chương trình, cho phép tạo sự tương tác giữa người sử dụng và máy. Xem ví dụ mẫu sau:

```
Program       VD;
Var    tieptuc:char;
       ... { các biến của chương trình }
Begin
       Repeat
              ...
              { đoạn chương trình thực hiện công việc và thay đổi điều kiện kiểm
       tra}
              write(' Ban co tiep tuc nua hay khong (C/K)');
              readln(tieptuc);
       UNTIL  (tieptuc='k')  or  (tieptuc='K');
End;
```

d. Lệnh nhảy vô điều kiện GOTO

Lệnh GOTO cho phép chương trình nhảy vô điều kiện tới một vị trí nào đó thông qua tên nhãn. Nhãn là một số nguyên hoặc một tên được khai báo trong phần LABEL ở đầu chương trình. Trong chương trình nhãn được đặt vào vị trí phù hợp kèm theo mộ dấu hai chấm (:).

Xem ví dụ sau:

```
Program       VDNHAN;
Uses   crt;
Label  N1,N2;
Var    a,b,s:real;
```

```
Begin
        a:=1;  b:=2;
    N1: a:=a+1;
        if  a<5  then GOTO  N1;
    N2: s:=a+b;
        if  s<10    then    GOTO  N2;
        writeln('a=', a, 'b=',b);
        readln;
End.
```

Thường người ta ít dùng lệnh GOTO trong chương trình PASCAL vì nó sẽ phá vỡ tính có cấu trúc của chương trình PASCAL. Nếu có thể, ta nên tránh dùng lệnh này.

Bài tập chương III

Viết chương trình cho các bài toán sau:

1. Tính n giai thừa: n! =1.2...n với n>1
2. Tính các tổng: S=1/2 + 1/4 +...+ 1/(2k)
   Q=1.1!+2.2!+...+n.n!
3. Tìm và in ra tất cả các số chính phương nhỏ hơn một số cho trước, cho biết có bao nhiêu số chính phương như vậy.
4. Viết chương trình giải bài toán cổ: " Vừa gà vừa chó, bó lại cho tròn, ba mươi sáu con, một trăm chân chẵn. Hỏi có bao nhiêu gà, bao nhiêu chó?"
5. Viết chương trình tìm ước số chung lớn nhất của 2 số nguyên dương cho trước.
6. Tính $Ex=1+\frac{x}{1!}+\frac{x^2}{2!}+\ldots+\frac{x^n}{n!}+\ldots$ với độ chính xác $\varepsilon=10^{-4}$ ($ABS(x^n/n!) < \varepsilon$), giá trị x được nhập vào từ bàn phím khi chạy chương trình.

7. Cần có 50000 đ từ các loại giấy bạc 1000đ, 2000đ và 5000đ. Tìm tất cả các phương án có thể.
8. Chuyển một số thập phân nguyên dương thành một số nhị phân, in ra màn hình dạng
   $X_{10} = Y_2$
9. Tính tích phân xác định của một hàm số trên một đoạn cho trước
10. Viết chương trình tìm và in ra màn hình các số nguyên tố nhỏ hơn một số cho trước.

CHƯƠNG IV

KIỂU DỮ LIỆU CÓ CẤU TRUC: KIỂU MẢNG, KIỂU XÂU KÍ TỰ, KIỂU TẬP HỢP

1 - Kiểu mảng

1.1- Khái niệm mảng (array)

a. Khái niệm mảng : Mảng là một kiểu dữ liệu có cấu trúc bao gồm một số xác định các phần tử có cùng kiểu, có một tên chung. Các phần tử của mảng được truy nhập thông qua các chỉ số.

Trong khái niệm này ta cần chú 2 điểm sau:

- Số phần tử của mảng phải là một số xác đinh, không được khai báo là biến. Ta có thể khai báo bằng một giá trị cụ thể chẳng hạn như 5,10,20,…; hoặc có thể khai báo là hằng.
- Các phần tử của mảng phải cùng kiểu. Có thể là các kiểu đơn giản hoặc kiểu có cấu trúc.

Ví dụ : Mảng A gồm 6 phần tử là các số nguyên: A[1] A[2] A[3] A[4] A[5] A[6] . Như vậy tên mảng là A, các chỉ số là 1,2,3,4,5,6.

b. Công dụng : Mảng là dùng để lưu trữ một dãy dữ liệu có cùng một tính chất. Ví dụ như họ tên của các thí sinh trong 1 lớp, lương của các nhân viên trong 1 cơ quan,...

Trong bộ nhớ của máy tính các phần tử của mảng được lưu trữ bởi các từ máy kế tiếp nhau. Trong ví dụ trên mảng A được lưu trữ trong bộ nhớ bằng 6 từ máy kế tiếp nhau, mỗi từ máy có độ dài là 2 bytes.

1.2- Khai báo mảng

Để khai báo mảng dùng cụm từ sau:

ARRAY [Kiểu_chỉ_số 1, Kiểu_chỉ_số 2, . . .] OF Kiểu_phần_tử ;

- Khai báo bằng định nghĩa kiểu

TYPE

Tên_kiểu_mảng = ARRAY [Kiểu_chỉ_số1, Kiểu_chỉ_số2, . . .] OF Kiểu_phần_tử ;

VAR

Tên_biến_mảng : Tên_kiểu_mảng ;

- Khai báo biến mảng trực tiếp qua khai báo VAR:

VAR

Tên_biến_mảng :ARRAY [Kiểu_chỉ_số1, Kiểu_chỉ_số2, . . .] OF Kiểu_phần_tử ;

Trong đó: Kiểu phần tử là kiểu của mỗi phần tử trong mảng. Kiểu phần tử có thể là kiểu bất kỳ.

Chỉ số để truy nhập đến các phần tử của mảng. Kiểu chỉ số chỉ cho phép là các kiểu đơn giản sau đây: Kiểu kí tự (CHAR), kiểu BOOLEAN, kiểu miền con (khoảng con), kiểu liệt kê. Kiểu chỉ số không được là kiểu REAL hoặc INTEGER.

Số chỉ số là số chiều của mảng, mảng 1 chiều có 1 chỉ số, mảng 2 chiều có 2 chỉ số, ... , mảng n chiều có n chỉ số.

Kích thước tối đa của mảng phải được khai báo là một số xác định (là hằng), chẳng hạn ta có thể khai báo là 5 hoặc 10 hay 100,... chứ không được khai báo là một biến như n,m,...

Ví dụ 1

```
var  B: array[ 1..5] of char ;
```

ở ví dụ 1 mảng B có kích thước tối đa là 5.

Ví dụ 2

```
TYPE    AB = ARRAY [1 .. 5]  OF INTEGER ;
        COLOR = ( Red, Blue, Green, While, Black );
VAR     X,Y,Z : AB;
        MAO, MKHAN : COLOR;
```

Ví dụ 3

```
VAR     DSHodem,DSTen : Array [1.. 200] of string [20] ;
        DSHeso, DSLuong, DSPhucap,DSTong : array [1.. 200] of real;
        So_lap : array [ 'a' .. 'z'] of integer;
```

1.3- Truy nhập mảng

Có thể truy nhập vào bất kỳ phần tử nào trong mảng. Để truy nhập vào một phần tử trong mảng ta viết theo qui cách sau:

Tên_biến_mảng [chỉ-số1, chỉ_số2, ..., chỉ_số]

Các phần tử của mảng được coi như một biến, có thể tham gia vào các thủ tục vào/ra, các biểu thức, lời gọi hàm.

Ví dụ:

```
Var  a:array[1..20] of integer;
A[1]:=1;
 readln(a[2];
 a[3]:=a[1]+a[2];
Writeln(a[3]);
```

1.4- Mảng 1 chiều

**** Khai báo mảng một chiều***: Dùng cụm từ sau

ARRAY [kiểu_chỉ_số] OF kiểu_phần_tử;

- Dùng khai báo kiểu:

```
TYPE    Tên_kiểu_mảng = ARRAY [ kiểu_chỉ_số ] OF kiểu_phần_tử ;
VAR     Tên_biến_mảng: Tên_kiểu_mảng ;
```

- Dùng khai báo biến :

```
VAR     Tên_biến_mảng : ARRAY [ kiểu-chỉ_số ] OF kiểu_phần_tử ;
```

Mảng một chiều chỉ có một chỉ số.

**** Cách dùng*** : Mảng 1 chiều thường được dùng cho dữ liệu ở dạng danh sách tuyến tính, ví dụ như dãy số, dãy xâu kí tự, ...

Ví dụ 1: Một dãy số nguyên a_1 , a_2 , ... , a_n ta khai báo như sau

VAR a: ARRAY [1. . 100] OF integer ;

Trong khai báo này n có giá trị tối đa là 100.

Ví dụ 2: Một danh sách có n tên học sinh ta khai báo như sau:

VAR Ten: ARRAY [1.. 200] OF String [25] ;

Trong khai báo này n có giá trị tối đa là 200, mỗi tên có tối đa là 25 kí tự.

Ví dụ 3: Danh sách số lần xuất hiện (tần số) của các chữ cái viết hoa trong một văn bản ta khai báo như sau:

VAR Tan_so : ARRAY ['A' . . 'Z'] OF integer ;

Trong khai báo này mảng có kích thước tối đa là 26 (26 chữ cái hoa trong bảng chữ cái tiếng Anh).

* Có thể truy nhập vào bất kỳ phần tử nào trong mảng. Để truy nhập vào một phần tử trong mảng một chiều ta viết theo qui cách sau:

Tên_biến_mảng [chỉ-số]

Ví dụ : Chỉ ra phần tử thứ 5 trong ví dụ 1 ta viết a[5]

Chỉ ra tên thứ 10 trong danh sách tên ta viết Ten[10]

Chỉ ra tần số của chữ 'B' ta viết Tan_so['B']

1.5. Các chương trình dùng mảng một chiều

Bài toán 1: Cho một dãy n số nguyên viết chương trình nhập dữ liệu vào, tính và in ra trung bình cộng, phần tử lớn nhất, phần tử nhỏ nhất của dãy số đó.

-Tư tưởng của thuật toán tìm phần tử max, phần tử min : Trước tiên gán phần tử đầu tiên là a[1] cho cả max và min, như vậy vị trí max, vị trí min đều là 1. Sau đó duyệt lần lượt các phần tử từ phần tử đầu cho tới phần tử cuối, nếu phần tử đang xét lớn hơn max thì gán giá trị phần tử đó cho max, vị trí của nó cho vị trí max, nếu phần tử đang xét nhỏ hơn min thì gán giá trị phần tử đó cho min, vị trí của nó cho vị trí min.

Chương trình

```
Program  Trung_binh_max_min;
 Uses  crt;
 Var  a:Array [ 1..100] of integer ;
       i,n,max,min,vtmax,vtmin : integer ; tb : real ;
 Begin
   Clrscr;
 { Nhap du lieu }
  Write(' Nhap  n: '); readln( n);
  for  i:=1 to n do
     begin
        Write( ' a[', i, ']=' );
         readln( a[i]);
      end;
  { Tinh toán }
    tb:=0; max:= a[1]; min:=a[1]; vtmax:=1; vtmin:=1;
   for i:=1 to n do
     begin
        tb:=tb + a[i];
        if  max <a[i]  then  begin max:=a[i]; vtmax:=i; end;
        if  min >a[i]  then  begin min :=a[i]; vtmin:=i; end;
     end;
  { in ket qua }
   writeln(' Trung binh = ', tb/n :8:2) ;
   writeln(' max= ', max, '  tai vi tri : ', vtmax);
   writeln(' min= ', min, ' tai vi tri : ', vtmin);
   readln;
end.
```

***Bài toán 2*:** Cho dãy n số thực a_1, a_2, .., a_n sắp xếp dãy theo thứ tự tăng dần.

Trong bài toán này ta dùng thuật toán tìm phần tử nhỏ nhất của dãy chưa sắp và đưa về đứng ở vị trí đầu tiên của dãy đó. Dãy có n phần tử thì ta phải thực hiện n-1 lần tìm phần tử lớn nhất.

Tư tưởng của thuật toán như sau:

-Ta so sánh phần tử đầu tiên của dãy chưa sắp lần lượt với các phần tử đứng sau nó , nếu có phần tử nào nhỏ hơn nó thì ta sẽ đổi giá trị cho nhau. Để đổi giá trị 2 phần tử cho nhau ta dùng một phần tử trung gian.

- Lặp lại bước trên cho đến khi dãy chưa sắp không chỉ còn một phần tử. Như vậy dãy có n phần tử thì ta lặp lại n-1 lần.

Chương trình

```
PROGRAM SAP_DAY_TANG;
USES CRT;
VAR   I,J,N: INTEGER; A:ARRAY[1..100] OF REAL; X:REAL ;
BEGIN
 CLRSCR;
 (*   NHAP SO LIEU    *)
 WRITE(' HAY NHAP SO PHAN TU N ? ');READLN(N);
 FOR I:=1 TO N DO
  BEGIN
    WRITE('A[',I,']= ');
    READLN(A[I]);
  END;
 (*  SAP XEP   *)
 FOR I:=1 TO N-1 DO
    FOR J:=I+1  TO N  DO  IF A[I]>A[J] THEN  BEGIN
                                    X:=A[I];
                                    A[I]:=A[J];
                                    A[J]:=X;
                                   END;
 (*  IN KET QUA   *)
 WRITELN('     DAY DA SAP THEO THU TU TANG');
 FOR I:=1 TO N DO WRITELN(A[I]:8:2);
 READLN;
END.
```

Bài toán 3: Nhập vào một xâu kí tự. Hãy in ra tần số xuất hiện của các chữ cái, không biệt chữ hoa và chữ thường. In ra chữ cái xuất hiện nhiều nhất.

Chương trình

```
program   Tan_so_chu_cai ;
uses    crt;
var  s:string; ts: array [‘A’..’Z’ ]  of integer ; n,j,m: integer ; i,vt: char ;
Begin
  clrscr ;
 { nhap xau ki tu }
Write(‘ Nhap xau ki tu : ‘); readln(s);
n:=length(s);
{ Tinh tan so }
for i:=’A’ to ‘’Z’  do ts[i] :=0 ;
```

```
 for j:= 1 to n do
    for i:='A' to 'Z' do
        if  upcase(s[j])=i  then ts[i]:=ts[i] + 1;
for  i:='A' to 'Z' do writeln(i,'   co tan so = ',ts[i]);
{ Tin tan so max }
m:=ts['A'] ; vt:='A' ;
for i:='B' to 'Z' do
    if  m<ts[j]  then  begin
                                        m:=ts[j];
                                        vt:=i ;
                            end ;
 Writeln('ki tu ',vt, 'co tan so lon nhat la : ',m);
 readln;
end.
```

1.6- Mảng nhiều chiều

Trong phần trình bày mảng 2 chiều, mảng nhiều hơn 2 chiều được suy diễn một cách tương tư.

**** Khai báo mảng 2 chiều***: Dùng cụm từ sau

ARRAY [kiểu_chỉ_số1, kiểu_chỉ_số2] OF kiểu_phần_tử;

- Dùng khai báo kiểu:

```
TYPE
Tên_kiểu_mảng = array [ kiểu_chỉ_số1, kiểu_chỉ_số2 ] OF kiểu_phần_tử ;
VAR      Tên_biến_mảng: Tên_kiểu_mảng ;
```

- Dùng khai báo biến :

```
VAR
Tên_biến_mảng : ARRAY [ kiểu_chỉ_số1,kiểu_chỉ_số2 ] OF kiểu_phần_tử ;
```

Mảng 2 chiều có 2 chỉ số.

**** Cách dùng*** : Mảng 2 chiều thường được dùng cho dữ liệu ở dạng bảng hay ma trận, ví dụ như ma trận số có m dòng ,n cột, ... Mảng hai chiều có 2 chỉ số, chỉ số 1 chỉ dòng, chỉ số 2 chỉ cột.

Ví dụ1: Một ma trận số nguyên a có 2 dòng, 3 cột được khai báo như sau

```
VAR  A: ARRAY [ 1. . 2, 1..3 ] OF  integer ;
```

Trong bộ nhớ máy lưu trữ các phần tử của mảng A kế tiếp nhau theo thứ tự sau:

A[1,1], A[1,2], A[2,1], A[2,2], A[3,1], A[3,2]

Ví dụ 2: Một bảng có 8 dòng, 8 cột, các ô của bảng chứa các chữ cái sẽ được khai báo như sau:

```
VAR  Bang: ARRAY [ 1. . 8, 1..8 ] OF  char ;
```

* Có thể truy nhập vào bất kỳ phần tử nào trong mảng. Để truy nhập vào một phần tử trong mảng hai chiều ta viết theo qui cách sau:

Tên_biến_mảng [chỉ_số1, chỉ_số2]

Cách viết trên để chỉ ra phần tử ở dòng có giá trị bằng chỉ số 1 và ở cột có giá trị bằng chỉ số 2.

Ví dụ:

```
Var A:array[1..10,1..5] of integer;
    A[1,1]:= 1;
    A[1,2]:=3*A[1,1];
    Readln(A[2,1]);
```

```
Writeln(A[1,2]);
```

1.7. Các chương trình dùng mảng 2 chiều

Bài toán 1: Viết chương trình nhập một ma trận m dòng , n cột và tìm giá trị lớn nhất, giá trị nhỏ nhất, tính tổng của các phần tử, đếm số phần tử âm.

Chương trình

```
 PROGRAM TINH_MA_TRAN;
(*TIM MAX, MIN, TINH TONG, DEM SO PHAN TU AM  *)
USES CRT;
VAR I,J,M,N,MAX,MIN,T,SOPTAM:INTEGER;
   A:ARRAY[1..30,1..20] OF INTEGER;
BEGIN
  CLRSCR;T:=0;SOPTAM:=0;
  (*  NHAP SO LIEU  *)
  WRITE('NHAP SO DONG M, SO COT N ');READLN(M,N);
  FOR I:=1 TO M DO
    FOR J:=1 TO N DO BEGIN
                 WRITE('A[',I,',',J,']=? ');
                 READLN(A[I,J]);
                END;
  (*  TINH TOAN  *)
  MAX:=A[1,1];
  MIN:=MAX;
  FOR I:=1 TO M DO
   FOR J:=1 TO N DO
    BEGIN
     T:=T+A[I,J];
     IF A[I,J]<0 THEN SOPTAM:=SOPTAM+1;
     IF MAX<A[I,J] THEN MAX:=A[I,J] ELSE IF MIN>A[I,J] THEN MIN:=A[I,J];
    END;
 (* IN KET QUA  *)
 WRITELN;
 WRITELN('TONG T= ',T);
 WRITELN('SO PHAN TU AM LA ',SOPTAM);
 WRITELN('GIA TRI LON NHAT = ',MAX);
 WRITELN('GIA TRI NHO NHAT = ',MIN);
 READLN;
END.
```

Bài toán 2: Nhân ma trận a có m dòng, n cột với ma trận b có n dòng, l cột.

Kết quả là ma trận c có m dòng, l cột. Các phần tử của ma trận c được tính theo công thức sau:

$$C[i,j] = \sum_{k=1}^{n} a[i,k] * b[k,j] \qquad \text{với } i: 1 \rightarrow m \text{ ; } j: 1 \rightarrow l$$

Chương trình

```
(*   nhan hai ma tram  *)
program nhan_ma_tran;
uses crt;
type mang=array[1..20,1..30] of integer;
var i,j,k,m,l,n:integer; a,b,c: mang;
begin
  clrscr;
  (* nhap ma tran a  *)
  write('nhap so dong, so cot m,n ? ');readln(m,n);
  for i:=1 to m do
    for j:=1 to n do
     begin
       write('a[', i, ',' ,j, ']= ');
       readln(a[i,j]);
     end;
 (* nhap ma tran b  *)
  write('nhap so cot l ? ');readln(l);
  for i:=1 to n do
    for j:=1 to l do
      begin
        write('b(',i,',',j,')= ');
        readln(b[i,j]);
      end;
 (*  Tinh toan  *)
  for i:=1 to m do
    for j:=1 to l do
      begin
        c[i,j]:=0;
        for k:=1 to n do c[i,j]:=c[i,j]+a[i,k]*b[k,j];
      end;
 (*  In ket qua  *)
 writeln(' Ma tran tich ');
 for i:=1 to m do
   begin
      for j:=1 to l do write( '   ',c[i,j]:4 );
      writeln;
   end;
 readln;
end.
```

2. Kiểu xâu kí tự

2.1. Khai báo kiểu xâu kí tự

a. Định nghĩa : Dữ liệu kiểu xâu là kiểu dữ liệu có cấu trúc, dùng để xử lý các xâu kí tự

Dữ liệu kiểu xâu được khai báo bằng từ khoá STRING , độ dài tối đa của một xâu kí tự là 255 kí tự. Có thể khai báo độ dài tối đa của một xâu kí tự trong dấu ngoặc vuông sau từ khoá STRING như sau: STRING[...]
Ví dụ: Tên người, quê quán, trình độ văn hóa là các dữ liệu kiểu xâu kí tự.

b. Khai xâu kí tự

Để khai báo xâu kí tự dùng từ khóa String nếu như khai báo độ dài xâu tối đa là 255 kí tự, dùng từ khóa String [n] nếu như khai báo độ dài xâu tối đa là n kí tự.

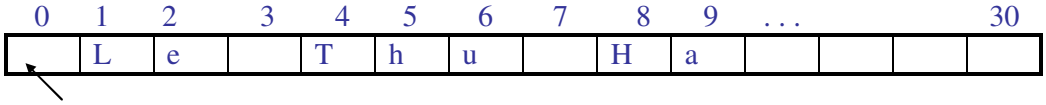
- Khai báo kiểu xâu:
  Type Tên_kiểu_xâu = String[n];
- Khai báo biến xâu
  Var Tên_biến: String[n];

Trong đó n là số kí tự tối đa của xâu, nếu không có phần [n] thì số kí tự tối đa của xâu mặc nhận là 255.
Ví dụ Var Hoten: string[30]; Ngaysinh: string[10]; Quequan: string;
Hằng xâu kí tự trong Pascal được viết trong ' '

```
Hoten:='Le Thu Ha';
Ngaysinh:='20-10-1960';
Quequan:='Gia lam - Ha noi ';
```

c. Truy nhập xâu kí tự

Mỗi kí tự trong xâu được chỉ ra bằng 1 số thứ tự được gọi là chỉ số viết trong [ ], kí tự đầu tiên có chỉ số là 1. Có thể truy nhập tới từng kí tự trong xâu theo cách viết sau: Tên_biến[chỉ_số]
Với ví dụ trên Hoten[2] là 'e', Ngaysinh[8] là 9.
Xâu kí tự được lưu trữ như sau: Byte đầu tiên chứa kí tự là độ dài thực của xâu, các byte tiếp theo là các kí tự của xâu.
Với ví dụ trên Hoten:='Le Thu Ha'; được lưu trữ như sau:



Kí tự đầu chứa độ dài thực của xâu là kí tự 9.
Độ dài của xâu = ORD(Hoten[0])

d. Hàm chuẩn Length(St) : Cho ta độ dài của xâu kí tự. Một xâu kí tự không chứa kí tự nào **là** rỗng**,** khi đó Length(St)=0, xâu kí tự rỗng được kí hiệu ''.

2.2. Các thao tác trên xâu kí tự

a. Phép cộng xâu

- Kí hiệu + là ghép nối các xâu kí tự
- Ví dụ Que:= 'Gia lam ' + 'Ha noi';

Cho kết quả Que= 'Gia lam Ha noi'

b. Phép So sánh hai xâu kí tự

Khi so sánh 2 xâu kí tự, so sánh mã ASCI I của từng cặp kí tự tương ứng từ 2 xâu theo trình tự từ trái sang phải, sẽ xuất hiện 1 trong các trường hợp sau:

- Nếu gặp một cặp có mã khác nhau thì xâu chứa kí tự có mã nhỏ hơn là xâu nhỏ hơn.
- Nếu tất cả các cặp kí tự đều có mã giống nhau thì 2 xâu bằng nhau.

- Nếu 2 xâu có độ dài khác nhau song các cặp kí tự đều có mã giống nhau đến độ dài của xâu ngắn thì xâu ngắn sẽ là xâu nhỏ hơn.

Kết quả của phép so sánh là giá trị logic True hoặc False.

'hang' < 'hun' cho giá trị True.

'Thoa' = 'Thoa' cho giá trị True.

'nhu' > 'nhung' cho giá trị Flase.

- Đọc xâu kí tự từ bàn phím: Readln(St); Độ dài thực của xâu St là số kí tự gõ vào từ bàn phím. Nếu không gõ kí tự nào mà gõ Enter luôn thì xâu St rỗng.
- Viết xâu kí tự ra màn hình: Write(St) và Writeln(St)

2.3. Các thủ tục và các hàm chuẩn xử lý xâu kí tự

Giả thiết St là xâu kí tự có độ dài chuẩn là 255 kí tự.

a. Hàm Length(St)

Hàm này cho độ dài của xâu kí tự.

St:='Tin học' ; Length(St) có giá trị là 7.

b. Thủ tục Delete(St,m,n)

Thủ tục này xoá đi n kí tự, bắt đầu tự vị trí m trong xâu St.

St:='Cong trinh khoa hoc';

Delete(St,6,5);

Kết quả St sẽ còn 'Cong khoa hoc' .

Nếu m+n > Length(St) thì chỉ xoá đi những kí tự nằm trong Length(St).

c. Thủ tục Insert (s1,St,m)

Thủ tục này chèn xâu s1 vào xâu St tại vị trí m.

St:='Hoc tot';

s1:='Toan ';

Insert(s1,St,5);

Kết quả St sẽ là 'Hoc Toan tot'.

Nếu Length(s1)+Length(St) > độ dài cực đại cho phép của xâu thì chỉ những kí tự nào nằm trong khoảng độ dài cực đại cho phép mới được giữ lại.

d. Hàm Copy(St,m,n)

Hàm này sao chép n kí tự của xâu St từ vị trí m.

St:='Lao dong';

s1:=Copy(St,5,2); Kết quả sẽ cho s1='do'.

Nếu m > Length(St) thì Copy sẽ cho 1 xâu rỗng.

Nếu m + n > Length(St) thì Copy chỉ nhận các kí tự nằm trong xâu St.

e. Hàm Concat(St1,St2,...,Stn)

Hàm này ghép nối tất cả các xâu kí tự St1,St2,...,Stn thành một xâu kí tự theo thứ tự đã viết.

Tổng số chiều dài của các xâu kí tự không được lớn hơn 255.

St1:='Lao dong ';

St2:='Hoc tap ';

St3:='Vui choi';

St:=Concat(St1,St1,St3); Kết quả St là 'Lao dong Hoc tap Vui choi'.

f. Hàm Pos(s1,St)

Hàm này cho vị trí đầu tiên của xâu s1 trong xâu St. Nếu không tìm thấy thì hàm cho giá trị là 0.

St:='12345abc';

s1:='345';
s2:='35ab';
Pos(s1,St) cho giá trị là 3.
Pos(s2,St) cho giá trị là 0.

g. Thủ tục Str(x,St)

Thủ tục này biến đổi giá trị số nguyên hoặc thực x thành một xâu kí tự St biểu diễn số đó. Cách biểu diễn của xâu St sẽ được qui định do qui cách của x.

I:=1234;
Str(I:5,St) sẽ cho St =' 1234'
x:=1234.56789111;
Str(x:9:4,St) sẽ cho St='1234.5678'

h. Thủ tục Var(St,x,m)

Thủ tục này biến đổi xâu kí tự St (biểu diễn số nguyên hoặc thực) thành một số nguyên hoặc thực chứa trong x. Biến m là biến nguyên chứa mã lỗi, nếu biến đổi đúng thì m=0, nếu biến đổi sai thì m cho giá trị là vị trí của kí tự sai.

Ví dụ 1: St:='1234.567'; x là biến thực, m là biến nguyên.
Var(St,x,m); cho ta x=1234.567 và m=0.
Ví dụ 2: St:='1234'; x là biến nguyên, m là biến nguyên.
Var(St,x,m); cho ta x=1234 và m=0
Ví dụ 3: St:='123ab'; x là biến thực, m là biến nguyên.
Var(St,x,m); kết quả sai và m≠0.

2.4. Các chương trình

Bài toán 1: Nhập 1 xâu kí tự từ bàn phím, kiểm tra xem xâu có đối xứng không.

Xâu kí tự S có độ dài n, là xâu đối xứng khi có tất cả các cặp kí tự S[i] = S[n-i+1] với i từ 1 đến (n Div 2). Chẳng hạn xâu '1234321' và xâu 'abccba' là các xâu đối xứng, xâu '12343321' là xâu không đối xứng.

Chương trình:

```
Program KT_xau_doi_xung;
Uses crt;
Var s: string; i,n : integer; t: boolean;
Begin
  clrscr;
  Witeln('Nhap xau ki tu'); readln(s);
  n:=length(s);
  t:=true;
  for i:=1 to  n div 2  do
       if  s[i] <> s[n-i+1] then t:=false;
  if  t  then writeln(' Xau ki tu doi xung')
      else writeln(' Xau khong doi xung');
  readln;
end.
```

Bài toán 2: Nhập vào 2 xâu kí tự có độ dài như nhau. Xây dựng xâu mới chứa các kí tự xen kẽ của 2 xâu nhập vào theo thứ tự từ trái sang phải,. Chẳng hạn s1='123' s2='abc' thì các xâu mới xây dựng là s3='1a2b3c', s4='a1b2c3'

Chương trình:

```
Program Ghep_xen_ke_xau_ki_tu;
Uses crt;
Var s1,s2,s3,s4: string; i,n : integer;
Begin
  clrscr;
  Witeln('Nhap xau ki tu mot s1: ‘); readln(s1);
  Witeln('Nhap xau ki tu hai s2: ‘); readln(s2);
  s3:=’’;
  s4:=’’;
  n:=length(s1);
  for i:=1 to  n   do
      begin
           s3:=s3+s1[i]+s2[i];
           s4:=s4+s2[i]+s1[i];
      end;
  Writeln(‘ Cac xau ki tu moi tao’);
  Writeln(s3);
  Writeln(s4);
  readln;
end.
```

Bài toán 3: Chuẩn hoá văn bản. Nôi dung chuẩn hoá văn bản như sau: Nhập vào các dòng văn bản từ bàn phìm, chuẩn hoá theo các qui định :

- Các từ chỉ cách nhau 1 khoảng cách.
- Sau dấu chấm phải viết hoa.
- Trước các dâu . , ; : không có khoảng cách.
- Sau các dâu . , ; : có 1 khoảng cách.

Chương trình

```
{ xuly tep van ban ve cac dau .,;: }
PROGRAM XU_LY_VAN_BAN;
USES CRT;
VAR S,T:STRING;TIEP:CHAR;
PROCEDURE SUA(VAR P:STRING);
 VAR I:INTEGER;
 BEGIN
   { xoa khoang cach thua }
   I:=1;
   WHILE I<LENGTH(P)-1 DO
     IF (P[I]=' ') AND (P[I+1]=' ') THEN DELETE(P,I+1,1) ELSE I:=I+1;
  { xoa khoang cach truoc cac dau , ; : . }
   I:=1;
   WHILE I<LENGTH(P) DO
     IF (P[I]=' ') AND ((P[I+1]=',') OR (P[I+1]=';') OR (P[I+1]=':') OR (P[I+1]='.'))
       THEN DELETE(P,I,1)  ELSE I:=I+1;
    { chen dau khoang trong sau cac dau neu thieu }
    I:=1;
    WHILE I< LENGTH(P)-1 DO
     IF ((P[I]=',') OR (P[I]=';') OR (P[I]=':') OR (P[I]='.')) AND (P[I+1]<>' ')
```

```
      THEN BEGIN INSERT(' ',P,I+1);
                 I:=I+1;
             END
     ELSE I:=I+1;
   { Viet hoa sau dau . }
    I:=1;
  WHILE I<LENGTH(P)-2 DO
    IF P[I]='.' THEN BEGIN P[I+2]:=UPCASE(P[I+2]);
                           I:=I+1;
                     END
    ELSE I:=I+1;

 END;
 { Than chuong trinh chinh}
BEGIN
 CLRSCR;
 TIEP:='C';
 WHILE UPCASE(TIEP)='C' DO
  BEGIN
       WRITELN('NHAP XAU KI TU ');
       READLN(S);
       SUA(S);
       WRITELN(S);
       WRITE('CO TIEP TUC KHONG(C/K) ');
       READLN(TIEP);
  END;
 READLN;
END.
```

3. Kiểu tập

3.1.Khái niệm tập

Tập là một bộ các đối tượng vô hướng và cùng kiểu.

Mỗi đối tượng gọi là một phần tử của Tập. Tập có tối đa là 256 phần tử. Nếu phần tử là kiểu số thì chỉ cho phép là các số nguyên có giá trị từ 0..255.

Khái niệm Tập gắn liền với khái niệm Tập hợp trong trong toán học.

Ví dụ: Tập các chữ cái hoa, tập này có 26 phần tử.

Tập các số nguyên dương có 2 chữ số, tập này có 90 phần tử.

3.2. Khai báo tập

Khai báo Tập dùng cụm từ : SET of Kiểu_phần_tử;

Kiểu phần tử phải là một kiểu vô hướng.

- ***Dùng khai báo kiểu***

```
Type   Tên_kiểu=Set  of  kiểu _phần_tử;
 Var    Tên_biến: Tên _kiểu;
Ví dụ:   Type  t=set of  1..200;
```

```
        Chu_hoa: Set of  'A'..'Z';
    Var  tuoi : t;
         Chu_in: chu_hoa;
```

- ***Khai báo trực tiếp trong khai báo biến***

```
Var     Tên_biến: Set of Kiểu_phần_tử;
Ví dụ:   Type  mau=(do,xanh,vang,tin,nau);
               hinh=(tam_giac,tron,chu_nhat);
         Var   mau_ao: Set of  mau;
               hinh_ve: set of  hinh;
               chu_so: 1..9;
```

3.3- Xây dựng một tập

Xây dựng một tập bằng cách liệt kê các phần tử của tập, chúng được cách nhau bởi dấu phẩy và được viết trong dấu móc vuông ([ ]) .

Tập rỗng là tập không chứa một phần tử nào, được viết là []

[1..99] ; tập này có 100 phần tử .

[2,4,6,8..12]; tập có các phần tử là 2,4,6,8,9,10,11,12.

['a'..'d',' m','n'] ; tập này có các phần tử 'a', 'b', 'c', 'd', 'm', 'n'.

[xanh,do,tim,vang];

Các phần tử của tập cũng có thể cho bằng biến hoặc biểu thức.

[2*i+j, i*j-2] ; nếu i=2 và j=3 thì tập có các phần tử là 7, 4.

Ta được dùng kiểu khoảng con để chỉ ra các phần tử của tập.

3.4- Các phép toán trên tập

a. Phép gán

Gán một tập cho biến kiểu tập.

Ví dụ

```
   var  chu: Set of  'A' .. 'Z';
        tuoi: Set of  1..200;
```

Khi đó có thể thực hiện các phép gán sau:

```
        chu:=['A', 'C'', 'M' . .'P'];
        tuoi:=[50..80, 90,100];
```

Tập rỗng có thể đem gán cho mọi biến kiểu tập khác nhau.

```
        Chu:=[];
        tuoi:=[];
```

Không thể gán các tập kiểu cơ bản không tương thích.

Chẳng hạn nếu gán chu:=[1..10] là sai.

b. Phép hợp

Phép hợp được kí hiệu bằng dấu +

Hợp của 2 tập là một tập có các phần tử thuộc hai tập.

Ví dụ A:=[1,3,5..10];

```
       B:=[2,4,6,8];
       C:=[1,5,9];
       D:=A+B;   tập D sẽ là [1..10]
       E:=B+C;   tập  E sẽ là [1,2,4,5,6,8,9]
```

c. Phép giao

Phép giao được kí hiệu bằng dấu *

Giao của 2 tập là 1 tập có các phần tử là các phần tử chung của cả 2 tập.

M:=A*B; tập M sẽ là [6,8]

N:=B*C; tạp N=[]

d. Phép hiệu

Phép hiệu được kí hiệu bằng dấu -.

Hiệu của 2 tập là 1 tập chứa các phần tử thuộc tập thứ nhất nhưng không thuộc tập thứ 2.

G:=A - B ; tập G sẽ là [1,3,5,7,9,10]

H:=C - A; tập H sẽ là tập rỗng.

e. Phép thuộc về

Phép thuộc về kí hiệu là IN

Phép thuộc về cho biết 1 phần tử hay 1 tập có thuộc về 1 tập khác không ?, kết quả của phép thuộc về là giá trị kiểu boolean (True hoặc False).

1 In A; kết quả là true.

B in A; kết qủa là False.

C in A; kết quả là True;

Phép so sánh =, <>, <=, >=

Hai tập đem so sánh phải cùng kiểu.

Kết quả của phép so sánh là giá trị kiểu boolean.

- Hai tập bằng nhau (=) nếu chúng có các phần tử như nhau từng đôi một.

Ví dụ: x:=['a' . .'d'];

y:=['a' .. 'd', 'e', 'f'];

z:=['a', 'b', 'c', 'd']

x=y; kết quả là False.

x=z; kết quả là True.

- Hai tập là khác nhau (<>) nếu chúng có ít nhất 1 cặp phần tử khác nhau.

x<> y ; kết quả là True;

x<> z ; kết quả là False.

- Tập thứ nhất <= tập thứ 2 nếu tất cả các phần tử của tập thứ nhất đều thuộc tập thứ 2.

x <= y; kết quả là true.

y <= x; kết quả là False.

- Tập thứ nhất >= tập thứ 2 nếu mọi phần tử của tập thứ 2 đều thuộc tập thứ 1.

y >= x; kết quả cho True.

z >= y; kết quả là False.

3.5- Các chương trình về tập

Bài toán : Tạo một tập chứa các số có tối đa là 2 chữ số nguyên dương< 256 chia hết cho 7, in tập đó ra. Sau đó nhập 1 số bất kỳ kiểm tra xem nó có thuộc vào tập đó không.

Chương trình

```
Program thao_tac_tap;
uses  crt;
var  a: set of  1..99; i,n: byte; lap: char;
Begin
  clrscr;
```

```
    { Tao tap cac so chia het cho 7 }
    a:=[];
    for  i:=7  to  255  do
      if  (i mod 7) = 0  then  a:=a+[i];
    { In tập }
    Writeln(' Tap cac so chia het cho 7');
    for  i:=7  to  255  do
      if  i  in  a  then  writeln(i);
    { Kiem tra so nhap thuoc tap chi het cho 7}
     Repeat
       write(' Nhap so bat ky ');
       Realn(n);
if i in a then writeln('Chia het cho 7') else witeln('Khong chia het cho 7');
       Write(' Co tiep tuc khong (C/K) ? '); readln(lap);
      Until  upcase(lap)= 'K';
  readln;
end.
```

Bài tập chương 4

Hãy viết chương trình cho các bài toán sau:

1. Cho dãy số sau: $a_1,a_2,....,a_n$. Viết chương trình tìm phần tử lớn nhất, phần tử nhỏ nhất của dãy số đó.

2. Cho dãy số sau: $a_1,a_2,....,a_n$. Viết chương trình sắp xếp dãy theo thứ tự tăng dần .

3. Cho dãy số sau: $a_1,a_2,....,a_n$. Viết chương trình đếm số phần tử dương và xoá đi phần tử thứ m trong dãy (m<=n) .

4. Cho dãy số sau: $a_1,a_2,....,a_n$. Viết chương trình tìm các phần tử có giá trị là x nhập vào từ bàn phím.

5. Cho dãy số sau: $a_1,a_2,....,a_n$. Viết chương trình thêm phần tử có giá trị là x, vào vị trí m trong dãy. Sau đó tính tổng các phần tử của dãy mới.

6. Cho ma trận có m dòng và n cột, các phần tử là nguyên. Tìm phần tử nhỏ nhất của ma trận.

7. Cho ma trận có m dòng và n cột, các phần tử là nguyên. Tính tổng và trung bình cộng các phần tử của ma trận.

8. Viết chương trình nhập vào một xâu ký tự. Hãy xây dựng xâu chứa các ký tự đảo của xâu đó, đếm xem có bao nhiêu ký tự 'a ' hoặc ‘A’ trong xâu.

9. Viết chương trình nhập vào một xâu ký tự. Hãy kiểm tra xem xâu đó có đối xứng không. In kí tự đầu và kí tự cuối của xâu đó.

10. Viết chương trình nhập vào hai xâu ký tự có độ dài bằng nhau. Hãy xây dựng xâu chứa các ký tự xen kẽ của hai xâu đó, theo thứ tự một kí tự của xâu1 rồi đến 1 kí tự của xâu 2.

CHƯƠNG V
CHƯƠNG TRÌNH CON: HÀM VÀ THỦ TỤC

1 - Cấu trúc của hàm và thủ tục

1.1- Chương trình con

Khi lập trình gặp đoạn chương trình cần dùng nhiều lần, để tránh viết lặp lại thì đoạn chương trình đó được tổ chức thành chương trình con và mỗi khi cần gọi tới chương trình con đó.

Mặt khác khi xây dựng chương trình cho các bài toán lớn, phức tạp, để dễ cho việc thiết kế chương trình, hiệu chỉnh chương trình, gỡ rối khi chạy chương trình, bài toán lớn được phân thành các phần nhỏ, mỗi phần xây dựng thành các chương trình con. Chương trình chính sẽ gọi tới các chương trình con .

Trong Pascal có 2 loại chương trình con đó là hàm (Function) và thủ tục (Procedure).

a - Cấu trúc của hàm (Function) và lời gọi hàm

- **Hàm có cấu trúc đầy đủ như sau:**

```
Function  Tên_hàm(Tham_số1 : kiểu; Tham_số2: kiểu; Var tham_số3: kiểu;. . .): kiểu;
  Label  {Khai báo các nhãn }
  Const   { Khai báo các hằng }
  Type     {Định nghĩa các kiểu dữ liệu  của người sử dụng }
  Var     { Khai báo các biến cục bộ}
     . . .
  Begin
    . . .  { Thân chương trình con }
         Tên_hàm:= Giá_trị ;
  End;
```

Các phần nếu có thì theo đúng thứ tự đã nêu.

Kiểu của tham số là các kiểu cơ bản, kiểu có cấu trúc như kiểu xâu kí tự và kiểu mang, nếu là kiểu mảng thì phải khai báo bằng định nghĩa kiểu ở phần định nghĩa khiểu ở đầu chương trình chính, không được khai báo trực tiếp.

Kiểu của hàm có thể là các kiểu cơ bản, kiểu xâu kí tự.

Các tham số khai báo trong hàm được gọi là tham số hình thức.

- ***Lời gọi hàm***

Trong thân chương trình chính sử dụng hàm phải có lời gọi hàm.

Lời gọi hàm được viết như sau:

Tên_hàm(danh sách các tham số thực sự)

Các tham số thực sự tương ứng cả về số lượng và cả về kiểu dữ liệu với các tham số hình thức khai báo trong hàm.

Lời gọi hàm được coi như 1 biến, có thể tham gia vào biểu thức, tham gia vào các thủ tục vào/ ra.

Ví dụ1: Chương trình có xây dựng Function

Bài toán : Tính diện tích của tam giác biết 3 cạnh a,b,c.

Chương trình

```
Program   Tinh_dien_tich;
  uses   crt;
  var  a,b,c : real;
  Function   DT(x,y,z : real) : real;
    var  s,p : real;
     begin
         p:=(x+y+z)/2;
         s:=sqrt(p*(p-a)*(p-b)*(p-c));
         DT:=s;
     end;
  begin  { than chuong trinh chinh }
    clrscr;
    a:=2;b:=3;c:=2;
    writeln(' dien tich tam giac 1 = ', DT(a,b,c):8:2);
    write('Nhap 3 canh của tam giac a,b,c:  ');
    readln(a,b,c);
    writeln(' dien tich tam giac 2 = ', DT(a,b,c):8:2);
    readln;
 end.
```

- Hàm đặt ở vị trí sau khai báo biến trong chương trình chính và trước thân chương trình chính.
- Chỉ được gọi tới hàm sau khi đã khai báo hàm.
- Ơ ví dụ 1 ta khai báo hàm có tên là DT có kiểu real, các tham số hình thức của hàm là: x,y,z. Trong thân chương trình có 2 lời gọi hàm, chúng đều nằm trong lệnh Writeln. Trong lời gọi hàm thứ nhất, 3 tham số thực sự là a=2, b=3, c=2. Trong lời gọi hàm thứ hai, 3 tham số thực sự là a,b,c có giá trị được nhập vào từ bàn phím.

b - Cấu trúc của thủ tục (Procedure) và lời gọi thủ tục

- **Thủ tục có cấu trúc đầy đủ như sau:**

```
Procedure   Tên_thủ_tục(Tham_số1 : kiểu; Tham_số2: kiểu; Var tham_số3: kiểu;. . .);
  Label  {Khai báo các nhãn }
  Const   { Khai báo các hằng }
  Type     { Định nghĩa các kiểu dữ liệu  của người sử dụng }
  Var     { Khai báo các biến cục bộ}
   Begin
     . . .   { thân chương trình con }
   End;
```

Các phần nếu có thì theo đúng thứ tự đã nêu.

Kiểu của tham số là các kiểu cơ bản, kiểu có cấu trúc như kiểu xâu kí tự và kiểu mang, nếu là kiểu mảng thì phải khai báo bằng định nghĩa kiểu ở phần định nghĩa khiểu ở đầu chương trình chính, không được khai báo trực tiếp.

Trong chương trình chính thủ tục đứng trước thân chương trình chính, sau khai báo biến.

Các tham số khai báo trong hàm được gọi là tham số hình thức.

- **Lời gọi thủ tục**

Trong thân chương trình chính sử dụng thủ tục phải có lời gọi thủ tục

Lời gọi thủ tục được viết như sau:

```
Tên_thủ_tục( danh sách các tham số thực sự);
```

Các tham số thực sự tương ứng cả về kiểu và số lượng với các tham số hình thức khai báo trong thủ tục.

Lời gọi thủ tục như 1 câu lệnh đứng độc lập.

Ví dụ 2: chương trình có xây dựng Procedure

Bài toán: Tính tổng và trung bình cộng của dãy số $a_1, a_2, \ldots, a_n$.

Chương trình

```
Program Tinh_tong_tb;
 uses  crt;
 type    mang= array[ 1 .. 50 ]  of  real ;
 var   i,n: integer; a: mang;  tg,tb: real;
 Procedure    tong(m: integer; x: mang; var  s, p : real);
   Var  j: integer ; t: real;
    begin
       t:=0;
       For  j:=1 to  m  do    t:=t + x[j] ;
       s:= t;
       p:=t/m;
    end;
begin  { than chuong trinh chinh }
  clrscr;
   write(' nhap so phan tu cua day n ');
   readln(n);
   for  i:= 1  to  n  do
     begin
       write(a[', i, ']=' ); readln(a[i]);
     end;
   tong(n,a,tg,tb);
   writeln(' tong= ', tg: 8: 2, 'trung binh = ', tb: 8: 2 );
   readln;
end.
```

1.2 - Sự khác nhau giữa hàm và thủ tục

- Hàm cho 1 giá trị thông qua tên hàm. Tên hàm trong lời gọi hàm được coi như một biến có thể tham gia vào biểu thức, các thủ tục vào ra. Cuối thân hàm phải có lệnh gán giá trị cho tên hàm.
- Tên thủ tục không cho 1 giá trị nào cả.
- Các tham số viết sau tên hàm, sau tên thủ tục được gọi là tham số hình thức. Tham số hình thức có 2 loại: + Tham số không có từ khoá Var đi kèm trước gọi là tham trị.
  + Tham số có từ khóa Var đi kèm trước gọi là tham biến.
- Trong hàm thường chứa các tham trị, ít khi chứa các tham biến. Trong thủ tục thường có các tham biến.
- Các tham số trong lời gọi hàm, lời gọi thủ tục gọi là tham số thực sự. Các tham số thực sự phải tương ứng về số lượng và kiểu với các tham số hình thức. Các tham số thực sự tương ứng vơi các tham trị để chứa các dữ liệu vào. Các tham số thực sự tương ứng vơi các tham biến để chứa kết quả của thủ tục.
- Hàm lấy kết quả ở tên hàm, thủ tục lấy kết quả ở các tham số thực sự tương ứng với các tham biến.

- Như vậy nếu để lấy 1 kết quả ta có thể tổ chức hàm hoặc thủ tục. Nếu muốn lấy nhiều hơn 1 kết quả thì phải tổ chức chương trình con dạng thủ tục.

2 - Biến toàn cục, biến cục bộ và truyền dữ liệu

2.1 - Biến toàn cục

Biến toàn cục là biến khai báo ở đầu chương trình chính.
Biến toàn cục tồn tại suốt thời gian làm việc của chương trình .
Biến toàn cục có thể sử dụng cả trong chương trình chính và chương trình con.
Ví dụ 1 mục 1 (tính diện tích tam giac) có a,b,c là biến toàn cục.
Ví dụ 2 mục 1 (tính tổng và trung bình) có i, n, a, tg, tb là biến toàn cục.

2.2 - Biến cục bộ

Biến cục bộ là các biến được khai báo ở đầu chương trình con.
Biến cục bộ được cấp phát bộ nhớ khi chương trình con được gọi tới và bị xoá khi ra khỏi chương trình con.
Biến cục bộ chỉ được dùng trong chương trình con.
Biến toàn cục và biến cục bộ có thể trùng tên nhau nhưng chương trình vẫn phân biệt 2 biến khác nhau.
Trong ví dụ 1 mục 1 (tính diện tích tam giác) có s, p là biến cục bộ.
Trong ví dụ 2 muc 2 (tính tổng và trung bình) có j, t là biến cục bộ.

2.3 - Truyền dữ liệu

Khi gặp lời gọi chương trình con máy sẽ thực hiện các bước sau:
- Cấp phát bộ nhớ cho các tham số và cho các biến cục bộ trong chương trình con.
- Truyền giá trị của các tham số thực sự cho tham trị và truyền địa chỉ cho các tham biến.
- Thực hiện các lệnh trong thân chương trình con.
- Thực hiện xong chương trình con máy giải phóng các tham số và các biến cục bộ, rồi trở về chương trinh chính.

3 - Tính đệ quy của chương trình con

Trong Function và Procedure có thể có lời gọi tới chính nó. Tính chất này được gọi là tính đệ qui.
Phương pháp đệ qui được áp dụng cho các bài toán thuật giải mang tính đệ qui.
Thuật giải đệ qui làm cho chương trình ngắn gọn, đẹp đẽ nhưng lại tốn thờ gian tính toán và bộ nhớ. Có những bài toán chỉ có thể giải quyết được bằng xây dựng các chương trình con đệ quy.

Ví dụ1: Bài toán tính giai thừa
- Trường hợp suy biến: n! = 1 khi n=0
- Trường hợp tổng quát: n! = (n-1)! . n khi n >= 1

Có thể xây dựng hàm Giaithua có tính chất đệ qui như sau:

```
Function   Giaithua( n: longint): longint ;
      begin
```

```
    if   n=0  then   Giaithua:= 1
       else   Giaithua:= Giaithua(n-1) * n ;
end;
```

* Muốn xây dựng được chương trình con đệ qui ta phải xác định được 2 trường hợp:
- Trường hợp suy biến , đó là trường hợp đặc biệt mà xác định được giá trị của hàm.
- Trường hợp tổng quát lần thứ n được tính dựa vào lần thứ (n-1).

Ví dụ 2: Tìm ước số chung lớn nhất của 2 số x và y có thể được định nghĩa như sau (x>y):
USCLN(x,y)= x nếu y=0 (đây là trường hợp suy biến)
USCLN(x,y)= USCLN(y, phần dư của x/y) nếu y<>0 (đây là trường hợp tổng quát).
Trong hàm xây dưng với x>y, nếu y>x thì chương trình tráo đổi giá trị giữ x và y.

Hàm USCLN được viết như sau:

```
Function   USCLN(x,y: integer) : integer ;
    Var t:Integer;
   Begin
      If y>x then begin t:=x; x:=y; y:=t; end;
      if  y=0  then   USCLN:= x
      else  USCLN := USCLN(y, x mod y) ;
    end;
```

4. Một số chương trình con của turbo pascal

Pascal đã xây dựng sẵn một số hàm và thủ thục, Ta có thể gọi tới các hàm, thủ tục đó theo đúng quy cách của nó để sử dụng.

Ngoài các hàn và thủ tục đã xét ở các phần trên, trong phần này bổ sung thêm một số thủ tục sau đây:

*** Procedure GotoXY(Xpos, YPos);**

Đưa con trỏ(cursor) của màn hình về vị trí có toạ độ Xpos và Ypos trên màn hình. Xpos, Ypos kiểu số nguyên.

*** Procedure ClrScr;**

Xoá toàn bộ màn hình và đặt con trỏ vào vị trí phía trên, bên trái.

*** Procedure ClrEof;**

Xoá toàn bộ các kí tự bên phải con trỏ màn hình. Sau khi xoá con trỏ vẫn ở tại chỗ.

*** Procedure Deline;**

Xoá toàn bộ dòng màn hình chứa con trỏ, sau đó dồn các dòng ở dưới lên.

*** Procedure InsLine;**

Xen một dòng trắng vào màn hình từ vị trí con trỏ.

*** Procedure LowVideo và NormVideo;**

Sau khi gọi **LowVideo** mọi kí tự viết ra màn hình đều có độ sáng yếu đi cho tới khi gọi thủ tục **NormVideo** (Normal Video).

*** Procedure Delay(Time);**

Tạo ra thời gian trễ Time (khoảng ms). Time là một số nguyên. Delay thường được dùng để làm chậm chương trình lại cho ta quan sát, khảo sát...

*** Procedure Sound(F) và NoSound;**

Tạo ra dao động âm thanh với tần số là F (F: số nguyên) cho đến khi ta gọi **NoSound;**

Bài tập chương 5

1. Viết chương trình tính tổ hợp chập m của n phần tử: C^m_n
   Chương trình viết có chương trình con.
2. Viết chương trình tính $P_n(x)=(. . . ((a_n*x+a_{n-1})*x+a_{n-2})*x+. . . +a_1)*x+a_0$
   Chương trình có chương trình con.
3. Cho dãy số sau: $a_1,a_2,....,a_n$. Viết chương trình tính tổng, trung bình cộng các phần tử của dãy số đó. Chương trình viết có chương trình con.
4. Lập chương trình tính diện tích và chu vi của các hình: Tam giác biết 3 cạnh a,b,c, hình chữ nhật biết hai cạnh a,b, hình tròn biết bán kính. Chọn hình để tính thông qua câu hỏi ' Ban tính cho hình gì TG=1, CN=2, TR =3 '. Chương trình viết có sử dụng chương trình con.
5. Cho hai số nguyên x1 và x2, lập chương trình nhập x1 và x2 từ bàn phím, sử dụng tính đệ quy của chương trình con để tìm ước số chung lớn nhất của x1 và x2.

CHƯƠNG VI

KIỂU DỮ LIỆU CÓ CẤU TRUC: KIỂU BẢN GHI VÀ KIỂU TỆP

1 - Kiểu bản ghi (record)

1.1 - Khái niệm

Kiểu bản ghi là một kiểu dữ liệu có cấu trúc gồm một số cố định các phần tử có kiểu khác nhau.

Kiểu bản ghi dùng để mô tả các dữ liệu có nhiều thành phần khác kiểu liên kết với nhau như dữ liệu của các bảng, các cột của bảng là các thành phần, mỗi cột có một kiểu dữ liệu khác nhau, các cột liên kết với nhau để biểu diễn một nội dung nhất định.

Ví dụ1: Bảng lương bao gồm các cột: Số thứ tự, Họ và tên, Ngày sinh, Hệ số, Lương, Bảo hiểm xã hội, Tổng lĩnh. Mỗi dòng của bảng lương thuộc kiểu dữ liệu bản ghi. Các cột là các thành phần còn được gọi là các phần tử.

1.2 - Khai báo kiểu dữ liệu bản ghi

Kiểu dữ liệu bản ghi có các phần tử liên kết với nhau. Phần tử được gọi là trường, mỗi trường có một tên, tên trường giống như tên biến. Mỗi trường thuộc một kiểu dữ liệu nào đó.

Khi báo kiểu dữ liệu bản ghi được viết trong cụm từ record ... end;

Trong cụm từ là danh sách tên các trường kèm theo sau là kiểu dữ liệu của nó.

Khai báo kiểu bản ghi như sau:

```
Type   Tên_kiểu= Record
                    Tên_trường1 : kiểu;
                    Tên_trường2 : kiểu;
                          . . .
                    Tên_trườngN : kiểu;
                End;
```

Ví dụ 1: Bảng lương trong ví dụ ở mục 1 được khai báo như sau:

```
Type  bang_luong = record
            Stt : Integer;
            Hoten : String[25];
            Ns : String[10];
            Heso,Luong,Bhxh,Tong : Real;
      End;
 Var  luong1,luong2:bang_luong;
```

Ví dụ 2: Danh sách khách hàng bao gồm các dữ liệu như họ và tên, số nhà, phố, quận, Thành phố, số điện thoại.

```
Type   khach_hang = record
            hoten : string[25];
            sonha : string[20];
            Pho, quan, thanhpho : string[30];
            tel : longint;
end;
 Var    Bangkh1,bangkh2: khach_hang;
```

1.3 - Sử dụng bản ghi

- Một trường của bản ghi coi như 1 biến, được sử dụng trong các biểu thức và các thủ tục vào ra.
- Một trường của bản ghi được chỉ định bằng cách viết sau:
  Tên_biến.Tên_trường

Ví dụ
```
luong1.stt:=1;
luong1.hoten:='Le Thu Ha';
luong1.luong:=luong1.heso * 210000;
bangkh1.pho:='Hang Dao';
bangkh1.tel:=8573980;
Readln(luong2.hoten);
Writeln(banghk2.pho);
```

Biến bản ghi không được tham gia vào các biểu thức, các thủ tục vào ra, các toán tử logic, các toán tử quan hệ >, >=, <, <= .

Nếu hai biến bản ghi có cùng kiểu thì có thể tham gia các phép so sánh = (bằng) và <> (khác).

Nếu hai biến bản ghi có cùng kiểu thì có thể gán giá trị cho nhau.

```
luong1:=luong2;
banghk1:=banghk2;
```

1.4 - Câu lệnh With ... do

Khi chỉ định một trường của bản ghi ta phải viết cả tên biến và tên trường, do vậy viết các tên sẽ dài. Để không phải viết tên biến, chỉ viết tên trường ta dùng câu lệnh With ... do như sau:

```
With Tên_biến do Chuỗi_lệnh;
```

Trong chuỗi lệnh nếu chỉ ra một trường nào đó của biến đã nêu tên thì không phải viết tên biến kèm theo.

Ví dụ thay cho cách viết ở ví dụ phần 3 ta có thể sử dụng câu lệnh With...do

```
With luong1 do
  begin
  stt:=1;
  hoten:='Le Thu Ha';
  luong:=heso * 210000;
end;

With banghk1 do
  begin
    pho:='Hang Dao';
    tel:=8573980;
  end;

With luong2 do Readln(hoten);
With bangkh2 do Writeln(pho);
```

1.5 - Các chương trình sử dụng bản ghi

Bài toán 1: Bài toán thống kê hàng nhập. Mỗi mặt hàng có các dữ liệu như tên hàng, ngày nhập, số lượng, đơn giá. Hãy nhập dữ liệu vào và in ra một bảng bao gồm các cột : tên hàng,

ngày nhập, số lượng, đơn giá, tiền của tất cả các mặt hàng đã nhập. Sau cùng in ra tổng số tiền đã nhập.

```
  Chương trình
  Program Thong_ke_hang;
  uses  crt;
  Type  hang=record
          ten: string[20];
          Ngay: string[10];
          sl,gia,tien : real;
     end;
  Var  bang:array[1..30] of  hang; i,n: integer; tong: real;
begin
    clrscr;
    Write)'Nhap so luong mat hang n ? ');
    readln(n);
    tong:=0;
    for i:=1  to n do
      with  bang[i]  do
        begin
          Write(' Tên hang '); readln(ten);
          Write(' Ngay nhap '); readln(ngay);
          Write(' So luong '); readln(sl);
          Write(' Gia  '); readln(gia);
           tien:=sl * gia;

         tong:=tong+tien;
      end;
   Writeln(' Bang thong ke hang nhap');
Writeln('| Ten hang':20,'| Ngay nhap':12,'| So luong':12,'| Don gia':12,'| Tien':12 );
   for  i:= 1 to n do
    with bang[i]  do
      Writeln(ten:20, ngay:12, sl:12:2, gia:12:2, tien:12:2);
   writeln;
   writeln('Tong so tien la: ', tong: 15:2);
   readln;
end.
```

2 - Kiểu tệp (File)

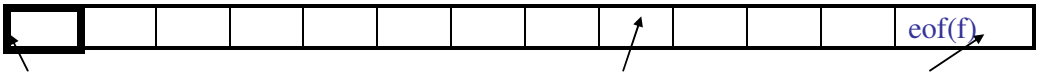
2.1 - Khái niệm tệp

Tệp dữ liệu là tập hợp các dữ liệu có liên quan với nhau và được nhóm lại với nhau tạo thành một dãy, được lưu trữ trên bộ nhớ ngoài ví dụ như đĩa từ.

Các phần tử của tệp cùng kiểu, được lưu trữ kế tiếp nhau, khi làm việc với các phần tử của tệp có con trỏ tệp. Khi mới mở tệp con trỏ tệp trỏ vào phần tử đầu tiên của tệp. Cuối tệp có dấu kết thuc tệp kí hiệu là eof(tệp).

Các phần tử của tệp f như sau:

Mỗi ô là một phần tử của tệp. Cuối tệp là dấu kết thúc tệp eof(f) (end of file)



Con trỏ tệp (cửa sổ tệp) chỉ vào phần tử đầu Phần tử i Kết thúc tệp

- Có thể có các loại tệp sau:
  - Tệp định kiểu.
  - Tệp văn bản (Text)
  - Tệp không định kiểu

Trong các phần sau chỉ xét tệp định kiểu và tệp văn bản.

- Tệp và mảng có những điểm giống và khác nhau sau đây:

* Điểm giống nhau giữa tệp và mảng : tập hợp các phần tử cùng kiểu.
* Điểm khác nhau giữa tệp và mảng : Mảng khai báo với số phần tử xác định, còn tệp với số phần tử không xác định, tệp có thể chứa số phần tử tuỳ ý theo dung lượng trên đĩa.

2.2 - Các cách truy nhập tệp

* Truy nhập tuần tự và truy nhập ngẫu nhiên:

- Truy nhập tuần tự: Việc đọc một phần tử bất kỳ của tệp bắt buộc phải tuần tự đi qua các phần tử trước đấy. Còn muốn thêm một phân tử vào tệp phải thêm vào cuối tệp.

Kiểu truy nhập này đơn giản trong việc tạo tệp, xử lý tệp, song nó kém linh hoạt

- Truy nhập tệp trực tiếp(direct access): Có thể truy nhập vào bất kỳ phần tử nào trong tệp thông qua chỉ số thứ tự của phần tử trong tệp. Tuỳ theo từng bộ nhớ ngoài mà có thể truy nhập trực tiếp được hay không, như đĩa từ có thể truy nhập trực tiếp được, còn băng từ chỉ có thể truy nhập tuần tự không truy nhập trực tiếp được. Như vậy trong truy nhập trực tiếp có thể đọc bất kỳ phần tử nào, thêm phần tử mới thì phải thêm vào cuối tệp.

2.3 - Khai báo tệp định kiểu

Khai báo tệp định kiểu dùng cụm từ sau:

File of kiểu_phần_tử;

- Khai báo kiểu tệp:

Type tên_kiểu = File of kiểu_phần_tử;

- Khai báo biến tệp:

Var tên_biến : File of kiểu_phần_tử;

```
Ví dụ 1    type   t= file  of  integer;
             var    f1,f2 : t;
Ví dụ 2:   type  bang= record
             ten: string[25];
              Ns: string[10];
            Que: string[30];
            luong,bhxh:real;
         end;
  var   f1,f2,f3: file of  bang;
```

2.4 - Tạo tệp để ghi dữ liệu

**** Mở tệp để ghi dữ liệu***

Dùng 2 thủ tục đi liền nhau theo thứ tự như sau:

- Thủ tục Assign

Assign(biến_tệp, tên_tệp);

Thủ tục này gán tên_tệp cho biến_tệp. Tên_tệp theo đúng qui tắc đặt tên trong DOS mà ta đã học ở phần trên.

- Thủ tục Rewrite

```
Rewrite(biến_tệp);
```

Thủ tục này thực hiện việc mở tệp để ghi.

Ví dụ: Mở tệp có tên là 'songuyen.dat' gán cho biến tệp f để ghi dữ liệu ta viết như sau:

```
Assign(f,'songuyen.dat');
Rewrite(f);
```

Sau khi mở tệp xong thì tệp chưa có phần tử nào, tệp rỗng. Con trỏ tệp (cửa sổ tệp) trỏ vào cuối tệp (eof) . Nếu tên_tệp trùng với tệp đã có thì tệp đó sẽ bị xoá.

* ***Ghi dữ liệu vào tệp dùng thủ tục Write***

Write(biến_tệp, biểu_thức1, biểu_thức2, . . ., biểu_thức n);

Các biểu_thức phải có giá trị cùng kiểu với kiểu của tệp. Giá trị của các biểu thức sẽ được ghi vào tệp theo như thứ tự đã viết.

```
Write(f, 2, 4, 6, i*j+3);
```

* ***Đóng tệp bằng thủ tục Close***

```
Close(biến _tệp);
```

* ***Các ví dụ chương trình tạo tệp để ghi dữ liệu***

Bài toán 1: Tạo tệp có tên là 'songuyen.dat' ghi các số nguyên dương <200 mà chia hết cho 3.

Chương trình

```
Program Tao_tep_so_nguyen;
  uses  crt;
  var    i : integer; f : file of integer ;
  Begin
    assign(f, 'songuyen.dat');
    rewrite(f);
    for  i:=3 to 199 do
        if  (i mod 3) = 0  then  write(f, i);
    close(f);
    readln;
  end.
```

Bài toán 2: Tạo tệp 'sach.dat' để ghi các thông tin cho các cuốn sách bao gồm các dữ liệu như tên sách, năm xuất bản, số trang, tác giả.

Chương trình

```
Program  Tao_tep_sach;
  uses  crt;
  Type  tin = record
          ten: string[25];
          nam: integer;
```

```
        trang: longint;
        tacgia: string[25];
      end;
Var  i,n : integer; f: file of tin; nhap: tin;
Begin
  clrscr;
  assign(f, 'sach.dat');
  rewrite(f);
  write(' Nhap so sach n : '); readln(n);
  for  i:= 1  to  n  do
   begin
    with  nhap  do
     begin
        write('Ten sach : ');  readln(ten);
        write('Nam xuat ban : ');  readln(nam);
        write('So trang : ');  readln(trang);
        write('Ten tac gia : ');  readln(tacgia);
     end;
    write(f, nhap);
   end;
 close(f);
end.
```

2.5 - Đọc dữ liệu từ tệp đã có

* ***Mở tệp để đọc***

Mở tệp để đọc dùng 2 thủ tục đi liền nhau theo thứ tự sau:

- Thủ tục Assign

  assign(biến_tệp, tên_tệp);

- Thủ tục Reset

  Reset(biến_tệp) ;

Thủ tục này thực hiện mở tệp để đọc.

Ví dụ1: Mở tệp 'songuyen.dat' gắn với biến tệp f để đọc dữ liệu.

```
assign(f, 'songuyen.dat');
reset(f);
```

Ví dụ 2: Mở tệp 'sach.dat' gắn với biến tệp f1 để đọc dữ liệu.

```
assign(f1, 'sach.dat');
reset(f1);
```

* ***Đọc dữ liệu từ tệp***

Đọc dữ liệu từ tệp được thực hiện bằng thủ tục Read như sau:

Read(biên_tệp, biến1, biến2, . . . , biến n);

Thủ tục này thực hiện đọc giá trị ở vị trí con trỏ gán cho các biến tương ứng như thứ tự đã viết, khi đọc xong con trỏ tệp lại chuyển sang phần tử tiếp theo đọc và gán cho biến khác, cứ thế đọc cho đến biến n .

Việc đọc chỉ được thực hiện khi tệp vẫn còn phần tử, tức là con trỏ chưa tới eof (cuối tệp). do vậy trước khi đọc phải kiểm tra xem đã kết thúc tệp chưa, dùng hàm chuẩn eof như sau:

eof(biến_tệp); hàm này cho giá trị True nếu con trỏ ở cuối tệp, ngược lại hàm cho giá trị False.

Có thể dùng 2 cấu trúc sau:

- Kiểm tra nếu tệp chưa kết thúc thì đọc

```
if not eof(biến_tệp) then read(biến_tệp, biến);
```

- Đọc tất cả các phần tử của tệp

```
While not eof(biến_tệp) do
   Begin
      read(biến_tệp, biến);
      . . .
   end;
```

Nếu con trỏ ở cuối tệp mà vẫn đọc thì máy sẽ báo lỗi, sau đó chương trình dừng lại. Do vậy phải kiểm tra trước khi đọc.

Ví dụ

```
while not eof(f) do
   begin
     read(f,x);
     writeln(x);
   end;
```

* ***Đóng tệp***

```
Close(biến_tệp);
```

* ***Các ví dụ chương trình đọc dữ liệu từ tệp***

Bài toán 1: Đọc dữ liệu từ tệp 'songuyen.dat' đã tạo ở trên và hiện kết quả trên màn hình.

Chương trình

```
program Doc_tep_songuyen;
 uses crt;
 var i: integer; f: file of integer;
 begin
  clrscr;
  assign(f, 'songuyen.dat');
  reset(f);
  while not eof(f) do
    begin
      read(f, i);
      writeln(i);
    end;
  close(f);
  readln;
end.
```

Bài toán 2: Viết chương trình thực hiện tạo tệp 'diem.dat' ghí lại điểm thi của thí sinh, dữ liệu bao gồm: họ và tên thí sinh, điểm toán, điểm lý, điểm hoá. Đồng thời thực hiện đọc tệp và in ra thí sinh trúng tuyển, điểm chuẩn đỗ được nhập vào từ bàn phím.

Chương trình

```
program Tao_doc_tep_diemts;
  uses crt;
  type hs = record
          ten: string[25];
          toan,ly,hoa : real;
         end;
```

```
  var   i,n: integer; f: file  of  hs;  nhap: hs; diemc: real;
{**************}
  procedure tao;   { thu tuc tao }
    begin
     clrscr;
     assign(f, 'diem.dat');
     rewrite(f);
     write(' So thi sinh: '); readln(n);
     for  i:=1  to  n  do
      begin
       with  hs  do
        begin
          write(' Ho va ten: '); readln(ten);
          write(' Diem toan: '); readln(toan);
          write(' Diem ly : '); readln(ly);
          write(' Diem hoa : '); readln(hoa);
        end;
       write(f, hs);
      end;
    close(f);
 end;  { ket thuc thu tuc tao}
{ ***************}
Procedure  doc;  { thu tuc doc }
  begin
    clrscr;
        Assign(f, 'Diem.dat' );
    reset(f);
    write(' Diem chuan : '); readln(diemc);
    writeln(' Danh sach thi sinh trung tuyen dai hoc ');
    while not  eof(f)  do
        begin
          read(f,hs);
          with  hs   do
 if   toan+ly+hoa >= diemc  then  writeln(ten:25,toan:10:1,ly:10:1,hoa:10:1);
        end;
   close(f);
  end;  { ket thuc thu tuc doc}
{******************}
 { than chuong trinh chinh}
repeat
  clrscr;
  writeln(' 1- Tao tep');
  writeln(' 2- Doc tep');
  writeln(' 3- Ket thuc');
  write(' Hay chon mot viec ? '); readln(i);
  case  i  of
    1: tao;
    2: doc;
```

```
  end;
 until  i=3;
 readln;
end.
```

2.6 - Truy nhập tệp trực tiếp

Các phần đã xét ở trên là truy nhập tuần tự tệp có định kiểu. Trong phần này ta xét cách truy nhập trực tiếp tệp có định kiểu.

Sử dụng tất cả các thủ tục và lệnh đã nêu ở trên, ngoài ra để truy nhập trực tiếp tệp còn sử dụng một số thủ tục và hàm sau.

* ***Thủ tuc Seek để dịch chuyển con trỏ tệp***

Seek(biến_tệp, n);

n có kiểu longint. Thủ tục này thực hiện chuyển con trỏ tệp tới phần tử thứ n.

Trong tệp phần tử đầu được đánh thứ tự là 0.

* ***Hàm Filepos***

Filepos(biến_tệp)

Hàm này cho vị trí hiện thời của con trỏ tệp. Ví trí đầu là 0.

* ***Hàm Filesize***

Filesize(biến_tệp)

Hàm này cho số lượng phần tử của tệp. Hàm cho giá trị 0 khi tệp rỗng.

Để thêm 1 phần tử vào tệp phải thêm vào cuối tệp. Như vậy phải dịch con trỏ tới cuối tệp bằng thủ tục seek như sau:

```
seek(biến_tệp, Filesize(biến_tệp)-1 );
```

* ***Ví dụ chương trình truy nhập tệp trực tiếp***

Bài toán 1: Tạo tệp 'sochan.dat' ghi các số nguyên dương chẵn <=20. Truy nhập để sửa một phần tử bất kỳ và thêm một phần tử vào tệp.

Chương trình

```
Program  truy_nhap_truc_tiep_tep;
 uses  crt;
 var    i,j : integer;  f: file  of  byte;
 { thu tuc tao tep }
 procedure tao;
  begin
    clrscr;
    assign(f, 'sochan.dat');
    rewrite(f);
    for  i:=1  to  20  do
      if  (i mod 2) =0  then  write(f, i);
    close(f);
    readln;
  end;  {ket thuc thu tuc tao }
{ thu tuc sua }
procedure  sua;
  begin
     clrscr;
     reset(f);
```

```
    write(‘ sua phan tu thu ? ‘); readln(i);
    seek(f, i-1);
    read(f,j);
    witeln(‘ gia trị cu: ‘, j);
    write(‘ nhap gia tri moi : ‘);
    readln(j);
    seek(f, i-1);
    write(f, j);
    close(f);
 end;  { ket thuc thu tuc sua }
{ thu tuc them phan tu }
 procedure  them;
   begin
    clrscr;
    reset(f);
    write(‘ gia tri moi them: ‘);
    readln(j);
    seek(f, filesize(f)-1);
    write(f,j);
    close(f);
    readln;
 end;  {ket thuc thu tuc sua }
{ thu tuc doc }
procedure doc;
 uses crt;
 clrscr;
 reset (f);
 while   not  eof(f)  do
   begin
     read(f,i);
     witeln(i);
  end;
 close(f);
end;     { ket thuc thu tuc doc}

{******************}
 { than chuong trinh chinh}
repeat
   clrscr;
   writeln(‘ 1- Tao tep’);
   writeln(‘ 2- Sua tep’);
   writeln(‘ 3- Them phan tu’);
   writeln(‘ 4- Doc tep’);
   writeln(' 5-  ket thuc ');
   write(‘ Hay chon mot viec ? ‘); readln(i);

   case  i  of
     1: tao;
```

```
    2: sua;
    3: them;
    4: doc;
  end;
 until  i=5;
 readln;
end.
```

2.7 - Tệp văn bản

**** Khai báo tệp văn bản***

Trong Pascal có một kiểu tệp đã được định nghĩa trước đó là tệp văn bản, được định nghĩa với từ chuẩn TEXT.

Khai báo kiểu tệp văn bản;

Var Ten_bien: TEXT;

Các phần tử của TEXT là các kiểu kí tự, được tổ chức thành các dòng với độ dài của các dòng khác nhau, nhờ có thêm các dấu hết dòng (End of Line). Đó là cặp kí tự điều khiển CR (Carriage Return, nhảy về đầu dòng, mã ASCII là 13) và LR(Line Feed: xuống dòng, mã ASCII là10), chúng dùng để ngăn cách các dãy kí tự tương ứng với 2 dòng khác nhau.

Ví dụ: Đoạn văn bản sau:

HANOI
123
THUDO

Được chứa trong tệp văn bản thành một dãy như sau:

HANOI	CR LF	123	CR LF	THUDO	CR LF	EOF

Do tệp văn bản được tổ chức thành từng dòng nên việc ghi và đọc tệp văn bản có thể tục ghi và đọc theo dòng.

**** Ghi vào tệp văn bản***

Có thể ghi các giá trị kiểu Integer, real, boolean, string vào tệp văn bản bằng lệnh Write hoặc Writeln, các lệnh này cho phép chuyển các giá trị kiểu nói trên sang dạng kí tự.

Có 3 dạng sau:

Dạng 1:

Write(biến _tệp, biểu_thức_1, biểu_thức_2,..., biểu_thức_n);

Ghi các giá trị của biểu thức vào tệp không có dấu hết dòng

Dạng 2:

Writeln(biến _tệp, biểu_thức_1, biểu_thức_2,..., biểu_thức_n);

Ghi các giá trị của biểu thức vào tệp có dấu hết dòng.

Dạng 3:

Writeln(biến _tệp);

Ghi dấu hết dòng vào tệp.

**** Đọc dữ liệu từ tệp vănbản***

Chúng ta có thể đọc các kí tự, các số nguyên, số thực, boolean từ tệp văn bản thông qua các thủ tục sau:

Dạng 1:

read(biến _tệp, biến1, biến2, biến3,..., biếnn);

Đọc các giá trị từ tệp gán cho các biến tương ứng và không sang đầu dòng tiếp theo.

Dạng 2:

readln(biến _tệp, biến1, biến2, biến3,..., biếnn);

Đọc các giá trị từ tệp gán cho các biến tương ứng và đưa con trỏ sang đầu dòng tiếp theo.

Dạng 3:

readln(biến _tệp);

Đưa con trỏ sang đầu dòng tiếp theo.

Hàm chuẩn EOLN(f) sẽ phát hiện ra dấu hết dòng của tệp f.

Hàm này cho giá trị là True nếu kết thúc dòng, ngược lại là False.

**** Các chương trình xử lý tệp văn bản***

Bài toán 1: Tạo tệp văn bản ghi lại n dòng văn bản nhập vào từ bàn phím. Đọc tệp đã tạo và hiện ra trên màn hình.

```
 Program  Tao_doc_tep_van_ban;
  uses  crt;
  var    i,n: integer; f: text;  s: string;
{***************}
  procedure tao;   { thu tuc tao }
    begin
      clrscr;
      assign(f, 'vanban.txt');
      rewrite(f);
      write(' So dong van ban: '); readln(n);
      for  i:=1  to  n  do
        begin
             write(' Nhap dong  ', i ); readln(s);
             writeln(f, s);
        end;
     close(f);
 end;  { ket thuc thu tuc tao}
{ ****************}
Procedure  doc;  { thu tuc doc }
  begin
    clrscr;
    Assign(f, 'vanban.txt' );
    reset(f);
    Writeln('   Tep van ban da tao');
    while not  eof(f)  do
        begin
          readln(f,s);
          writeln(s);
        end;
   close(f);
  end;  { ket thuc thu tuc doc}
{*******************}
 { than chuong trinh chinh}
repeat
  clrscr;
```

```
   writeln(‘ 1- Tao tep’);
   writeln(‘ 2- Doc tep’);
   writeln(‘ 3- Ket thuc’);
   write(‘ Hay chon mot viec ? ‘); readln(i);
   case  i  of
     1: tao;
     2: doc;
   end;
 until  i=3;
 readln;
end.
```

Bài toán 2: Đọc xâu văn bản từ tệp, kiểm tra xem xâu đó có đối xứng không, sau đó sắp xếp các chữ số trong xâu theo thứ tự tăng dần.

chương trình

```
  PROGRAM DOC_XAU_SAP_TT_SO;
   USES CRT;
   TYPE MANG=ARRAY[1..200] OF CHAR;
   VAR I,K,L,N:INTEGER;T:BOOLEAN;S:STRING;A:MANG;F:TEXT;
   PROCEDURE SAPTANG(VAR X:MANG;M:INTEGER);
   VAR J,Z:INTEGER;P:CHAR;
   BEGIN
     FOR J:=1 TO M-1 DO
       FOR Z:=1 TO M-J DO IF X[Z]>X[Z+1] THEN BEGIN P:=X[Z];
                                          X[Z]:=X[Z+1];
                                          X[Z+1]:=P;
                                          END;
   END;
   BEGIN   { thân chương trinh chính}
   CLRSCR;
   ASSIGN(F,'VB.TXT');
   RESET(F);
   READLN(F,S);
   WRITELN(' XAU DOC DUOC');
   WRITELN(S);
   N:=LENGTH(S);
   {tim xau doi xung}
   K:=1;L:=N;T:=TRUE;
   WHILE (K<L) AND T DO
       IF S[K]=S[L] THEN BEGIN K:=K+1; L:=L-1; END
                    ELSE T:=FALSE;
IF T THEN WRITELN(' XAU DOI XUNG ') ELSE WRITELN(' XAU KHONG DOI XUNG ');
   { sap cac so trong xau tang dan }
   K:=0;
   FOR I:=2 TO N DO IF (ORD(S[I])>=48) AND (ORD(S[I])<=57) THEN
        BEGIN
          K:=K+1;
```

```
      A[K]:=S[I];
    END;
SAPTANG(A,K);
L:=0;
FOR I:=1 TO N DO IF (ORD(S[I])>=48) AND (ORD(S[I])<=57) THEN
    BEGIN
      L:=L+1;
      S[I]:=A[L];
    END;
WRITELN('XAU DA SAP THU TU CAC SO ');
WRITELN(S);
READLN;
CLOSE(F);
END.
```

Bài tập chương VI

1. Viết chương trình gồm 2 CT con làm các công việc sau: Ghi 1 bài thơ có n dòng, nội dung nhập vào từ bàn phím lên tệp văn bản có tên 'BAITHO.TXT' và đọc tệp văn bản đã tạo từ đĩa và cho hiện nội dung lên màn hình. Dùng câu lệnh lựa chọn để chọn công việc: 1= Ghi, 2=Doc.

2. Viết chương trình tạo tệp dữ liệu 'DIEM.DAT' chứa bảng điểm của n người và in kết quả ra màn hình. Bảng điểm gồm các cột: Họ tên, Điểm Toán, Điểm Lý, Điểm Hoá , Tổng, Loại.
Nhập vào: Họ tên, Điểm toán, Điểm lý, Điểm hoá.
Tính Tổng, Loại :
Tổng = Điểm toán + Điểm lý + Điểm hoá .
Loại = ' DO ' nếu Tổng > 19
Loại = ' TRUOT' nếu Tổng <= 19

3. Viết chương trình tạo tệp dữ liệu 'LUONG.DAT' chứa bảng lương của n người và in kết quả ra màn hình.
Bảng lương gồm các cột: Họ tên, Ngày công, Lương ính,Thưởng, Tổng
Nhập vào : Họ tên, Ngày công, Lương chính .
Tính Thưởng, Tổng :
Thưởng = Lương chính x 2 nếu Ngày công > 25
Thưởng = Lương chính nếu Ngày công <= 25
Tổng = Lương chính + Thưởng .

4. Viết chương trình tạo tệp dữ liệu 'HANG.DAT' chứa bảng Thống kê hàng nhập của n mặt hàng và in kết quả ra màn hình. Bảng Thống kê hàng nhập gồm các cột:
Tên , Số lượng, Đơn giá, Tiền, Loại.
Nhập vào: Tên , Số lượng, Đơn giá.
Tính Tiền, Loại : Tiền = Số lương x Đơn giá .
Loại = 'Cao' nếu Tiền > 500000.
Loại = 'THAP' nếu Tiền <= 500000.

5. Viết chương trình tạo tệp dữ liệu 'SACH.DAT' chứa bảng danh mục của n quyển sách và in kết quả ra màn hình. Bảng danh mục sách bao gồm các cột:
Tên, Năm, Tác giả, Giá, Loại.
Nhập vào : Tên, Năm, Tác giả, Giá.
Tính Loại theo công thức sau:
Loại = 'Cao' nếu Tiền > 50000
Loại = 'TB' nếu 20000<= Tiền <= 50000
Loại = 'THAP' nếu Tiền < 20000.

CHƯƠNG VII
ĐỒ HOẠ VÀ ÂM THANH

1- Đồ hoạ

1.1 - Khái niệm chung

Màn hình máy vi tính hiện nay có nhiều loại khác nhau, các máy AT-386, AT-486, AT-586... thường dùng loại màn hình VGA (Video Graphic Adapter), TVGA (Top of VGA) hoặc SVGA (Supper VGA), đây là loại màn hình có thể dùng ở một trong hai chế độ: chế độ TEXT - hiển thị văn bản và chế độ GRAPHIC - hiển thị đồ hoạ.

Trong chế độ TEXT màn hình được chia thành 25 dòng và 80 cột, nếu viết kín màn hình ta có thể viết được 2000 ký tự. Chúng ta có thể thay đổi chế độ phân giải để viết ra 25 dòng x 40 cột hoặc 80 dòng x 50 cột....

Muốn vẽ hình, tô màu các hình ta phải chuyển sang chế độ đồ hoạ, trong chế độ này màn hình được xem là một ma trận điểm, tuỳ thuộc độ phân giải ta có thể có ma trận 640x480 điểm hoặc 1024x720 điểm.... Mỗi điểm trên màn hình được gọi là 1 Pixel tức là một phần tử ảnh
(Picture Element), ta có thể hoàn toàn chủ động trong việc thay đổi màu sắc, cường độ sáng của từng điểm để tạo ra một bức tranh theo ý muốn. Vị trí của mỗi điểm trên màn hình được biểu diễn bởi hai toạ độ: Hoành độ và Tung độ, gốc toạ độ (0,0) là điểm ở góc trên bên trái màn hình còn điểm góc dưới bên phải có toạ độ là 639,479.

Muốn chuyển sang làm việc ở chế độ đồ hoạ, trong thư mục hiện hành (thư mục chứa chương trình Pascal) phải có các tệp GRAPH.TPU, *.BGI và *.CHR. Lời gọi đơn vị chương trình đồ hoạ phải đặt ở đầu chương trình sau từ khoá USES như ví dụ 43.

Ví dụ 43
```
Program Ve_hinh;
Uses GRAPH;
.......
```

1.2 - Khởi động chế độ đồ hoạ

Trong phần thân chương trình cần phải đưa vào các thông báo về kiểu màn hình, chế độ đồ hoạ (MODE) tương ứng. Những người làm tin học ứng dụng thường không quan tâm lắm đến các thông số này do vậy dễ lúng túng khi cần khai báo. Để khắc phục nhược điểm đó trong Turbo Pascal đã có sẵn một thủ tục khởi tạo chế độ đồ hoạ là **Initgraph(var GD,GM: Integer, DP:string[n]);** Khi gọi thủ tục này với các tham số hợp lệ Initgraph sẽ tự xác định kiểu màn hình và Mode đồ hoạ tối ưu .

Người lập trình có thể tự mình khai báo các tham số GD, GM (Trong đó GD: Graph Driver - là một số nguyên xác định kiểu màn hình; GM: Graph Mode - cũng là một số nguyên xác định Mode đồ hoạ).

Trước hết trong phần khai báo biến các tham số này phải khai thuộc kiểu Integer, sau đó trong phần thân chương trình phải chọn các giá trị thích hợp cho kiểu màn hình và Mode đồ hoạ.

Thông thường chúng ta không biết chính xác kiểu màn hình và Mode đồ hoạ của máy tính đang sử dụng nên sẽ gặp lúng túng khi khai báo. Trong trường hợp này tốt nhất là để Pascal tự xác định giúp chúng ta các tham số này.

Nếu ngay sau từ khoá Begin của phần thân chương trình chúng ta khai báo

GD:= DETECT;

thì Initgraph hiểu là nó phải tự đi xác định kiểu màn hình và Mode đồ hoạ sao cho đạt kết quả tối ưu. Nói chung trừ những trường hợp đặc biệt, chúng ta không nên tự xác định những thông số này làm gì.

Bảng .... cho biết giá trị của các tham số GD,GM và độ phân giải màn hình của ba loại màn hình đã có, loại thông dụng nhất hiện nay là VGA.

Bảng ...

Kiểu màn hình		**Mode đồ hoạ**		**Độ phân giải**
Tên	Giá trị	Tên	Giá trị	
CGA	1	CGAC0 CGAC1 CGAC2 CGAC3 CGACHi	0 1 2 3 4	320x200 320x200 320x200 320x200 640x200
EGA	3	EGALO EGAHi	0 1	640x200 640x350
IBM8514	6	IBM8514LO IBM8514Hi	0 1	640x480, 256 màu 1024x768, 256 màu
VGA	9	VGALO VGAMed VGAHi	0 1 2	640x200 640x350 640x480

Ví dụ 44 trình bày cách sử dụng thủ tục **Initgraph** để thiết kế chương trình vẽ một đường tròn có tâm tại chính giữa màn hình và bán kính là 50 Pixel. Các tham số GD,GM sẽ do Pascal tự xác định, các tệp điều khiển đồ hoạ đặt trong thư mục TP trên đĩa cứng C.

```
Ví dụ 44
Program Ve_hinh_tron;
Uses graph;
Var
GD,DM: Integer;
BEGIN
GD:= detect;
Initgraph(GD,GM,'C:\tp\bgi');
If graphresult <> grok then halt(1);
Circle(320,240,50);
Readln;
CloseGraph;
END.
```

Nếu chúng ta muốn tự mình khai báo GD và GM (Ví dụ: màn hình VGA, kiểu đồ hoạ là độ phân giải cao, tệp tin đồ hoạ chứa trong thư mục TP5 trên đĩa A) thì bỏ lệnh GD:=DETECT Và viết lại đoạn đầu của ví dụ 44 như sau:

.....

GD:=VGA;
GM:=VGAHi;
INITGRAPH(GD,GM,'C:\TP\BGI');

Chú ý: *Có thể thay thế tên màn hình bằng giá trị tương ứng đã có trong bảng.*

Tham số DP (Driver Path) là đường dẫn tới thư mục chứa các tệp điều khiển kiểu màn hình đồ hoạ, trong ví dụ trên được khai là 'c:\tp\bgi' nghĩa là ổ đĩa C thư mục TP, còn BGI là đuôi tệp đều khiển. Trong trường hợp chúng ta không khai báo gì cho DP, nghĩa là để một chuỗi rỗng " thì Pascal sẽ tự tìm các tệp điều khiển đồ hoạ trong thư mục chủ (là nơi mà Pascal được khởi động).

1.3 - Kiểm tra lỗi đồ hoạ

Dòng thứ 8 trong ví dụ 44

If graphresult <> grok then halt(1);

Là câu lệnh kiểm tra lỗi đồ hoạ. Việc kiểm tra được thực hiện thông qua hàm của hàm **graphresult**. Lỗi đồ hoạ thường phát sinh khi không tìm thấy các trình điều khiển đồ hoạ, hoặc khi chúng ta khai báo không đúng các tham số GD, GM, DP. Nếu phát hiện ra lỗi hàm Graphresult sẽ nhận một trong các giá trị cho trong bảng sau, mỗi giá trị ứng với một hằng chuỗi ký tự:

Bảng lỗi của hàm Graphresult

Giá trị	Hằng chuỗi	ý nghĩa
0	Grok	Tốt, không có lỗi
-1	Grnoinitgraph	Không tìm thấy đơn vị đồ hoạ
-2	Grnotdetected	Không có phần cứng đồ hoạ
-3	GrFilenotfound	Không tìm thấy các tệp điều khiển màn hình đồ hoạ
-4	GrInvalidDriver	Trình điều khiển hỏng, không hợp lệ
-5	GrNoloadMem	Bộ nhớ động (RAM) bị thiếu
-6	GrNoscanMem	Không thực hiện được việc kiểm tra bộ nhớ
-7	GrNoFloodMem	
-8	GrFontNoFound	Không tìm thấy các tệp chứa font chữ
-9	GrNoFontMem	Không đủ bộ nhớ để nạp Font chữ
-10	GrInvalidMode	Kiểu đồ hoạ không hợp lệ
-11	GrError	Lỗi đồ hoạ (tổng quát)
-12	GrIOError	Lỗi các đường nhập, xuất đồ hoạ
-13	GrInvalidFont	Kiểu chữ không hợp lệ
-14	GrInvalidFontNum	Số hiệu kiểu chữ không hợp lệ

Như vậy nếu giá trị của hàm Graphresult khác Grok nghĩa là có một lỗi nào đó thì chương trình sẽ bị dừng bằng lệnh Halt(1), còn nếu tốt thì tiếp tục làm việc.

1.4 - Một số thủ tục cơ bản để vẽ hình

* ***MOVETO(x,y)*** : Di chuyển con trỏ đến toạ độ x,y (x là hoành độ, y là tung độ)
x,y là các giá trị kiểu Integer, với màn hình VGA thì 0<= x <=639, 0<= y <= 479

* ***LINETO(x,y)***: Vẽ một đường thẳng từ vị trí con trỏ hiện thời tới toạ độ x,y kết thúc quá trình vẽ con trỏ nằm tại toạ độ mới.

Pascal có sẵn hai hàm để xác định toạ độ góc dưới bên phải màn hình đó là Getmaxx và Getmaxy. Để vẽ đường chéo của màn hình từ toạ độ 0,0 ta có thể viết các lệnh

```
MOVETO(0,0);
LINETO(Getmaxx,Getmaxy);
```

* ***LINE(x1,y1,x2,y2*** : Thủ tục này vẽ một đường thẳng từ toạ độ x1,y1 đến toạ độ x2,y2
* ***LINEREL(dX,dY)***: Vẽ đường thẳng từ vị trí hiện thời (toạ độ x,y) tới toạ độ x+dx, y+dy.
* ***CIRCLE(x,y,r)***: Vẽ đường tròn tâm tại toạ độ x,y bán kính bằng r Pixel
* ***PUTPIXEL(x,y, n)***: Thủ tục này sẽ vẽ một điểm sáng tại toạ độ x,y với màu là n. Giá trị n lấy trong khoảng 0-15 hoặc viết trực tiếp tên màu theo tiếng Anh.

• *Thiết lập màu đồ hoạ*

Để định màu trong chế độ đồ hoạ ta có thể dùng hai thủ tục sau đây :

g. SETCOLOR(n) : Định màu cho các nét vẽ

h. SETBKCOLOR(n) : Định màu nền cho nét vẽ

Tham số n cho các thủ tục 7 và 8 xem trong bảng sau

Bảng .. Mã màu và tên màu

Mã màu	Tên tiếng Anh	Tên tiếng Việt
0	Back	Đen
1	Blue	Xanh da trời
2	Green	Xanh lá mạ
3	Cyan	Xanh lơ
4	Red	Đỏ
5	Magenta	Tím
6	Brown	Nâu
7	Lightgray	Xám nhạt
8	Darkgray	Xám xẫm
9	Lightblue	Xanh da trời nhạt
10	Lightgreen	Xanh lá mạ nhạt
11	Lightcyan	Xanh lơ nhạt
12	Lightred	Đỏ nhạt
13	Lightmagenta	Tím nhạt
14	Yellow	Vàng
15	White	Trắng

Chú ý:

- Lệnh định màu phải đặt trước lệnh vẽ hình
- Một màu đã định sẽ có tác dụng cho đến khi có lệnh định màu mới

- **Vẽ hình khép kín và tô màu**

* ***RECTANGLE(x1,y1,x2,y2) :*** Vẽ khung hình chữ nhật toạ độ góc trên bên trái là x1,y1 , toạ độ góc dưới bên phải là x2,y2.

* ***BAR(x1,y1,x2,y2)*** : Vẽ một hình chữ nhật góc trên bên trái có toạ độ x1,y1 góc dưới bên phải có toạ độ x2,y2. Khi dùng kết hợp với thủ tục số 11 sẽ đồng thời cho phép kẻ các vân hoa trên nền và tô màu cho nền.

* ***SETFILLSTYLE(n1,n2) :*** Thủ tục định vân hoa và màu nền.

n1 là một giá trị nguyên với 0<= n1 <= 11 : định kiểu vân hoa (xem bảng ...)

n2 là số hiệu mã màu đã giới thiệu 0<= n2 <= 15

Chú ý:

- Thủ tục 11 chọn màu nền và vân hoa chỉ có hiệu lực khi chúng ta sử dụng kèm theo các thủ tục vẽ hình phẳng sau đây:

Hình chữ nhật (Bar), hình hộp chữ nhật (Bar3D), hình quạt (Pieslice), đa giác có toạ độ nhập vào mảng (Fillpoly), Tô màu cho miền (Floodfill).

- Khi vẽ đường tròn hoặc khung chữ nhật thì không thể tạo hoa văn và tô màu trong nền.

- Thủ tục SETFILLSTYLE(n1,n2) cần đưa vào trước các thủ tục vẽ hình đã nêu trong mục chú ý. Giá trị của màu và kiểu vân hoa sẽ được giữ cho đến khi ta định nghĩa lại.

Bảng... Giá trị và ý nghĩa của tham số n1

Giá trị	Tên tiếng Anh	ý nghĩa
0	EmptyFill	Không tạo hoa văn bên trong hình
1	SolidFill	Tạo hoa văn bằng nét liền
2	LineFill	Tạo bằng nét gạch nối ----
3	LTSlashFill	Tạo hoa văn bằng nét ///
4	SlashFill	Tạo hoa văn bằng nét /// đậm
5	BKSlashFill	Tạo hoa văn bằng nét \\\ đậm
6	LTBKSlashFill	Tạo hoa văn bằng nét \\\
7	HatchFill	Kẻ lưới hình ô vuông
8	XHatchFill	Kẻ lưới hình ô vuông xoay 90 độ
9	InterleaveFill	Tô bằng nét đứt quãng
10	WideDotFill	Tạo hoa văn bằng dấu chấm thưa
11	CloseDotFill	Tạo hoa văn bằng dấu chấm dày (liền nhau)

1.5 - Viết chữ trong chế độ đồ hoạ

Khi đã chuyển sang làm việc ở chế độ đồ hoạ ta không thể viết chữ bình thường như trong chế độ văn bản. Muốn viết chữ trong các hình vẽ ta sử dụng một số thủ tục sau đây:

* ***OUTTEXT(chuỗi)*** : Thủ tục này sẽ cho hiện chuỗi ký tự tại vị trí con trỏ hiện thời. Chuỗi có thể viết trực tiếp hoặc thông qua biến chuỗi như trong ví dụ 45 sau đây:

Ví dụ 45

```
.....
Var
chuviet : string[30]
Begin
outtext('cong hoa xa hoi chu nghia ...');
```

```
chuviet:='Viet nam dan chu cong hoa';
outtext(chuviet);
.....
```

* ***OUTTEXTXY(x,y,chuoi) :*** thủ tục này sẽ viết ra chuỗi ký tự tại toạ độ x,y.
* ***SETTEXTSTYLE(Kiểu chữ, Chiều viết, Kích thước);***
  Kiểu chữ là một tham số nguyên nhận giá trị trong khoảng 0-4
  Chiều viết chỉ nhận 1 trong hai giá trị : 0 nằm ngang; 1 thẳng đứng
  Kích thước Là hệ số phóng to chữ có thể chọn từ 0-10

Để chấm dứt chế độ đồ hoạ trở về chế độ văn bản ta dùng thủ tục CLOSEGRAPH. Sau đó muốn quay lại chế độ đồ hoạ ta lại phải gọi lại INITGRAPH.

Trong một số trường hợp để chuyển nhanh giữa chế độ đồ hoạ và văn bản chúng ta có thể dùng hai thủ tục sau đây:

- RESTORECRTMODE; Tạm ngừng chế độ đồ hoạ chuyển sang chế độ văn bản.
- SETGRAPHMODE(n); Ngắt chế độ văn bản đã tạo ra bởi Restorecrtmode thiết lập trở lại chế độ đồ hoạ. Tham số n có thể lựa chọn trong khoảng 0-2. Ví dụ 46 dưới đây trình bày cách sử dụng các thủ tục này.

```
Program dohoa_text;
uses crt,graph;
var
gd,gm:integer;
begin
gd:=detect;
initgraph(gd,gm,'a:\tp5\bgi');
if graphresult<>grok then halt(1);
moveto(0,0); setcolor(5);
lineto(300,300); delay(2500);
circle(400,300,100); delay(1500);
restorecrtmode;                 (* Chuyển về chế độ văn bản *)
gotoxy(20,20);textcolor(9);
write('Happy New Year');
readln;
setgraphmode(2);        (* Trở về chế độ đồ hoạ với n=2 cho màn hình VGA*)
setcolor(blue);
circle(100,100,50);
delay(2000);
restorecrtmode;         (* Chuyển sang chế độ văn bản lần thứ hai*)
textcolor(3);
gotoxy(20,0);write('DAI HOC NONG NGHIEP I HA NOI');
readln;
closegraph;              (* Kết thúc chế độ đồ hoạ*)
end.
```

Việc sử dụng các thủ tục đồ hoạ không có gì phức tạp, với một chút cố gắng bạn có thể vẽ đọc những hình rất đẹp theo mong muốn. Dưới đây là một chương trình vẽ đồ thị hình sin. Chạy chương trình ta sẽ thấy ba đường hình sin với các biên độ và màu sắc khác nhau.

Ví dụ 46

```
Program Do_thi_hinh_sin;
uses graph,crt;
const  m=0.1;
Var  t3,t4,t1,n,t2,gd,gm:integer;    t,x,y,z:real;
  Begin
  gd:=detect;
  Initgraph(gd,gm,'a:\tp5\bgi');
  if graphresult<>grok then Halt(1);
  x:=0;     t3:=100;     n:=0;     t2:=10;
   while t2<=600 do
   Begin
  setcolor(green);
  y:=sin(x);
  t1:=round(y*50);
  t3:=round(y*70);
  t4:=round(y*100);
  t1:=200-t1;
  t3:=200-t3;
   t4:=200+t4;
  moveto(10,200);
  lineto(620,200);
  line(10,80,10,300);
  settextstyle(3,0,3);
  outtextxy(610,205,'x');
  settextstyle(3,0,3);
  outtextxy(15,75,'y');
  settextstyle(4,0,3);
  setcolor(5);
  outtextxy(200,300,'do thi ham sin(x)');
  setcolor(12);
  moveto(10,200);
  putpixel(t2,t1,11);
  putpixel(t2,t3,14);
  setcolor(red);
  putpixel(t2,t4,random(14));
  setcolor(12);
  delay(5);
   x:=x+0.07;
   t2:=t2+1;
   end;
   repeat until keypressed;
   t1:=1;
   t2:=200;
   while t1<=220 do
   begin
   line(340,240,round(sqrt(440*440-t1*t1)),t1);
   t1:=t1+1;
   delay(15);
```

```
  end;
  repeat until keypressed;
  closegraph;
  End.
```

Chương trình dưới đây thiết kế một đồng hồ ba kim , tốc độ chạy của kim giây tuỳ thuộc vào lệnh DELAY(n) , nếu chọn DELAY(1000) thì cứ 1 giây kim giây chuyển một vị trí. Khi nhập chương trình vào máy cần lưu ý khai báo lại đường dẫn đến thư mục chứa các tệp *.BGI

```
Program VEDONGHO;
uses crt,graph;
 var
x,y, maxx,maxy, gd,gm,color,i,j,t:integer;
 N:real;
 LAM,TT:CHAR;
 begin
  gd:=detect;
  initgraph(gd,gm,'c:\tp5\BGI');
  setcolor(5);
  rectangle(30,20,610,450);
  rectangle(31,21,609,449);
  rectangle(32,22,608,448);
  setfillstyle(9,2);
  bar(33,23,607,447);
  setcolor(red);
  setbkcolor(red);
  for i:=1 to 10 do circle(320,240,i);
  setcolor(11);
  setbkcolor(white);
  for i:=11 to 80 do circle(320,240,i);
   setcolor(14);
   setbkcolor(white);
  for i:=80 to 160 do circle(320,240,i);
   setcolor(white);
  for i:=160 to 200 do circle(320,240,i);
  setcolor(11);
  circle(320,240,79);
  circle(320,240,80);
  setcolor(4);
  circle(320,240,159);
  circle(320,240,160);
  settextstyle(3,0,4);
  outtextxy(310,40,'XII');
  outtextxy(405,60,'I');
  outtextxy(470,120,'II');
  outtextxy(490,200,'III');
  outtextxy(480,290,'IV');
```

```
outtextxy(410,370,'V');
outtextxy(310,400,'VI');
outtextxy(210,370,'VII');
outtextxy(135,290,'VIII');
outtextxy(130,210,'IX');
outtextxy(155,130,'X');
outtextxy(220,60,'XI');
setcolor(blue);
Settextstyle(4,0,5);
outtextxy(230,100,'DIAMON');
setcolor(random(14));
for i:=1 to 20 do
circle(320,360,i );
settextstyle(1,0,2);
setcolor(5);
outtextxy(200,450,'Copyright by Dr. Duong Xuan Thanh');
    for i:=1 to 20 do
  begin
  setcolor(random(14));
  circle(320,360,i );
  end;
for i:=1 to 20 do
begin
setcolor(random(14));
circle(320,360,i );

end;
for t:=0 to 12 do   {-------- Kim gio --------}
begin
setcolor(12);
moveto(320,240);
setlinestyle(0,0,3);
SetWriteMode(xorput);
linerel(round(110*cos((t*30-89)*pi/180)),round(110*sin((t*30-89)*pi/180)));
 moveto(320,240);
linerel(round(110*cos((t*30-90)*pi/180)),round(110*sin((t*30-90)*pi/180)));
 moveto(320,240);
linerel(round(110*cos((t*30-91)*pi/180)),round(110*sin((t*30-91)*pi/180)));
 moveto(320,240);
linerel(round(110*cos((t*30-92)*pi/180)),round(110*sin((t*30-92)*pi/180)));
   for i:=0 to 60 do  { ---------Kim phut -------}
   begin
   setcolor(12);
   moveto(320,240);
   setlinestyle(0,0,3);
   SetWriteMode(xorput);
   linerel(round(130*cos((i*6-89)*pi/180)),round(130*sin((i*6-89)*pi/180)));
   moveto(320,240);
```

```
  linerel(round(130*cos((i*6-90)*pi/180)),round(130*sin((i*6-90)*pi/180)));
  moveto(320,240);
  linerel(round(130*cos((i*6-91)*pi/180)),round(130*sin((i*6-91)*pi/180)));
              (*--------Kim giay--------*)
   for j:=0 to 360 do
   begin
   moveto(320,240);
   setlinestyle(0,0,3);
   SetWriteMode(XORPut);
   setcolor(12);
   linerel(round(150*cos((j-90)*pi/180)),round(150*sin((j-90)*pi/180)));
   moveto(320,240);
   linerel(round(150*cos((j-91)*pi/180)),round(150*sin((j-91)*pi/180)));
   delay(1000);       moveto(320,240);
   linerel(round(150*cos((j-90)*pi/180)),round(150*sin((j-90)*pi/180)));
   moveto(320,240);
   linerel(round(150*cos((j-91)*pi/180)),round(150*sin((j-91)*pi/180)));
   end;
   moveto(320,240);
   linerel(round(130*cos((i*6-89)*pi/180)),round(130*sin((i*6-89)*pi/180)));
   moveto(320,240);
   linerel(round(130*cos((i*6-90)*pi/180)),round(130*sin((i*6-90)*pi/180)));
   moveto(320,240);
   linerel(round(130*cos((i*6-91)*pi/180)),round(130*sin((i*6-91)*pi/180)));
 end;
 moveto(320,240);
linerel(round(110*cos((t*30-89)*pi/180)),round(110*sin((t*30-89)*pi/180)));
moveto(320,240);
linerel(round(110*cos((t*30-90)*pi/180)),round(110*sin((t*30-90)*pi/180)));
moveto(320,240);
linerel(round(110*cos((t*30-91)*pi/180)),round(110*sin((t*30-91)*pi/180)));
moveto(320,240);
linerel(round(110*cos((t*30-92)*pi/180)),round(110*sin((t*30-92)*pi/180)));
 end;
repeat  until keypressed;
END.
```

2 – Âm thanh

Trong các máy PC thông dụng công suất của loa rất nhỏ do đó việc tạo âm thanh chỉ có tính chất biểu diễn. Muốn tạo ra âm thanh cao thấp khác nhau ta chỉ cần đưa vào loa các xung điện với tần số khác nhau. Turbo Pascal đã có ba thủ tục thiết kế sẵn để làm việc này

SOUND(n) : tạo ra âm thanh với tần số n , ở đây n phải là một số nguyên dương

DELAY(n) : Kéo dài tín hiệu âm thanh trong khoảng thời gian n miligiây nếu sau đó có thủ tục ngắt âm Nosound.

NOSOUND : ngắt tín hiệu âm thanh

Cần chú ý rằng khi có thủ tục sound và Delay tín hiệu âm thanh sẽ được phát ra chừng nào chưa có Nosound mặc dù ta đã định khoảng thời gian trễ qua thủ tục Delay.

Để tạo một bản nhạc chúng ta cũng cần có một chút kiến thức về nhạc lý. Độ cao thấp trong âm nhạc được phân thành các quãng tám. Quãng tám trung tần thì nốt Đô có tần số 512 Hz, quãng tám trầm hơn tần số của nốt Đồ sẽ là 256 Hz, còn quãng tám cao hơn nốt Đố có tần số 1024 Hz.

Tên các nốt nhạc theo quy ước là:

Quãng tám trầm : đồ, rề, mì, fà, sòn, là, xì

Quãng tám trung : đô, rê, mi, fa, son, la , xi

Quãng tám cao : đố, rế, mí, fá, són, lá, xí

Về cao độ thì các nốt Mi - Fa và Xi - Đô cách nhau nửa cung các nốt còn lại cách nhau một cung. Nếu tính từ nốt Đồ (tần số 256) đến nốt Đố (tần số 512) khoảng cách là 6 cung như vậy sự chênh leechj về tần số trong mỗi cung sẽ là (512-256)/6 = 43 Hz. Giữa các nốt Mi - Fa và Xi - Đô chênh lệch chỉ là 22 Hz (vì các nốt này cách nhau nửa cung). Ngoài ra cũng cần biết cách tạo ra các nốt thăng, giáng để có thể tạo ra một bản nhạc theo ý muốn.

Nếu chúng ta chỉ muốn nghe một bản nhạc phát ra từ máy thì có thể tạo nên một chương trình thiết kế các nốt nhạc sau đó ghép chúng lại thành bản nhạc tuỳ ý.

Chương trình dưới đây thiết kế bản nhạc Làng tôi, các chương trình con T0, T1...T5 tạo ra trường độ, ví dụ T0 là nốt tròn, T1 là nốt đen... các chương trình con khác tạo ra các nốt nhạc của hai quãng tám trầm và trung, trong đó có một số nốt thăng và giáng.

```
Program Vancao;
uses crt;
   procedure T0;    Begin   delay(20000); Nosound; End;
   procedure T1;    Begin   delay(10000); Nosound; End;
   procedure T2;    Begin   delay(5000); Nosound; End;
   procedure T3;    Begin   delay(2500); Nosound; End;
   procedure T4;    Begin   delay(1250); Nosound; End;
   procedure T5;    Begin   delay(30000); Nosound; End;

   Procedure nt(i:integer);
    Begin   Sound(i);  End;
      Procedure do1; Begin nt(256); End;  {quang tam tram}
      Procedure do11; Begin nt(270); End;  {not do thang}
      Procedure re1; Begin nt(299); End;
      Procedure mi1; Begin nt(342); End;
      Procedure fa1; Begin nt(363); End;
      Procedure son1; Begin nt(406); End;
      Procedure la1; Begin nt(449); End;
      Procedure si1; Begin nt(492); End;

      Procedure do2; Begin nt(534); End;  {quang tam trung}
      Procedure re2; Begin nt(600); End;  {not Re giang}
      Procedure mi2; Begin nt(685); End;
      Procedure fa2; Begin nt(634); End;
      Procedure son2; Begin nt(677); End;
      Procedure la2; Begin nt(712); End;
      Procedure si2; Begin nt(755); End;

   Procedure Langtoi;
    Begin
```

```
  clrscr; gotoxy(25,12); textcolor(14); textbackground(red);
  Write(' LANG TOI * Nhac tien chien ');
  Repeat
  do11;t0;mi1;t1;son1;t0;la1;t1;son1;t0;
  son1;t1;do2;t1;si1;t1;la1;t1;son1;t0;
  la1;t1;son1;t1;fa1;t1;mi1;t1;son1;t0;t1;
  do11;t0;re1;t1;mi1;t0;son1;t1;do2;t0;re2;t1;mi2;t0;t1;
  re2;t1;do2;t1;re2;t1;do2;t1;son1;t0;mi1;t1;son1;t1;do2;t5;t1;
  do2;t1;do2;t2;la1;t1;la1;t1;si1;t1;la1;t1;son1;t0;t1;
  fa1;t1;fa1;t1;la1;t0;t4;la1;t1;mi1;t0;re1;t1;son1;t5;t1;
  do1;t0;re1;t1;mi1;t0;fa1;t1;son1;t0;mi1;t1;re1;t5;
  do1;t1;do2;t0;si1;t1;re2;t0;son1;t1;do2;t5;

  Until keypressed; Nosound;
  End;
 BEGIN
  Langtoi;
END.
```

Dưới đây là ví dụ thiết kế bàn phím thành các phím của một chiếc đàn dương cầm. Các nốt đô, rê, mi, fa, son, la , xi đố sẽ bấm các chữ cái tương ứng (D, R, M, F, S, L, X, Z). Muốn dừng âm ta bấm phím P (PAUSE), còn muốn dừng chương trình bấm E (EXIT).

```
Program nhac;
uses crt,graph;
var
n :char; i,j:integer;
begin
   clrscr;
   textcolor(14); textbackground(white);
   for i:=1 to 6 do
   begin
   gotoxy(16,i+3);
   for j:=1 to 51 do write(chr(177));
   end;
   gotoxy(17,5);
   textcolor(red);
   write('Do-D, Re-R, Mi-M, Fa-F, Son-S, La-L, Xi-X, Do2-Z');
   gotoxy(33,6); write(' Re2-W,  Mi2-T ');
   gotoxy(22,7);
   textcolor(blue);
   writeln('   P->Ngat am,  E-> Dung chuong trinh   ');
   gotoxy(22,8);
textcolor(5);
writeln('Moi ban choi moi ban nhac minh ua thich ');
repeat
n:= readkey;
if n = 'd' then begin nosound; delay(3); sound(523); end;
```

```
if n = 'r' then begin nosound; delay(10); sound(587); end;
if n = 'm' then begin  nosound; delay(10); sound(659); end;
if n = 'f' then begin  nosound; delay(10); sound(698); end;
if n = 's' then begin  nosound; delay(10); sound(784); end;
if n = 'l' then begin nosound; delay(10); sound(880); end;
if n = 'x' then begin nosound; delay(10); sound(988); end;
if n = 'z' then begin  nosound; delay(3); sound(1050); end;
if n = 'w' then begin  nosound; delay(3); sound(1190); end;
if n = 't' then begin nosound; delay(3); sound(1300); end;
if n = 'p' then nosound;
until upcase(n) ='E';
nosound;
end.
```

Bài tập chương VII

1. Viết chương trình vẽ n hình vuông lồng nhau.

2. Viết chương trình vẽ bàn cờ hình vuông có 64 ô (mỗi chiều là 8), xen kẽ cứ một ô đỏ rồi đến 1 ô trắng.

3. Viết chương trình vẽ đồ thị hình Cosin trong lương giác.

Tài liệu tham khảo

1. Bùi Thế Tâm. Windows 2000, Word 2000, Excel 2000, Powerpoint 2000. Nhà xuất bản Giao thông vận tải , năm 2003.

2. Bùi Thế Tâm. Turbo Pascal 7.0. Nhà xuất bản thống kê, năm 1996

3. Charles rubin. Microsoft Word (bản dịch). Nhà xuất bản trẻ, năm 1999.

4. Đỗ Xuân Lôi. Cấu trúc dữ liệu và Giải thuật. Nhà xuất bản Khoa học và Kỹ thuật, năm 1998.

5. Đỗ Xuân Thụ - Hồ Khánh Lâm. Kĩ thuật vi xử lý và máy vi tính. Nhà xuất bản giáo dục, năm 2000.

6. Kain Richard. Advance Computer architecturre: A system Design Approach. Prentice Hall USa, 1996

7. Hoàng Hồng. Giáo trình tin học văn phòng. Nhà xuất bản Giao thông vận tải, năm 2002.

8. Hayes John. Computer Architecturre and Orgnizassion. McGaw Hill USA. 1998

9. Nguyễn Nam Trung. Cấu trúc máy tính và thiết bị ngoại vi. Nhà xuất bản Khoa học và kỹ thuật, năm 2003.
10. Nguyễn Xuân Phong - Trương Lê Ngọc Thường. Tin học đại cương. Nhà xuất bản Thanh niên, năm 2002.

11. Nguyễn Xuân Huy. Thuật toán . Nhà xuất bản Thống kê, năm 1988

12. Xuân Thành. Internet, Explorer 6.0. Nhà xuất bản Thống kê, năm 2002.