

Mục lục

Chương 1. Giải thuật	7
1.1. Khái niệm giải thuật	7
1.2. Các tính chất của giải thuật	8
1.3. Các cách viết giải thuật.....	8
1.3.1. Liệt kê từng bước.....	8
1.3.2. Lưu đồ.....	8
1.3.3. Giả mã lệnh.....	10
1.4. Ví dụ	10
Câu hỏi và bài tập	11
Chương 2. Căn bản về Turbo Pascal	13
2.1. Khởi động và kết thúc phiên làm việc với Turbo Pascal.....	14
2.1.1. Các tệp tin cần thiết	14
2.1.2. Khởi động Turbo Pascal	15
2.1.3. Kết thúc.....	15
2.2. Các thao tác xử lý tệp tin	15
2.3. Cấu trúc chương trình Turbo Pascal.....	15
2.3.1. Giới thiệu cấu trúc chung	15
2.3.2. Bảng ký tự	16
2.3.3. Từ khóa, tên chuẩn và qui tắc đặt tên	16
2.4. Các thao tác soạn thảo và thực thi chương trình	17
2.4.1. Di chuyển con trỏ	17
2.4.2. Chọn văn bản	17
2.4.3. Sao chép, di chuyển, xóa văn bản.....	17
2.4.4. Các bước thực thi chương trình	18
2.5. Chương trình Pascal đơn giản	18
2.5.1. Bài toán.....	18
2.5.2. Xác định các biến	18
2.5.2. Soạn thảo chương trình.....	18
2.5.3. Biên dịch và thực thi chương trình	18
2.6. Các câu lệnh cơ bản.....	19
2.6.1. Lệnh gán	19
2.6.2. Các câu lệnh viết ra màn hình	19
2.6.3. Các câu lệnh nhập dữ liệu	19
2.6.4. Cách viết có qui cách.....	20
2.7. Kiểu dữ liệu và các phép toán	20
2.7.1. Khai báo biến.....	20
2.7.2. Kiểu số nguyên	20
2.7.3. Kiểu số thực.....	21
2.7.4. Các hàm số học chuẩn	21

2.7.5. Kiểu ký tự	21
2.7.6. Kiểu logic	22
Câu hỏi và bài tập	22
Chương 3. Cấu trúc điều khiển cơ bản	24
3.1. Cấu trúc lệnh tuần tự	24
3.2. Cấu trúc rẽ nhánh.....	24
3.2.1. Cấu trúc If .. then	24
3.2.2. Cấu trúc Case ... of	26
3.3. Cấu trúc lặp.....	28
3.3.1. Cấu trúc For .. to	28
3.3.2. Cấu trúc Repeat và While.....	30
Câu hỏi và bài tập	31
Chương 4. Chương trình con.....	34
4.1. Các khái niệm cơ bản	34
4.2. Cấu trúc chương trình con	34
4.2.1. Chương trình con hàm.....	35
4.2.2. Chương trình con thủ tục.....	35
4.3. Biến toàn cục, biến địa phương	36
4.4. Tham số chương trình con.....	37
4.5. Chương trình con đệ qui.....	38
4.5.1. Khái niệm đệ qui	38
4.5.2. Viết chương trình con đệ qui.....	38
4.5.3. Mô tả hoạt động của chương trình con đệ qui.....	39
4.6. Các hàm và thủ tục của unit CRT.....	39
Câu hỏi và bài tập	40
Chương 5. Lập trình xử lý giao diện	43
5.1. Giới thiệu bảng mã ký tự mở rộng	43
5.2. Các hàm xử lý phím.....	43
5.3. Tạo và xử lý giao diện	45
5.3.1. Tạo cửa sổ.....	45
5.3.2. Tạo thực đơn.....	46
Câu hỏi và bài tập	49
Chương 6. Dữ liệu kiểu mảng	50
6.1. Khái niệm và phân loại mảng.....	50
6.2. Mảng một chiều.....	50
6.2.1. Khai báo mảng.....	50
6.2.2. Truy xuất phần tử	51
6.2.3. Tìm kiếm phần tử	52
6.2.4. Sắp xếp mảng	53
6.3. Mảng hai chiều	54

6.3.1. Khai báo mảng hai chiều	54
6.3.2. Một số bài toán về ma trận	55
Câu hỏi và bài tập	56
Chương 7. Dữ liệu kiểu xâu ký tự	59
7.1. Khái niệm xâu ký tự	59
7.2. Khai báo và truy xuất	59
7.3. Các hàm về xâu ký tự	60
7.4. Bài toán.....	60
Câu hỏi và bài tập	61
Chương 8. Dữ liệu kiểu bản ghi	63
8.1. Khái niệm về bản ghi.....	63
8.2. Khai báo bản ghi.....	63
8.3. Truy xuất thành phần trong bản ghi	63
8.4. Ví dụ áp dụng	64
Câu hỏi và bài tập	65
Chương 9. Dữ liệu kiểu tệp tin	66
9.1. Khái niệm và phân loại tệp tin.....	66
9.2. Tệp có kiểu	66
9.2.1. Khai báo.....	67
9.2.2. Các hàm và thủ tục xử lý tệp tin.....	67
9.2.3. Ví dụ áp dụng	69
9.3. Tệp văn bản	70
9.3.1. Khai báo.....	70
9.3.2. Ví dụ áp dụng	70
9.4. Tệp không định kiểu.....	76
9.4.1. Khai báo.....	76
9.4.2. Các hàm và thủ tục xử lý tệp tin.....	76
9.4.3. Ví dụ áp dụng	76
Câu hỏi và bài tập	77
Tài liệu tham khảo	78

Lời nói đầu

Trong hệ thống kiến thức chuyên ngành trang bị cho sinh viên, giáo trình Lập trình cấu trúc góp phần cung cấp những nội dung kiến thức chung nhất về lập trình, thực nghiệm trên ngôn ngữ lập trình Pascal. Giúp người học nắm bắt được những thuật toán cơ bản để giải quyết những bài toán trong từng chương và biết vận dụng những giải thuật đó để giải quyết các bài toán yêu cầu cao hơn.

Để phục vụ công tác giảng dạy và học tập, qua những năm trực tiếp giảng dạy, đồng thời tham khảo thêm một số tài liệu khác, tôi biên soạn cuốn giáo trình Lập trình cấu trúc nhằm cung cấp tới người học một số kiến thức cơ bản nhất về lĩnh vực này.

Mặc dù đã cố gắng nhiều trong biên soạn, do kiến thức và thời gian có hạn, nên giáo trình không tránh khỏi những thiếu sót. Tôi mong muốn nhận được sự thông cảm và ý kiến đóng góp của các thầy cô, các bạn học sinh, sinh viên và bạn đọc để cuốn giáo trình được tốt hơn.

Liên hệ: Vũ Văn Minh

Khoa Công nghệ thông tin

Trường Cao đẳng Công nghiệp Nam Định

Chương 1. Giải thuật

Mục tiêu bài học:

- Xác định được tập dữ liệu vào, dữ liệu ra, biết phân chia công việc thành các bước. Sau mỗi bước bao giờ cũng cho 1 kết quả xác định không phụ thuộc vào người hay máy thực hiện mà chỉ phụ thuộc vào dữ liệu vào.

- Chỉ ra tính khả thi của các bước thực hiện. Tính dừng sau một số hữu hạn bước. Nắm được 3 cách biểu diễn thuật toán.

Trong toán học, để giải quyết một bài toán ta luôn tìm cách áp dụng những định lý, tính chất, tiên đề, hệ quả... nhằm biến đổi dữ kiện đề bài để đưa về kết quả cuối cùng. Trong tin học việc giải các bài toán trước hết là đi tìm thuật giải của bài toán đó.

1.1. Khái niệm giải thuật

Thuật giải giải một bài toán nào đó là một dãy các thao tác đơn giản được sắp xếp theo một trình tự xác định rõ ràng và kết thúc sau một số hữu hạn bước nhằm biến đổi dữ liệu vào (input) của một bài toán thành dữ liệu ra (output) mô tả lời giải bài toán đó.

Ví dụ. Bài toán tìm UCLN của hai số nguyên dương.

Cho: hai số nguyên dương a, b ;

Cần biết: UCLN của hai số a, b ?

Ý tưởng để giải quyết bài toán

- Để giải quyết bài toán này ta phải hiểu UCLN là gì?

➤ UCLN của a, b là một số lớn nhất mà cả a và b đều chia hết.

- Hình dung ra giải thuật để giải quyết

Đối với mỗi bài toán có rất nhiều cách giải quyết khác nhau, vấn đề ta tìm ra được một cách giải quyết tối ưu như thế nào? vậy ta cần phải phân tích thuật toán.

➤ Cho biến x bắt đầu từ a hoặc b lùi tới 1, nếu cả a và b đều chia hết cho x thì x là UCLN của a và b .

➤ Thực hiện vòng lặp {nếu $a > b$ thì $a = a - b$ nếu $a < b$ thì $b = b - a$ } đến khi nào $a = b$ thì UCLN là a hoặc b .

➤ Thực hiện phép toán chia dư, như trình bày dưới đây:

Thuật toán chia dư giải bài toán trên như sau:

<p>- Input: a, b nguyên dương - Output: UCLN của a và b Bước 1: nhận vào số a và số b Bước 2: chia a cho b tìm số dư r</p>
--

Bước 3: Nếu $r = 0$ thì chuyển đến bước 5
Bước 4: gán giá trị b cho a , gán giá trị r cho b . Quay về bước 2
Bước 5: thông báo kết quả UCLN là b ;
Bước 6: Kết thúc.

- Phân tích cả 3 thuật toán và chỉ ra ưu điểm của từng thuật toán.

1.2. Các tính chất của giải thuật

1.2.1. Có dữ liệu vào (input)

Mỗi thuật giải có thể có một hoặc nhiều dữ liệu vào.

1.2.2. Xác định dữ liệu ra (output)

Sau khi thuật giải đã được thực hiện xong, tùy theo chức năng mà thuật giải đảm nhiệm ta có thể thu được một số dữ liệu ra xác định

1.2.3. Tính xác định

Tính xác định của thuật giải đòi hỏi ở mỗi bước các thao tác phải hết sức rõ ràng, không thể gây nên sự nhập nhằng, lẫn lộn, tùy tiện.

1.2.4. Tính kết thúc (tính dừng)

Thuật giải phải dừng sau một số hữu hạn bước thực hiện.

1.2.5. Tính hiệu quả

Một yêu cầu quan trọng là với input đúng thuật giải phải cho output đúng.

1.2.6. Tính phổ dụng

Một thuật giải được xem là có tính phổ dụng cao nếu nó có thể giải bất kỳ bài toán nào trong một lớp lớn các bài toán.

1.3. Các cách viết giải thuật

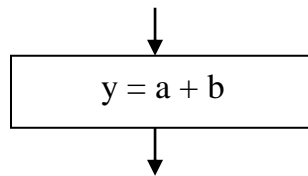
1.3.1. Liệt kê từng bước

- Thuật giải UCLN ở trên được diễn tả theo hình thức liệt kê từng bước.

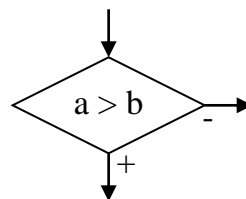
1.3.2. Lưu đồ (sơ đồ khối)

Lưu đồ là công cụ giúp ta diễn tả thuật giải một cách trực quan. Lưu đồ được tạo bởi 4 loại khối nối với nhau bằng các cung

- Khối thao tác được biểu diễn bằng hình chữ nhật. Trong khối này ta viết một hoặc một dãy các thao tác như gán trị, tính toán biểu thức v.v... Khối thao tác có 1 cung đi đến và 1 cung đi ra:



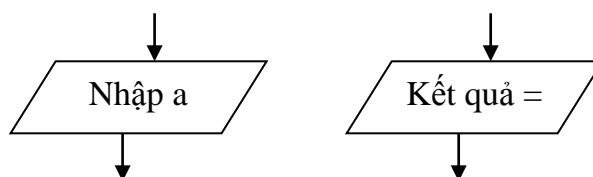
- Khối điều kiện được biểu diễn bằng hình thoi. Trong khối này ta viết một biểu thức logic. Tùy theo giá trị của biểu thức logic là đúng hay sai mà việc thực hiện tiếp theo sẽ được chỉ dẫn bởi một trong hai cung đi ra mang dấu + (cho trường hợp đúng) hoặc dấu - (cho trường hợp sai). Như vậy khối điều kiện có 1 cung đi đến và 2 cung đi ra:



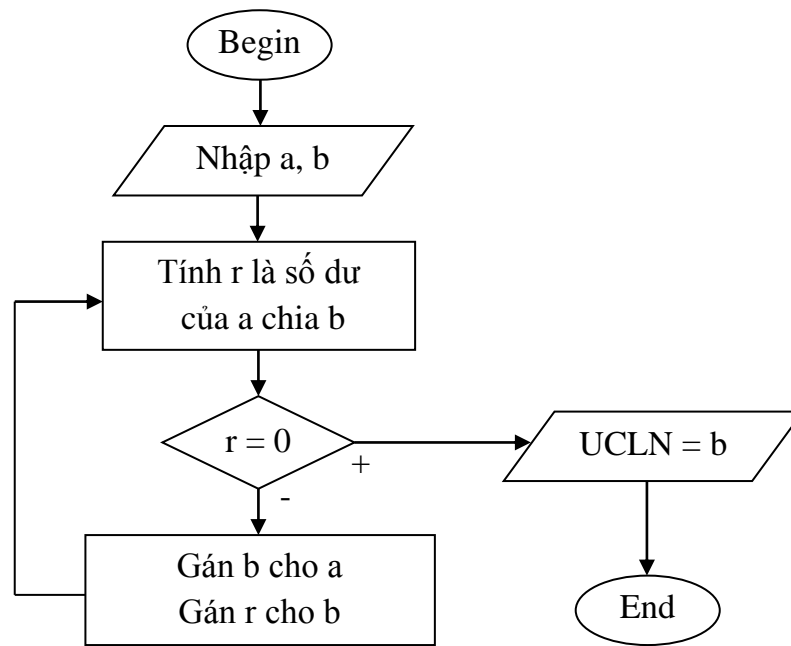
- Hai khối đặc biệt là khối bắt đầu và khối kết thúc được biểu diễn bằng hình ellip chỉ rõ điểm bắt đầu và điểm kết thúc (điểm dừng) của thuật giải. Khối bắt đầu không có cung đi đến và có 1 cung đi ra. Khối kết thúc có 1 cung đi đến và không có cung đi ra:



- Hình bình hành dùng để biểu diễn thao tác vào ra của thuật toán:



Dùng lưu đồ diễn tả thuật giải Oclid tìm UCLN của hai số nguyên dương:



1.3.3. Giải mã lệnh

Khi thể hiện thuật giải bằng giải mã lệnh, ta sẽ vay mượn các cú pháp của một ngôn ngữ lập trình nào đó. Ở đây chúng ta vay mượn các khái niệm của ngôn ngữ lập trình pascal.

1.4. Ví dụ

Người A nghĩ trong đầu một số nguyên X trong đoạn từ 1 đến 100. Người B hỏi, người A trả lời hoặc đúng hoặc sai. Sau không quá 7 lần hỏi đáp người B biết số X là số nào. Viết thuật giải cho bài toán này.

1.4.1. Dùng ngôn ngữ liệt kê từng bước

Bước 1 Gán $T := 1$; $P := 100$;

Bước 2 Lấy thương nguyên của tổng $(T + P)$ chia cho 2 rồi gán cho G.

Bước 3 Kiểm tra điều kiện $X > G$ nếu đúng thì chuyển đến bước 4, còn sai thì chuyển đến bước 5;

Bước 4 Lấy $G + 1$ gán cho T; chuyển đến bước 6;

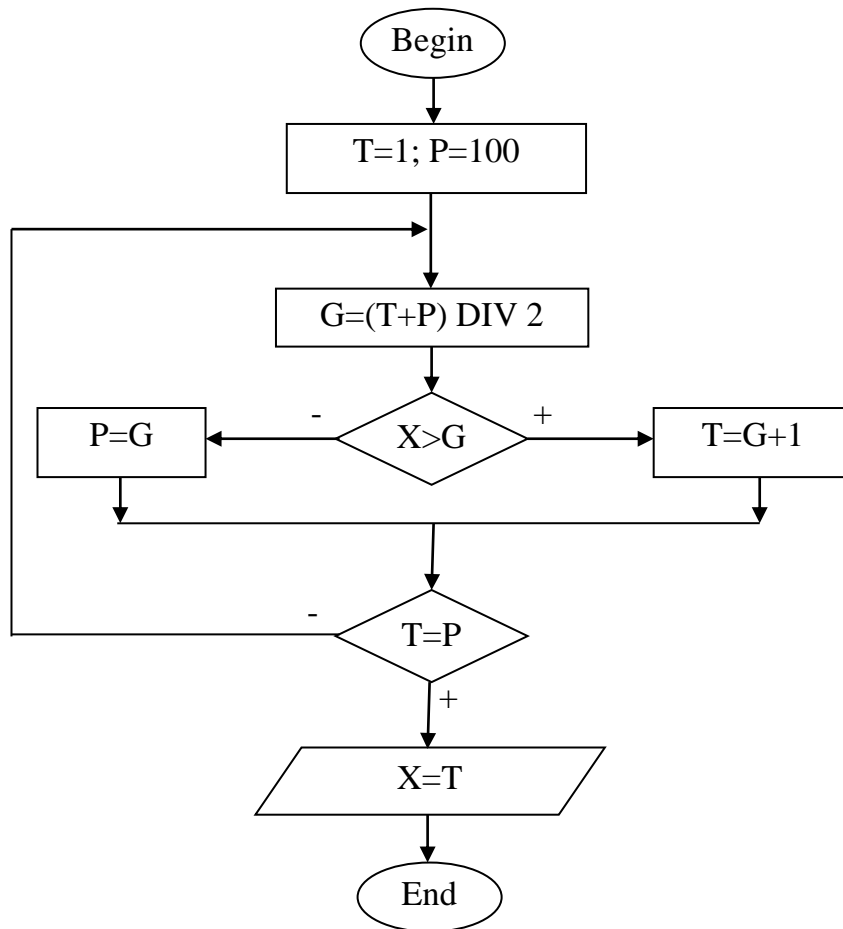
Bước 5 Lấy G gán cho P;

Bước 6 Kiểm tra điều kiện $T = P$ nếu sai thì chuyển đến bước 2;

Bước 7 Trả lời $X = T$;

Bước 8 Kết thúc;

1.4.2. Dùng sơ đồ khối



1.4.3. Dùng giả mã lệnh

Biến nguyên không âm T, P, G, X;

Bắt đầu

T :=1; P :=100;

Lặp

G := (P+T) div 2;

nếu X > G thì T :=G+1 ngược lại P :=G;

đến khi T = P;

Thông báo X = P;

Kết thúc.

Câu hỏi và bài tập

1. Thuật giải là gì? Thuật giải có những tính chất cơ bản nào?
2. Có mấy cách biểu diễn thuật giải

3. Hãy viết thuật giải vẽ đồ thị của hàm số $y = |ax|$ (với a khác 0) thông qua đồ thị của hàm số $y = ax$.

4. Trình bày tính chất xác định của thuật giải và nêu rõ ý nghĩa của tính chất này

5. Hãy phát biểu thuật giải để giải bài toán sau: "Có một số quả táo. Dùng cân hai đĩa (không có quả cân) để xác định quả táo nặng nhất"(giả sử mỗi đĩa cân có thể đựng được nửa số quả táo).

6. Xác định dữ liệu vào và dữ liệu ra cho các thuật giải sau đây:

a. Rút gọn một phân số.

b. Kiểm tra xem ba số cho trước a , b và c có thể là độ dài ba cạnh của một tam giác hay không?

Tính trung bình cộng của hai số.

Dùng một cốc phụ để trao nước ở hai cốc cho trước.

Tìm chu vi và diện tích của hình tròn có bán kính cho trước

7. Có hai bình A và B. Bình A có dung tích 8 lít, bình B có dung tích 5 lít. Trình bày các bước thực hiện để lấy được 2 lít nước.

8. Có 3 bình A, B, C. Bình A có dung tích 8 lít và đựng đầy 8 lít rượu, bình B có dung tích 5 lít, bình C có dung tích 3 lít. Trình bày các bước thực hiện để có được 4 lít rượu ở bình A và 4 lít rượu ở bình B.

9. Một người có 1 con gấu, 1 con dê và 1 cái bắp cải. Nếu không có người ở bên chúng thì con gấu sẽ ăn thịt con dê hoặc con dê sẽ ăn bắp cải. Thuyền chỉ có thể chở được người đó với con gấu hoặc con dê hoặc bắp cải. Người đó làm thế nào để mang chúng sang sông.

10. Có 4 người phải qua một cái cầu, trời tối họ chỉ có một chiếc đèn. Cầu chỉ đi được tối đa 2 người. Như vậy qua cầu phải có đèn và nhiều nhất là chỉ đi được 2 người cùng một lúc. Biết rằng người thứ nhất đi qua cầu hết 1 phút. Người thứ hai đi qua cầu hết 2 phút. Người thứ ba đi qua cầu hết 5 phút. Người thứ tư đi qua cầu hết 10 phút. Hãy tìm cách cho 4 người này qua cầu sao cho tổng số thời gian ít nhất

Chương 2. Căn bản về Turbo Pascal

Mục tiêu bài học:

- Biết khai thác môi trường làm việc của Turbo Pascal.
- Nắm được cấu trúc của 1 chương trình Pascal đơn giản.
- Biết viết 1 chương trình Pascal đơn giản với các câu lệnh đơn giản.

Sau khi đã có thuật giải cho bài toán, một câu hỏi đặt ra là làm thế nào để máy thực thi thuật giải đó để đưa ra output của bài toán? Chính là ta cần một công cụ lập trình. Turbo Pascal là một công cụ như thế.

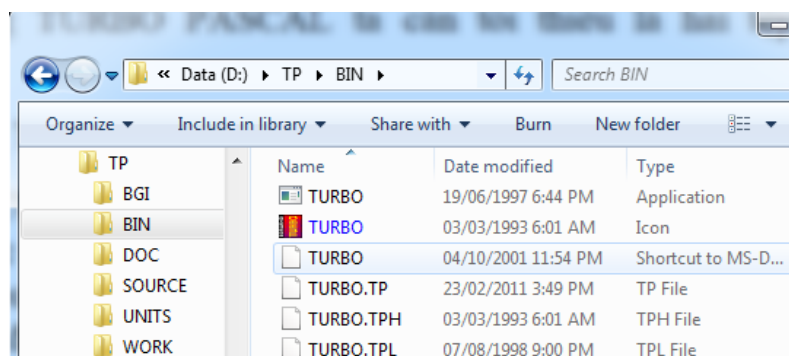
Sau khi chương trình đã soạn thảo xong, ta dùng TP để kiểm tra xem trong chương trình đó có lỗi cú pháp (viết sai quy cách câu lệnh hoặc mô tả) hay không.

Khi không còn các thông báo lỗi nữa, nghĩa là chương trình đã đúng đắn về mặt cú pháp, ta có thể chạy chương trình, nạp dữ liệu và thu nhận kết quả.

2.1. Khởi động và kết thúc phiên làm việc với Turbo Pascal

2.1.1. Các tệp tin cần thiết

Để sử dụng TURBO PASCAL ta cần tối thiểu là hai tệp: TURBO.EXE và TURBO.TPL.



Ngoài ra, để làm việc được đầy đủ với Turbo Pascal, cần phải có đầy đủ các thư mục sau:

BGI thường lưu trữ các tệp tin làm việc về đồ họa

BIN thư mục chứa các tệp tin phục vụ chương trình TP.

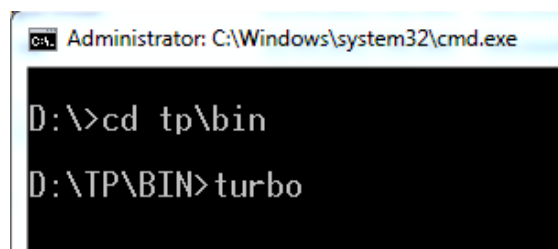
SOURCE chứa các chương trình mẫu

UNITS chứa các thư viện chương trình

WORK chứa các tệp tin được sinh ra khi thi hành chương trình.

2.1.2. Khởi động Turbo Pascal

Khởi động TURBO PASCAL, giả sử ta đang ở thư mục có hai tệp nói trên ta gõ TURBO tiếp theo là phím ENTER (có nhiều cách khởi động TURBO PASCAL, nếu trên màn hình Windows chúng ta thấy biểu tượng của TURBO PASCAL thì ta chỉ cần kích chuột vào đó)



2.1.3. Kết thúc

Để kết thúc phiên làm việc, cần phải tắt chương trình ứng dụng. Trước khi tắt chương trình ứng dụng, chú ý lưu các tệp tin đang làm việc và nhấn tổ hợp phím Alt+X hoặc vào thực đơn File, chọn Exit.

2.2. Các thao tác xử lý tệp tin

Ctrl + N hoặc File \ New	Tạo tệp tin mới
F3 hoặc File \ Open	Mở tệp tin có sẵn
F2 hoặc File \ Save	Lưu trữ tệp tin
Alt + F3 hoặc File \ Close	Đóng cửa sổ tệp tin hiện hành
F6	Chuyển đổi giữa các cửa sổ tệp tin
F5	Phóng to cửa sổ tệp tin hiện hành

2.3. Cấu trúc chương trình Turbo Pascal

2.3.1. Giới thiệu cấu trúc chung

```
{Tieu de chuong trinh}

Program ten_chuong_trinh;

{Cac khai bao}

Uses CRT, GRAPH; {Khai bao thu vien}

LABEL L1, L2; {Khai bao nhan}

CONST C1, C2; {Khai bao hang}

TYPE Nguyen = Integer; {Khai bao kieu du lieu theo y nguoi su dung}

VAR v1, v2 : Nguyen ; {Khai bao bien}

BEGIN

    ... {Cac lenh}

END.
```

2.3.2. Bảng ký tự

Bảng chữ gốc của TP là bảng ASCII (American Standard Codes for Information Interchanges). ASCII có 128 kí hiệu bao gồm bộ chữ cái la tinh, các chữ số, các dấu và một số kí hiệu đặc biệt.

Bộ chữ cái La tinh

Gồm 26 chữ cái tiếng Anh in hoa A-Z và in thường a-z. TP không phân biệt chữ thường và chữ hoa trong tên các đối tượng, ngoại trừ trường hợp trong các xâu văn bản.

Bộ chữ số

Gồm các chữ số thập phân: 0, 1, ... , 9. Để tránh lẫn 0 (chữ số không) và O (chữ O) TP quy định gạch chéo trong chữ số không.

Những dấu phép toán số học

+ (cộng), - (trừ), * (nhân), / (chia)

Các dấu so sánh

= (bằng) , > (lớn hơn) , < (nhỏ hơn),

>= (lớn hơn hoặc bằng), <= (nhỏ hơn hoặc bằng), <> (khác)

Những kí hiệu khác:

. , ; : ' " : ! @ # \$ % \ ^ & () [] { }

Trong bảng chữ có một kí hiệu gọi là dấu cách hay khoảng trống. Dấu cách được dùng để tạo một khoảng trắng ngăn cách các từ.

2.3.3. Từ khóa, tên chuẩn và qui tắc đặt tên

Từ khóa:

Trong từ vựng Pascal có một số từ dành riêng mà nghĩa đã được quy định sẵn, chẳng hạn program, begin, end, mà ta đã biết được gọi là từ khóa. chúng tôi giới thiệu dưới đây một số từ khóa sẽ dùng trong giáo trình này
AND ARRAY BEGIN CASE CONST DIV DO DOWNTON ELSE END FOR
FUNCTION GOTO IF IN LABEL MOD NOT OF OR PROCEDURE PROGRAM
REPEAT STRING THEN TO TYPE UNTIL USES VAR WHILE

Tên chuẩn:

Những tên được đặt sẵn trong TP, chẳng hạn pi, byte, word, integer, longint, read, readln, write, writeln, char, boolean, ... được gọi là tên chuẩn.

Qui tắc đặt tên:

Tên chương trình, tên biến và các tên sau này ta gặp phải bắt đầu bằng chữ cái, tiếp theo có thể có một số kí tự nữa nhưng chỉ lấy trong tập gồm chữ cái, chữ số, dấu gạch nối. Tên có thể nhận đến 127 kí tự.

Chú ý: TP không phân biệt ký tự hoa hoặc ký tự thường. Ví dụ các cách viết sau là có ý nghĩa như nhau: Begin, BEGIN, begin, beGIN, ...

2.4. Các thao tác soạn thảo và thực thi chương trình

2.4.1. Di chuyển con trỏ

- Các phím lên, xuống, phải, trái (có hình những mũi tên ở bên phải bàn phím): dịch con trỏ từng kí tự theo chiều mũi tên.

- Ctrl và phím mũi tên sang trái (phải) : dịch chuyển con trỏ theo từng từ sang trái (phải) của dòng văn bản.

- Home: đưa con trỏ về đầu dòng.

- End: đưa con trỏ về cuối dòng.

- PgUp (PgDn): dịch con trỏ lên (xuống) theo từng trang màn hình.

- Ctrl-PgUp hoặc Ctrl-PgDn: đưa con trỏ về đầu hoặc cuối văn bản.

2.4.2. Chọn văn bản

- Ctrl-K, B. Đánh dấu đến đầu khối.

- Ctrl-K, K. Đánh dấu đến cuối khối.

- Ctrl-K, Y. Xóa khối dòng đã đánh dấu.

- Nhấn phím Shift + Các phím di chuyển con trỏ

2.4.3. Sao chép, di chuyển, xóa văn bản

- Ctrl-K, C hoặc Ctrl + Ins Sao chép khối dòng tới vị trí mới của con trỏ.

- Ctrl-K, V hoặc Shift + Ins Chuyển khối dòng tới vị trí mới của con trỏ.

- Ctrl-K, W. Ghi khối dòng vào một tệp.

- Ctrl-K, R. Đọc một tệp từ đĩa vào và xen vào chỗ con trỏ.

- Phím Del để xóa một kí tự tại vị trí hiện thời của con trỏ.

- Phím lùi (Backspace) để xóa đi một kí tự nằm bên trái con trỏ.

- Phím INSERT để chọn chế độ chèn hoặc đè.

- Ctrl-Y. Xóa cả dòng đang chứa con trỏ.

- Ctrl-Q, Y. Xóa từ vị trí con trỏ đến cuối dòng

- Ctrl-Q, A. Tìm kiếm một dãy kí tự và thay thế.

2.4.4. Các bước thực thi chương trình

- Alt + F9: biên dịch chương trình

- Ctrl + F9: thi hành chương trình

- Theo mặc định, Turbo Pascal dịch chương trình nguồn ra tệp thi hành (exe) ở trên RAM và thực thi trên RAM. Người lập trình có thể lấy tệp exe đó bằng cách chọn lại nơi lưu trữ tệp exe khi dịch của TP bằng cách chọn thực đơn COMPILE, chọn DESTINATION là DISK và chọn nơi lưu trữ bằng cách vào thực đơn OPTION chọn DIRECTORY, thiết lập đường dẫn tại dòng EXE and TPU

2.5. Chương trình Pascal đơn giản

2.5.1. Bài toán

Tính chu vi và diện tích hình vuông có độ dài cạnh là một số nguyên được nhận vào từ bàn phím.

Xây dựng thuật giải

Bước 1. Nhận vào a;

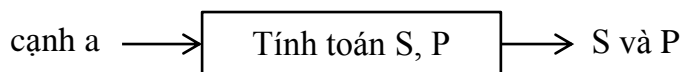
Bước 2. Viết ra giá trị của biểu thức chu vi là $4 * a$;

Bước 3. Viết ra giá trị của biểu thức diện tích là $a * a$;

Bước 4. Kết thúc

2.5.2. Xác định các biến

a được nhận giá trị mỗi khi thực hiện chương trình (phù hợp với cạnh hình vuông khác nhau) nên a là biến



2.5.2. Soạn thảo chương trình

```
Program HìnhVuong;  
Var a : Word;  
Begin  
    readln (a);  
    writeln ('Chu vi S= ', 4*a);  
    write ('Diện tích P= ', a*a);  
    Readln;  
End.
```

2.5.3. Biên dịch và thực thi chương trình

Gõ phím F9 để máy dịch chương trình ta vừa viết ở trên sang mã máy. Nếu có lỗi thì máy thông báo cho ta sửa. Sửa xong lại gõ phím F9 để máy báo lỗi tiếp theo (nếu

còn). Khi nào gõ phím F9 mà máy không báo có lỗi thì ta cho chạy chương trình bằng cách gõ tổ hợp hai phím Ctrl-F9 (giữ phím Ctrl và gõ phím F9) sau đó gõ giá trị của a từ bàn phím (thực hiện lệnh Readln(a)).

2.6. Các câu lệnh cơ bản

2.6.1. Lệnh gán

Tên_biến := Biểu_thức;

Thực hiện lệnh này, máy tính giá trị của biểu thức ở bên phải dấu gán (:=), sau đó gán giá trị này cho biến ở bên trái dấu gán, tức là đưa giá trị đó vào địa chỉ được kí hiệu bởi tên biến ở bên trái dấu gán. Sau lệnh gán, giá trị cũ của biến bị mất và biến nhận giá trị mới.

Ví dụ ta khai báo Var a : Integer ;

Ở phần thân chương trình ta có hai lệnh gán a := - 6; a := a + 8 thì thực hiện lệnh gán thứ nhất a có giá trị là -6.

Tiếp đến sau khi thực hiện lệnh gán thứ hai biến a có giá trị là 2. Giá trị cũ là -6 bị mất, biến nhận giá trị mới.

Chú ý: Biểu thức và biến phải cùng kiểu dữ liệu

2.6.2. Các câu lệnh viết ra màn hình

Write(biến, hằng, biểu thức);

Viết lên màn hình tại vị trí hiện hành các giá trị của biến, hằng, biểu thức. Sau đó con trỏ đặt ở cuối dòng

Writeln(biến, hằng, biểu thức);

Viết lên màn hình tại vị trí hiện hành các giá trị của biến, hằng, biểu thức. Sau đó con trỏ đặt ở đầu dòng tiếp theo

Writeln;

Xuống dòng, về đầu dòng tiếp theo.

Ví dụ:

Write('Nhập cạnh hình vuông a=');

readln (a);

2.6.3. Các câu lệnh nhập dữ liệu

Read(biến, biến, biến);

Nhập giá trị cho các biến, mỗi biến cách nhau ít nhất một dấu cách.

Readln(biến, biến, biến);

Nhập giá trị cho các biến, mỗi biến cách nhau ít nhất một dấu cách, nhập xong ấn phím Enter.

Readln;

Dừng chương trình, chờ ấn phím Enter.

2.6.4. Cách viết có qui cách

Write(giá_trị : m);	Hiện lên màn hình tại vị trí hiện hành giá_trị ở một vùng có m ký tự được căn phải. Nếu giá_trị là số thực thì được làm tròn.
Writeln(giá_trị : m);	Tương tự, xuống dòng.
Write(giá_trị : m:n);	Chỉ áp dụng cho số thực, hiển thị lên màn hình có m ký tự trong đó có n ký tự là phần lẻ của giá_trị
Ví dụ	
Write(pi);	Cho kết quả 3.1492635432E00
Writeln(pi:4:2);	Cho kết quả: 3.14

2.7. Kiểu dữ liệu và các phép toán

2.7.1. Khai báo biến

Mỗi biến có một kiểu dữ liệu khác nhau tùy theo mục đích sử dụng biến của lập trình viên. Các biến phải được khai báo trước khi sử dụng.

Cú pháp: **ten_bien : KIEU_DU_LIEU;**

ten_bien được đặt theo quy tắc đặt tên đã trình bày ở mục 2.3.3

KIEU_DU_LIEU là các kiểu dữ liệu chuẩn hoặc kiểu của người sử dụng định nghĩa.

2.7.2. Kiểu số nguyên

Có 5 kiểu số nguyên với kích thước ô nhớ khác nhau, tùy theo tính chất sử dụng biến trong chương trình, người lập trình khai báo biến cho phù hợp (tránh lãng phí bộ nhớ).

Tên kiểu	Phạm vi giá trị	Kích thước ô nhớ (byte)
Byte	0 ... 255	1
ShortInt	-128 ... 127	1
Word	0 ... 65535	2
Integer	-32768 ... 32767	2
LongInt	- 2147483648 ... 2147483647	4

Các phép toán với số nguyên

- Phép toán số học: + (cộng) - (trừ) * (nhân) / (chia)
 DIV (phép chia nguyên) MOD (phép chia dư)
- Phép so sánh: < > <= >= = <>
- Phép toán trên bit: AND (và) OR (hoặc) NOT (phủ định)
 SHL (dịch trái) SHR (dịch phải)

2.7.3. Kiểu số thực

Trong chế độ làm việc mặc định của TP, ta chỉ có thể làm việc được với kiểu REAL, muốn làm việc với kiểu dữ liệu khác, ta phải có **khai báo xa**.

Tên kiểu	Phạm vi giá trị	Kích thước ô nhớ (byte)
Single	1.5E-45 ... 3.4E38	4
Real	2.9E-39 ... 1.7E38	6
Double	5.0E324 ... 1.7E308	8

Các phép toán trên kiểu số thực

- Phép toán số học: + (cộng) - (trừ) * (nhân) / (chia)
- Phép so sánh: < > <= >= = <>

Vì số thực được lưu trữ trong các ô nhớ dạng cấu trúc nên không có phép toán DIV, MOD và các phép toán xử lý BIT.

2.7.4. Các hàm số học chuẩn

Abs(x): Lấy giá trị tuyệt đối của x. Giá trị của hàm cùng kiểu với kiểu đối số x.

Sqr(x) : Cho x bình phương. Giá trị của hàm cùng kiểu với kiểu của đối số x.

Sqrt(x): Lấy căn bậc 2 của x. Giá trị của hàm thuộc kiểu thực.

Int(x) : Cho phần nguyên của x. Giá trị của hàm thuộc kiểu thực.

Trunc(x): Cho phần nguyên của x. Giá trị của hàm thuộc kiểu nguyên.

Round(x): Làm tròn x đến số nguyên gần x nhất. Giá trị của hàm có kiểu nguyên

Sin(x): Cho giá trị của $\sin x$, ở đây x tính bằng đơn vị là radian.

Cos(x): Cho giá trị của $\cos x$, ở đây x tính bằng đơn vị là radian.

Arctan(x): Cho góc tính bằng radian có tang bằng x.

Exp(x): Tính e^x . Giá trị của hàm thuộc kiểu thực.

Ln(x): Tính lôgarit cơ số e của x. Giá trị của hàm thuộc kiểu thực.

Frac(x): Cho phần thập phân của x. Giá trị của hàm thuộc kiểu thực

Random(x): đối số x phải là nguyên. Hàm này cho ngẫu nhiên một số nguyên lớn hơn hoặc bằng 0 và nhỏ hơn x.

2.7.5. Kiểu ký tự

Kiểu ký tự (CHAR) để biểu diễn các giá trị là các ký tự thuộc bảng chữ cái ASCII. Bảng mã ASCII có 256 ký tự được gán thứ tự từ 0...255, ví dụ ký tự 'A' có mã là 65, 'a' có mã là 97.

Bảng mã ASCII được chia thành 3 loại ký tự: ký tự điều khiển, ký tự văn bản, ký tự đồ họa:

Loại ký tự	Mã ký tự	Ví dụ
Ký tự điều khiển	0..31	xuống dòng: 13 về đầu dòng: 10 tab: 8
Ký tự văn bản	32..127	'A': 65; 'a':97; '0':48
Ký tự đồ họa	128..255	' ':179

Các hàm và thủ tục với kiểu Char

Readkey: hàm đọc một ký tự từ bộ đệm bàn phím, nếu bộ đệm bàn phím rỗng, hàm chờ người sử dụng nhấn phím Enter.

Keypress: hàm kiểm tra nếu có một phím được nhấn thì hàm trả về giá trị TRUE, ngược lại hàm cho giá trị FALSE.

chr(x) hàm trả về ký tự có mã là x

ord(c) hàm trả về mã của ký tự c

pred(c) hàm trả về ký tự đứng trước ký tự c.

succ(c) hàm trả về ký tự đứng sau ký tự c.

2.7.6. Kiểu logic

Kiểu logic (BOOLEAN) để biểu diễn trạng thái đúng hoặc sai của phép toán; biến có kiểu logic nhận một trong hai giá trị là TRUE hoặc FALSE. Mặc định FALSE<TRUE

Các phép toán:

x	y	x and y	x or y	not y
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	TRUE
FALSE	TRUE	FALSE	TRUE	FALSE
FALSE	FALSE	FALSE	FALSE	TRUE

Câu hỏi và bài tập

1. Những tên sau đây viết đúng hay sai:

A, AA, BIC, 1B, St , A*, A-B, CHU VI, In, TO, READ, _10A, F(x), M[, \$Q, P^

2. Hãy lấy các ký tự đồ họa bằng cách ấn phím Alt với phím số bên phải bàn phím (ví dụ: Alt 219: giữ phím Alt, gõ các số 2, 1, 9 ở các phím phía bên phải bàn phím) để viết khối hình chữ nhật. Viết chương trình để máy in lên màn hình thông tin sinh viên như sau:

3. Viết chương trình tính diện tích xung quanh, diện tích toàn phần, thể tích hình lập phương khi cho độ dài cạnh là một số nguyên.
4. Viết chương trình tính giá trị của biểu thức $X^3 - 4X^2 + 3X - 5$ cho biết X là một số nguyên.
5. Lập chương trình tính chu vi và diện tích hình tròn theo bán kính r .
6. Viết chương trình tính diện tích, thể tích hình cầu khi cho bán kính.
7. Viết chương trình cho máy nhận vào ba cạnh a, b, c của tam giác ABC .
8. Tính độ dài các đường trung tuyến của tam giác ABC .
9. Tính bán kính đường tròn ngoại tiếp tam giác ABC .
10. Tính chu kỳ dao động của một con lắc đơn có chiều dài dây là l .
11. Biết cạnh huyền a , cạnh góc vuông b của tam giác vuông ABC . Tính bán kính đường tròn nội tiếp tam giác đó.
12. Cho tam giác vuông ABC có góc $A = 90^\circ$. Lập chương trình tính góc C , cạnh AC , cạnh AB khi cho biết cạnh huyền BC và góc B (góc B , góc C tính bằng độ và phút).
13. Viết chương trình giải tam giác ABC khi biết góc A , góc B và cạnh c (các góc cho bằng độ và phút).

Chương 3. Cấu trúc điều khiển cơ bản

Mục tiêu bài học:

- Hiểu được cú pháp và ngữ nghĩa của các câu lệnh rẽ nhánh, lệnh lặp.
- Biết vận dụng các câu lệnh này để giải các bài toán đơn giản.

3.1. Cấu trúc lệnh tuần tự

Thực hiện lần lượt các lệnh theo thứ tự khi lập trình, câu lệnh nào viết trước thực hiện trước, lệnh nào viết sau thực hiện sau.

Ví dụ:

Nên	Không nên
Write('Nhập giá trị a, b, c: '); Readln(a, b, c);	Readln(a, b, c); Write('Nhập giá trị a, b, c: ');
Readln(a, b, c); $p := (a+b+c)/2$; $s := \text{sqrt}(p*(p-a)*(p-b)*(p-c))$;	$s := \text{sqrt}(p*(p-a)*(p-b)*(p-c))$; $p := (a+b+c)/2$; Readln(a, b, c);

3.2. Cấu trúc rẽ nhánh

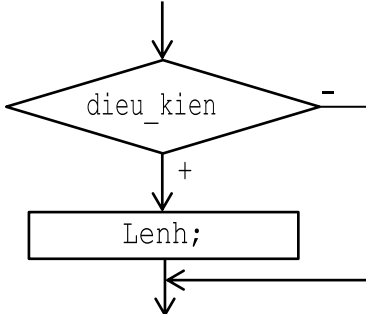
Trong quá trình giải bài toán, ta cần phải giải tổng quát, vì thế ta phải xét tất cả các trường hợp của bài toán có thể xảy ra.

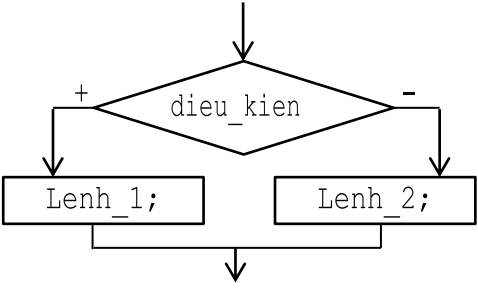
Ví dụ để giải bài toán $ax^2 + bx + c = 0$ ($a \neq 0$) ta phải xét các trường hợp của Delta:

- Trường hợp $\Delta = 0$
- Trường hợp $\Delta < 0$
- Trường hợp $\Delta > 0$

Vì thế cấu trúc rẽ nhánh rất hay dùng trong lập trình, sau đây ta xét hai cấu trúc rẽ nhánh cơ bản của TP.

3.2.1. Cấu trúc If ... then ...

Cú pháp	Lưu đồ	Nguyên tắc hoạt động
IF dieu_kien THEN Lenh;		Khi gặp câu lệnh này, TP kiểm tra điều kiện: <ul style="list-style-type: none">- Nếu điều kiện đúng thì thực hiện lệnh- Nếu điều kiện sai thì bỏ qua lệnh

<pre>IF dieu_kien THEN Lenh_1 ELSE Lenh_2;</pre>	 <pre>graph TD Entry(()) --> Decision{dieu_kien} Decision -- "+" --> Lenh1[Lenh_1;] Decision -- "-" --> Lenh2[Lenh_2;] Lenh1 --> Exit(()) Lenh2 --> Exit</pre>	<p>Khi gặp câu lệnh này, TP kiểm tra điều kiện:</p> <ul style="list-style-type: none"> - Nếu điều kiện đúng thì thực hiện lệnh 1; - Nếu điều kiện sai thì thực hiện lệnh 2;
--	---	---

Chú ý:

- Lenh, Lenh_1, Lenh_2 là lệnh đơn hoặc lệnh ghép. Nếu là lệnh ghép thì các lệnh phải được đặt trong cặp từ khóa BEGIN và END
- Lệnh trước từ khóa ELSE không có dấu chấm phẩy.

Ví dụ:

Viết chương trình Pascal tìm nghiệm của phương trình bậc 2: $ax^2 + bx + c = 0$ ($a > 0$)

Giải thuật:

- Xác định biến, tùy theo ý đồ lập trình của lập trình viên mà có thể khai báo các biến để sử dụng, với mục tiêu dễ hiểu và sử dụng ít ô nhớ nhất. Các biến nhận giá trị các hệ số là a, b, c; biến tính giá trị delta, nếu có thể có biến chứa nghiệm của phương trình...

- Giá trị đầu vào: nhập giá trị cho các hệ số a, b, c.
- Tính giá trị delta.
- Xét các trường hợp xảy ra của delta.

+ Nếu delta = 0: kết luận “phương trình có nghiệm kép” và hiển thị giá trị nghiệm.

+ Nếu delta < 0: kết luận “phương trình vô nghiệm”

+ Nếu delta > 0: kết luận “phương trình có hai nghiệm” và hiển thị giá trị hai nghiệm.

Kết thúc chương trình.

Viết mã Pascal cho bài toán:

```
Program phuong_trinh_b2;
Uses Crt;
Var a, b, c: Integer;
    delta: Real;
Begin
    clrscr;
    writeln('Giai phuong trinh bac 2');
```

```

write('Nhap he so a: '); readln(a);
write('Nhap he so b: '); readln(b);
write('Nhap he so c: '); readln(c);
delta := b * b - 4 * a * c;
if delta < 0 then writeln('Phuong trinh vo nghiem.');
```

if delta = 0 then writeln('Phuong	trinh co	nghiem	kep:
	x1 = x2 = ',	-b/(2*a));	

```

if delta > 0 then
begin
  writeln('Phuong trinh co hai nghiem phan biet:');
  writeln('x1 = ', (-b - sqrt(delta))/(2*a));
  writeln('x2 = ', (-b + sqrt(delta))/(2*a));
end;
readln;
end.
```

Trên đây là một cách giải quyết bài toán đơn giản, còn có rất nhiều cách giải quyết khác như khai báo thêm hai biến x1, x2 để nhận giá trị tính toán, hoặc tính nghiệm phức của phương trình... Tuy nhiên để làm được bất cứ một bài toán nào, lập trình viên của phải hiểu được bản chất bài toán, các bước giải bài toán.

3.2.2. Cấu trúc Case ... of

```

CASE biểu_thức OF
  nhãn 1 : lệnh 1 ;
  nhãn 2 : lệnh 2 ;
  { .....; }
  nhãn n : lệnh n;
[ELSE
  lệnh n+1]
END;
```

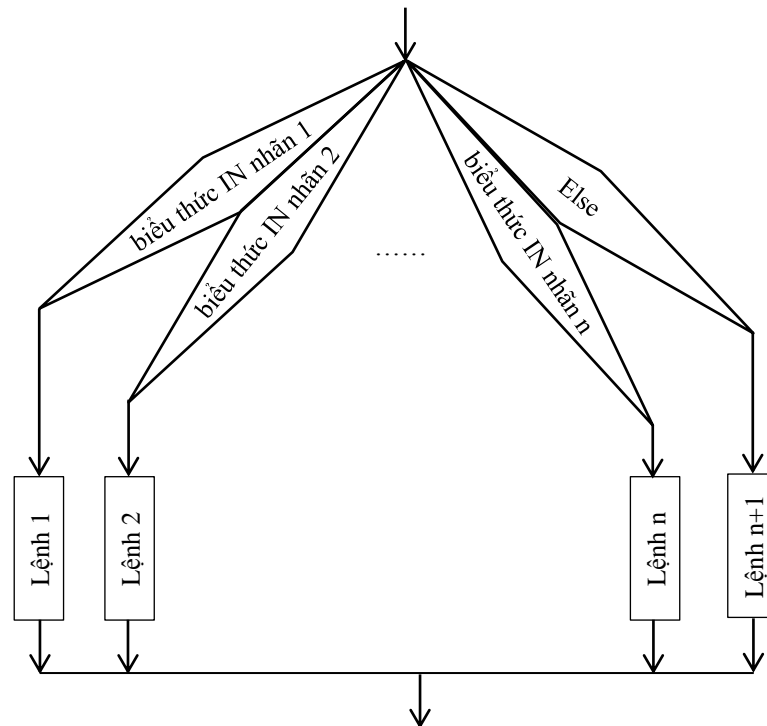
- Nhãn của lệnh CASE và biểu thức lựa chọn phải tương thích về kiểu và phải là kiểu vô hướng (trừ kiểu số thực).

- Lệnh trước từ khóa ELSE không có dấu chấm phẩy.
- Các lệnh trong biểu CASE có thể là lệnh đơn hoặc lệnh ghép.
- Nhãn của lệnh CASE có thể là số, dãy số hoặc đoạn con.

Nguyên tắc hoạt động:

Khi gặp câu lệnh Case .. of chương trình sẽ kiểm tra giá trị của biểu thức tương ứng với nhãn nào thì sẽ thực hiện lệnh tương ứng trong nhãn đó, sau khi thực hiện xong thì kết thúc câu lệnh Case .. of.

Lưu đồ hoạt động:



Ví dụ:

Viết chương trình Pascal nhập vào tháng và năm. In ra trong tháng đó, năm đó có bao nhiêu ngày?

Phân tích bài toán:

- Khai báo biến vào mm: tháng; yy: năm. Biến ra dd: số ngày
- Hiển thị thông báo số ngày
- Xét từng trường hợp với các tháng có số ngày như sau:
 - 1, 3, 5, 7, 8, 10, 12 có 31 ngày
 - 4, 6, 9, 11 có 30 ngày
 - 2 có 28 hoặc 29 ngày (năm nhuận có 29 ngày, cứ 4 năm nhuận có 1 năm nhuận)
- Trong trường hợp này, các nhãn là dãy các giá trị hoặc là 1 số.

Viết chương trình:

```
Program Ngay_trong_thang;  
Uses Crt;  
Var dd, mm, yy: Word;  
Begin  
  clrscr;  
  writeln('In ra so ngay trong thang');  
  write('Nhap thang: '); readln(mm);  
  write('Nhap nam: '); readln(yy);  
  CASE mm OF
```



```

1,3,5,7,8,10,12 : dd:=31;
4,6,9,11: dd:=30;
2: if (yy mod 4 =0) then dd:=29 else dd:=28;
ELSE
    dd:=0;
END;
If (dd=0) then
    writeln('Ban nhap thang hoac nam khong hop le!')
Else
    writeln('Nam ', yy, ' thang ', mm, ' co ', dd, ' ngay. ');
readln;
end.

```

3.3. Cấu trúc lặp

Cấu trúc lặp là không thể thiếu khi lập trình, trong nhiều trường hợp có những công việc phải xử lý lặp lại liên tục, với mỗi lần lặp giá trị các biến có thể thay đổi phù hợp với mục đích lập trình. Ví dụ người nông dân tưới rau, công việc chính là tưới rau; mỗi luống rau cần tưới 5 gánh nước; trong công việc chính đó có rất nhiều công việc con được lặp đi lặp lại 5 lần như đi ra giếng nước, cúi xuống lấy nước, gánh về vườn rau, tưới rau. Như vậy các công việc con đó có thể được đặt vào vòng lặp để chương trình tự động thực hiện. Trong quá trình thực hiện, luôn luôn kiểm tra giá trị các biến, đặc biệt là biến nhận biết khi nào thì vòng lặp kết thúc. Trong cấu trúc lặp có 2 loại lặp đó là lặp với số lần biết trước và lặp không biết trước số lần lặp.

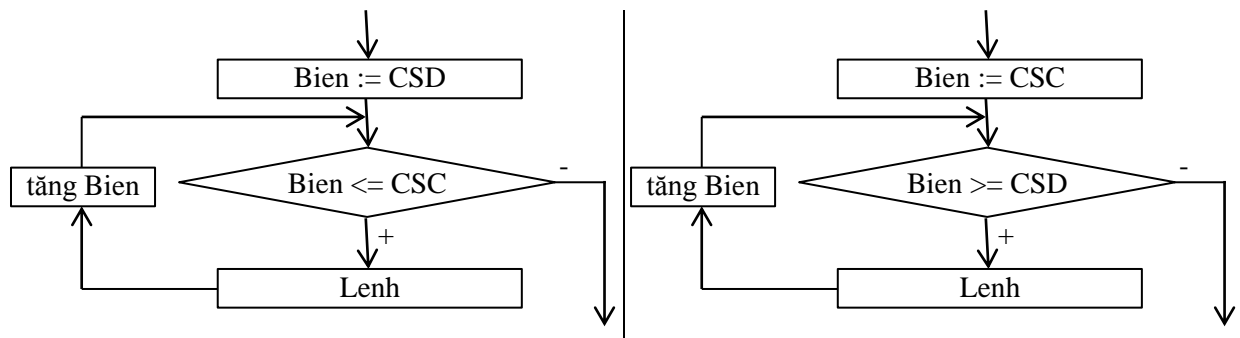
Thông thường lặp với số lần biết trước ta sử dụng một biến gọi là biến đếm, khi đếm đủ số lần lặp thì vòng lặp kết thúc.

Với vòng lặp không biết trước số lần lặp, ta luôn kiểm tra giá trị kết quả, nếu thỏa mãn thì kết thúc vòng lặp.

3.3.1. Cấu trúc For ... do

Cấu trúc For ... do là dạng lặp có số lần biết trước. Có hai dạng lặp kiểu này đó là lặp tiến và lặp lùi.

Lặp tiến	Lặp lùi
For bien:=CSD to CSC do Lenh	For bien:=CSC downto CSD do Lenh
- Ban đầu bien nhận giá trị CSD	- Ban đầu bien nhận giá trị CSC
- Kiểm tra nếu bien <= CSC thì thực hiện Lenh ngược lại kết thúc vòng lặp	- Kiểm tra nếu bien >= CSD thì thực hiện Lenh ngược lại kết thúc vòng lặp
- Sau mỗi lần lặp giá trị của bien tự động tăng 1 đơn vị.	- Sau mỗi lần lặp giá trị của bien tự động giảm 1 đơn vị.



Chú ý:

- Biến và CSD, CSC phải có kiểu vô hướng (trừ kiểu số thực)
- Có thể kết thúc vòng lặp bất cứ lúc nào gặp câu lệnh BREAK;

Bài toán

Trong đợt quyên góp sắt vụn để gây quỹ giúp các bạn nghèo có hoàn cảnh khó khăn, tổ 3 có 12 bạn, mỗi bạn nộp một khối lượng sắt vụn nào đó tính bằng kg. Viết chương trình cho máy nhận vào khối lượng sắt vụn của từng bạn và tính tổng khối lượng sắt vụn của cả 12 bạn trong tổ.

Thuật giải:

Ta dùng ngôn ngữ giả mã lệnh để viết thuật giải cho bài toán theo ý tưởng đã nói ở trên như sau:

Biến thực S, X; Biến nguyên K;
 Bắt đầu
 S := 0;
 Lặp K := 1 đến 12 mỗi lần lặp nhập vào (X) và tính S := S + X;
 Thông báo S;
 Kết thúc.

Chương trình

```

Program SatVun;
Uses Crt;
Var S, X : Real; K : byte;
BEGIN
  ClrScr; S := 0;
  Writeln ( 'Nhập vào kg sắt vụn của từng người' );
  For K := 1 TO 12 DO
  Begin
    Write ( 'Nhập KLSV của bạn thu ' , k , '=' );
    Readln (X); S := S + X;
  End;
  Writeln ( 'Đáp số ' , S : 9 : 2 ); Readln;
END.

```

Bài toánViết chương trình giải bài toán 100 con trâu 100 bó cỏ, một con trâu đứng ăn 5 bó cỏ một con trâu nằm ăn 3 bó cỏ , 3 con trâu già ăn 1 bó cỏ. Hỏi mỗi loại trâu có mấy con? **Thuật giải**

Biến nguyên không âm td, tn;

Bắt đầu

Với td := 0 đến 20 Làm

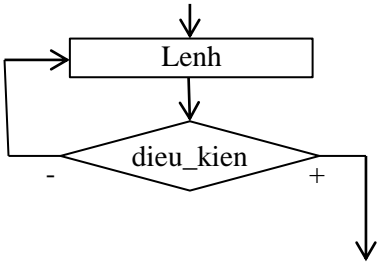
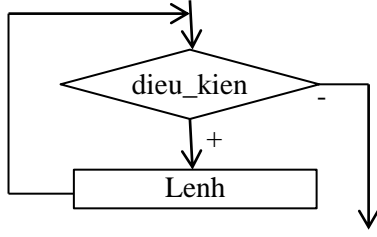
Với tn := 0 đến 33 Làm

Nếu $5*td+3*tn+(100-td-tn) \div 3 = 100$ và $(100-td-tn) \bmod 3 = 0$

Thì viết ra (td, tn, 100-td-tn);

Kết thúc.

3.3.2. Cấu trúc Repeat ... until và Cấu trúc While ... do

Cấu pháp	Lưu đồ	Ý nghĩa
Repeat Lệnh; Until (điều_kiện);		<ul style="list-style-type: none"> - Thực hiện lệnh trước khi kiểm tra điều kiện. - Lệnh được thực hiện ít nhất 1 lần
While (điều_kiện) do Lệnh;		<ul style="list-style-type: none"> - Kiểm tra điều kiện thỏa mãn rồi thực hiện lệnh. - Lệnh được thực hiện ít nhất 0 lần

Chú ý:

- Có thể kết thúc vòng lặp bất cứ lúc nào gặp câu lệnh BREAK;

Bài toán.

Tìm ước chung lớn nhất của hai số nguyên dương m và n.

Thuật giải

Ta dùng ngôn ngữ giả mã lệnh để viết thuật giải theo tinh thần trên.

Biến nguyên không âm m, n ;

Bắt đầu

Lặp

Nhận vào (m , n) ;
 Nếu (m = 0) hoặc (n = 0) Thì thông báo mời vào lại
 Đến khi (m > 0) và (n > 0) ;
 Khi m < n Làm
 Nếu m > n Thì m := m - n Ngược lại thì n := n - m ;
 Thông báo đáp số là m;
 Kết thúc.

Chương trình

```

Program UCLN;
Uses Crt ;
Var m , n : Word ;
BEGIN
  writeln ( ' cho 2 so nguyen duong ' );
  Repeat
    Readln( m , n );
    If ( m = 0 ) or ( n = 0 ) Then Writeln ( ' moi vao lai ' );
  Until ( m > 0 ) and ( n > 0 );
  While m <> n DO
    If m > n Then m := m - n Else n := n - m ;
  Write ( ' UCLN cua chung la ' , m );
  Readln;
END.
  
```

Câu hỏi và bài tập

- Hãy lập bảng mô phỏng việc thực hiện chương trình giải phương trình bậc hai với các giá trị của a , b , c nhập vào như sau:
 - lần thứ nhất a = 1, b = 14, c = 49,
 - lần thứ hai a = 1, b = -2, c = -15,
 - lần thứ ba a = 2, b = 1, c = 3.
- Viết chương trình cho máy nhận vào hai số nguyên bất kì. Sau đó máy hỏi tổng bình phương của chúng là bao nhiêu? Nếu ta nhập đúng đáp số thì máy thông báo "Bạn đã làm đúng đáp số". Nếu sai máy thông báo "Bạn làm sai rồi, đáp số là:" và máy cho hiện đáp số lên màn hình.
- Viết thuật giải và chương trình giải phương trình bậc nhất $ax + b = 0$ với mọi số thực a, b.
- Viết chương trình kiểm tra xem 1 năm nào đó có là năm nhuận không? Chú ý. Năm nhuận là bội của 4 trừ những năm bắt đầu thế kỉ như 1900, 2000, ... những năm này chỉ nhuận khi nó chia hết cho 400.
- Lập trình để thông báo những mốc lịch sử của nước ta từ năm 1945 đến nay, khi một người gõ vào một số có hai chữ số là hai chữ số sau cùng chỉ năm từ

1945 đến 1995 thì máy thông báo lên màn hình sự kiện lịch sử đáng ghi nhớ của dân tộc Việt Nam. Chẳng hạn 45 Cách mạng tháng 8 thắng lợi. Từ 46 đến 54 Kháng chiến chống Pháp. Từ 55 đến 74 đánh Mỹ. 75 thống nhất đất nước. 76 đến 95 cả nước đi lên CNXH.

6. Giả sử hôm nay là thứ ba. Lập chương trình để ta nhập vào từ bàn phím số ngày tiếp theo thì máy sẽ trả lời hôm đó là thứ mấy.

7. Viết chương trình theo yêu cầu dưới đây: Khi chạy chương trình trên màn hình xuất hiện năm dòng chữ như sau:

- | |
|----------------------------------|
| 1 Tìm số bé nhất trong ba số |
| 2 giải phương trình $ax + b = 0$ |
| 3 Tính a^x |
| 4 Tính logarit cơ số a của x |

Bạn chọn công việc nào thì gõ vào phím số bên cạnh. Khi người ta gõ vào phím nào thì yêu cầu họ vào dữ liệu cần thiết để đưa lên màn hình kết quả tương ứng. Ví dụ người gõ vào số 2 thì hỏi giá trị của hệ số a và hệ số b rồi viết lên màn hình thông báo nghiệm của phương trình.

8. Viết chương trình giải bài toán:

Vừa gà vừa chó bó lại cho tròn 36 con 100 chân chẵn. Hỏi có mấy con chó, mấy con gà?

9. Viết chương trình để MTĐT nhận vào 10 số thực bất kì (mỗi lần chỉ nhận một số). Sau 10 lần nhận vào máy thông báo số bé nhất, lớn nhất, trung bình cộng của chúng.

10. Viết chương trình để MTĐT tìm tất cả các số tự nhiên có ba chữ số. Biết rằng số đó chia cho 6 dư 2 và tổng các chữ số của nó bằng 20.

11. Viết chương trình cho MTĐT in ra màn hình các số có bốn chữ số mà tổng các chữ số hàng trăm và hàng ngàn bằng tổng các chữ số hàng chục và hàng đơn vị. Yêu cầu in từng trang màn hình.

12. Cho một số tự nhiên n lớn hơn 5.

- Tìm các số tự nhiên x, y, z sao cho $n = x^3 + y^3 + z^3$
- Tìm các số tự nhiên x, y, z, t sao cho $n = x^2 + y^2 + z^2 + t^2$
- Hãy tìm tất cả các bộ 5 số tự nhiên x_1, x_2, x_3, x_4, x_5 sao cho số sau không nhỏ hơn số trước và tổng của chúng bằng n .

13. Viết chương trình tính tổng của các số khác không được nhập vào từ bàn phím. Điều kiện kết thúc là khi máy nhận được số 0.

14. Tính tổng

a. $S_1 = 1 + 1/2 + 1/3 + 1/4 + 1/5 + \dots$

b. $S_2 = 1 - 1/2 + 1/4 - 1/6 + \dots - 1/102$

c. $S_3 = 1! + 2! + 3! + \dots + 10!$

d. $S_4 = 1 - 1/3 + 1/5 - 1/7 + \dots + 1/n$ với điều kiện $1/n < 0.0001$

15. Nhận vào số tự nhiên n lớn hơn 100 và nhỏ hơn 60000.

Tính số chữ số và tổng các chữ số của n .

Chương 4. Chương trình con

Mục tiêu bài học

- Biết chia bài toán thành những bài toán nhỏ hơn, trong số các bài toán nhỏ hơn có thể được chia thành những bài toán nhỏ hơn nữa, sao cho mỗi bài toán con chỉ làm cụ thể 1 công việc nào đó.
- Biết cách viết thủ tục, hàm, biết cách truyền đối tượng thực sự vào tham chiếu khi gọi chương trình con.

Với một số bài toán và phức tạp, người ta thường chia nó thành nhiều đoạn chương trình nhỏ. Ta tiến hành giải các bài toán nhỏ này rồi ghép vào được kết quả của bài toán lớn ban đầu. Trong Turbo Pascal cũng cho phép chúng ta làm như vậy. Bài học này chúng ta sẽ đề cập đến:

4.1. Các khái niệm cơ bản

Khi lập trình, có những đoạn chương trình cần dùng nhiều lần. Để tránh việc viết lại đoạn chương trình này, ta nên chuyển đoạn chương trình này thành một chương trình con và mỗi lần cần thực hiện công việc đó thì ta gọi nó thông qua tên.

Chương trình con còn để mẫu hóa một chương trình làm công việc nào đó. *Người khác dùng chương trình con chỉ cần biết truyền số liệu vào và lấy kết quả ra như thế nào mà không cần phải quan tâm đến thuật toán trong chương trình con như thế nào.*

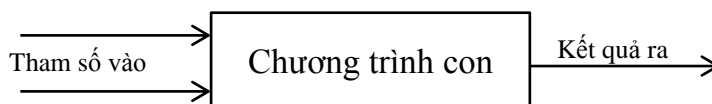
Khi viết những chương trình lớn, để dễ dàng quản lý, gỡ rối và hiệu chỉnh chương trình, ta nên phân chương trình thành nhiều *công việc độc lập*, mỗi công việc là một chương trình con. Chương trình con gồm có hai loại là HÀM và THỦ TỤC.

4.2. Cấu trúc chương trình con

Cấu trúc chung chương trình con

```
PROCEDURE/FUNCTION Ten_CTC(tham_so) [:kiểu_CTC];  
{Các khai báo biến, hằng, kiểu ... cục bộ}  
Begin  
    Các lệnh;  
End;
```

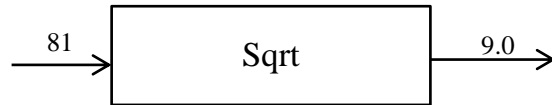
Mô tả



Ta có thể coi Chương trình con là một hộp đen. Để sử dụng chương trình con, người lập trình phải xác định các tham số đầu vào và lấy kết quả đầu ra. Còn phần xử lý như nào là do người lập trình viết mã lệnh bên trong.

Để viết một chương trình con tốt, trước hết cần phải xác định đúng các tham số đầu vào và kết quả đầu ra. Sau đó khi cần các biến khác thì ta dùng các biến cục bộ.

Ví dụ



4.2.1. Chương trình con hàm

```
FUNCTION Ten_ham(Tham_số):Kiểu_hàm;  
{Khai báo biến, hằng, kiểu, nhãn...}  
Begin  
    Các_lệnh;  
    Ten_ham := Biểu_thức_kết_quả;  
End;
```

- Tham số: là các tham số đầu vào để khi gọi hàm ta truyền giá trị cho hàm

- Tên_hàm: có kiểu hàm là các kiểu chuẩn hoặc kiểu được định nghĩa bởi người sử dụng. Vì vậy có thể coi Tên_hàm là một biến có thể nhận giá trị trước khi kết thúc hàm.

- Tên_hàm có kiểu hàm nên có thể gán cho một biến khác cùng kiểu, có thể được viết trong biểu thức tính toán ...

Ví dụ 1: $a = 7 + \text{sqrt}(6) - \text{ord}('a');$

Ví dụ 2: Viết chương trình con tính n giai thừa

```
FUNCTION giai thua(n:integer):Longint;  
Var i:integer; s:longint;  
Begin  
    s:=1;  
    for i:=1 to n do s:=s*i;  
    giai thua:=s;  
End;
```

4.2.2. Chương trình con thủ tục

```
PROCEDURE Ten_tt(Tham_số);  
{Khai báo biến, hằng, kiểu, nhãn...}  
Begin  
    Các_lệnh;  
End;
```

- Tham số: là các tham số đầu vào để khi gọi thủ tục ta truyền giá trị cho nó.

- Thủ tục chỉ đơn thuần là thực hiện một công việc, nó không có kiểu nên không thể nhận giá trị, không được viết trong biểu thức.

Ví dụ 1: Writeln('Chương trình tìm UCLN');

Ví dụ 2: Viết thủ tục hoán vị hai giá trị nguyên x, y

```
PROCEDURE hoanvi(Var x, y: Integer);  
  Var z: Integer;  
  Begin  
    z:=x; x:=y; y:=z;  
  End;
```

4.3. Biến toàn cục, biến địa phương

Biến là các ô nhớ để lưu trữ giá trị trung gian hoặc kết quả tính toán. Có hai loại biến đó là biến toàn cục và biến địa phương:

Biến toàn cục là biến khai báo ở đầu chương trình chính, tồn tại suốt thời gian làm việc của chương trình. Ta có thể sử dụng và làm thay đổi giá trị của biến toàn cục tại bất cứ nơi nào trong chương trình.

Biến cục bộ là biến khai báo ở đầu chương trình con. Chúng được cấp phát bộ nhớ khi chương trình con được gọi và bị xóa khi kết thúc chương trình con. Biến cục bộ có giá trị trong chương trình con và tất cả các chương trình con trong nó.

Nếu tên biến cục bộ của một chương trình con trùng với một tên biến toàn cục thì máy không bị nhầm lẫn, máy sẽ dùng hai ô nhớ khác nhau để lưu trữ hai biến, khi ra khỏi chương trình con, biến cục bộ tự động được xóa.

Khi gặp một lời gọi chương trình con, máy sẽ thực hiện các bước sau:

- Cấp phát bộ nhớ có các tham số và biến cục bộ.
- Truyền giá trị cho các tham số.
- Thực hiện các lệnh trong chương trình con, trong khi thực hiện chương trình con, giá trị các biến cục bộ và các tham số có thể bị thay đổi nhưng không làm ảnh hưởng các biến bên ngoài. Tuy nhiên các tham biến sẽ bị thay đổi khi thực hiện lệnh.
- Thực hiện xong các lệnh của chương trình con, máy xóa tất cả các tham số và biến cục bộ và trả về lệnh sau lời gọi chương trình con.

Ví dụ 1: ở ví dụ tính n giai thừa ta có 1 tham số và hai biến cục bộ (i, s)

Ví dụ 2: ở ví dụ hoán vị ta có hai tham biến (x, y) và 1 biến cục bộ (z);

Ví dụ 3: Lập hàm tính diện tích hình thang. Nhập dữ liệu của hai thửa ruộng hình thang và tính tổng diện tích hai thửa ruộng đó.

```
Var a1, b1, h1, a2, b2, h2, s : Real;  
(***** Bat dau Function *****)  
Function DTHinhThang(a, b, h: Real): Real;  
  Begin  
    DTHinhThang := (a + b) * h / 2;  
  End;  
(***** Bat dau chương trình chính *****)
```

```

Begin
  Write(' Canh dai, ngan va cao cua thua ruong thu nhât: ');
  Readln(a1, b1, h1);
  Write(' Canh dai, ngan va cao cua thua ruong thu hai: ');
  Readln(a2, b2, h2);
  s := DTHinhThang(a1, b1, h1) + DTHinhThang(a2, b2, h2);
  Writeln(' Tong dien tich hai thua ruong = ', s : 0 : 3);
  Readln;
End.

```

4.4. Tham số chương trình con

Có hai loại tham số chương trình con đó là tham biến và tham trị. Ví dụ cái quạt điện là một chương trình con và nó có hai tham số là hai sợi dây điện; người sử dụng chỉ cần đấu nối hai dây điện đó là chương trình con hoạt động và trả về giá trị (gió).

Vậy tham số chương trình con được coi như dữ liệu đầu vào, khi vào chương trình con sẽ xử lý để cho kết quả ra.

Tham trị là tham số truyền theo giá trị, giá trị của tham số này không bị thay đổi khi chương trình con kết thúc.

Tham biến là tham số truyền theo địa chỉ ô nhớ, giá trị của tham số này có thể bị thay đổi khi chương trình con kết thúc. Người ta có thể dùng tham biến để lấy kết quả xử lý của chương trình con khi có nhiều giá trị trả về (trong khi tên hàm chỉ có thể lấy được 1 giá trị trả về).

Ví dụ 1: Lập hàm tính ước số chung lớn nhất (UCLN). Sau đó tính bội số chung nhỏ nhất của hai số được nhập từ bàn phím

```

Var m, n, usc, bsc: Integer;
(***** Function USCLN *****)
Function UCLN(a, b : Integer): Integer;
Var r: Integer;
Begin
  While b <> 0 do
  Begin
    r := a mod b;
    a := b;
    b := r;
  End; { a hiện tại là UCLN của a và b ban đầu }
  UCLN := a;
End;
(***** bắt đầu chương trình chính *****)
Begin
  Write(' Nhập số thu nhât: ');
  Readln(m);
  Write(' Nhập số thu hai: ');
  Readln(n);

```

```

usc := USCLN(m, n);
bsc := m * n div USCLN(m, n);
Writeln('Uoc so chung lon nhat cua ', m, ' va ', n, ' la :', usc);
Writeln('Boi so chung nho nhat cua ', m, ' va ', n, ' la :', bsc);
Readln;
End.

```

Ví dụ 2: Lập thủ tục để tính đồng thời diện tích và thể tích hình cầu

```

Var r, s, v : Real;
Reply : Char;
(***** Function *****)
Procedure SVHinhCau( r : Real; Var s, v :Real);
Begin
  s := 4 * pi * r * r;
  v := 4 * pi * r * r * r / 3;
End;
(***** bat dau chuong trinh chinh *****)
Begin
  Repeat
    Write( ' Nhập bán kính hình cầu : ' ); Readln(r);
    SVHinhCau(r, s, v);
    Writeln('Diện tích = ', s : 0 : 4, ' . Thể tích = ', v : 0 : 4 );
    Write(' Bạn có tiếp tục không?(C/K) ');
    Readln(Reply);
  Until Upcase(Reply) = 'K';
End.

```

4.5. Chương trình con đệ qui

4.5.1. Khái niệm đệ qui

Khi tính toán, có những bài toán lớn được chia thành nhiều bài toán con, có bài toán con có cách tính giống như bài toán lớn. Như vậy các lệnh trong chương trình con có lệnh gọi đến chính chương trình con đó.

Có thể coi chương trình con đệ qui như một vòng lặp, mỗi lần gọi đệ qui sẽ phải biến đổi phạm vi hay cấu trúc chương trình con để có thể kết thúc vòng lặp.

4.5.2. Viết chương trình con đệ qui

Trong bài toán tính $n! = 1*2*3*\dots*n$ có thể được tính theo công thức đệ qui như sau $n! = n*(n-1)!$

Ta có thể coi dấu chấm than (!) là tên của một hàm, thì ta có thể viết:

$giaithua(n) = n * giaithua(n-1);$

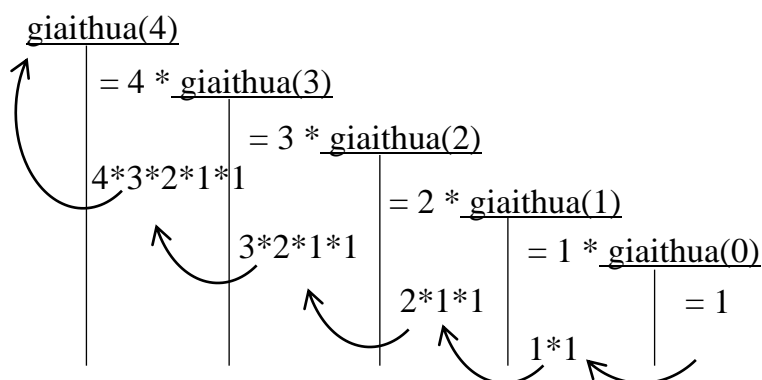
Vấn đề ở đây là $n-1$ thì trừ đến bao giờ? vì thế trong thân chương trình con phải có lệnh kiểm tra điểm tới hạn khi tính toán n .

$$n! = \begin{cases} 1 & \text{nếu } n = 0 \\ n * (n - 1)! & \text{nếu } n > 0 \end{cases}$$

```
Function giaithua(n : Integer): Longint;
Begin
    if (n=0) then giaithua:=1
    else giaithua:=n*giaithua(n-1);
End;
```

4.5.3. Mô tả hoạt động của chương trình con đệ quy

Với chương trình con tính $n!$ ta có thể mô phỏng với $n=4$ xem chương trình con đệ quy hoạt động như thế nào:



4.6. Các hàm và thủ tục của unit CRT

Unit CRT có nhiều hàm, thủ tục dùng để điều khiển màn hình, bàn phím và âm thanh. Nó cho phép mở các cửa sổ với màu sắc khác nhau, thay đổi màu của các dòng chữ trên màn hình, giúp cho việc trình bày màn hình đẹp và hấp dẫn hơn, tổ chức hội thoại giữa người với máy thuận tiện. Khi dùng các hàm và thủ tục này, ở đầu chương trình chính cần phải khai báo USES CRT; Các thủ tục của unit CRT gồm:

ClrScr xóa màn hình và con trỏ về vị trí (1,1) trên màn hình. Màn hình mặc định được chia thành 25 dòng và 80 cột. ô đầu tiên ở góc trái trên màn hình có vị trí (1,1); ô cuối cùng có vị trí (25,80) ở vị trí góc phải dưới.

ClrEOL xóa từ vị trí con trỏ đến cuối dòng hiện hành. Sai khi thực hiện xong, con trỏ đứng ngay vị trí trước khi gọi thực hiện thủ tục.

DelLine xóa dòng con trỏ đang đứng, các dòng sau sẽ được dịch chuyển lên.

InsLine chèn dòng trắng vào vị trí hiện hành của con trỏ trên màn hình.

GotoXY(x, y: Byte) đưa con trỏ đến cột x dòng y.

WhereX: Byte cho vị trí hiện hành cột x của con trỏ.

WhereY: Byte cho vị trí hiện hành dòng y của con trỏ.

Sound(Hz : Word) phát âm thanh có tần số Hz.

NoSound tắt âm thanh được phát bởi thủ tục Sound

TextBackGround(Color : Byte) đặt màu màn hình trong chế độ văn bản, có 8 màu được sử dụng; tham số Color nhận giá trị từ 0 đến 7.

TextColor(Color : Byte) đặt màu chữ trong chế độ văn bản, Color có thể nhận trị từ 1 đến 16 (có 16 màu)

KeyPressed: Boolean hàm kiểm tra có phím nào được nhấn. Nếu có hàm trả về giá trị TRUE, nếu không hàm trả về giá trị FALSE.

ReadKey: Char hàm đọc một ký tự trong bộ đệm bàn phím, nếu bộ đệm bàn phím rỗng, hàm chờ người sử dụng nhấn phím.

Câu hỏi và bài tập

1. Viết một chương trình trong đó có bốn thủ tục như sau:

Thủ tục mang tên VAO, trong đó máy nhận vào một số thực a dương. Thủ tục mang tên HINHVUONG là một chương trình tính chu vi, diện tích và đường chéo của hình vuông cạnh a. Thủ tục mang tên HINHTRON là một chương trình tính chu vi và diện tích của hình tròn bán kính a. Thủ tục mang tên TAMGIAC là một chương trình tính chu vi, diện tích, đường cao của tam giác đều cạnh a.

Phần thân chương trình gọi cả bốn thủ tục trên.

2. Viết một chương trình trong đó có hai thủ tục như sau:

Thủ tục mang tên GPTB1 là một chương trình giải phương trình bậc nhất $ax + b = 0$ với a, b là những số bất kì.

Thủ tục mang tên NHATRON là một chương trình tính diện tích của phần bên trong hình tròn và bên ngoài hình chữ nhật có hai cạnh a, b (hình chữ nhật nội tiếp hình tròn), tính chu vi đường tròn và chu vi hình chữ nhật.

Phần thân chương trình có lệnh nhận vào hai số thực a, b bất kì. Gọi thủ tục GPTB1. Nếu cả hai số a, b đều dương thì gọi thủ tục NHATRON.

3. Viết 3 thủ tục không có tham biến để giải phương trình bậc hai $ax^2 + bx + c = 0$, ứng với 3 trường hợp ($D > 0$, $D < 0$, $D = 0$).

4. Viết chương trình cho máy thực hiện hai lần những việc như sau:
Lần thứ nhất. Nhận vào mảng A có m dòng và n cột ($m=3, n=4$), các phần tử là những số nguyên lớn hơn 5 và nhỏ hơn 500. Yêu cầu trên màn hình các phần tử cùng dòng ở trên một dòng, cùng cột ở trên một cột, có kiểm tra điều kiện phần tử lớn hơn 5 và nhỏ hơn 500. Sau khi nhận vào mảng A thì biến đổi mỗi phần tử của A được thay bằng tổng các ước của nó. Sau đó in mảng A đã biến đổi lên màn hình, cũng yêu cầu các phần tử cùng dòng ở trên một dòng, cùng cột ở trên một cột. Lần thứ hai. Việc nhận vào mảng A và in ra nó sau khi biến đổi cũng như lần thứ nhất. Sau khi nhận vào mảng A thì biến đổi mỗi phần tử của A được thay bằng tổng các chữ số của nó.

5. Lớp 10A có 4 tổ (giả sử số người ở các tổ không bằng nhau). Trong đợt thi lập trình, mọi người đều tham gia và mỗi người đạt một điểm nào đó từ 0 đến 20. Lập

chương trình cho MTĐT tính:

- a. Điểm trung bình của từng tổ
- b. Số người trong từng tổ đạt từ 15 điểm trở lên
- c. Số người trong từng tổ bị dưới 5 điểm
- d. Tổng số điểm của cả lớp.

Ghi chú. Điểm của từng người theo tổ được nhập từ bàn phím khi chương trình được thực hiện.

6. Viết chương trình theo yêu cầu sau đây:

Trên màn hình có năm dòng chữ như sau:

- a. Tính diện tích và trung tuyến ứng với cạnh a.
- b. Tính diện tích và trung tuyến ứng với cạnh b.
- c. Tính diện tích và trung tuyến ứng với cạnh c.

0. Ngừng chương trình

Go phím tương ứng để chọn công việc.

Cho máy nhận vào 3 cạnh a, b, c của tam giác ABC sau đó tùy theo người dùng gõ vào phím nào trong các ký tự a, b, c, 0 thì công việc tương ứng đã chỉ ra được thực hiện.

7. Viết chương trình theo yêu cầu sau đây:

Có một thủ tục mang tên GPTB1 là thủ tục truyền theo hai trị a và b, làm công việc giải phương trình bậc nhất $ax + b = 0$ với mọi a, b. Một thủ tục mang tên CHUNHAT là thủ tục truyền theo hai trị a và b, làm công việc tính chu vi, diện tích, đường chéo của hình chữ nhật theo hai kích thước a, b. Một thủ tục mang tên TRON là thủ tục truyền theo trị r, làm công việc tính chu vi, diện tích của hình tròn theo bán kính r của nó.

Trong thân chương trình có lệnh nhận vào ba số x, y, z. Tiếp theo thủ tục GPTB1 được gọi ba lần với ba cặp giá trị thay cho cặp (a, b) là (x, y), (y, z) và (z, x). Thủ tục TRON được gọi hai lần với $r = 5$ và $r = |x+y| + 4$. Thủ tục CHUNHAT được gọi với $a = \sin(x - z) + 6$ và $b = y^2 + 3$.

8. Viết chương trình cho máy nhận vào 2 lần, mỗi lần 5 ký tự. Trong mỗi lần nhận vào, nếu ký tự thứ 2 rơi vào đoạn từ ký tự 'p' đến ký tự 't' thì nó được thay bởi dấu sao (*), nếu ký tự thứ 4 là nguyên âm thì nó được thay theo quy tắc sau đây:

a -> e ; e -> i ; i -> o ; o -> u ; u -> y ; y -> a ;

Để đơn giản ta giả sử các ký tự đưa vào không có chữ hoa. Sau 2 lần vào dữ liệu, máy phải in lên màn hình 10 ký tự đã được thay đổi theo yêu cầu đầu bài.

9. Cho hai phân số a/b và c/d hãy viết chương trình cộng và nhân hai phân số đó.

Trong đó một hàm tìm ước chung lớn nhất của hai số để tìm mẫu số chung của hai phân số và giản ước phân số kết quả.

10. Giải hệ phương trình

$$a_1x + b_1y + c_1 = 0$$

$$a_2x + b_2y + c_2 = 0$$

11. Có ba vật dùng điện với điện trở tương ứng là R_1 , R_2 , R_3 và thời gian tương ứng là t_1 , t_2 , t_3 . Ba vật này cùng được mắc song song vào mạng điện 220V.

Viết chương trình tính tổng điện năng mà ba vật đó đã tiêu thụ.

12. Viết chương trình tính chu vi và diện tích của hình viên phân khi máy nhận vào bán kính của hình tròn và số đo của cung tính bằng độ. Yêu cầu có các hàm truyền theo trị: đổi độ sang radian, tính độ dài cung tròn theo bán kính và số đo cung, tính cạnh đáy và diện tích của tam giác cân theo cạnh bên và góc ở đỉnh, tính diện tích hình quạt theo bán kính và góc ở tâm.

13. Trong mặt phẳng, cho năm điểm, mỗi điểm được xác định bởi toạ độ của nó. Năm điểm này tạo thành đường gấp khúc bốn đoạn. Viết chương trình để máy nhận vào toạ độ của các điểm, sau đó tính độ dài của đường gấp khúc.

14. Tam giác được cho bởi ba cạnh. Tính ba đường cao và ba đường trung tuyến của nó. Lập chương trình theo yêu cầu sau: Một hàm có tham biến trị để tính đường cao theo ba cạnh. Một hàm có tham biến trị để tính đường trung tuyến theo ba cạnh. Một thủ tục không có tham biến nhận vào ba số kiểm tra ba số là ba cạnh của tam giác.

15. Viết chương trình nhận vào số tự nhiên n chẵn lớn hơn 2. Chỉ ra hai số nguyên tố x và y sao cho $n = x + y$.

16. Ba đỉnh của tam giác ABC có toạ độ như sau: $A(-1, 5)$, $B(-2, -7)$, $C(3, 8)$. Tính những yếu tố sau của tam giác:

Diện tích, ba đường cao, ba đường trung tuyến, ba góc, ba đường phân giác.

17. Viết chương trình để máy in lên màn hình phương trình có dạng $y = ax + b$ của ba đường thẳng đi qua ba điểm A, B, C khi cho toạ độ của ba điểm đó.

18. Viết chương trình cho máy Tìm ước số chung lớn nhất của hai số nguyên dương

a. Dùng hàm đệ qui

b. Dùng thủ tục đệ qui.

19. Viết chương trình cho máy tính nhận vào số tự nhiên n . Tìm các số tự nhiên nhỏ hơn n mà nguyên tố cùng nhau với n .

20. Viết chương trình cho máy tính nhận vào hai số tự nhiên n và k .

Tính $n! / (k! * (n-k)!)$

21. Viết hàm tính số hạng thứ n của dãy Fibonacci

Chương 5. Lập trình xử lý giao diện

5.1. Giới thiệu bảng mã ký tự mở rộng

Trong bảng mã ASCII có 256 ký tự, trong phần các kiểu dữ liệu cơ bản ta đã đề cập đến bảng mã này. Sau đây tác giả trình bày chi tiết các ký tự mở rộng của bảng mã ASCII như bảng sau :

128	Ç	144	É	160	á	176	░	192	Ł	208	⌚	224	α	240	≡
129	ù	145	æ	161	í	177	▒	193	ł	209	⌞	225	β	241	≠
130	é	146	Æ	162	ó	178	▓	194	Ṛ	210	⌘	226	Γ	242	≥
131	â	147	ô	163	ú	179		195	ṛ	211	ℓ	227	π	243	≤
132	ã	148	õ	164	ñ	180	┆	196	—	212	ℓ	228	Σ	244	∫
133	à	149	ò	165	Ñ	181	┆	197	+	213	ℓ	229	σ	245	∫
134	ä	150	û	166	²	182	┆	198	┆	214	ℓ	230	μ	246	+
135	ç	151	ù	167	°	183	π	199	┆	215	┆	231	τ	247	≈
136	ê	152	ÿ	168	¿	184	Ṛ	200	ℓ	216	┆	232	Φ	248	°
137	ë	153	Ö	169	┐	185	┆	201	ℓ	217	┆	233	Θ	249	.
138	è	154	Ü	170	┐	186	┆	202	ℓ	218	┆	234	Ω	250	.
139	ï	155	◊	171	½	187	Ṛ	203	Ṛ	219	■	235	δ	251	√
140	î	156	£	172	¼	188	┆	204	┆	220	■	236	∞	252	∞
141	ì	157	₣	173	¡	189	┆	205	=	221	■	237	φ	253	²
142	Ä	158	₤	174	«	190	┆	206	┆	222	■	238	ε	254	■
143	Å	159	₥	175	»	191	┆	207	┆	223	■	239	∩	255	

Để lập trình xử lý giao diện, muốn có giao diện đẹp ta cần phải sử dụng các mã ký tự mở rộng này. Để lấy được các ký tự này ta có 2 cách:

- Dùng hàm chr để hiển thị ký tự (thông thường trong lập trình ta dùng cách này).
- Hiển thị trực tiếp ký tự đồ họa bằng cách sử dụng bàn phím số bên phải bàn phím.

Ví dụ để vẽ hình hộp đường kẻ đôi ta dùng các ký tự sau:

201 Ṛ 200 ℓ 187 Ṛ 188 Ṛ

5.2. Các hàm xử lý phím

- Hàm Readkey; đọc một phím từ bộ đệm bàn phím, nếu bộ đệm bàn phím có phím thì hàm trả về mã của phím, nếu bộ đệm rỗng thì hàm chờ người sử dụng nhập một phím.

- Hàm Keypress; kiểm tra có một phím được nhấn hay không, nếu có hàm trả về giá trị True, ngược lại hàm trả về giá trị False.

Các phím điều khiển mở rộng đặc biệt của bảng mã ASCII:

HEX	DEC	keys	HEX	DEC	keys	HEX	DEC	keys	HEX	DEC	keys
03	3	nul	0F	15	Shift-tab	72	114	Ctrl-PrtSc			
47	71	Home				77	119	Ctrl-Home	8C	140	Alt-Home
48	72	Up				85	133	Ctrl-Up	8D	141	Alt-Up
49	73	PgUp				84	132	Ctrl-PgUp	8E	142	Alt-PgUp
4A	74	-(num)				86	134	Ctrl--(nu)	8F	143	Alt--(num)
4B	75	Left				73	115	Ctrl-Left	90	144	Alt-Left
4C	76	Center				87	135	Ctrl-Centr	91	145	Alt-Center
4D	77	Right				74	116	Ctrl-Right	92	146	Alt-Right
4E	78	+(num)				88	136	Ctrl++(nu)	93	147	Alt++(num)
4F	79	End				75	117	Ctrl-End	94	148	Alt-End
50	80	Down				89	137	Ctrl-Down	95	149	Alt-Down
51	81	PgDn				76	118	Ctrl-PgDn	96	150	Alt-PgDn
52	82	Ins				8A	138	Ctrl-Ins	97	151	Alt-Ins
53	83	Del				8B	139	Ctrl-Del	98	152	Alt-Del
3B	59	F1	54	84	Shift-F1	5E	94	Ctrl-F1	68	104	Alt-F1
3C	60	F2	55	85	Shift-F2	5F	95	Ctrl-F2	69	105	Alt-F2
3D	61	F3	56	86	Shift-F3	60	96	Ctrl-F3	6A	106	Alt-F3
3E	62	F4	57	87	Shift-F4	61	97	Ctrl-F4	6B	107	Alt-F4
3F	63	F5	58	88	Shift-F5	62	98	Ctrl-F5	6C	108	Alt-F5
40	64	F6	59	89	Shift-F6	63	99	Ctrl-F6	6D	109	Alt-F6
41	65	F7	5A	90	Shift-F7	64	100	Ctrl-F7	6E	110	Alt-F7
42	66	F8	5B	91	Shift-F8	65	101	Ctrl-F8	6F	111	Alt-F8
43	67	F9	5C	92	Shift-F9	66	102	Ctrl-F9	70	112	Alt-F9
44	68	F10	5D	93	Shift-F10	67	103	Ctrl-F10	71	113	Alt-F10
1E	30	Alt-A	31	49	Alt-N	78	120	Alt-1			
30	48	Alt-B	18	24	Alt-O	79	121	Alt-2			
2E	46	Alt-C	19	25	Alt-P	7A	122	Alt-3			
20	32	Alt-D	10	16	Alt-Q	7B	123	Alt-4			
12	18	Alt-E	13	19	Alt-R	7C	124	Alt-5			
21	33	Alt-F	1F	31	Alt-S	7D	125	Alt-6			
22	34	Alt-G	14	20	Alt-T	7E	126	Alt-7			
23	35	Alt-H	16	22	Alt-U	7F	127	Alt-8			
17	23	Alt-I	2F	47	Alt-V	80	128	Alt-9			
24	36	Alt-J	11	17	Alt-W	81	129	Alt-0			
25	37	Alt-K	2D	45	Alt-X	82	130	Alt--			
26	38	Alt-L	15	21	Alt-Y	83	131	Alt==			
32	50	Alt-M	2C	44	Alt-Z						

Các phím điều khiển mở rộng đặc biệt này của ASCII thường có hai mã : mã đầu tiên có giá trị 0, mã thứ hai có giá trị tương ứng với các phím. Ví dụ phím F1 có mã là 0 và 59, phím Home có mã 0 và 71, phím Left có mã 0 và 75...

Để lập trình xử lý được các phím này ta dùng hàm readkey để đọc từng mã phím trong bộ đệm, kiểm tra nếu phím đầu tiên có giá trị 0 thì ta đọc tiếp mã tiếp theo của phím.

Sau đây là chương trình yêu cầu người sử dụng gõ phím, chương trình sẽ cho biết đó là phím gì (phím thường hay phím mở rộng)

```
Prcedure TestKey ;
Var ch : char ;
Begin
  repeat
    write ('Ban go mot phim <ESC-ket thuc> : ');
    ch := Readkey;
    if (ord(ch)=0) then
      begin
        ch := Readkey;
        writeln('Ban go phim mo rong co ma 0-',ord(ch));
      end else
        writeln('Ban go phim thuong co ma ',ord(ch));
    until ord(ch)=27;
  end;
```

5.3. Tạo và xử lý giao diện

Sau đây tác giả trình bày một số đoạn mã để tạo giao diện chương trình bằng các ký tự đồ họa và xử lý phím

5.3.1. Tạo cửa sổ

```
{Chuong trinh ve mot khung}
procedure vekhung(x1,y1,x2,y2:byte);
var i:byte;
begin
  textcolor(0);
  gotoxy(x1,y1);write(chr(201));
  gotoxy(x2,y1);write(chr(187));
  gotoxy(x2,y2);write(chr(188));
  gotoxy(x1,y2);write(chr(200));
  for i:=x1+1 to x2-1 do
    begin
      gotoxy(i,y1);write(chr(205));
      gotoxy(i,y2);write(chr(205));
    end;
  for i:=y1+1 to y2-1 do
    begin
      gotoxy(x1,i);write(chr(186));
      gotoxy(x2,i);write(chr(186));
    end;
  end;
```

```

{Vẽ cửa sổ có tiêu đề}
procedure cuaso(x1,y1,x2,y2:byte; tieude:string);
begin
    vekhung(x1,y1,x2,y2);
    gotoxy(x1+(x2-x1-length(tieude))div 2,y1);
    write(tieude);
end;

```

5.3.2. Tạo thực đơn

Chương trình con vietxy hiển thị dòng chữ s ở tọa độ x,y có màu nền là mn. Trong chuỗi s, nếu gặp ký tự & thì ký tự đứng ngay sau đó có màu đỏ.

```

procedure vietxy(x,y,mn:byte; s:string);
var i:byte;
begin
    textbackground(mn);
    for i:= 1 to length(s) do
    begin
        gotoxy(x+i-1,y);
        if s[i]='&' then
        begin
            textcolor(4);
            i:=i+1;
            x := x-1;
        end else textcolor(0);
        write(s[i]);
    end;
end;

```

Chương trình con HMenu hiển thị thực đơn ngang tại tọa độ x,y với các đề mục thực đơn là mni, n là số các mục thực đơn .

```

Menu=array[1..20] of string[20];
Function HMenu(x,y:byte;mni:Menu; n:byte; var xx:byte):byte;
Const mn=7;ms=3;
Var tdx:array[1..NoMenu] of byte;
    chon, chon1, i:byte;
    ch:char;
Begin
    tdx[1]:=x;
    for i:=2 to n do tdx[i]:=tdx[i-1] + length(mni[i-1])-1;
    for i:=1 to n do vietxy(tdx[i],y,mn,mni[i]);
    chon:=1; chon1:=1;
    vietxy(tdx[chon],y,ms,mni[chon]);
    repeat
        ch:=readkey;
        if ord(ch)=0 then
        begin
            ch:=readkey;
            case ord(ch) of

```

```

77: begin
    chon:=chon+1;
    if chon>n then chon:=1;
end;
75: begin
    chon:=chon-1;
    if chon<1 then chon:=n
end;
end;
end;
if ord(ch)=13 then break;
if ord(ch)=27 then begin chon:=0; break; end;
if chon<>chon1 then
begin
    vietxy(tdx[chon],y,ms,mni[chon]);
    vietxy(tdx[chon1],y,mn,mni[chon1]);
    chon1:=chon;
end;
until ord(ch)=13;
HMenu:=chon;
xx:=tdx[chon];
end;

```

Chương trình con VMenu hiển thị thực đơn dọc tại tọa độ x,y với các đề mục thực đơn là mni, n là số các mục thực đơn .

```

Function VMenu(x,y:byte; mni:Menu; n:byte):byte;
Const mn=7;ms=3;
Var tdy:array[1..25] of byte;
    chon, chon1, i:byte;
    ch:char;
Begin
    SaveScreen(x,y,x+length(mni[1]),y+n);
    if x+length(mni[1])>79 then x:=79-length(mni[1]);
    tdy[1]:=y;
    for i:=2 to n do tdy[i]:=tdy[i-1] + 1;
    for i:=1 to n do vietxy(x,tdy[i],mn,mni[i]);
    chon:=1; chon1:=1;
    vietxy(x,tdy[chon],ms,mni[chon]);
    repeat
        ch:=readkey;
        if ord(ch)=0 then
            begin
                ch:=readkey;
                case ord(ch) of
                    80: begin
                        chon:=chon+1;
                        if chon>n then chon:=1;
                    end;
                end;
            end;
    until ord(ch)=13;
    chon1:=chon;
    vietxy(x,tdy[chon1],mn,mni[chon1]);
    chon1:=chon;
end;

```

```

        chon:=chon-1;
        if chon<1 then chon:=n
        end;
    end;
end;
if ord(ch)=27 then begin chon:=0; break; end;
if (ord(ch)>=49)and(ord(ch)<=57)then
if (abs(48-ord(ch))<=n) then chon:=abs(48-ord(ch));
if chon<>chon1 then
begin
    vietxy(x,tdy[chon],ms,mni[chon]);
    vietxy(x,tdy[chon1],mn,mni[chon1]);
    chon1:=chon;
end;
until ord(ch)=13;
RestoreScreen(x,y,x+length(mni[1]),y+n);
VMenu:=chon;
end;

```

Chương trình con MenuControl điều khiển thực đơn.

```

Procedure MenuControl;
Var HMni:Menu;
    VMni:array[1..NoMenu]of Menu;
    hchon, vchon, x:byte;
Begin
    hmni[1]:=' Cau truc &Lenh ';
    hmni[2]:=' &Chuong trinh con ';
    hmni[3]:=' &Mang ';
    hmni[4]:=' Chuoi &ky tu ';
    hmni[5]:=' &Ban ghi ';
    hmni[6]:=' &Tep tin ';
    hmni[7]:=' Tro &giup';
    vmni[1][1]:='&1. Lenh tuan tu    ';
    vmni[1][2]:='&2. Lenh If then    ';
    vmni[1][3]:='&3. Lenh Case of    ';
    vmni[1][4]:='&4. Lenh For to do   ';
    vmni[1][5]:='&5. Lenh Repeat until';
    vmni[1][6]:='&6. Lenh While do   ';
    vmni[2][1]:='&1. Bai toan co ban  ';
    vmni[2][2]:='&2. CTC de qui      ';

    vmni[2][3]:='&3. Hien thi so Fibo  ';
    vmni[2][4]:='&4. Lenh For to do   ';
    vmni[2][5]:='&5. Lenh Repeat until';
    vmni[2][6]:='&6. Lenh While do   ';
    Repeat
        hchon:=HMenu(1,1,hmni,7,x);
        case hchon of
            1: Begin

```

```

        vchon:=VMenu(x,2,vmni[1],6);
    End;
2: Begin
    vchon:=VMenu(x,2,vmni[2],6);
    End;
3: Begin
    vchon:=VList(x,2,4,vmni[2],6);
    gotoxy(10,20); write(vchon);
    End;
End;
Until hchon=0;
End;

```

Câu hỏi và bài tập

Xây dựng chương trình có hệ thống menu, cửa sổ... tích hợp các chương trình con trước đó thành một chương trình lớn.

Chương 6. Dữ liệu kiểu mảng

Mục tiêu bài học:

- Biết khai báo với mảng 1 chiều, 2 chiều.
- Biết phân biệt địa chỉ ô nhớ và nội dung ô nhớ. Biết sử dụng vòng lặp để truy nhập đến địa chỉ từng phần tử để nhập thông tin.
- Biết xử lý trên 1 số dòng 1 số cột của mảng 2 chiều sắp xếp được mảng 1 chiều có thể lấy ví dụ về mảng nhiều chiều.

6.1. Khái niệm và phân loại mảng

Mảng là kiểu có cấu trúc dùng để chỉ định một nhóm đối tượng có cùng một kiểu dữ liệu nào đó. Ta có thể truy nhập đến từng phần tử của mảng thông qua chỉ số của chúng.

Có thể phân loại thành mảng một chiều, mảng hai chiều, mảng nhiều chiều. Thông thường mảng 1 chiều được coi là một dãy các phần tử. Mảng hai chiều được coi là một ma trận.

6.2. Mảng một chiều

6.2.1. Khai báo mảng

VAR tên_mảng : ARRAY [chỉ số] OF kiểu_phần_tử ;

Trong đó: VAR, ARRAY và OF là từ khoá,

Tên biến_mảng là một tên gọi tự đặt theo qui tắc đặt tên.

Kiểu_phần_tử là kiểu mà các phần tử của mảng lưu trữ trong từng ô nhớ.

chỉ_số: Chỉ số thường dùng kiểu dữ liệu miền con: cận đầu .. cận cuối, cận đầu, cận cuối xác định giá trị chỉ số đầu và chỉ số cuối của mảng.

Ví dụ

```
VAR SV: ARRAY [1..12] OF REAL ;
    M : ARRAY [ 'a'..'k' ] of Integer ;
    A, B: ARRAY [5 .. 19] of Word ;
    H: ARRAY[ Char ] of Boolean ;
Procedure Nhap;
var i:byte;
Begin
    Writeln('Nhap cac phan tu mang');
    For i :=1 to 19 do
    Begin
        write('a[',i,']='); readln(a[i]);
    end ;
```

```

End ;
Procedure Xuat;
var i:byte;
Begin
    Writeln('Cac phan tu trong mang la');
    For i :=1 to 19 do
        Begin
            write(a[i]:7) ;
        end ;
    End ;

```

6.2.2. Truy xuất phần tử

Tên_mảng [chỉ số]:= giá_trị;

Biểu_thức := Tên_mảng[chỉ_số];

Ví dụ mảng SV đã khai báo ở trên, sau khi nhập các phần tử vào các địa chỉ, ta có hình ảnh như sau:

8.5	10.5	6.5	8.5	11.5	9.5	15.5	11.5	15.5	15	15.5	13.5
1	2	3	4	5	6	7	8	9	10	11	12

Ta có $SV[2] = 10.5$; $SV[11 \text{ MOD } 4] = 6.5$; $SV[\text{SUCC}(9)] = 15$;

Ví dụ:

Các số hạng F của dãy Fibonacci được thiết lập theo quy tắc như sau:

$F(1) = F(2) = 1$;

$F(i) = F(i - 1) + F(i - 2)$ với $i > 2$.

Hãy lập chương trình để viết ra 48 số hạng đầu của dãy Fibonacci, sau đó tính tổng của các số hạng này.

Cách viết n số hạng đầu của dãy Fibonacci là rõ ràng bởi vì quy tắc thiết lập số hạng thứ n đã được cho theo công thức quy nạp ở trên.

Ta lưu trữ các số hạng này vào mảng một chiều để có thể truy nhập từng phần tử theo tên mảng và chỉ số của phần tử đó.

Ta có chương trình như sau:

```

Program Fibonacci;
Uses Crt ;
Const n = 48 ;
Var    F: array[1 .. n] of real ;
        i : Byte ; s : real ;
BEGIN
    ClrScr ; F[1] := 1 ; F[2] := 1 ; s := 2 ;

```



```

For i := 3 To n Do
Begin
    F[i] := F[i-1] + F[i-2];
    s := s + F[i];
End;
Window (1, 1, 35, 25);
For i := 1 To 24 Do
    Writeln ('so hang thu ', i:3, ' cua day la ', F[i]:6:0);
Window (42, 1, 80, 25);
For i := 25 To n Do
    Writeln ('so hang thu ', i:3, ' cua day la ', F[i]:12:0);
Write ('Tong ', n, ' so hang dau tien la ', s:13:0);
Readln;
End.

```

6.2.3. Tìm kiếm phần tử

Trong giải thuật đã trình bày, việc tìm kiếm phần tử trong mảng có hai phương pháp đó là tìm kiếm tuần tự và tìm kiếm nhị phân.

Tư tưởng của thuật toán tìm kiếm tuần tự là lần lượt so sánh các phần tử trong mảng, từ phần tử đầu tiên đến phần tử cuối cùng. Nếu bắt gặp một phần tử nào đó cần tìm thì quá trình tìm kiếm thành công và kết thúc. Nếu duyệt đến hết mảng mà không thấy phần tử cần tìm thì quá trình tìm kiếm là thất bại.

```

Function TimkiemTT(M:Mang;n:integer; X:Item):Index;
Var i: Integer;
BEGIN
    For i := 1 To n Do
        If (M[i] = X) then
            Begin
                TimkiemTT := i;
                Break;
            End;
        If (i>n) then TimkiemTT := 0;
    End;

```

Thuật toán tìm kiếm nhị phân phải tuân theo hai quy tắc sau:

- Dãy số phải được sắp xếp theo thứ tự tăng dần hoặc giảm dần.
- Dãy số được chia đôi, lấy phần tử ở giữa và so sánh với phần tử cần tìm, xảy ra 3 tình huống sau:
 - + Nếu nhỏ hơn: Tìm ở nửa trước
 - + Nếu lớn hơn: Tìm ở nửa sau
 - + Nếu bằng: tìm kiếm thành công.

Sau đây là chương trình con mô tả thuật toán tìm kiếm nhị phân

```

Function TimkiemNP(M:Mang; l,r:Integer; X:Item):Index;
Var k:Integer; timthay:integer;
Begin
  timthay:=0;
  While l<r do
  Begin
    k := (l+r) div 2;
    if (M[k] = X) then
      Begin timthay := k; break; End;
    if (M[k] < X) then l := k+1;
    if (M[k] > X) then l = k-1;
  End;
  TimkiemNP := timthay;
end;

```

6.2.4. Sắp xếp mảng

Sắp xếp mảng có rất nhiều thuật toán sắp xếp khác nhau, trong phạm vi bài giảng này tác giả giới thiệu 3 thuật toán sắp xếp cơ bản đó là: Sắp xếp chọn (Selection sort), sắp xếp chèn (Insertion sort), sắp xếp nổi bọt (Bubble sort).

Sắp xếp chọn

Tư tưởng thuật toán: Duyệt từ vị trí k+1 về cuối dãy, chọn ra chỉ số có phần tử nhỏ nhất sau đó đổi chỗ cho phần tử thứ k.

```

Function SelectionSort(var M:Mang; n:Integer);
Var k, i, x:Integer; tg:Item;
Begin
  For k:=1 to n-1 do
  Begin
    x:=k;
    For i:=k+1 to n do;
      If M[i] < M[x] then x := i;
    If (x<>k) then
      Begin
        tg:=M[x]; M[x]:=M[k]; M[k]:=tg;
      End;
    End;
  End;
end;

```

Sắp xếp chèn

Tư tưởng thuật toán: Duyệt từ vị trí k-1 về 1, tìm được vị trí thích hợp với phần tử thứ k và rồi chèn phần tử k vào đúng vị trí đó.

```

Function InsertionSort(var M:Mang; n:Integer);
Var k, i:Integer;
Begin

```

```

For k:=2 to n do
Begin
  i:=k-1;
  While (M[k] < M[i])and(i>0) do
  Begin
    M[i+1] := M[i];
    i:=i-1;
  End;
  M[i+1]:=M[k];
End;
end;

```

Sắp xếp nổi bọt

Tư tưởng thuật toán: Duyệt từ cuối dãy về đầu dãy, trên đường đi nếu có hai phần tử ngược thứ tự thì đổi chỗ.

```

Function BubbleSort(var M:Mang; n:Integer);
Var j, i:Integer; tg:Item;
Begin
  For i:=2 to n do
  Begin
    For j:=n downto i do;
    If M[j]<M[j-1] then
    Begin
      tg := M[j]; M[j] := M[j-1]; M[j-1] := tg;
    End;
  End;
End;
end;

```

6.3. Mảng hai chiều

6.3.1. Khai báo mảng hai chiều

VAR tên_mảng : ARRAY [chỉ số 1, chỉ số 2] OF kiểu_phần_tử ;

Mảng hai chiều tương tự như mảng một chiều, chỉ khác số chiều trong mảng là hai. Mảng hai chiều được coi như một ma trận có n dòng và n cột:

$$\begin{array}{ccc}
 a_{11} & a_{12} & a_{13} \\
 a_{21} & a_{22} & a_{23} \\
 a_{31} & a_{32} & a_{33}
 \end{array}$$

Ví dụ về mảng hai chiều: nhập ma trận 5x5 và hiển thị ma trận

```

TYPE Matran = ARRAY [1..5, 1..5] OF REAL;
VAR MT: Matran;
Procedure Nhap;
var i,j:byte;
Begin

```

```

        Writeln('Nhap cac phan tu trong ma tran');
        For i :=1 to 5 do
        For j :=1 to 5 do
        Begin
            write('a[',i,',',j,']='); readln(mt[i,j]);
        end ;
    End ;
    Procedure Xuat;
    var i,j:byte;
    Begin
        Writeln('Cac phan tu trong mang la');
        For i :=1 to 5 do
        Begin
            For j :=1 to 5 do
            Begin
                write(mt[i,j]:6);
            end ;
        end ;
        writeln;
    End;
End ;

```

6.3.2. Một số bài toán về ma trận

Bài toán 1. Tính tổng các phần tử nằm trên đường chéo chính của ma trận vuông n.

$$\begin{array}{ccc}
 \boxed{a_{11}} & a_{12} & a_{13} \\
 a_{21} & \boxed{a_{22}} & a_{23} \\
 a_{31} & a_{32} & \boxed{a_{33}}
 \end{array}$$

Ta nhận thấy chỉ số của các phần tử nằm trên đường chéo chính bằng nhau, và số phần tử bằng n. Vì vậy ta cần một vòng lặp từ 1 đến n và tính tổng các phần tử có chỉ số i=j

```

Function Tong1(M: Matran, n: integer):Longint;
var i:byte; t:longint;
Begin
    t:=0;
    For i :=1 to n do t := t + m[i,i];
    Tong1 := t;
End;

```

Bài toán 2. Tính tổng các phần tử nằm phía trên đường chéo chính của ma trận vuông n.

$$\begin{array}{ccc}
 a_{11} & \boxed{a_{12}} & \boxed{a_{13}} \\
 a_{21} & a_{22} & \boxed{a_{23}} \\
 a_{31} & a_{32} & a_{33}
 \end{array}$$

Ta nhận thấy tại mỗi dòng i thì ta cần hiển thị phần tử tại cột $i+1$ về cuối mỗi dòng

```
Function Tong2(M: Matran, n: integer):Longint;  
var i,j:byte; t:longint;  
Begin  
  t:=0;  
  For i :=1 to n do  
    For j :=i+1 to n do t := t + m[i,i];  
  Tong2 := t;  
End;
```

Câu hỏi và bài tập

1. Viết chương trình cho máy nhận vào mảng $A [1..n]$ ($n > 8$), các phần tử là số tự nhiên. Xác định số lượng các phần tử thoả mãn:

- Là số chính phương.
- $2^k < A[k] < k!$
- Chỉ số là số lẻ, giá trị phần tử là số chẵn.
- $a[k] < k^2$

2. Viết chương trình cho máy làm những việc sau:

- Nhận vào mảng $A [1..n]$ ($n > 9$), các phần tử là số thực lớn hơn -2 và nhỏ hơn 2.
- Tính trung bình cộng của các phần tử dương của mảng.
- So sánh số phần tử dương với số phần tử âm của mảng.
- Tìm phần tử nhỏ nhất và lớn nhất của mảng. Chỉ ra vị trí và giá trị của chúng.
- Tính $a[1] + a[2]^2 + a[3]^3 + \dots + a[n]^n$

3. Cho n số thực ($n > 7$) a_1, a_2, \dots, a_n . Hãy tính

4. Có n người xếp hàng một để mua hàng. Thời gian người bán hàng phục vụ khách hàng thứ i là t_i . Viết chương trình nhận vào các t_i và tính thời gian c_i mà khách hàng thứ i phải chờ để bắt đầu đến lượt mình.

5. Cho $A [1..n]$ ($n > 7$), các phần tử là số tự nhiên với $A [i] = i^2 + 2$.

- Tính tổng các căn bậc 2 của các phần tử chia cho 7 dư 3
- Tìm các phần tử là số nguyên tố của mảng. Chỉ ra vị trí và giá trị của chúng.
- Xây dựng mảng B có các phần tử là các số nguyên tố của mảng A .

6. Có các loại tiền 1.000, 2.000, 5.000, 10.000, 20.000 và 50.000 đồng. Viết chương trình in lên màn hình cách trả tiền cho n nghìn đồng sao cho tổng số tờ giấy bạc là nhỏ nhất.

7. Có n thiết bị dùng điện với điện trở tương ứng là R_1, R_2, \dots, R_n .

Tính điện trở tương đương của n thiết bị trên khi:

- Các thiết bị mắc song song với nhau.
- Các thiết bị mắc nối tiếp với nhau.

8. Cho mảng A các số nguyên với các dòng được đánh số từ 1 đến m và các cột được đánh số từ 1 đến n ($m > 4$, $n > 6$), trong đó $A[i, j] = i^2 + j$.

- Tính tổng S1 gồm các phần tử ở hai cột 3 và 5.
- Tính tổng S2 gồm các phần tử ở dòng 5 và cột 7 (phần tử ở ô là giao của dòng 5 và cột 7 chỉ cộng một lần).
- Tính tổng các phần tử là số nguyên tố của mảng.

9. Cho hai số thực a, b và n cặp số thực $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. a, b là tọa độ trường học. x_i, y_i (i nhận giá trị từ 1 đến n) là tọa độ của n ngôi nhà. Tính khoảng cách từ những ngôi nhà đến trường học, Khoảng cách trung bình của chúng, khoảng cách nhỏ nhất, lớn nhất.

10. Mảng A có n dòng n cột các phần tử là những số thực ($n > 4$)

- Tính tổng các phần tử nằm trên đường chéo phụ.
- Tính tổng các phần tử nằm song song sát trên đường chéo chính.
- Tính tổng các phần tử nằm song song sát dưới đường chéo chính.
- Tính tổng các phần tử nằm song song sát trên đường chéo phụ.
- Tính tổng các phần tử nằm song song sát dưới đường chéo phụ.
- Tính tổng các phần tử nằm trên đường chéo chính và ở phía trên đường chéo chính.
- Tính tổng các phần tử nằm ở phía dưới đường chéo phụ.

11. Lập chương trình cho máy tính viết lên màn hình tam giác PASCAL gồm K+1 dòng là hệ số của khai triển nhị thức Niu Tơn $(x+y)^n$ với n nhận các giá trị từ 0 đến K. K là một số tự nhiên không quá 12 nhận vào từ bàn phím. Ví dụ k=4 thì màn hình sẽ in ra 5 dòng như sau

12. Viết chương trình nhận vào chiều cao của các bạn Mai, Lan, Cúc, Thu, Bích Thủy, Hồng Nhung, Anh. Sắp xếp và in ra màn hình chiều cao của các bạn theo thứ tự tăng dần. Yêu cầu khai báo kiểu liệt kê tương ứng với tên các bạn để làm chỉ số cho mảng lưu trữ chiều cao của họ.

13. Viết chương trình cho máy nhận vào mảng A có n phần tử là những số thực ($n > 9$). Xây dựng mảng B các phần tử là những số nguyên được tạo thành từ các phần tử của mảng A theo cách $B[i]$ là trị tuyệt đối của phần nguyên của $A[i]$. Sắp xếp và in ra mảng A, mảng B theo thứ tự tăng dần.

14. Ba khối lớp 10, 11, 12 nộp sắt vụn, các lớp đặt tên theo khối bắt đầu từ A. Viết chương trình cho máy nhận vào lượng sắt vụn của từng lớp theo từng khối. Sau đó sắp xếp số lượng sắt vụn của các lớp trong từng khối theo thứ tự giảm dần.

4. Cho mảng m dòng n cột ($m > 5$, $n > 6$), các phần tử là những số nguyên ngẫu nhiên từ 0 đến 30000. Cho máy chọn ra các phần tử là số nguyên tố và sắp xếp chúng theo thứ tự tăng dần

15. Vòng chung khảo thi tiếng hát truyền hình, có m thí sinh, có n giám khảo. Mỗi thí sinh hát hai bài, sau mỗi bài các giám khảo cho điểm ngay. Ban thư kí tính điểm trung bình cho từng bài. Điểm kết quả tổng hợp của thí sinh đó được tính như sau: điểm trung bình của bài thứ nhất cộng với hai lần điểm trung bình của bài thứ hai rồi lấy tổng này chia cho 3.

16. Viết chương trình cho máy nhận vào điểm của từng bài cho mỗi thí sinh. Có bao nhiêu thí sinh có điểm kết quả tổng hợp cao nhất?

Yêu cầu viết chương trình theo ba cách sau đây:

- Dùng hai mảng hai chiều, mỗi mảng có m dòng, $n + 1$ cột để lưu trữ n điểm thành phần và điểm trung bình của từng bài. Sau đó dùng mảng một chiều có m phần tử để lưu trữ điểm tổng hợp của mỗi thí sinh.
- Dùng mảng một chiều có m phần tử, mỗi phần tử là mảng hai chiều có 2 dòng và n cột để lưu trữ điểm thành phần cho từng bài hát. Có thể dùng thêm mảng hai chiều KQ như chương trình đã làm.
- Chỉ dùng một mảng ba chiều (không dùng thêm mảng hai chiều như chương trình đã làm) để giải bài toán trên. Lưu điểm thành phần của từng bài, điểm trung bình của từng bài điểm tổng hợp của 2 bài của mỗi thí sinh cùng trên mảng ba chiều.

17. Mảng A bốn chiều. Để theo dõi tình hình phát nầm nhóm thức ăn (tính bằng gam) cho mười vận động viên quốc gia trong một tuần, mỗi ngày ba bữa, như mục 6 mảng nhiều chiều ở trên. Viết chương trình nhận vào các phần tử của mảng A. Tính xem nhóm thức ăn nào tiêu thụ nhiều nhất, ít nhất (tính đến hết tuần). Ngày thứ ba, bữa chiều, vận động viên nào dùng nhiều nhất thức ăn nhóm 2.

18. Mảng TG năm chiều. Viết chương trình mô phỏng việc nhận tiền gửi tiết kiệm trong mười năm từ 2001 đến 2010 của ba ngân hàng là: Công thương, Nông nghiệp và phát triển nông thôn, Thương mại dịch vụ. Mỗi năm truy nhập đến từng tháng. Mỗi tháng truy nhập đến từng loại tiền Việt Nam đồng, Đô la Mỹ, Ô rô. Mỗi loại tiền có kì hạn gửi gồm: không kì hạn, kì hạn ba tháng, sáu tháng, chín tháng và một năm.

Hướng dẫn. Khai báo mảng TG (tiền gửi) 5 chiều có các chỉ số năm, ngân hàng, tháng, tiền, kì hạn sau đó nhận vào từng phần tử của mảng. Tính số tiền Đô la Mỹ gửi kì hạn một năm ở quí 2 của ngân hàng Công thương năm 2002 (dùng hàm ngẫu nhiên cho việc nhận vào từng phần tử của mảng TG).

Chú ý. Khi thực hiện có thể hạn chế kích thước của mảng, vì số phần tử của mảng sẽ là tích của số lượng các phần tử của năm chỉ số.

Chương 7. Dữ liệu kiểu xâu ký tự

Mục tiêu bài học:

- Biết 2 cách khai báo xâu dùng từ khóa string và hạn chế số ký tự tối đa.
- Biết cách truy cập đến từng ký tự của xâu khi đã có giá trị.
- Biết sử dụng được các thủ tục chèn, xóa, đổi số thành xâu và ngược lại.
- Đặt thuộc tính cho màu chữ, màu nền.
- Biết sử dụng các hàm cho độ dài của xâu. Tìm vị trí xâu con trong xâu, cộng xâu...

7.1. Khái niệm xâu ký tự

Một dãy các ký tự trong bảng mã ASCII liên tiếp nhau gọi là xâu ký tự hay chuỗi ký tự. Khai báo xâu ký tự trong Pascal ta dùng từ khóa String.

Cấu trúc một xâu ký tự lưu vào biến s được mô tả như sau với các ký tự “Khoa công nghệ thông tin”.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	K	h	o	a		c	o	n	g		n	g	h	e		t	h	o	n	g		t	i	n

Khi đó ta có:

$s[1] = 'K'$ $s[2] = 'h'$ $s[9] = 'g'$.

$s[0]$ chứa ký tự có mã là độ dài của chuỗi s. ký tự có mã 24 vì độ dài của chuỗi s là 24. Và ta có $ord(s[0]) = 24$.

Như vậy một chuỗi ký tự có thể coi là một mảng ký tự, cách thức truy xuất các ký tự tương tự như truy xuất phần tử mảng.

7.2. Khai báo và truy xuất

- Khai báo chuỗi ký tự có độ dài tối đa 255 ký tự:

VAR s1: String;

- Khai báo chuỗi ký tự có độ dài tối đa là N ($N \leq 255$)

VAR s2: String[N];

- Nếu dùng giới hạn chuỗi ký tự làm đối của chương trình con ta có thể khai báo kiểu xâu ký tự:

TYPE ST25: String[25];

Procedure CTC(s1, s2 : ST25);

7.3. Các hàm về chuỗi ký tự

LENGTH(st): Cho số ký tự của chuỗi st kể cả dấu cách.

Uppcase(st[i]): Đổi ký tự thứ i của chuỗi st sang viết hoa (nếu nó là ký tự viết thường).

COPY(st, p, n): Cho kết quả là một chuỗi con gồm n ký tự của chuỗi st kể từ ký tự thứ p tính từ trái sang phải.

POS(st1, st2): Cho vị trí xuất hiện lần đầu tiên của chuỗi st1 trong chuỗi st2.

Phép cộng chuỗi ký tự (st1 + st2) cho chuỗi kết quả nhận được bằng cách viết chuỗi st2 sau chuỗi st1.

DELETE(st, p, n): Xóa khỏi chuỗi st một chuỗi con gồm n ký tự kể từ ký tự thứ p và kéo các ký tự còn lại ở phía cuối chuỗi về vị trí p.

INSERT(st1, st2, p): Chèn chuỗi st1 vào chuỗi st2 kể từ ký tự có vị trí thứ p của chuỗi st2, tính từ trái sang phải. Các ký tự của chuỗi st2 từ vị trí p bị đẩy về cuối chuỗi.

Val(st, v, c): Thủ tục biến đổi biểu thức chuỗi ký tự số st thành số và gán kết quả cho biến số v. Biến c là biến nguyên dùng để ghi vị trí ký tự đầu tiên gây ra lỗi khi chuyển đổi từ st sang v. Không có lỗi thì c = 0. Việc có đổi được hay không là phụ thuộc vào biểu thức chuỗi và khai báo kiểu của biến số v.

Str(v, st): Thủ tục biến đổi giá trị biểu thức số v thành chuỗi ký tự và gửi kết quả vào biến chuỗi st. Chuỗi st sẽ nhận được từ số v theo qui cách in. Ví dụ: str((3.12 + 5.23547) :10:2, st) thì st sẽ là chuỗi ' 8.36' như vậy trước ký tự 8 còn 6 dấu cách.

7.4. Bài toán

Viết hàm nhận vào hai chuỗi số có thể dài tối đa 255 số, sau đó cộng hai chuỗi số.

```
Function cong2s(s1,s2:string):string;  
var i,d,c,nho,m,n:byte; s:string;  
begin  
  m:=length(s1); n:=length(s2);  
  s:="";nho:=0;  
  while (m>0)and(n>0) do  
  begin  
    d:=ord(s1[m])-48;  
    c:=ord(s2[n])-48;  
    s:=chr(48 + (nho+d+c) mod 10)+s;  
    nho:=(nho+d+c) div 10;  
    n:=n-1; m:=m-1;  
  end;  
  for i:=n downto 1 do  
  begin  
    d:=ord(s2[i])-48;  
    s:=chr(48 + (nho+d) mod 10)+s;
```

```

    nho:=(nho+d) div 10;
end;
for i:=m downto 1 do
begin
    d:=ord(s1[i])-48;
    s:=chr(48 + (nho+d) mod 10)+s;
    nho:=(nho+d) div 10;
end;
if nho > 0 then s:=chr(48 + nho)+s;
cong2s:=s;
end;

```

Câu hỏi và bài tập

- Viết chương trình thay tất cả các kí tự a trong xâu bằng kí tự z và tìm số lần xuất hiện kí tự a trong xâu đó.
- Viết chương trình cho máy thông báo lên màn hình tên 6 nước châu Á, người đối thoại với máy phải nhập tên thủ đô của nước đó. Nếu vào đúng thì máy khen, nếu nhập sai thì máy báo "Sai rồi" và máy cho hiện tên thủ đô nước đó lên màn hình.
- Viết chương trình đảo ngược các kí tự của một xâu.
- Cho máy nhận vào một xâu. Kiểm tra xem xâu có đối xứng không? Ví dụ 'abcdcba' là đối xứng.
- Sắp xếp tên của n người theo thứ tự tăng dần, hãy viết chương trình theo yêu cầu có ba thủ tục: một thủ tục in mảng, một thủ tục sắp xếp, một thủ tục nhận vào danh sách. Trong đó thủ tục vào danh sách gọi thủ tục sắp xếp, thủ tục sắp xếp gọi thủ tục in mảng.
- Mỗi buổi chào cờ đầu tuần, các lớp xếp hàng theo tổ theo thứ tự từ thấp đến cao. Viết chương trình cho máy nhận vào tên và chiều cao của từng người theo tổ. Sau đó máy in ra tên theo chiều cao tăng dần.
- Trong danh sách có tên n người. Tìm trong danh sách đó những nhóm người trùng tên theo yêu cầu: Tên là gì, có bao nhiêu người, thứ tự của họ trong danh sách.
- Cho biết năm 1945 là năm Ất Dậu. Có nghĩa can là ất, chi là dậu. 12 hàng chi là Thân, Dậu, Tuất, Hợi, Tý, Sửu, Dần, Mão, Thìn, Tị, Ngọ, Mùi. 10 hàng can là Canh, Tân, Nhâm, Quý, Giáp, Ất, Bính, Đinh, Mậu, Kì.
- Viết chương trình để ta vào năm dương lịch. Máy thông báo lên màn hình can và chi tương ứng.
- Đội tuyển thi học sinh giỏi tin học có các bạn Mai, Chi, Lan, Thu, Nghi, Anh, Minh, Hoa, Dung. Viết chương trình cho máy tính làm những việc sau:
 - Nhập điểm cho từng người, với yêu cầu máy cho hiện tên của người nào đó để ta nhập vào điểm cho họ từ bàn phím.

b. Bao nhiêu người đạt điểm cao nhất, thấp nhất?

c. Ta hỏi ai đó thì máy cho biết họ được điểm mấy. Việc hỏi được lặp đi lặp lại nhiều lần, mỗi khi trả lời xong cho một người, máy hỏi có hỏi nữa không và nhận tên người tiếp theo. Việc hỏi kết thúc khi máy hỏi có hỏi nữa không và nhận được kí tự K hoặc k từ bàn phím báo cho máy biết là không.

d. Sắp xếp theo tên tăng dần (tăng dần của xâu ví dụ 'Dung' < 'Mai') và in lên màn hình tên và điểm của họ sau khi sắp xếp.

10. Viết chương trình nhận vào danh sách có N người, mỗi người được cho bởi họ và tên. Sắp xếp theo tên, nếu hai người có tên trùng nhau thì sắp theo họ (có thể có nhiều người cùng tên).

11. Viết chương trình cho máy nhận vào n câu thơ ($n > 5$), sau đó máy cho hiện lên màn hình những câu có từ "hoa" (mỗi câu chỉ in một lần. Chú ý từ được phân biệt nhờ dấu cách).

12. Cho xâu st trong đó có duy nhất một dấu mở ngoặc "(" và một dấu đóng ngoặc ")". Giữa hai dấu ngoặc này có một số kí tự. Hãy xoá khỏi xâu st các kí tự từ dấu mở ngoặc đến dấu đóng ngoặc.

13. Nhận vào xâu st. Hãy tìm trong st có bao nhiêu từ 'ong' thì thay bằng bấy nhiêu xâu 'con ong ngoan'.

14. Viết chương trình cho máy nhận vào xâu X. Hãy kiểm tra xem trong X có hai từ mà từ này là đảo ngược của từ kia không?

15. Viết chương trình cho xâu 'CONG NGHE THÔNG TIN' chạy từ phải sang trái màn hình. Đầu tiên các kí tự lần lượt xuất hiện, tiếp đến toàn bộ xâu chạy qua màn hình, rồi từng kí tự lại khuất vào trong màn hình.

16. Viết chương trình cho máy nhận vào xâu X gồm họ, đệm và tên của một người (xâu X có thể có nhiều dấu cách ở đầu xâu, cuối xâu, giữa các từ, các kí tự viết thường, viết hoa là tùy tiện). Tách ra từ X để có xâu T là tên (tên là từ cuối cùng trong xâu X), xoá ở xâu X tên của người đó để còn lại là họ và đệm.

17. Tìm tất cả các số tự nhiên không lớn hơn n sao cho số đó trùng với phần cuối của bình phương chính nó. Ví dụ $6^2 = 36$; $25^2 = 625$;

18. Viết chương trình cho máy nhận vào một số tự nhiên n.

a. Chữ số 3 có xuất hiện trong cách viết thập phân của n^2 không?

b. Viết n theo thứ tự ngược lại.

c. Tráo đổi chữ số đầu tiên và chữ số sau cùng của n.

19. Viết chương trình cho máy nhận vào một số thực n. Loại bỏ các chữ số 0 ở phần thập phân của n.

20. Viết chương trình cho máy nhận vào hai số tự nhiên m và n. Hãy tìm các số tự nhiên bé hơn n mà có tổng các bình phương các chữ số của nó bằng m.

Chương 8. Dữ liệu kiểu bản ghi

Mục tiêu:

- Biết cách khai báo bản ghi
- Biết 2 cách truy nhập vào 1 trường của bản ghi.

8.1. Khái niệm về bản ghi

Kiểu dữ liệu bản ghi (RECORD) bao gồm nhiều thông tin mô tả về một đối tượng. Mỗi thông tin thành phần của một bản ghi được gọi là một trường của bản ghi đó. Một bản ghi có thể có nhiều trường, các trường có thể thuộc các kiểu khác nhau. Trường của bản ghi có tên gọi do người dùng đặt cho nó.

Ví dụ mô tả về sinh viên gồm các thông tin: họ tên, lớp có kiểu string, điểm có kiểu số thực, ngày sinh có kiểu cấu trúc... Các thông tin đó được gom lại thành một bản ghi.

8.2. Khai báo bản ghi

Để khai báo bản ghi ta dùng từ khoá RECORD và END; giữa hai từ khoá này là khai báo các trường. Trường được khai báo như sau:

Tên trường : Kiểu của trường;

Nếu có nhiều trường cùng kiểu thì có thể khai vào một nhóm trường, trong đó các tên trường được ngăn cách nhau bởi dấu phẩy (,).

Sau từ khoá END kết thúc khai báo bản ghi là dấu chấm phẩy (;).

Ví dụ khai báo bản ghi mô tả về sinh viên:

```
Type SINHVIEN = Record
    hoten: string;
    tuoi: byte;
    diem1, diem2, diem3: real;
    hocbong: real;
END;
Var sv: SINHVIEN;
```

8.3. Truy xuất thành phần trong bản ghi

Ta cần truy nhập đến trường nào của bản ghi thì ra lệnh cho chương trình bằng cách chỉ ra tên biến bản ghi, tiếp đó là dấu chấm (.) và sau đó là tên trường mà ta muốn truy nhập.

Ví dụ: sv.hoten := 'Nguyen Van Khuong';

Turbo Pascal cho phép truy nhập đến tất cả các trường của bản ghi bằng cách dùng câu lệnh WITH kèm theo DO. Câu lệnh WITH được viết theo dạng sau:

WITH Biến bản ghi DO Câu lệnh ;
Trong câu lệnh sau từ khoá DO ta muốn truy nhập đến trường nào của biến bản ghi thì chỉ cần viết tên trường đó.

Ví dụ:

```
with sv do  
    hoten := 'Nguyen Van Khuong';  
    tuoi := 26;  
    diem1 := 8; diem2 := 7; diem3 := 7;  
end;
```

8.4. Ví dụ áp dụng

Viết chương trình quản lý điểm và tính học bổng cho sinh viên.

- Khai báo cấu trúc dữ liệu:

```
type SINHVIEN=Record  
    hoten:string;  
    tuoi:byte;  
    diem:real;  
    hocbong:real;  
END;  
Var sv: array[1..100] of SINHVIEN; n:byte;
```

- Thủ tục nhập sinh viên tại vị trí x, y.

```
procedure Nhap(x,y:byte);  
begin  
    n:=0;  
    repeat  
        clrscr; n:=n+1;  
        gotoxy(x,y); Write('Nhap thong tin sinh vien');  
        gotoxy(x,y+1); write('Ho va ten:');  
        gotoxy(x,y+2); write('Tuoi: ');  
        gotoxy(x,y+3); write('Diem: ');  
        gotoxy(x+15,y+1); readln(sv[n].hoten);  
        gotoxy(x+15,y+2); readln(sv[n].tuoi);  
        gotoxy(x+15,y+3); readln(sv[n].diem);  
        writeln('ESC->ket thuc, Enter -> tiep tục');  
    until readkey=#27;  
end;
```

- Thủ tục hiển thị danh sách sinh viên

```
procedure Hienthi;  
var i:byte;  
begin  
    clrscr;
```

```

for i:=1 to n do
begin
    write(sv[i].hoten, ' ', sv[i].tuoi, ' ');
    writeln(sv[i].diem:5:2, ' ', sv[i].hocbong:8:1);
end;
end;

```

- Tính học bổng cho sinh viên

```

Procedure Tinhhocbong;
var i:byte;
Begin
    for i:=1 to n do
    begin
        if (sv[i].diem >=7)and(sv[i].diem<8) then sv[i].hocbong:=120000;
        if (sv[i].diem >=8)and(sv[i].diem<9) then sv[i].hocbong:=180000;
        if (sv[i].diem >=9)and(sv[i].diem<=10) then sv[i].hocbong:=240000;
    end;
end;

```

Câu hỏi và bài tập

1. Một cửa hàng tạp hóa nhập và bán các mặt hàng theo hóa đơn, hàng tháng chủ cửa hàng thống kê các mặt hàng bán chạy và tính số tiền lãi theo từng mặt hàng. Viết chương trình quản lý hàng hóa theo yêu cầu của chủ cửa hàng trên.
2. Thư viện trường học muốn quản lý sách mượn trả của sinh viên. Mỗi ngày làm việc chương trình cho biết các sinh viên mượn sách quá hạn và in phiếu yêu cầu trả sách. Viết chương trình quản lý mượn trả sách theo yêu cầu của thư viện trên.

Chương 9. Dữ liệu kiểu tệp tin

Mục tiêu bài học:

- Biết sử dụng 2 loại tệp tuần tự và truy nhập trực tiếp.
- Phân biệt sự giống nhau và khác nhau của chúng.
- Biết qui trình làm việc với tệp đó là định vị tệp, mở tệp (tệp mới hoặc tệp đã tồn tại), đóng tệp, cập nhật thông tin vào tệp.

9.1. Khái niệm và phân loại tệp tin

Tệp cũng là dãy các dữ liệu cùng kiểu. Nhưng khác với mảng, số phần tử của tệp không được xác định trước, tệp được lưu trữ trong bộ nhớ ngoài.

Việc sử dụng tệp đáp ứng được 3 yêu cầu:

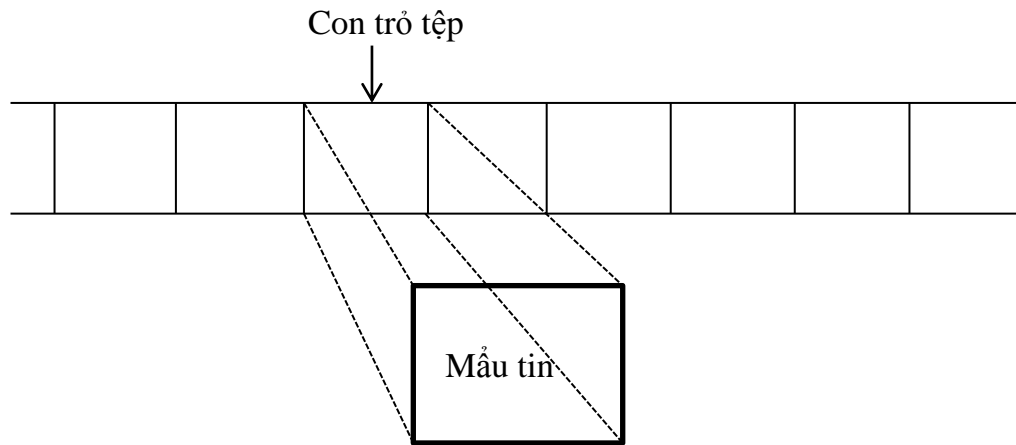
- Cất trữ được thông tin trong khi bộ nhớ (RAM) không giữ được thông tin khi máy mất điện.
- Lưu trữ một lượng thông tin lớn vượt khỏi sức chứa của bộ nhớ trong.
- Sao chép được thông tin từ đĩa này sang đĩa khác.

Hai thao tác cơ bản đối với tệp là ghi dữ liệu vào tệp và đọc dữ liệu từ tệp. Điểm khác của tệp so với dữ liệu lưu bộ nhớ trong:

- Kiểu tệp không thể là thành phần của bất cứ kiểu nào.
- Không được dùng lệnh gán đối với các biến tệp
- Chương trình không thể truy nhập trực tiếp dữ liệu trên tệp, mà phải thông qua hai thao tác cơ bản là đọc dữ liệu từ tệp ở trên đĩa xuống bộ nhớ trong và ghi dữ liệu ở bộ nhớ trong lên tệp ở trên đĩa.
- Tại một thời điểm thao tác đọc hoặc ghi chỉ làm việc với một thành phần của tệp ở cửa sổ tệp.
- Tham chiếu kiểu tệp trong đối số của các chương trình con bắt buộc phải là tham biến (không thể là tham trị).

9.2. Tệp có kiểu

Qui định kiểu dữ liệu của mẫu tin trong tệp, là các kiểu dữ liệu cơ bản hay kiểu của người sử dụng; tệp số nguyên (integer) qui định mỗi mẫu tin đọc được là một số nguyên (integer), tệp ký văn bản (text) qui định mỗi mẫu tin là một ký tự hay một dòng văn bản, tệp kiểu record có thể dùng thay cho mảng trong chương trình quản lý...



9.2.1. Khai báo

Muốn làm việc với tệp, ta phải khai báo biến tệp. Khai báo biến tệp thông qua kiểu hoặc khai báo trực tiếp trong var.

```
Type Tepsonguyen = FILE OF INTEGER;
Var F: Tepsonguyen; {khai bao thong qua kieu}
    G: File of Integer; {khai bao truc tiep}
```

9.2.2. Các hàm và thủ tục xử lý tệp tin

a. Định nghĩa tệp tin

Assign(biến tệp, tên tệp);

Trong đó biến tệp được khai báo ở trên (F hoặc G), tên tệp là tên hoặc đường dẫn tới tệp cần truy xuất (tên được dùng theo qui tắc đặt tên của DOS).

Hàm Assign tham chiếu biến tệp tới vùng dữ liệu của tệp tin trên đĩa. Các thao tác đọc, ghi dữ liệu ta làm việc thông qua biến tệp.

Ví dụ: Assign (F, 'songuyen.dat');

Trong trường hợp này, tệp songuyen.dat được lưu ở thư mục hiện hành.

b. Mở tệp tin

Sau khi định nghĩa tệp tin, ta có thể mở tệp tin bằng thủ tục:

REWRITE (biến tệp); mở một tệp mới và đặt cửa sổ tệp vào vị trí đầu tiên để sẵn sàng làm việc. Chú ý nếu biến tệp trở vào một tệp đã tồn tại thì nội dung cũ của tệp này sẽ bị xóa.

RESET (biến tệp); mở một tệp đã tồn tại được tham chiếu bởi biến tệp và đặt con trỏ tệp vào vị trí đầu tiên của tệp, nếu tệp này không tồn tại thì thủ tục sẽ báo lỗi.

Để kiểm tra lỗi khi mở tệp ta có thể dùng dẫn hướng {\$I-} và {\$I+}. dẫn hướng {\$I-} để tắt chế độ tự kiểm tra vào ra của TP, dẫn hướng {\$I+} mở lại chế độ tự kiểm

tra của TP. Kết hợp với dẫn hướng này, lập trình viên có thể sử dụng hàm IOResult để kiểm tra trạng thái vào ra.

Hàm IOResult cho kết quả bằng 0 nếu thao tác mở tệp thành công, ngược lại sẽ cho mã lỗi tương ứng.

c. Đóng tệp tin

Sau khi thao tác với tệp tin xong, ta có thể đóng tệp tin bằng thủ tục CLOSE(biến tệp). Sau khi sử dụng thủ tục Close(biến tệp) hệ thống sẽ đẩy toàn bộ dữ liệu cần ghi đang lưu ở bộ nhớ tạm vào tệp.

d. Ghi dữ liệu lên tệp

WRITE (biến tệp, Danh sách mẫu tin); (1)

WRITELN (biến tệp, Danh sách mẫu tin); (2)

Ghi giá trị của các mẫu tin vào tệp được xác định bởi biến tệp. Thủ tục (2) khác thủ tục (1) ở chỗ: sau khi ghi giá trị của các mẫu tin thì chuyển cửa sổ tệp xuống dòng dưới. Trong danh sách, các biểu thức phân cách nhau bởi dấu phẩy (,).

e. Đọc dữ liệu từ tệp

READ (biến tệp, Danh sách biến) ; (1)

READLN (biến tệp, Danh sách biến) ; (2)

READLN (biến tệp); (3)

Đọc giá trị ở cửa sổ tệp hiện hành (xác định bởi biến tệp) vào danh sách biến. Các biến này có kiểu tương thích với các dữ liệu đã ghi trên tệp.

Thủ tục (2) khác thủ tục (1) ở chỗ: sau khi đọc giá trị cho danh sách biến thì chuyển cửa sổ tệp xuống dòng dưới.

Thủ tục (3) chuyển cửa sổ tệp tới đầu dòng tiếp theo.

f. Hàm EOF

EOF (biến tệp). Hàm có kiểu boolean, nhận giá trị TRUE (đúng) nếu cửa sổ tệp trở đến vị trí kết thúc tệp, ngược lại, hàm nhận giá trị FALSE (sai). Vị trí kết thúc tệp ở sau thành phần cuối cùng của tệp

g. Thủ tục đổi tên tệp

Rename(biến tệp, tên mới): đổi tên tệp tin được mở bởi biến tệp thành tên mới

h. Thủ tục xóa tệp tin

Erase(biến tệp): xóa tệp tin được mở bởi biến tệp.

9.2.3. Ví dụ áp dụng

Quay trở lại với ví dụ quản lý điểm sinh viên ở trên, nếu ta dùng mảng để lưu trữ dữ liệu, tức là sử dụng bộ nhớ RAM để lưu trữ, vậy nếu ta kết thúc chương trình thì toàn bộ dữ liệu sẽ bị mất. Vậy nên có thể dùng kiểu tệp tin để lưu trữ, dữ liệu sẽ tồn tại trong bộ nhớ thứ cấp nên không bị mất khi kết thúc chương trình.

Sau đây là đoạn mã chương trình quản lý điểm sinh viên:

```
type SINHVIEN=Record
    hoten:string;
    tuoi:byte;
    diem:real;
    hocbong:real;
END;
Var sv:SINHVIEN; n:byte;
    F:File of SINhVIEN;
procedure Nhap(x,y:byte; Tentep:String);
begin
    assign(F,tentep);
    {$I-} Reset(F); {$I+}
    if ioresult<>0 then rewrite(F);
    repeat
        clrscr;
        gotoxy(x,y); Write('Nhap thông tin sinh vien');
        gotoxy(x,y+1); write('Ho va ten:');
        gotoxy(x,y+2); write('Tuoi: ');
        gotoxy(x,y+3); write('Diem: ');
        gotoxy(x+15,y+1); readln(sv.hoten);
        gotoxy(x+15,y+2); readln(sv.tuoi);
        gotoxy(x+15,y+3); readln(sv.diem);
        write(F,sv);
        writeln('ESC->ket thuc, Enter -> tiep tục');
    until readkey=#27;
    close(F);
end;
procedure DocSV(tentep:string);
begin
    assign(F,tentep);
    {$I-} Reset(F); {$I+}
    if ioresult<>0 then
        begin
            writeln('Loi tep'); halt;
        end;
    while not eof(F) do
        begin
            read(F,sv);
            write(sv.hoten,' ', sv.tuoi, ' ');
            writeln(sv.diem:5:2, ' ',sv.hocbong:8:1);
```

```

end;
close(F);
end;
{----Tinh hoc bong cho sinh vien-----}
Procedure Tinhhocbong(tentep:string);
var G:File of sinhvien;
Begin
  assign(G,'tam.txt');
  rewrite(G);
  assign(F,tentep);
  {$I-} Reset(F); {$I+}
  if ioresult<>0 then
  begin
    writeln('Loi tep'); halt;
  end;
  while not eof(F) do
  begin
    read(F,sv);
    if (sv.diem >=7)and(sv.diem<8) then sv.hocbong:=120000;
    if (sv.diem >=8)and(sv.diem<9) then sv.hocbong:=180000;
    if (sv.diem >=9)and(sv.diem<=10) then sv.hocbong:=240000;
    write(G,sv);
  end;
  close(F);
  erase(F);
  close(G);
  rename(G,tentep);
end;

```

9.3. Tập văn bản

Tập văn bản lưu trữ thông tin dưới dạng văn bản, có thể là ký tự, số, hoặc một dòng văn bản. Việc đọc, ghi thông tin vào tập văn bản phụ thuộc vào biến hoặc mẫu tin cần ghi.

9.3.1. Khai báo

```

Type VANBAN = TEXT;
Var F: VANBAN; {khai bao thong qua kieu}
    G: TEXT; {khai bao truc tiep}

```

Các hàm và thủ tục truy xuất tệp tin tương tự như trong tệp có kiểu.

9.3.2. Ví dụ áp dụng

Ví dụ sau mô tả bài hát “tóc em đuôi gà”, chương trình này chỉ có hiệu quả với các máy tính có loa trong, phát ra các âm thanh có tần số giống như các nốt nhạc. Chương trình sẽ đọc một tệp văn bản và hiển thị lên màn hình.

```

Const
  t0=800; t1=600; t2=400; t3=300; t4=200; t5=150; t6=100; t7=25;

```

```

sol0=196; solt0=208; sol1=392; solt1=415; sol2=784; solt2=831;
la0=220; la1=440; la2=880; lat0=233; lat1=446; lat2=932;
si0=247; si1=494; si2=988;
do0=262; do1=523; do2=1047; dot0=277; dot1=554; dot2=2109;
re0=294; re1=587; re2=1175; ret0=311;ret1=622; ret2=1245;
mi0=330; mi1=659; mi2=1319;
fa0=349; fa1=698; fa2=1397; fat0=370; fat1=470; fat2=1480;
Var F:text; ok:Char; i,j,n,k:integer;
    tu:array[1..500] of string[7];
    cao, truong, lang: array[1..500] of integer;
    x:string;
{-----}
Procedure ngat(xau:string;var cc,tt,ll:integer;var xx:string);
var x1, x2: String[10]; ii,jj:integer;
    xau2:string[20]; xau5:string[9];
Begin
    ii:=pos(#32,xau)+1;
    if ii<>2 then xx:=copy(xau, 1, ii-2)
    else xx:=#32;
    x1:=copy(xau,ii,20);
    xau2:=copy(x1,1,2);
    Case xau2[2] of
        '0': tt:=t0;
        '1': tt:=t1;
        '2': tt:=t2;
        '3': tt:=t3;
        '4': tt:=t4;
        '5': tt:=t5;
        '6': tt:=t6;
        '7': tt:=t7;
    End;
    xau2:=copy(x1,3,2);
    Case xau2[2] of
        '0': ll:=t0;
        '1': ll:=t1;
        '2': ll:=t2;
        '3': ll:=t3;
        '4': ll:=t4;
        '5': ll:=t5;
        '6': ll:=t6;
        '7': ll:=t7;
    End;
    xau5:=copy(x1,5,5);
    if xau5[1] in ['s','S'] then
        if xau5[2] in ['i','I'] then
            case xau5[3] of
                '0': cc:=si0;
                '1': cc:=si1;

```

```

    '2': cc:=si2;
end else
case xau5[4] of
    '0': cc:=sol0;
    '1': cc:=sol1;
    '2': cc:=sol2;
    't','T': case xau5[5] of
        '0': cc:=solt0;
        '1': cc:=solt1;
        '2': cc:=solt2;
    end;
end else
case xau5[1] of
    'm','M': case xau5[3] of
        '0': cc:=mi0;
        '1': cc:=mi1;
        '2': cc:=mi2;
    end;
    'd','D': case xau5[3] of
        '0': cc:=do0;
        '1': cc:=do1;
        '2': cc:=do2;
        't','T': case xau5[4] of
            '0': cc:=do0;
            '1': cc:=do1;
            '2': cc:=do2;
        end;
    end;
    'r','R': case xau5[3] of
        '0': cc:=re0;
        '1': cc:=re1;
        '2': cc:=re2;
        't','T': case xau5[4] of
            '0': cc:=ret0;
            '1': cc:=ret1;
            '2': cc:=ret2;
        end;
    end;
    'f','F': case xau5[3] of
        '0': cc:=fa0;
        '1': cc:=fa1;
        '2': cc:=fa2;
        't','T': case xau5[4] of
            '0': cc:=fat0;
            '1': cc:=fat1;
            '2': cc:=fat2;
        end;
    end;
end;

```

```

    'l','L': case xau5[3] of
        '0': cc:=la0;
        '1': cc:=la1;
        '2': cc:=la2;
        't','T': case xau5[4] of
            '0': cc:=lat0;
            '1': cc:=lat1;
            '2': cc:=lat2;
        end;
    end;
End;
End;
{-----}
Procedure Choi(cc,tt,ll:integer;k:integer);
Begin
    if ll<>t7 then
        Begin
            sound(cc*k); delay(tt*k); nosound;
        End else delay(ll*k);
    End;
{-----}
Procedure Viet(xauviet:string; mc,mn:byte);
Var xx,yy:byte;
Begin
    Textcolor(mc); Textbackground(mn);
    xx:=wherex; yy:=wherey;
    if xauviet<>#32 then
        if length(xauviet)+xx < 79 then write(xauviet,#32)
        else
            if length(xauviet)+xx=79 then write(xauviet)
            else
                begin gotoxy(1,yy+1); write(xauviet,#32);end;
        End;
{-----}
Procedure Doc(var sopt:integer);
Begin
    clrscr;
    sopt:=0;
    While not eof(F) do
        Begin
            Readln(F,x); sopt:=sopt+1;
            Ngat(x,cao[sopt],truong[sopt],lang[sopt],x);
            tu[sopt]:=copy(x,1,7);
        End;
    End;
{-----}
Begin
    Assign(F,'tocem.txt');

```

```

{$I-}Reset(F);{$I+}
if ioresult=0 then
Begin
  k:=12;
  Doc(n);
  textbackground(1);
  clrscr;
  writeln('Bai hat: Toc em duoi ga':30);
  gotoxy(1,2);
  For i:=1 to n do
  Begin
    clrscr;
    writeln('Bai hat: Toc em duoi ga':30);
    For j:=1 to n do
    if i<>j then viet(tu[j], 14, 1)
    else viet(tu[i],14,5);
    choi(cao[i], truong[i], lang[j],k);
    if keypressed then break;
  End;
  Nosound;
End else begin writeln; write('Loi mo tep'); end;
End;
{-----}
Begin
if paramcount=2 then
Begin
  Assign(F,paramstr(1));
  {$I-}Reset(F);{$I+}
  if ioresult=0 then
  Begin
    val(paramstr(2),k,i);
    Doc(n);
    For i:=1 to n do
    Begin
      clrscr;
      writeln('Bai hat: Toc em duoi ga':30);
      For j:=1 to n do
      if i<>j then viet(tu[i], 14, 1)
      else viet(tu[j],14,5);
      choi(cao[i], truong[i], lang[j],k);
      if keypressed then break;
    End;
    Nosound;
  End else begin writeln;write('Loi mo tep:',paramstr(1));end;
End else
  Begin writeln; write('baihat tentep heso');end;
End;

```

Sau đây là định dạng và nội dung tệp tin tocem.txt

Nay t4l7mi0
 co t2l7fat0
 be t4l7sol0
 t2l7sol0
 co t4l7sol0
 mai t2l7sol0
 toc t4l7sol0
 duoi t2l7fat0
 ga. t4l7mi0
 t2l7mi0
 t2l2mi0
 Dap t4l7si0
 xe t4l7mi0
 tren t2l7sol0
 pho t2l7si1
 t1l7si1
 pho t4l7si0
 dong t4l7la0
 nguoi t2l7sol0
 qua. t4l7la0
 t1l7la0
 t1l2la0
 Cho t4l7fat0
 anh t2l7sol0
 voi t4l7la0
 t2l7sol0
 neu t4l7la0
 co t4l7la0
 lo t4l7la0
 ai t2l7sol0
 cuoi t4l7la0
 t2l7fat0
 t2l2fat0
 thi t4l7re0
 anh t4l7fat0
 se t2l7sol0
 noi t4l7la0
 t1l7la0
 "Tinh t4l7fat0
 co t4l7fat0
 sanh t2l7re0
 doi". t4l7si0
 t1l7si1
 t1l2si1
 Nay t4l7mi0
 co t2l7fat0
 be t4l7sol0
 t2l7sol0

co t4l7sol0
 mai t2l7sol0
 toc t4l7sol0
 duoi t2l7fat0
 ga. t4l7mi0
 t2l7mi0
 t2l2mi0
 Dap t4l7si0
 xe t4l7mi0
 nhanh t2l7sol0
 qua t4l7si1
 t1l7si1
 khien t4l7si1
 anh t4l7la0
 hut t2l7sol0
 hoi. t4l7la0
 t1l7la0
 t1l2la0
 Cho t4l7fat0
 anh t2l7sol0
 voi t4l7la0
 t2l7la0
 sap t4l7la0
 toi t4l7la0
 nga t4l7la0
 tu t2l7sol0
 roi t4l7fat0
 t2l7fat0
 t2l2fat0
 thi t4l7si0
 xin t4l7ret0
 em t2l7mi0
 hay t4l7fat0
 t1l7fat0
 cham t4l7si0
 cham t4l7la0
 ma t4l7si0
 thoi t4l7mi0
 t1l7mi0
 t0l7mi0
 Toc t4l7si1
 duoi t4l7si1
 ga t4l7la1
 trong t4l7si1
 gio t4l7do1
 bay t4l7si1
 nhe t4l7la1
 ngay t4l7si1

ngat t4l7do1
 long t2l7la1
 anh. t4l7do1
 t1l7do1
 t1l4do1
 Nang t4l7si1
 xuan t4l7la1
 hong t4l7sol1
 moi t4l7la1
 tham t4l7si1
 em t4l7la1
 cuoi t4l7sol1
 chum t4l7la1
 chim t4l7si1
 that t2l7sol1
 xinh. t4l7si1
 t1l7si1
 t4l4si1
 Uoc t4l7la1
 chi t4l7sol1
 lam t4l7fat0
 con t4l7sol1
 gio t4l7la1
 hay t4l7sol1
 la t4l7fat0
 tia t4l7sol1
 nang t0l7la1
 dao t4l7la1
 quanh t2l7si1
 pho t4l7si1
 vui t1l7si1
 a t4l7do1
 ha. t1l7re0
 t4l4re0
 Cho t4l7do1
 voi t4l7si1
 vang t4l7la1
 em t4l7si1
 hay t4l7do1
 coi t4l7si1
 chung t4l7la1
 keo t4l7si1
 khong t4l7do1
 kip t2l7si1
 thang. t4l7do1
 t1l7do1
 t1l4do1
 Giua t4l7si1

con t4l7la1
duong t4l7sol1
thanh t4l7la1
pho t4l7si1
dong t4l7la1
nguoi t4l7sol1
dung t4l7la1
nen t4l7si1
chay t2l7sol
nhanh. t4l7si1
t1l7si1
t4l4si1
De t4l7la1
anh t4l7sol1
duoc t4l7fat0
nhin t4l7sol1
ngam t4l7la1
em t4l7sol1
cuoi t4l7fat0
trong t4l7sol1
nang t0l7sol1
toc t5l7do1
em t5l7si1
duoi t5l7sol1
ga! t0l7mi1
t0l7mi0
Nay t4l7mi0
co t2l7fat0
be t4l7sol0
t2l7sol0
co t4l7sol0
mai t2l7sol0
toc t4l7sol0
duoi t2l7fat0
ga. t4l7mi0
t2l7mi0
t2l2mi0
Chieu t4l7si0
nay t4l7mi0
tren t2l7sol0
pho t2l7si1
t1l7si1
don t4l7si0
em t4l7la0
ve t2l7sol0
qua. t4l7la0

t1l7la0
t1l2la0
Ta t4l7sol0
ao t2l7sol0
trang t4l7la0
t2l7sol0
van t4l7sol0
trang t4l7la0
mai t4l7la0
trong t2l7la0
anh t4l7fat0
t2l7fat0
t2l2fat0
nu t4l7re0
cuoi t4l7fat0
ai t2l7sol0
do t4l7la0
t1l7la0
ma t4l7fat0
long t4l7fat0
van t2l7re0
vuong. t4l7si0
t1l7si1
t1l2si1
Nay t4l7mi0
co t2l7fat0
be t4l7sol0
t2l7sol0
be t4l7sol0
co t2l7sol0
biet t4l7sol0
chang t2l7fat0
la t4l7mi0
t2l7mi0
t2l2mi0
vong t4l7mi0
xe t4l7mi0
lan t2l7sol0
banh t4l7si1
t1l7si1
khien t4l7si1
anh t4l7la0
ngan t2l7sol0
ngo. t4l7la0
t1l7la0
t1l2la0

Nao t4l7sol0
ai t2l7sol0
biet t4l7la0
t2l7la0
co t4l7la0
nhung t4l7la0
noi t4l7la0
mong t2l7sol0
cho. t4l7fat0
t2l7fat0
t2l2fat0
Vi t4l7si0
anh t4l7ret0
yeu t2l7mi0
mai t4l7fat0
t1l7fat0
toc t4l7la0
em t4l7sol1
duoi t4l7fat0
ga. t4l7mi0
t0l7mi0
t0l7mi0
t0l7mi0
Nao t4l7fat0
ai t2l7sol0
biet t4l7la1
t2l7la1
co t4l7la1
nhung t4l7la1
noi t4l7la1
mong t2l7sol1
cho. t4l7fat0
t2l7fat0
t2l4fat0
Vi t4l7si0
anh t4l7re0
yeu t2l7mi0
mai t4l7fat0
t2l7fat0
t4l7fat0
t4l4fat0
toc t2l7la1
em t2l7sol1
duoi t2l7fat1
ga t0l7mi0
t0l7mi0

9.4. Tập không định kiểu

Không qui định kiểu dữ liệu của tệp, tức là mẫu tin trong tệp không có kiểu. Mỗi lần truy xuất tệp tin ta có thể đọc, ghi theo khối dữ liệu. Sau khi có một khối dữ liệu trong bộ đệm, có thể xử lý từng byte hoặc nhiều byte theo mục đích người sử dụng.

9.4.1. Khai báo

```
Var F: FILE;
```

9.4.2. Các hàm và thủ tục xử lý tệp tin

Các hàm và thủ tục truy xuất tệp tin tương tự như trong tệp có kiểu, chỉ khác thủ tục đọc, ghi tệp là đọc/ghi theo khối dữ liệu.

a. Thủ tục đọc dữ liệu

```
BlockRead(File,buffer,Sizeof(buffer),NumRead);
```

BlockRead đọc Sizeof(buffer) dữ liệu và lưu vào biến buffer, NumRead là số byte đọc được.

b. Thủ tục ghi dữ liệu

```
BlockWrite (File,Buffer,NumRead,NumWritten);
```

BlockWrite ghi NumRead byte trong buffer vào tệp File, NumWritten là số byte ghi được.

9.4.3. Ví dụ áp dụng

Chương trình sao chép dữ liệu, mỗi lần đọc hoặc ghi dữ liệu là 2048 byte (2KB).

```
Var Fin, Fout : File;  
    NumRead,NumWritten : Word;  
    Buf : Array[1..2048] of byte;  
    Total : Longint;  
begin  
    Assign (Fin, Paramstr(1));  
    Assign (Fout,Paramstr(2));  
    Reset (Fin,1);  
    Rewrite (Fout,1);  
    Total:=0;  
    Repeat  
        BlockRead (Fin,buf,Sizeof(buf),NumRead);  
        BlockWrite (Fout,Buf,NumRead,NumWritten);  
        inc(Total,NumWritten);  
    Until (NumRead=0) or (NumWritten<>NumRead);  
    Write ('Copied ',Total,' bytes from file ',paramstr(1));  
    Writeln (' to file ',paramstr(2));  
    close(fin);  
    close(fout);  
end.
```

Câu hỏi và bài tập

1. Viết chương trình nhập ngẫu nhiên 500 số nguyên vào tệp songuyen.dat, sau đó đọc và in lên màn hình các số nguyên tố có trong tệp.
2. Viết chương trình cho học sinh cấp I luyện tập phép cộng, trừ. Mỗi lần máy đưa ra phép cộng hai số nguyên được lấy ngẫu nhiên trong khoảng từ 0 đến 100. Yêu cầu học sinh nạp vào máy tổng của hai số đó. Nếu đáp số đúng thì sẽ có lời khen trên màn hình là “Bạn làm đúng rồi, hoan hô!” và máy phát ra âm thanh với tần số lấy ngẫu nhiên trong khoảng từ 0 đến 800 và trường độ lấy ngẫu nhiên trong khoảng từ 0 đến 110.
3. Chuyển các bài toán quản lý sử dụng mảng thành lưu trữ trên tệp.

Tài liệu tham khảo

1. TS. Quách Tuấn Ngọc, *Ngôn ngữ lập trình Pascal*, NXB Thống kê, 2001.
2. PGS.TS Lê Khắc Thành, *Bài giảng Lập trình Pascal*, ĐHSP Hà Nội, 2005.
3. PGS. TS Bùi Thế Tâm, *Turbo Pascal 7.0*, NXB Giao thông vận tải, 2006.
4. PGS. TS Bùi Thế Tâm, *Turbo Pascal. Lý thuyết cơ bản, bài tập, những chương trình mẫu trong khoa học lý thuật và kinh tế*. NXB Giao thông vận tải, 1995.