

100 bài tập Turbo Pascal

ỦY BAN NHÂN DÂN HUYỆN QUẾ SƠN
PHÒNG GIÁO DỤC VÀ ĐÀO TẠO

*** ❁ ❁ ❁ **

100 BÀI TẬP TURBO PASCAL LỚP 8

(Tài liệu bồi dưỡng học sinh giỏi môn tin học
dành cho học sinh THCS)

Quế Sơn, tháng 11 năm 2010

ê Nho Duyệt – Phòng Giáo dục & Đào tạo huyện Quế Sơn



GIỚI THIỆU TÓM TẮT TẬP TÀI LIỆU

1. Sự cần thiết:

Các trường THCS đang có xu hướng dạy môn tự chọn tin học thay cho chủ đề tự chọn khi mà cơ sở vật chất phục vụ cho việc dạy học môn học này được trang bị ngày càng tốt hơn. Rất nhiều học sinh, phụ huynh học sinh, lãnh đạo các trường muốn bản thân, con mình, học sinh mình tham gia các kỳ thi học sinh giỏi môn tin học (Kỳ thi tin học trẻ, Kỳ thi học sinh giỏi lớp 9...) bởi tính mới mẻ, hấp dẫn, thiết thực của bộ môn. Lần đầu tiên bộ môn tin học được đưa vào dạy học tại các trường THCS nên tài liệu bồi dưỡng cho học sinh giỏi **dành cho học sinh THCS** hầu như chưa có. Đáp ứng nhu cầu bồi dưỡng tại các trường, tập tài liệu bồi dưỡng học sinh giỏi môn tin học cấp THCS với tên gọi **100 bài tập Turbo Pascal** được bộ phận chuyên môn Phòng GD&ĐT Quế Sơn biên soạn.

2. Nội dung:

Tập tài liệu được biên soạn theo định hướng 10 x 10. Nội dung bồi dưỡng được chia làm 10 chương. Mỗi chương gồm tóm tắt lý thuyết và 10 bài tập xoay quanh nội dung của chương. Mỗi bài tập được trình bày theo cấu trúc:

a. Đề bài.

b. Hướng dẫn, thuật toán.

c. Mã chương trình.

d. Nhận xét: Nhấn mạnh nội dung mới, quan trọng cần nắm sau khi thực hiện bài tập, giải quyết bài toán theo thuật toán khác, điểm được và chưa được của thuật toán ...

Nội dung các bài tập ở các chương đã được lựa chọn theo hướng kế thừa, tăng dần độ khó. Nhiều bài toán cần giải quyết trong thực tế được đưa vào các bài tập nhằm tăng hứng thú học tập

Hầu hết các bài tập có độ khó vừa phải, phù hợp với nội dung bồi dưỡng ở cấp trường. Nội dung liên quan với môn toán và các môn khác từ lớp 8 trở xuống. Nội dung bồi dưỡng được chia thành 10 chương như sau:

Lớp 8:

I. Làm quen với chương trình Pascal – Khai báo, sử dụng biến – Các thủ tục vào ra.

II. Cấu trúc lựa chọn: if ... then ... else

Case ... of ...

III. Cấu trúc lặp với số lần lặp đã biết: For ... to ... do

IV. Cấu trúc lặp với số lần lặp chưa biết.

V. Dữ liệu kiểu mảng (một chiều).

VI. Chương trình con.

VII. Chuyên đề: Tính chia hết- Số nguyên tố.

VIII. Chuyên đề dãy con.

IX. Chuyên đề chữ số - hệ cơ số.

X. Chuyên đề đa thức.

3. Đề nghị:

Chắc chắn tập tài liệu cần hiệu chỉnh, bổ sung để có thể đưa vào sử dụng. Rất mong Hội đồng thẩm định cho ý kiến cụ thể về:

- Những hiệu chỉnh về cấu trúc của tập tài liệu.
- Những nội dung cần bổ sung thêm, những nội dung cần giảm bớt cho phù hợp với thực tế bồi dưỡng tại các đơn vị trường.
- Đề xuất các bài tập hay hơn để thay thế cho các bài tập đã có ở tập tài liệu. Các bài tập mà tập tài liệu còn thiếu.

CHƯƠNG I
CÁC KIỂU DỮ LIỆU CƠ BẢN
KHAI BÁO HẰNG, BIẾN, KIỂU, BIỂU THỨC VÀ CÂU LỆNH

A. LÝ THUYẾT:**I. CÁC KIỂU DỮ LIỆU CƠ BẢN****1. Kiểu logic**

- Từ khóa: **BOOLEAN**
- miền giá trị: (**TRUE, FALSE**).
- Các phép toán: phép so sánh (=, <, >) và các phép toán logic: AND, OR, XOR, NOT.

Trong Pascal, khi so sánh các giá trị boolean ta tuân theo qui tắc: FALSE < TRUE.

Giả sử A và B là hai giá trị kiểu Boolean. Kết quả của các phép toán được thể hiện qua bảng dưới đây:

A	B	A AND B	A OR B	A XOR B	NOT A
TRUE	TRUE	TRUE	TRUE	FALSE	FALSE
TRUE	FALSE	FALSE	TRUE	TRUE	FALSE
FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE	FALSE	TRUE

2. Kiểu số nguyên**2.1. Các kiểu số nguyên**

Tên kiểu	Phạm vi	Dung lượng
Shortint	-128 → 127	1 byte
Byte	0 → 255	1 byte
Integer	-32768 → 32767	2 byte
Word	0 → 65535	2 byte
LongInt	-2147483648 → 2147483647	4 byte

2.2. Các phép toán trên kiểu số nguyên**2.2.1. Các phép toán số học:**

+, -, *, / (phép chia cho ra kết quả là số thực).

Phép chia lấy phần nguyên: **DIV** (Ví dụ : 34 DIV 5 = 6).

Phép chia lấy số dư: **MOD** (Ví dụ: 34 MOD 5 = 4).

3. Kiểu số thực

3.1. Các kiểu số thực:

Tên kiểu	Phạm vi	Dung lượng
Single	$1.5 \times 10^{-45} \rightarrow 3.4 \times 10^{+38}$	4 byte
Real	$2.9 \times 10^{-39} \rightarrow 1.7 \times 10^{+38}$	6 byte
Double	$5.0 \times 10^{-324} \rightarrow 1.7 \times 10^{+308}$	8 byte
Extended	$3.4 \times 10^{-4932} \rightarrow 1.1 \times 10^{+4932}$	10 byte

Chú ý: Các kiểu số thực Single, Double và Extended yêu cầu phải sử dụng chung với bộ đồng xử lý số hoặc phải biên dịch chương trình với chỉ thị $\{ \$N+\}$ để liên kết bộ giả lập số.

3.2. Các phép toán trên kiểu số thực: +, -, *, /

Chú ý: Trên kiểu số thực không tồn tại các phép toán DIV và MOD.

3.3. Các hàm số học sử dụng cho kiểu số nguyên và số thực:

SQR(x):	Trả về x^2
SQRT(x):	Trả về căn bậc hai của x ($x \geq 0$)
ABS(x):	Trả về $ x $
SIN(x):	Trả về $\sin(x)$ theo radian
COS(x):	Trả về $\cos(x)$ theo radian
ARCTAN(x):	Trả về arctang(x) theo radian
TRUNC(x):	Trả về số nguyên gần với x nhất nhưng bé hơn x.
INT(x):	Trả về phần nguyên của x
FRAC(x):	Trả về phần thập phân của x
ROUND(x):	Làm tròn số nguyên x
PRED(n):	Trả về giá trị đứng trước n
SUCC(n):	Trả về giá trị đứng sau n
ODD(n):	Cho giá trị TRUE nếu n là số lẻ.
INC(n):	Tăng n thêm 1 đơn vị ($n:=n+1$).
DEC(n):	Giảm n đi 1 đơn vị ($n:=n-1$).

4. Kiểu ký tự

- Từ khoá: **CHAR**.
- Kích thước: 1 byte.
- Để biểu diễn một ký tự, ta có thể sử dụng một trong số các cách sau đây:
 - Đặt ký tự trong cặp dấu nháy đơn. Ví dụ 'A', '0'.
 - Dùng hàm CHR(n) (trong đó n là mã ASCII của ký tự cần biểu diễn). Ví dụ CHR(65) biểu diễn ký tự 'A'.
 - Dùng ký hiệu #n (trong đó n là mã ASCII của ký tự cần biểu diễn). Ví dụ #65.
- Các phép toán: =, >, >=, <, <=, <>.

*** Các hàm trên kiểu ký tự:**

- **UPCASE(ch):** Trả về ký tự in hoa tương ứng với ký tự ch. Ví dụ: UPCASE('a') = 'A'.
- **ORD(ch):** Trả về số thứ tự trong bảng mã ASCII của ký tự ch. Ví dụ ORD('A')=65.
- **CHR(n):** Trả về ký tự tương ứng trong bảng mã ASCII có số thứ tự là n. Ví dụ: CHR(65)='A'.
- **PRED(ch):** cho ký tự đứng trước ký tự ch. Ví dụ: PRED('B')='A'.
- **SUCC(ch):** cho ký tự đứng sau ký tự ch. Ví dụ: SUCC('A')='B'.

II. KHAI BÁO HẰNG

- Hằng là một đại lượng có giá trị không thay đổi trong suốt chương trình.
- Cú pháp:

CONST <Tên hằng> = <Giá trị>;

III. KHAI BÁO BIẾN

- Biến là một đại lượng mà giá trị của nó có thể thay đổi trong quá trình thực hiện chương trình.
- Cú pháp:

VAR <Tên biến>[, <Tên biến 2>, ...] : <Kiểu dữ liệu>;

Ví dụ:

VAR x, y: Real; {Khai báo hai biến x, y có kiểu là Real}
a, b: Integer; {Khai báo hai biến a, b có kiểu integer}

Chú ý: Ta có thể vừa khai báo biến, vừa gán giá trị khởi đầu cho biến bằng cách sử dụng cú pháp như sau:

CONST <Tên biến>: <Kiểu> = <Giá trị>;

Ví dụ:

CONST x:integer = 5;

Với khai báo biến x như trên, trong chương trình giá trị của biến x có thể thay đổi. (Điều này không đúng nếu chúng ta khai báo x là hằng).

IV. BIỂU THỨC

Biểu thức (expression) là công thức tính toán mà trong đó bao gồm các phép toán, các hằng, các biến, các hàm và các dấu ngoặc đơn.

Ví dụ: $(x + y) / (5 - 2 * x)$ biểu thức số học
 $(x + 4) * 2 = (8 + y)$ biểu thức logic

Trong một biểu thức, thứ tự ưu tiên của các phép toán được liệt kê theo thứ tự sau:

- LỜI GỌI HÀM.
- DẤU NGOẶC ()
- PHÉP TOÁN MỘT NGÔI (NOT, -).
- PHÉP TOÁN *, /, DIV, MOD, AND.
- PHÉP TOÁN +, -, OR, XOR

- Phép toán so sánh =, <, >, <=, >=, <>, IN

V. CÂU LỆNH

6.1. Câu lệnh đơn giản

- **Câu lệnh gán** (:=): <Tên biến>:=<Biểu thức>;
- Các lệnh xuất nhập dữ liệu: **READ/READLN, WRITE/Writeln.**
- Lời gọi hàm, thủ tục.

6.2. Câu lệnh có cấu trúc

- Câu lệnh ghép: **BEGIN ... END;**
- Các cấu trúc điều khiển: **IF..., CASE..., FOR..., REPEAT..., WHILE...**

6.3. Các lệnh xuất nhập dữ liệu

6.3.1. Lệnh xuất dữ liệu

Để xuất dữ liệu ra màn hình, ta sử dụng ba dạng sau:

- (1) **WRITE**(<tham số 1> [, <tham số 2>,...]);
- (2) **Writeln**(<tham số 1> [, <tham số 2>,...]);
- (3) **Writeln**;

Các thủ tục trên có chức năng như sau:

- (1) Sau khi xuất giá trị của các tham số ra màn hình thì con trỏ không xuống dòng.
- (2) Sau khi xuất giá trị của các tham số ra màn hình thì con trỏ xuống đầu dòng tiếp theo.
- (3) Xuống dòng.

Các tham số có thể là các hằng, biến, biểu thức. Nếu có nhiều tham số trong câu lệnh thì các tham số phải được phân cách nhau bởi dấu phẩy.

Khi sử dụng lệnh **WRITE/Writeln**, ta có hai cách viết: **không qui cách** và **có qui cách**:

- **Viết không qui cách**: dữ liệu xuất ra sẽ được canh lề ở phía bên trái. Nếu dữ liệu là số thực thì sẽ được in ra dưới dạng biểu diễn khoa học.

Ví dụ:

```
Writeln(x); WRITE(sin(3*x));
```

- **Viết có qui cách**: dữ liệu xuất ra sẽ được canh lề ở phía bên phải.

Ví dụ:

```
Writeln(x:5); WRITE(sin(13*x):5:2);
```

Câu lệnh	Kết quả trên màn hình
Writeln('Hello');	Hello
Writeln('Hello':10);	Hello
Writeln(500);	500
Writeln(500:5);	500
Writeln(123.457)	1.2345700000E+02
Writeln(123.45:8:2)	123.46

6.3.2. Nhập dữ liệu

Để nhập dữ liệu từ bàn phím vào các biến có kiểu dữ liệu chuẩn (trừ các biến kiểu BOOLEAN), ta sử dụng cú pháp sau đây:

READLN(<biến 1> [,<biến 2>,...,<biến n>]);

Chú ý: Khi gặp câu lệnh READLN; (không có tham số), chương trình sẽ dừng lại chờ người sử dụng nhấn phím ENTER mới chạy tiếp.

6.4. Các hàm và thủ tục thường dùng trong nhập xuất dữ liệu

- Hàm **KEYPRESSED**: Hàm trả về giá trị TRUE nếu như có một phím bất kỳ được nhấn, nếu không hàm cho giá trị là FALSE.
- Hàm **READKEY**: Hàm có chức năng đọc một ký tự từ bộ đệm bàn phím.
- Thủ tục **GOTOXY(X,Y:Integer)**: Di chuyển con trỏ đến cột X dòng Y.
- Thủ tục **CLRSCR**: Xoá màn hình và đưa con trỏ về góc trên bên trái màn hình.
- Thủ tục **CLREOL**: Xoá các ký tự từ vị trí con trỏ đến hết dòng.
- Thủ tục **DELLINE**: Xoá dòng tại vị trí con trỏ và dồn các dòng ở phía dưới lên.
- Thủ tục **TEXTCOLOR(color:Byte)**: Thiết lập màu cho các ký tự. Trong đó color $\in [0,15]$.
- Thủ tục **TEXTBACKGROUND(color:Byte)**: Thiết lập màu nền cho màn hình.

B. BÀI TẬP:

Bài tập 1.1:

Viết chương trình tính chu vi và diện tích của hình chữ nhật có chiều dài hai cạnh là a, b (được nhập từ bàn phím).

a. Hướng dẫn:

- Nhập hai cạnh vào hai biến a, b.
- Chu vi hình chữ nhật bằng $2*(a+b)$; Diện tích hình chữ nhật bằng $a*b$.

b. Mã chương trình:

```
Program Chu_nhat;
uses crt;
Var a, b, S, CV: real;
Begin
    Write('Nhap chieu dai:'); readln(a);
    Write('Nhap chieu rong:'); readln(b);
    S := a*b;
    CV := (a+b)*2;
    Writeln('Dien tich hinh chu nhat la:',S);
    Writeln('Chu vi hinh chu nhat la:',CV:10:2);
    readln
end.
```


c. Nhận xét: Lệnh write cho phép in ra màn hình một hoặc nhiều mục. Có thể định dạng được số in ra bằng cách qui định khoảng dành cho phần nguyên, khoảng dành cho phần thập phân.

Bài tập 1.2:

Viết chương trình tính chu vi, diện tích hình vuông có cạnh a (được nhập từ bàn phím).

a. Hướng dẫn:

- Nhập cạnh vào biến canh.
- Chu vi hình vuông bằng $4 * \text{canh}$; Diện tích hình vuông bằng $\text{canh} * \text{canh}$.

b. Mã chương trình:

```
Program HINH_VUONG;
uses crt;
Var canh: real;
Begin
    clrscr;
    Write('Nhap do dai canh:');readln(canh);
    Writeln('Chu vi hinh vuong la:',4*canh:10:2);
    Writeln('Dien tich hinh vuong
la:', canh*canh:10:2);
    readln
end.
```

c. Nhận xét: Bài tập 1.2 tiết kiệm được hai biến là CV và S vì lệnh write cho phép in một biểu thức. Trong lập trình việc tiết kiệm biến là cần thiết nhưng đôi lúc gây khó hiểu khi đọc, kiểm tra chương trình.

Bài tập 1.3:

Viết chương trình tính chu vi và diện tích hình tròn có bán kính r (được nhập từ bàn phím).

a. Hướng dẫn:

- Nhập bán kính vào biến r.
- Chu vi đường tròn bằng $2 * \pi * r$.
- Diện tích hình tròn bằng $\pi * r * r$.

b. Mã chương trình:

```
Program HINH_TRON;
uses crt;
Var r: real;
Begin
    clrscr;
    Write('Nhap ban kinh:'); readln(r);
    Writeln('Chu vi duong tron la:',2*pi*r:10:2);
    Writeln('Dien tich hinh tron la:',pi*r*r:10:2);
    readln
end.
```

c. Nhận xét: pi là hằng số. Một hằng số có thể được người dùng khai báo hoặc do Pascal tự tạo. Pi là hằng do Pascal tự tạo nên người dùng không cần khai báo.

Bài tập 1.4:

Viết chương trình tính diện tích của tam giác có ba cạnh là a,b,c (được nhập từ bàn phím)

a. Hướng dẫn:

- Nhập ba cạnh của tam giác vào ba biến a,b,c.
- Nửa chu vi của tam giác $p = (a+b+c)/2$.
- Diện tích của tam giác: $s = \sqrt{p(p-a)(p-b)(p-c)}$.

b. Mã chương trình:

```
Program TAM_GIAC;
uses crt;
Var a,b,c,p,S: real;
Begin
  clrscr;
  Write('Nhap canh a:');readln(a);
  Write('Nhap canh b:');readln(b);
  Write('Nhap canh c:');readln(c);
  p:=(a+b+c)/2;
  S:= sqrt(p*(p-a)*(p-b)*(p-c));
  Write('Dien tich tam giac la:',s:10:2);
  readln
end.
```

b. Nhận xét: Ở đây ta lại hai lần dùng biến trung gian p, s để chương trình sáng sủa, dễ theo dõi. sqrt là hàm có sẵn của turbo pascal. Nó cho phép tính căn bậc hai của một số không âm.

Bài tập 1.5:

Viết chương trình cho phép tính trung bình cộng của bốn số.

a. Hướng dẫn:

- Nhập bốn số vào bốn biến a, b, c, d
- Trung bình cộng của a, b, c, d bằng $(a + b + c + d)/4$.

b. Mã chương trình:

```
Program TB_Cong_4_So;
uses crt;
Var a, b, c, d: real;
Begin
  Clrscr;
  Write('Nhap so thu nhat:');readln(a);
  Write('Nhap so thu hai:');readln(b);
  Write('Nhap so thu ba:');readln(c);
  Write('Nhap so thu tu:');readln(d);
```

```
Writeln('Trung binh cong: ', (a+b+c+d)/4):10:2);  
Readln  
end.
```

Bài tập 1.6:

Viết chương trình cho phép tính trung bình cộng của bốn số với điều kiện chỉ được sử dụng hai biến.

a. Hướng dẫn:

- Dùng một biến S có giá trị ban đầu bằng 0.
- Dùng một biến để nhập số.
- Sau khi nhập một số cộng ngay vào biến S.

b. Mã chương trình:

```
Program TB_Cong_4_So;  
uses crt;  
Var s,a: real;  
Begin  
    Clrscr;  
    S:=0;  
    Write('Nhap so thu nhat:');readln(a); S:=S+a;  
    Write('Nhap so thu hai:');readln(a); S:= S+a;  
    Write('Nhap so thu ba:');readln(a); S:=S+a;  
    Write('Nhap so thu tu:');readln(a); S:=S+a;  
    Writeln('Trung binh cong: ',S/4:10:2);  
    readln  
end.
```

b. Nhận xét: Câu lệnh gán $S:=S+a$ thực hiện việc cộng thêm a vào biến S . Thực chất là thực hiện các bước: lấy giá trị của S cộng với a rồi ghi đè vào lại biến S . Ở đây ta cũng đã sử dụng biến a như là một biến tạm để chứa tạm thời giá trị được nhập từ bàn phím.

Bài tập 1.7:

Viết chương trình cho phép tính trung bình nhân của bốn số với điều kiện chỉ được sử dụng hai biến.

a. Hướng dẫn:

- Dùng một biến S có giá trị ban đầu bằng 1.
- Dùng một biến để nhập số.
- Sau khi nhập một số nhân ngay vào biến S.
- Trung bình nhân bốn số là căn bậc 4 tích của chúng (Dùng hai lần căn bậc hai).

b. Mã chương trình:

```
Program TB_nhan;  
uses crt;  
Var a, S: real;  
Begin  
    clrscr;
```

```

S:=1;
Write('Nhap so thu nhât: '); readln(a); S:=S*a;
Write('Nhap so thu hai: '); readln(a); S:=S*a;
Write('Nhap so thu ba: '); readln(a); S:=S*a;
Write('Nhap so thu tu: '); readln(a); S:=S*a;
Write('Trung binh nhan cua bon so
la:',sqrt(sqrt(s)));
  readln
End.

```

b. Nhận xét: Ta đã dùng hai lần khai phương để lấy căn bậc 4 của một số. Để cộng dồn giá trị vào một biến thì biến đó có giá trị ban đầu là 0. Để nhân dồn giá trị ban đầu vào biến thì biến đó cần có giá trị ban đầu là 1.

Bài tập 1.8:

Viết chương trình nhập hai số, đổi giá trị hai số rồi in ra hai số.

a. Hướng dẫn:

- Dùng các biến a, b để lưu hai số được nhập từ bàn phím;
- Gán cho biến tạm giá trị của a.
- Gán giá trị của b cho a. (Sau lệnh này a có giá trị của b).
- Gán giá trị của tạm cho b (Sau lệnh này b có giá trị của tạm = a).

b. Mã chương trình:

```

Program Doi_Gia_Tri;
uses crt;
var a, b, tam:real;
Begin
  clrscr;
  write('nhap a: '); readln(a);
  write('nhap b: '); readln(b);
  writeln('Truoc khi doi a =',a,' va b= ',b);
  readln;
  tam:=a;
  a:=b;
  b:=tam;
  writeln('Sau khi doi a =',a,' va b= ',b);
  readln
end.

```

Nhận xét: Nếu thực hiện hai lệnh $a:=b$; $b:=a$ để đổi giá trị hai biến thì sau hai lệnh này hai biến có giá trị bằng nhau và bằng b. Thực chất sau lệnh thứ nhất hai biến đã có giá trị bằng nhau và bằng b rồi! Trong thực tế để đổi chỗ số đầu ở hai bình cho nhau ta phải dùng thêm một bình phụ.

Bài tập 1.9

Giải bài tập 1.8 mà chỉ được sử dụng hai biến (Tức không được dùng thêm biến tạm).

a. Hướng dẫn:

- Cộng thêm b vào a. (Giá trị hai biến sau lệnh này là: $a+b$, b)

- Gán b bằng tổng trừ đi b (Sau lệnh này b có giá trị bằng a);
- Gán giá trị a bằng tổng trừ đi b mới (Sau lệnh này a có giá trị bằng b).

b. Mã chương trình:

```
Program Doi_Gia_Tri;
uses crt;
var a, b:real;
Begin
  clrscr;
  write('nhap a: '); readln(a);
  write('nhap b: '); readln(b);
  writeln('Truoc khi doi a =',a,' va b= ',b);
  readln;
  a:=a+b;
  b:=a-b;
  a:=a-b;
  writeln('Sau khi doi a =',a,' va b= ',b);
  readln
end.
```

Nhận xét:Giống sang đầu giữa hai bình nhưng không giống hoàn toàn!!!Kỹ thuật đổi giá trị biến cho nhau sẽ được sử dụng nhiều trong phần sắp xếp.

Bài tập 1.10:

Viết chương trình cho biết chữ số hàng trăm, hàng chục, hàng đơn vị của một số có ba chữ số. Ví dụ khi nhập số 357 thì máy in ra:

- Chữ số hàng trăm: 3.
- Chữ số hàng chục: 5.
- Chữ số hàng đơn vị: 7.

a. Hướng dẫn:

Sử dụng hàm mod để lấy số dư. Khi chia cho 10 để lấy số dư ta được chữ số hàng đơn vị. Sử dụng DIV để lấy phần nguyên. Khi chia cho 10 để lấy phần nguyên ta đã bỏ đi chữ số hàng đơn vị để số có ba chữ số còn số có hai chữ số.

b. Mã chương trình:

```
Program CHU_SO;
uses crt;
var n:integer;
begin
  clrscr;
  write('Nhap so n: ');readln(n);
  writeln('Chu so hang don vi: ',n mod 10);
  n:=n div 10;
  writeln('Chu so hang chục: ',n mod 10);
  n:=n div 10;
  writeln('Chu so hang tram: ',n mod 10);
  readln
end.
```

c. Nhận xét:

Hãy sửa chương trình để có kết quả là hàng trăm, hàng chục, hàng đơn vị.

Mã chương trình:

```
Program CHU_SO;
uses crt;
var n:integer;
begin
    clrscr;
    write('Nhap so n: ');readln(n);
    writeln('Chu so hang trm: ',n div 100);
    n:=n mod 100;
    writeln('Chu so hang chuc: ',n div 10);
    n:=n div 10;
    writeln('Chu so hang tram: ',n);
    readln
end.
```


☞ Const i: Hằng thứ i, có thể là một giá trị hằng, các giá trị hằng (phân cách nhau bởi dấu phẩy) hoặc các đoạn hằng (dùng hai dấu chấm để phân cách giữa giá trị đầu và giá trị cuối).

☞ Giá trị của biểu thức và giá trị của tập hằng i (i=1..n) phải có cùng kiểu.

Khi gặp lệnh CASE, chương trình sẽ kiểm tra:

- Nếu giá trị của biểu thức B nằm trong tập hằng const i thì máy sẽ thực hiện lệnh S_i tương ứng.

- Ngược lại:

+ Đối với dạng 1: Không làm gì cả.

+ Đối với dạng 2: thực hiện lệnh S_{n+1}.

B. BÀI TẬP:

Bài tập 2.1:

Viết chương trình in ra số lớn hơn trong hai số (được nhập từ bàn phím).

a. Hướng dẫn:

- Nhập hai số vào hai biến a, b.
- Nếu a > b thì in a. Nếu a ≤ b thì in b.
- Hoặc: Nếu a > b thì in a. Ngược lại thì in b.

b. Mã chương trình:

```
Program SO_SANH1;
uses crt;
var a,b: real;
begin
  clrscr;
  write('nhap so thu nhat: '); readln(a);
  write('nhap so thu hai: '); readln(b);
  if a > b then writeln(' So lon la:',a);
  if a <= b then writeln(' So lon la:',b:10:2);
  readln
end.
```

Hoặc:

```
Program SO_SANH2;
uses crt;
var a,b: real;
begin
  clrscr;
  write('nhap so thu nhat: '); readln(a);
  write('nhap so thu hai: '); readln(b);
  if a > b then writeln(' So lon la:',a:10:2)
  else writeln(' So lon la:',b:10:2);
  readln
end.
```


c. Nhận xét: Khi hai số bằng nhau thì mỗi số được xem là số lớn. Hãy sửa chương trình để khắc phục yếu điểm này.

Nói chung nên sử dụng lệnh `if ... then ... else` vì chương trình sáng sủa, dễ hiểu hơn. Tuy nhiên trong vài trường hợp sử dụng các lệnh `if ... then` rồi lại dễ diễn đạt hơn. Hãy xem ví dụ sau:

Bài tập 2.2:

Viết chương trình in ra số lớn nhất trong bốn số nhập từ bàn phím.

a. Hướng dẫn:

Nếu $a \geq b$ và $a \geq c$ và $a \geq d$ thì a là số lớn nhất.

Tương tự như thế xét các trường hợp còn lại để tìm số lớn nhất.

b. Mã chương trình:

```
Program So_Lon_Nhat_1;
Uses crt;
Var a,b,c,d: real;
Begin
    Clrscr;
    Write('Nhap so thu nhat:');readln(a);
    Write('Nhap so thu hai:');readln(b);
    Write('Nhap so thu ba:');readln(c);
    Write('Nhap so thu tu:');readln(d);
    if (a>=b) and (a>=c) and (a>= d) then
writel('So lon nhat la:',a:10:2);
    if (b>=a) and (b>=c) and (b>= d) then
writel('So lon nhat la:',b:10:2);
    if (c>=a) and (c>=b) and (c>= d) then
writel('So lon nhat la:',c:10:2);
    if (d>=a) and (d>=b) and (d>= c) then
writel('So lon nhat la:',d:10:2);
    readln
end.
```

c. Nhận xét: Hãy sử dụng cấu trúc `if ... then ... else` để giải bài tập trên.

Độ khó của bài toán sẽ tăng lên nhiều nếu thêm yêu cầu có thông báo khi hai số, ba số, bốn số bằng nhau.

Bài tập 2.3:

Viết chương trình in ra số lớn nhất trong bốn số nhập từ bàn phím với điều kiện chỉ được dùng hai biến.

a. Hướng dẫn:

Sử dụng một biến `max` và một biến `a` để chứa số vừa nhập. Cho `max` bằng số đầu tiên. Sau khi nhập một số thực hiện so sánh nếu số vừa nhập lớn hơn `max` thì lưu số vừa nhập vào `max`. Sau khi nhập xong ta có `max` là số lớn nhất

(Giải thuật này gọi là kỹ thuật lính canh cần hiểu rõ để sử dụng sau này).

b. Mã chương trình:

```
Program So_Lon_Nhat_2;
Uses crt;
Var a,max: real;
Begin
    Clrscr;
    Write('Nhap so thu nhat:');readln(a);Max:=a;
    Write('Nhap so thu hai:');readln(a);if a>=Max
then Max:=a;
    Write('Nhap so thu ba:');readln(a);if a>=Max
then Max:=a;
    Write('Nhap so thu tu:');readln(a);if a>=Max
then Max:=a;
    Write('So lon nhat la:',Max:10:2);
    readln
end.
```

Bài tập 2.4

Viết chương trình xét xem một tam giác có là tam giác đều hay không khi biết ba cạnh của tam giác.

a. Hướng dẫn:

- Nhập ba cạnh của tam giác vào ba biến a,b,c.
- Nếu $a = b$ và $b = c$ thì tam giác là tam giác đều và ngược lại tam giác không là tam giác đều.

b. Mã chương trình:

```
Program Tam_giac_deu;
uses crt;
var a,b,c: real;
begin
    clrscr;
    write('Nhap a = '); readln(a);
    write('Nhap b = '); readln(b);
    write('Nhap c = '); readln(c);
    if (a = b) and (b = c) then writeln('La tam
giac deu')
    else
        writeln('Khong phai la tam giac deu');
    readln
end.
```

Bài tập 2.5

Viết chương trình xét xem một tam giác có là tam giác cân hay không khi biết ba cạnh của tam giác.

a. Hướng dẫn:

- Nhập ba cạnh của tam giác vào ba biến a,b,c.

- Nếu $a = b$ hoặc $b = c$ hoặc $a = c$ thì tam giác là tam giác cân và ngược lại tam giác không là tam giác cân.

b. Mã chương trình:

```
Program Tam_giac_can;
uses crt;
var a,b,c: real;
begin
  clrscr;
  write('Nhap a = '); readln(a);
  write('Nhap b = '); readln(b);
  write('Nhap c = '); readln(c);
  if (a = b) or (b = c) or (a = c) then
writel('La tam giac can')
  else
  writel('Khong phai la tam giac can');
  readln
end.
```

Bài tập 2.6

Viết chương trình xét xem một tam giác có là tam giác vuông hay không khi biết ba cạnh của tam giác.

a. Hướng dẫn:

- Nhập ba cạnh của tam giác vào ba biến a,b,c.
- Nếu $a^2 = b^2 + c^2$ hoặc $b^2 = c^2 + a^2$ hoặc $c^2 = a^2 + b^2$ thì tam giác là tam giác vuông và ngược lại tam giác không là tam giác vuông.

b. Mã chương trình:

```
Program Tam_giac_can;
uses crt;
var a,b,c: real;
begin
  clrscr;
  write('Nhap a = '); readln(a);
  write('Nhap b = '); readln(b);
  write('Nhap c = '); readln(c);
  if (a*a = b*b+c*c) or (b*b = c*c+a*a) or (c*c=
a*a+b*b) then writel('La tam giac vuong')
  else
  writel('Khong phai la tam giac vuong');
  readln
end.
```

Bài tập 2.7:

Viết chương trình giải phương trình $ax + b = 0$ (Các hệ số a, b được nhập từ bàn phím).

a. Hướng dẫn:

- Nếu $a \neq 0$ thì phương trình có nghiệm $x = \frac{-b}{a}$
- Nếu $a = 0$ và $b = 0$ thì phương trình có vô số nghiệm
- Nếu $a = 0$ và $b \neq 0$ thì phương trình vô nghiệm

Hoặc:

- Nếu $a = 0$ thì xét b . Nếu $b = 0$ thì phương trình có vô số nghiệm ngược lại ($b \neq 0$) thì phương trình vô nghiệm ngược lại ($a \neq 0$) phương trình có nghiệm $x = \frac{-b}{a}$.

a. Mã chương trình:

```

Program Phuong_trinh_2;
uses crt;
var a,b:real;
begin
  clrscr;
  Writeln('          CHUONG TRINH GIAI PT ax + b =
0');
  Write('Nhap he so a = ');readln(a);
  Write('Nhap he so b = ');readln(b);
  if (a<>0) then writeln('phuong trinh',a:10:2,'x
+ ',b:10:2,'= 0',' co nghiem x =;',-b/a:10:2);
  if (a=0) and (b=0) then writeln('Phuong trinh co
vo so nghiem');
  if (a=0) and (b<>0) then writeln('Phuong trinh
vo nghiem');
  readln
end.

```

Hoặc:

```

Program Phuong_trinh_2;
uses crt;
var a,b:real;
begin
  clrscr;
  Writeln('          CHUONG TRINH GIAI PT ax + b =
0');
  Write('Nhap he so a = ');readln(a);
  Write('Nhap he so b = ');readln(b);
  if (a<>0) then writeln('phuong trinh',a:10:2,'x
+ ',b:10:2,'= 0',' co nghiem x =;',-b/a:10:2)

```

```
else
  if (b=0) then writeln('Phuong trinh co vo so
nghiem')
  else
    writeln('Phuong trinh vo nghiem');
readln
end.
```

Bài tập 2.8: (HSG lớp 8 -TP Huế 2006-2007)

Ba bạn An, Bình và Cường cùng tham gia một trò chơi như sau: Mỗi bạn nắm trong tay một đồng xu, mỗi đồng xu có hai trạng thái : sấp và ngửa. Theo hiệu lệnh, cả ba bạn cùng đưa đồng xu của mình ra phía trước. Nếu cả ba đồng xu cùng sấp hoặc cùng ngửa thì chưa phát hiện người thua cuộc (hòa nhau). Nếu một bạn có trạng thái đồng xu khác với hai bạn kia(nghĩa là đồng xu của bạn ấy sấp còn hai người kia ngửa và ngược lại đồng xu của bạn ấy ngửa thì hai người kia sấp) thì bạn đó thắng cuộc. Hãy viết chương trình mô phỏng trò chơi trên.

Thuật toán:

- Để mô phỏng trạng thái sấp, ngửa của đồng xu ta dùng hàm Random(1) hoặc dùng $\text{Random}(n) \bmod 2$ với $n > 2$.
- Xét tám trường hợp xảy ra để tìm người thắng cuộc.

Cài đặt:

```
Program Sap_ngua;
uses crt;
Var A, B, C: byte;
Begin
  clrscr;
  Writeln('Go phim de xem ket qua: ');
  A:=Random(10); A:=A mod 2;
  B:=Random(10); B:=B mod 2;
  C:=Random(10); C:=C mod 2;
  Write('Ket qua: ',a,b,c);
  if (A=0) and (B=0) and (C=0) then Write(' Hoa');
  if (A=0) and (B=0) and (C=1) then Write(' C Thang');
  if (A=0) and (B=1) and (C=0) then Write(' B Thang');
  if (A=0) and (B=1) and (C=1) then Write(' A Thang');
  if (A=1) and (B=0) and (C=0) then Write(' A Thang');
  if (A=1) and (B=0) and (C=1) then Write(' B Thang');
  if (A=1) and (B=1) and (C=0) then Write(' C Thang');
  if (A=1) and (B=1) and (C=1) then Write(' Hoa');
  Readln;
  Readln
End.
```

Bài tập 2.9:

Viết chương trình dịch các ngày trong tuần sang tiếng anh

2	3	4	5	6	7	8
Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday

a. Hướng dẫn:

- Dùng biến a kiểu byte để chứa thứ (2 đến 8)
- Trường hợp a = 2: Monday
- Trường hợp a = 3: Thursday
- ...
- Trường hợp a = 8: Sunday
- Ngoài ra không còn thứ nào.

b. Mã chương trình:

```
Program dich;
uses crt;
Var thu:byte;
begin
  clrscr;
  write('nhap thu can dich 2→8: '); readln(thu);
  case thu of
    2: Write('--> Monday');
    3: Write('--> Tuesday');
    4: Write('--> Wednesday');
    5: Write('--> Thursday');
    6: Write('--> Friday');
    7: Write('--> Saturday');
    8: Write('--> Sunday');
  else
    Write(' Khong co thu nay');
  end;
  readln
end.
```

Bài tập 2.10

Viết chương trình cho phép tính diện tích các hình: Hình vuông; Hình chữ nhật; Hình tròn; Tam giác; Hình thang. Người dùng chọn hình cần tính diện tích từ bảng chọn, sau đó khai báo các thông số liên quan và nhận được diện tích của hình:

MOI BAN CHON HINH CAN TINH DIEN TICH

1. Hình vuông.
2. Hình chữ nhật.
3. Hình tròn.
4. Tam giác.
5. Hình thang.

Muốn tính diện tích tam giác, người dùng gõ 4 và khai báo đường cao, đáy. Chương trình tính và thông diện tích đến người dùng.

a. Hướng dẫn:

- Dùng cấu trúc chọn Case chọn of với chọn có kiểu Char để tạo bảng chọn.
- Dùng 3 biến a,b,c để lưu các thông số của hình; Biến S để lưu diện tích của hình.
- Thực hiện chung câu thông báo diện tích (Nằm ngoài Case . . . of) để gọn

chương trình.

b. Mã chương trình:

```
Program Dien_Tich_cac_hinh;
uses crt;
var chon: byte;
    a,b,c,S: real;
Begin
    clrscr;
    writeln('CHUONG TRINH TINH DIEN TICH CAC HINH');
    Writeln('          -----');
    writeln('1.   DIEN TICH HINH TAM GIAC');
    writeln('2.   DIEN TICH HINH VUONG');
    writeln('3.   DIEN TICH HINH CHU NHAT');
    writeln('4.   DIEN TICH HINH THANG');
    writeln('5.   DIEN TICH HINH TRON');
    write('Moi ban chon hinh can tinh dien tich:
');readln(chon);
    case chon of
    1 : Begin
        Write('Cho biet canh day: '); readln(a);
        Write('Cho biet chieu cao: '); readln(b);
        S:=(a*b)/2;
        end;
    2:Begin
        Write('Cho biet chieu dai canh: ');
readln(a);
        S:=a*a;
        end;
    3:Begin
        Write('Cho biet chieu dai: '); readln(a);
        Write('Cho biet chieu rong: '); readln(b);
        S:=a*b;
        end;
    4:Begin
        Write('Cho biet day lon: '); readln(a);
        Write('Cho biet day nho: '); readln(b);
        Write('Cho biet chieu cao: ');
readln(c);
        S:=(a+b)*c/2;
    End;
```

```
5:Begin
    Write('Cho biet ban kinh: '); readln(a);
    S:=a*a*pi;
    End;
else
    Writeln('Chon sai roi!!!');
end;
Writeln('Dien tich cua hinh la: ',S:8:2);
readln
end.
```

c. Nhận xét: Với mỗi trường hợp thỏa Case **biến chọn** of thực hiện **một lệnh**. Vì thế, muốn thực hiện nhiều lệnh ta cần ghép nhiều lệnh thành một lệnh ghép.

CHƯƠNG III CÂU LỆNH CÓ CẤU TRÚC LẶP XÁC ĐỊNH

A. LÝ THUYẾT:

II. CÂU LỆNH LẶP

2.1. Vòng lặp xác định

Có hai dạng sau:

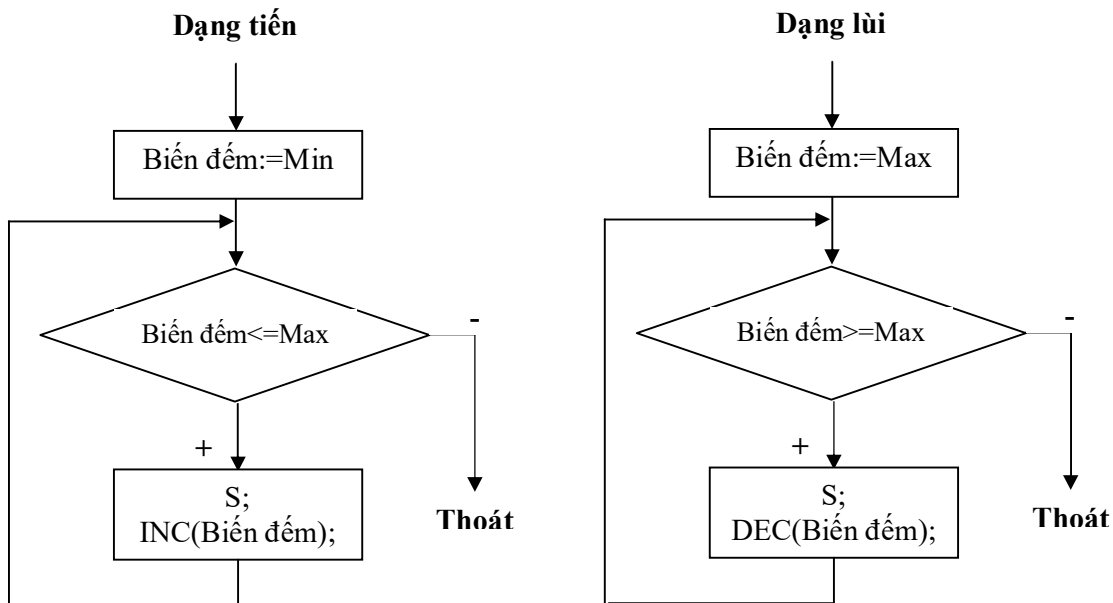
❶ **Dạng tiến**

FOR <biến đếm>:=<giá trị Min> TO <giá trị Max> DO S;

❷ **Dạng lùi**

FOR <biến đếm>:=<giá trị Max> DOWNTO <giá trị Min> DO S;

Sơ đồ thực hiện vòng lặp FOR:



Chú ý: Khi sử dụng câu lệnh lặp FOR cần chú ý các điểm sau:

- Không nên tùy tiện thay đổi giá trị của biến đếm bên trong vòng lặp FOR vì làm như vậy có thể sẽ không kiểm soát được biến đếm.
- Giá trị Max và Min trong câu lệnh FOR sẽ được xác định ngay khi vào đầu vòng lặp. Do đó cho dù trong vòng lặp ta có thay đổi giá trị của nó thì số lần lặp cũng không thay đổi.

B. BÀI TẬP

Bài tập 3.1:

Viết chương trình in ra các số lẻ nhỏ hơn hoặc bằng n (Với n được nhập).

a. Hướng dẫn:

- Cho biến i chạy từ 1 đến n.
- Nếu i chẵn (i chia 2 dư 0) thì in ra số n.

b. Mã chương trình:

```
Program In_So_Le;
Uses crt;
var i,n: integer;
Begin
    Clrscr;
    Write('Nhap so n ='); readln(n);
    For i:=1 to n do if i mod 2 =1 then
Write(i:3, ', ');
    readln
end.
```

Bài tập 3.2:

Viết chương trình in ra các số lẻ nhỏ hơn hoặc bằng n (Với n được nhập). Sao cho 15 số lẻ được in trên một dòng.

a. Hướng dẫn:

- Cho j=0.
- Cho biến i chạy từ 1 đến n.
- Nếu i chẵn (i chia 2 dư 0) thì in ra số n và tăng dem lên 1
- Nếu dem chia hết cho 15 thì thực hiện xuống dòng (Dùng Writeln).

b. Mã chương trình:

```
Program In_So_Le;
Uses crt;
var Dem,i,n: integer;
Begin
    Clrscr;
    Write('Nhap so n ='); readln(n);
    Dem:= 0;
    For i:=1 to n do if i mod 2 =1 then
Begin
        Write(i:3, ', ');
        Dem:= Dem + 1;
        if Dem mod 15 = 0 then Writeln;
    end;
    readln
end.
```

c. Nhận xét: Lệnh writeln không có tham số cho phép xuống hàng.

Nếu có nhận xét trong 30 số thì có 15 số lẻ, ta có thể không cần thêm biến đếm mà chỉ cần kiểm tra biến i để xuống hàng.

Bài tập 3.3:

Viết chương trình in ra tổng các số lẻ nhỏ hơn hoặc bằng n (Với n được nhập).

a. Hướng dẫn:

- Cho S = 0.
- Cho biến i chạy từ 1 đến n.
- Nếu i chẵn (i chia 2 dư 0) thì cộng thêm i vào S.
- In ra S.

b. Mã chương trình:

```
Program In_So_Le;  
Uses crt;  
var S,i,n: integer;  
Begin  
    Clrscr;  
    Write('Nhap so n ='); readln(n);  
    S:= 0;  
    For i:=1 to n do if i mod 2 =1 then S:= S+i;  
    Writeln('Tong cac so le nho hon ',n,' la: ',S);  
    readln  
end.
```

c. Nhận xét: Ta dùng biến S để cộng dồn nên nó được khởi tạo giá trị đầu bằng 0.

Bài tập 3.4:

Viết chương trình in ra tất cả các ước của một số n (Với n được nhập từ bàn phím)

a. Hướng dẫn:

- Cho biến i chạy từ 1 đến n. Nếu n chia hết cho i thì in ra i.

b. Mã chương trình:

```
Program Tim_uoc;  
uses crt;  
Var n, i: integer;  
Begin  
    clrscr;  
    Write('Nhap so n ='); readln(n);  
    For i:=1 to n do if n mod i = 0 then  
write(i:3, ', ');  
    readln  
end.
```

c. Nhận xét: Chương trình trên có hai điểm yếu:

- In dư một dấu phẩy ở cuối dãy các ước.
 - Khi n = 0 thì không in một số nào trong khi mọi số đều là ước của 0.
- Hãy tìm cách khắc phục yếu điểm này.

Bài tập 3.5:

Một số có tổng các ước nhỏ hơn nó bằng chính nó được gọi là số hoàn chỉnh.

Ví dụ: 6 có các ước nhỏ hơn nó là 1, 2, 3. Tổng là $1 + 2 + 3 = 6$.

Viết chương trình xét xem một số n được nhập từ bàn phím có phải là số hoàn chỉnh không.

a. Hướng dẫn:

- Dùng biến n lưu số cần xét.
- Biến S có giá trị ban đầu bằng 0.
- Cho i chạy từ 1 đến $n-1$. nếu i là ước của n thì cộng thêm i vào S .
- Nếu $S = n$ thì S là số hoàn chỉnh.

b. Mã chương trình:

```
Program So_Hoan_Chinh;
uses crt;
var n, i, s: integer;
begin
    write('nhap so n: ');readln(n);
    s:=0;
    for i:= 1 to n -1 do if n mod i = 0 then s:=s+i;
    if s = n then write(n, ' la so hoan chinh')
    else writeln(n, ' khong phai la so hoan chinh');
    readln
end.
```

Bài tập 3.6:

Viết chương trình tìm các số hoàn chỉnh nhỏ hơn n (Với n được nhập từ bàn phím).

a. Hướng dẫn:

Cho biến i chạy từ 1 đến n . Xét i . Nếu nó là số hoàn chỉnh thì in ra.

b. Mã chương trình:

```
Program Tim_uoc_2;
uses crt;
Var S, n, i,j: longint;
Begin
    clrscr;
    Write('Nhap so n ='); readln(n);
    For i:=1 to n do
    Begin
        S:=0;
        For j:=1 to i do if i mod j = 0 then S:=S+j;
        if S = 2*i then write(i:6, ', ');
    end;
    readln
end.
```

c. Nhận xét: Ở đây ta sử dụng hai vòng lặp lồng nhau.

Bài tập 3.7

In bảng cửu chương n (Với n nhập từ bàn phím)

a. Hướng dẫn :

b. Mã chương trình:

```
Program Cuu_Chuong_1;
uses crt;
var n, i : integer;
begin
  clrscr;
  write('Nhap n: '); readln(n);
  for i:= 1 to 9 do writeln(n, ' x ', i, ' = ',
n*i);
  readln
end.
```

Bài tập 3.8

Lần lượt in các bảng cửu chương.

a. Hướng dẫn :

- Cho biến i chạy từ 2 đến 9
- In bảng cửu chương i.

b. Mã chương trình:

```
Program Cuu_Chuong_1;
uses crt;
var i, j : integer;
begin
  clrscr;
  for i:= 2 to 9 do
  Begin
  Writeln('Bang cuu chuong ', i);
  For j := 1 to 9 do writeln(i, ' x ', j, ' = ',
j*i);
  readln
  end;
  readln
end.
```

c. **Nhận xét:** Chương trình này in bảng cửu chương dọc (Hết bảng này đến bảng khác tính từ trên xuống). Hãy sửa chương trình để in theo kiểu ngang thường thấy.

Bài tập 3.9

Viết chương trình xét xem một số n có phải là số nguyên tố không?

a. Hướng dẫn:

- Nếu n không chia hết mọi số i có giá trị từ 2 đến n - 1 thì n là số nguyên tố.
- Sử dụng biến ok có kiểu boolean và có giá trị ban đầu là true.

- Cho biến i chạy từ 2 đến $n - 1$. Xét $n \bmod i$. Nếu bằng 0 thì gán $ok = false$. Ngược lại vẫn để nguyên ok .

b. Mã chương trình:

```
Program Nguyen_to_1;
uses crt;
var n, i: integer;
    ok: boolean;
begin
    ok:=true;
    write('Nhap n: ');readln(n);
    for i:= 2 to n - 1 do if n mod i = 0 then
        ok :=false;
    if ok then write(n,' la so nguyen to')
    else write(n, ' khong la so nguyen to');
    readln
end.
```

c. Nhận xét: Ở đây ta sử dụng biến có kiểu logic (Đúng, sai). Chỉ cần một lần $n \bmod i = 0$ thì sau khi thực hiện xong vòng lặp ok có giá trị là $false$.

Bài tập 3.10

Viết chương trình in ra tất cả các số nguyên tố bé hơn hoặc bằng n ?

a. Hướng dẫn:

- Cho i chạy từ 2 đến n .
- Xét i . Nếu nó là số nguyên tố thì in nó ra.

b. Mã chương trình:

```
Program Nguyen_to_1;
uses crt;
var n, i, j: integer;
    ok: boolean;
begin
    clrscr;
    write('Nhap n: '); readln(n);
    for i := 2 to n do
        begin
            ok:=true;
            for j:= 2 to i - 1 do if i mod j = 0 then
                ok :=false;
            if ok then write(i, ';')
            end;
        readln
    end.
```

CHƯƠNG IV CẤU TRÚC LẶP KHÔNG XÁC ĐỊNH

A. LÝ THUYẾT

Dạng REPEAT	Dạng WHILE
Repeat S; Until B;	While B Do S;

Ý nghĩa:

- **Dạng REPEAT:** Lặp lại công việc S cho đến khi biểu thức B=TRUE thì dừng. Trước Repeat có thể B chưa được xác định nhưng khi thực hiện S thì B xác định.
- **Dạng WHILE:** Trong khi biểu thức B=TRUE thì tiếp tục thực hiện công việc S. Trước While cần có các lệnh để B được xác định.
- **Yêu cầu quan trọng:** Trong S phải có một lệnh làm thay đổi dữ liệu liên quan đến điều kiện B.

B. BÀI TẬP

Bài tập 4.1:

Viết chương trình in ra các số lẻ nhỏ hơn hoặc bằng số nguyên dương n (Với n được nhập). Yêu cầu nhập lại nếu n <=0

a. Hướng dẫn:

- Sử dụng kiến thức số lẻ đầu tiên bằng 1. Số lẻ sau bằng số trước cộng với 2.
- Cho biến i có giá trị ban đầu bằng 1.
- Dùng vòng lặp while do với điều kiện $i < n$ và công việc bên trong là in i và tăng i lên 2.

b. Mã chương trình:

```

Program In_So_Le;
uses crt;
var i,n:integer;
begin
  clrscr;
  Repeat
  write('Nhap so n: ');readln(n);
  until n>0;
  i:=1;
  while i<=n do
  begin
    write(i:3,' ', ' ');
    i:=i+2;
  end;
  readln
end.

```

Nhận xét:

- Mọi vòng lặp For ... to ... do đều có thể thay thế bằng vòng lặp while ... do.

- Đoạn mã:

```
Repeat  
write('Nhập số n: ');readln(n);  
until n>0;
```

Dùng để kiểm tra, không chế điều kiện của dữ liệu vào.

- Trong vòng lặp while nhất thiết phải có một câu lệnh làm thay đổi điều kiện lặp.

Ở đây là $i:=i+2$. Nếu không có sẽ dẫn đến trường hợp lặp vô hạn. Chương trình chạy mãi mà không có lối ra (Không thoát ra khỏi vòng lặp được).

Bài tập 4.2:

Viết chương trình tính $n!$ với $n!$ được định nghĩa như sau:

- $n! = 1$ với $n = 0$

- $n! = 1.2.3...n$ (Tích của n số từ 1 đến n).

Yêu cầu: Sử dụng vòng lặp với số lần chưa biết trước:

a. Hướng dẫn:

- Có thể viết lại: $n! = n.(n-1)... 3.2.1$.

- Lặp $gt = gt*n; n = n-1$ với điều kiện $n>0$.

b. Mã chương trình:

```
Program Giai_Thua_while;  
uses crt;  
var n, gt:longint;  
begin  
  clrscr;  
  Repeat  
  write('Nhập số n: ');readln(n);  
  until n>0;  
  gt:=1;  
  while n>0 do  
  begin  
  gt:=gt*n;  
  n:=n-1;  
  end;  
  writeln('Giai thua của n là: ',gt);  
  readln  
end.
```

c. **Nhận xét:** Tiết kiệm được một biến i để chạy nhưng làm thay đổi n nên khi xuất ra chỉ có thể xuất một câu chung chung “Giai thừa của n là:”

Bài tập 4.3:

Viết chương trình tính $n!!$ (giai thừa kép) với $n!!$ được định nghĩa như sau:

- $n!! = 1$ với $n = 0$

- $n!! = 1.3.5...n$ với n lẻ.

- $n! = 2.4.6..n$ với n chẵn.

Yêu cầu: Sử dụng vòng lặp với số lần chưa biết trước:

a. Hướng dẫn:

- Hai số chẵn liên tiếp hơn kém nhau 2. Hai số lẻ liên tiếp cũng vậy.

- Thực hiện tính như giai thừa đơn nhưng với bước nhảy là 2.

b. Mã chương trình:

```
Program Giai_thua_kep;
uses crt;
var n,gt:longint;
begin
  Repeat
    write('Nhap so n: ');readln(n);
  until n>0;
  gt:=1;
  while n>0 do
  begin
    gt:=gt*n;
    n:=n-2;
  end;
  write('Giai thua la: ',gt);
  readln
end.
```

c. Nhận xét:

- Với thuật toán trên ta không cần xét n là chẵn hay lẻ.

Bài tập 4.4:

Viết chương trình cho phép tính tổng của nhiều số (Chưa biết bao nhiêu số). Nhập số 0 để kết thúc quá trình nhập.

a. Hướng dẫn:

b. Mã chương trình:

```
Program Tong_Repeat;
uses crt;
var i: byte;
    so, tong: real;
begin
  write('NHAP CAC SO - NHAP 0 DE NGUNG ');
  readln;
  repeat
    clrscr;
    write('Nhap so thu ',i,': ');
    readln(so); tong:=tong+so; i:=i+1;
  until so=0;
  write('Tong la: ',tong:6:1);
  readln
end.
```

Bài tập 4.5

Viết chương trình tìm ước chung lớn nhất (UCLN) của hai số với yêu cầu sử dụng thuật toán Euclid.

Thuật toán Euclid: Nếu a chia hết cho b (a chia b dư 0) thì $UCLN(a,b)$ bằng b

Nếu a chia b dư r thì $UCLN(a,b) = UCLN(b,r)$

a.Hướng dẫn:

- Nhập a, b và gán $r = a \bmod b$.
- Lặp với điều kiện $r \neq 0$: $b = r$, $a = b$, $r = a \bmod b$.

b.Mã chương trình:

```
Program UCLN;
uses crt;
var a,b,r:byte;
begin
  clrscr;
  writeln('CHUONG TRINH TIM UCLN CUA HAI SO');
  write('Nhap a: ');readln(a);
  write('Nhap b: ');readln(b);
  r:=a mod b;
  while r<> 0 do
  begin
    b:=r;
    a:=b;
    r:=a mod b;
  end;
  write('UCLN cua hai so la: ',b);
  readln
end.
```

Bài tập 4.6

Dãy Fibonacy có hai phần tử đầu là 1, 1. Các phần tử sau bằng tổng hai phần tử đứng ngay trước nó: 1, 1, 2, 3, 5, 8, 13, 21, ...

Viết chương trình in ra dãy Fibonacy có phần tử lớn nhất nhỏ hơn n?

a.Hướng dẫn:

- Cần hai biến F₁ và F có giá trị đầu là 1, 1.
- Thực hiện lặp cho đến khi $F \geq n$.
- Do yêu cầu chỉ in các số bé hơn n nên khi in cần thêm một lệnh kiểm tra.

b.Mã chương trình:

```
Program Fi_Bo_na_xi;
Var n, F_2,F_1, F: Longint;
Begin
  Write('Nhap n: ');Readln(n);
  F_1:=1; F_2:=1;
  Write(F_2,',',F_1,',');
  Repeat
    F:=F_2+F_1;
```

```
if F < n then Write(F,');  
F_2:=F_1; F_1:=F;  
Until F > n;  
Readln;  
End.
```

Nhận xét: Giữa Repeat ... until có thể chứa nhiều lệnh mà không cần ghép.
Hãy phát triển bài tập theo hướng chỉ in một phần tử trong dãy lớn nhất nhưng bé hơn n hoặc theo hướng phân tử thứ k của dãy.

CHƯƠNG V DỮ LIỆU KIỂU MẢNG (Một chiều)

A. LÝ THUYẾT

I. KHAI BÁO MẢNG

Cú pháp:

VAR <Biến mảng> : ARRAY [chỉ số] OF <Kiểu dữ liệu>;

Ví dụ:

```
VAR      M: Array[1..100] of Integer;
          C: Array['A'..'Z'] of byte;
```

Trong đó chỉ số có kiểu vô hướng đếm được (như: nguyên, kí tự ...)

II. QUẢN LÝ MỘT MẢNG:

- Để quản lý một dãy cần hai biến nguyên và một biến mảng. Một biến kiểu nguyên để lưu số phần tử của dãy, một biến nguyên khác để lưu chỉ số và một biến mảng để lưu giá trị các phần tử của dãy.

- Để truy cập đến phần tử thứ k trong mảng một chiều A, ta sử dụng cú pháp: A[k].

- Có thể sử dụng các thủ tục READ(LN)/WRITE(LN) đối với các phần tử của biến kiểu mảng.

B. BÀI TẬP

Bài tập 5.1

Viết chương trình cho phép nhập n số và in ra theo thứ tự ngược lại. Ví dụ nhập 3, 5, 7 thì in ra 7, 5, 3.

a. Hướng dẫn:

- Dùng biến n để lưu lượng số cần nhập.
- Dùng mảng để lưu các số vừa nhập.
- Cho i chạy từ n về 1 để in các số vừa nhập.

b. Mã chương trình:

```
Program mang_1;
uses crt;
var n, i: integer;
    M: array[1..100] of real;
Begin
  write('Nhập số n: ');readln(n);
  for i:=1 to n do
  Begin
    write('M[' ,i, ']='); readln(M[i]);
  end;
  for i:= n downto 1 do write(m[i], ' ');
  readln
end.
```

Bài tập 5.2

Viết chương trình nhập dãy n số và in ra tổng các số lẻ trong dãy số vừa nhập.

a. Hướng dẫn:

Thực hiện cộng dồn các số lẻ bằng lệnh:

```
if M[i] mod 2 =1 then tong:=tong+M[i]
```

b. Mã chương trình:

```
Program Mang_Tong_Le;
uses crt;
var i,n:byte;
    M:array[1..100] of integer;
    tong:longint;
begin
    write('Nhap so phan tu cua day: ');readln(n);
    for i:=1 to n do
    begin
        write('M[' ,i, ']); readln(M[i]);
    end;
    tong:=0;
    for i:=1 to n do if M[i] mod 2 =1 then
    tong:=tong+M[i];
    write('Tong cac so le trong day la: ',tong);
    readln
end.
```

c. Nhận xét: Với yêu cầu của bài toán thì không cần sử dụng biến mảng vẫn giải được. Hãy thử nhé.

Bài tập 5.3

Viết chương trình nhập n số, xoá số thứ k trong n số vừa nhập. In ra n-1 số còn lại.

n= 10 (Nhập 10 phần tử)

Ví dụ: Nhập 2, 3, 4, 5, 6, 8, 7, 6, 5, 4.

k= 8 (Xoá phần tử thứ 8).

In ra: 2, 3, 4, 5, 6, 8, 7, 5, 4.

a. Hướng dẫn:

Xoá phần tử k bằng cách ghi đè phần tử thứ k+1 lên nó.

b. Mã chương trình:

```
Program Xoa_mang;
uses crt;
var m:array[1..100] of integer;
    n,i,k:byte;
begin
    Write('So phan tu cua day: ');readln(n);
    for i:=1 to n do
    Begin
        write('M[' ,i, ']=');
```

```
        readln(M[i]);
    end;
    write('Nhap phan tu can xoa: '); readln(k);
    for i:=k to n-1 do m[i]:=m[i+1];
    for i:=1 to n-1 do write(m[i],', ');
    readln
end.
```

c. Nhận xét: Với yêu cầu của bài tập trên chỉ cần kiểm tra chỉ số khi in (VD nếu $i = 8$ thì không in). Tuy nhiên trong khi sử dụng mảng để lưu dữ liệu giải toán, nhiều khi ta có nhu cầu xóa bớt hoặc chèn thêm phần tử vào dãy.

Bài tập 5.4

Viết chương trình cho phép nhập một dãy gồm n số nguyên. Nhập thêm một số và chèn thêm vào dãy sau phần tử k .

a. Hướng dẫn:

- Dời các phần tử từ vị trí k về sau một bước.
- Nhập giá trị cần chèn vào vị trí k .

b. Mã chương trình:

```
Program Mang_chen;
uses crt;
var M: array[1..100] of integer;
    i,n,k:integer;
begin
    clrscr;
    write('Nhap : '); readln(n);
    for i:=1 to n do
    begin
        write('M[' ,i, ']='); readln(M[i]);
    end;
    write('Vi tri chen: '); readln(k);
    for i:=n+1 downto k+1 do M[i]:=M[i-1];
    write('Nhap so can chen: '); readln(M[k]);
    for i:=1 to n+1 do write(M[i],', ');
    readln
end.
```

Bài tập 5.5

Viết chương trình cho phép nhập n số và cho biết số nhỏ nhất trong các số vừa nhập là số thứ mấy.

a. Hướng dẫn:

- Dùng biến n để lưu lượng số cần nhập.
- Dùng mảng để lưu các số vừa nhập.
- Cho $\text{Min} = M[1]$, $j = 1$ (Xem phần tử đầu tiên là bé nhất)

- So sánh Min với n-1 số còn lại. Trong quá trình so sánh nếu $Min > M[i]$ thì gán $Min = M[i]$, $j=i$ và tiếp tục so sánh.

b. Mã chương trình:

```
Program TIM_NHO_NHAT;
uses crt;
var n,i,nhonhat:integer;
    m: array[1..100] of real;
    min:real;
begin
    write('Nhap n: '); readln(n);
    for i:=1 to n do
    begin
        write('M[' ,i, ']='); readln(m[i]);
    end;
    min:=m[1];
    nhonhat:=1;
    for i:=2 to n do if m[i] < min then
    begin
        min:=m[i]; nhonhat:=i;
    end;
    writeln('phan tu nho nhat la phan tu
thu',nhonhat);
    readln
end.
```

Bài tập 5.6

Viết chương trình cho phép nhập n số sắp xếp và in ra các số đã nhập theo thứ tự tăng dần.

a. Hướng dẫn:**b. Mã chương trình:**

```
Program Sap_xep_mang;
Var M: array[1..10] of integer;
    i,j,n: byte;
    tam: integer;
Begin
Write('Nhap so phan tu n:');Readln(n);
For i:=1 to n do
Begin Write('M[' ,i, ']='); Readln(M[i]); End;
For i:=1 to n-1 do
For j:=i+1 to n do if M[j] <=M[i] then
Begin Tam:= M[i]; M[i]:=M[j]; M[j]:=tam; End;
Write('Sau khi sap xep: ');
For i:=1 to n do Write(M[i],',');
Readln;
```

```
End.
```

Bài tập 5.7:

Viết chương trình in dãy n số fibonacci.

a. Hướng dẫn:

- Sử dụng mảng M để chứa dãy n số fibonacci. Tạo lập hai phần tử đầu tiên là 1, 1.
- Cho i chạy từ 3 đến n. $M[i] = M[i-1] + M[i-2]$.
- In n phần tử đầu tiên của mảng.

b. Mã chương trình:

```
Program Fibonacci_mang;
uses crt;
var i, n: integer;
    m: array[1..100] of longint;
begin
    write('Nhap so phan tu can in:'); readln(n);
    m[1]:=1;
    m[2]:=1;
    for i:=3 to n do m[i]:=m[i-1]+m[i-2];
    for i:=1 to n do write(m[i], ' ');
    readln
end.
```

Bài tập 5.8

Viết chương trình in ra màn hình tam giác Pascal. Ví dụ, với n=4 sẽ in ra hình sau:

```

1      1
1      2      1
1      3      3      1
1      4      6      4      1
```

... Hàng thứ n được xác định từ hàng n-1:

- Phần tử đầu tiên và phần tử cuối cùng đều bằng 1.
- Phần tử thứ 2 là tổng của phần tử thứ nhất và thứ 2 của hàng n-1
- Phần tử thứ k của hàng thứ n là tổng của phần tử thứ k-1 và k của hàng thứ n-1.

Thuật toán:

Bước 1: Khởi tạo một mảng một chiều n phần tử có giá trị 0.

Bước 2: Khởi tạo giá trị cho hàng thứ nhất $M[1,1] = 1$.

Bước 3: - Đối với hàng thứ i tính giá trị phần tử từ phần tử thứ i + 1 xuống phần tử thứ 2:
 $M[j] := M[j] + M[j-1]$

- In ra hàng thứ i.

Chương trình:

```
Program Tam_giac_Pascal_mot_chieu;
Var n, i, j: integer;
    M: array[1..10] of integer;
Begin
```



```
Write('Nhap n: '); Readln(n);
For i:= 1 to n do M[i]:=0;
M[1]:=1;
For i:=1 to n do
Begin
    For j:=i+ 1 downto 2 do M[j]:=M[j]+M[j-1];
    For j:=1 to i+ 1 do Write(M[j]:3);
    Writeln;
End;
Readln
End.
```

Bài tập 5.9

Viết chương trình cho phép nhập.

- a. Hướng dẫn:
- b. Mã chương trình:

Bài tập 5.10

Viết chương trình cho phép nhập.

- a. Hướng dẫn:
- b. Mã chương trình:

CHƯƠNG VI

CHƯƠNG TRÌNH CON: THỦ TỤC VÀ HÀM

A. LÝ THUYẾT

I. KHÁI NIỆM VỀ CHƯƠNG TRÌNH CON

Chương trình con (CTC) là một đoạn chương trình thực hiện trọn vẹn hay một chức năng nào đó. Trong Turbo Pascal, có 2 dạng CTC:

- Thủ tục (PROCEDURE): Dùng để thực hiện một hay nhiều nhiệm vụ nào đó.
- Hàm (FUNCTION): Trả về một giá trị nào đó (có kiểu vô hướng, kiểu string). Hàm có thể sử dụng trong các biểu thức.

Ngoài ra, trong Pascal còn cho phép các CTC lồng vào nhau.

II. CẤU TRÚC CHUNG CỦA MỘT CHƯƠNG TRÌNH CÓ SỬ DỤNG CTC

```
PROGRAM Tên_chương_trình;
USES CRT;
CONST .....;
VAR .....;

PROCEDURE THUTUC[(Các tham số)];
[Khai báo Const, Var]
BEGIN
    .....
END;

FUNCTION HAM[(Các tham số)]:<Kiểu dữ liệu>;
[Khai báo Const, Var]
BEGIN
    .....
    HAM:=<Giá trị>;
END;

BEGIN {Chương trình chính}
    .....
    THUTUC[(...)]
    .....
    A:= HAM[(...)]
    .....
END.
```

Chú ý: Trong quá trình xây dựng CTC, khi nào thì nên dùng thủ tục/hàm?

Dùng hàm	Dùng thủ tục
<ul style="list-style-type: none"> - Kết quả của bài toán trả về 1 giá trị duy nhất (kiểu vô hướng, kiểu string). - Lệnh gọi CTC cần nằm trong các biểu thức tính toán. 	<ul style="list-style-type: none"> - Kết quả của bài toán không trả về giá trị nào hoặc trả về nhiều giá trị hoặc trả về kiểu dữ liệu có cấu trúc (Array, Record, File). - Lệnh gọi CTC không nằm trong các biểu thức tính toán.

B. BÀI TẬP

Bài tập 6.1:

Viết chương trình giải phương trình bậc hai với yêu cầu sử dụng các chương trình con để giải quyết các trường hợp xảy ra của delta.

a. Hướng dẫn:

b. Mã chương trình:

```

Program Giai_PT_bac_hai;
uses crt;
var a,b,c,delta:real;
{-----}
Procedure delta_duong;
begin
write('Phuong trinh co hai nghiem x1=',(-
b+sqrt(delta))/(2*a), 'x2=', (-b+sqrt(delta))/(2*a));
end;
{-----}
Procedure delta_khong;
begin
write('Phuong trinh co nghiem kep x=',-b/(2*a):3:1);
end;
{-----}
Procedure delta_am;
begin
writeln('Phuong trinh vo nghiem');
end;
{Chương trình chính}
Begin
clrscr;
write('Nhap a: ');readln(a);
write('Nhap b: ');readln(b);
write('Nhap c: ');readln(c);
delta:=b*b-4*a*c;
if delta>0 then delta_duong
else if delta = 0 then delta_khong
else delta_am;
readln
end.

```

c. Nhận xét: Chương trình con cho phép chia nhỏ công việc nhằm đơn giản hoá. Ngoài ra một công việc nào đó lặp lại nhiều lần trong chương trình nên dùng CTC vì khi đó mã chương trình sẽ gọn, dễ theo dõi, gỡ rối hơn.

Bài tập 6.2:

Viết chương trình cho phép nhập hai số vào hai biến, thực hiện đổi giá trị của hai biến cho nhau. Yêu cầu dùng chương trình con để thực hiện chức năng đổi giá trị.

a. Hướng dẫn:

b. Mã chương trình:

```
Program CTC_1;
uses crt;
var a,b: real;
{----CTC doi gia tri----}
Procedure swap(var x,y:real);
var tam:real;
begin
tam:=x; x:=y; y:=tam;
end;
{-----Ket thuc CTC-----}
begin
clrscr;
write('Nhap so a: ');readln(a);
write('Nhap so b: ');readln(b);
swap(a,b);
write('Sau khi doi a =',a:3:1);
write('Sau khi doi b =',b:3:1);
readln
end.
```

b. Nhận xét:

- Nếu bỏ từ var ở khai báo var x, y: real thì chương trình vẫn không báo lỗi nhưng chức năng đổi giá trị của hai biến không thực hiện được.

Bài tập 6.3:

Tìm số lớn thứ nhì trong n số được nhập từ bàn phím.

Ví dụ: Nhập các số 10; 10; 9; 9; 8 thì được số lớn nhì là 9.

Giải thuật:

- Nhập 2 số, Xác định giá trị cho hai biến Max, Nhi.
- Lần lượt nhập các số. Với mỗi số được nhập xét:
 - Nếu So > Max thì gán Nhi = Max, Max = So.
 - Nếu Nhi < So < Max thì gán Nhi = So.

```
Program So_lon_nhi;
Var n,i,so, nhi,max,tam:integer;
Procedure swap(var x,y:real);
```

```
var tam:real;
begin
tam:=x;
x:=y;
y:=tam;
end;
Begin
  Write('Nhap n: ');Readln(n);
  Write('nhap so:');Readln(max);
  Write('nhap so: ');Readln(nhi);
  if nhi > Max then swap(Max,nhi)
  For i:= 3 to n do
  Begin
    Write('Nhap so: ');Readln(so);
    if (so>nhi) and (so<max) then nhi:=so;
    if so>max then
    Begin
      nhi:=max;
      Max:=so;
    End;
  End;
  Write('So thu nhi la: ',nhi);
  Readln
End.
```

Bài tập 6.3:

Viết chương trình tính giai thừa của số n (Viết là n!). Với yêu cầu:

- Nếu người dùng nhập số $n < 0$ thì yêu cầu nhập lại.
- Sử dụng chương trình con để tính giai thừa của một số.
 - $n! = 1$ nếu $n = 0$;
 - $n! = 1.2.3.4.5...n$ (Tích của n thừa số).

a. Hướng dẫn:

b. Mã chương trình:

```
Program CTC_2;
uses crt;
var n:integer;x:longint;
Procedure Giai_Thua(var GT:longint; n:byte);
begin
  GT:=1;
  while n>0 do
  begin
    GT:=GT*n;
    n:=n-1;
  end;
end;
```

```
begin
  repeat
    clrscr;
    write('Nhap so n: ');readln(n);
    if n < 0 then begin write('Nhap so n>=0');readln;
  end;
  until n>=0;
  Giai_Thua(x,n);
  writeln('Giai thua cua ',n,'la:',x);
  readln
end.
```

Nhận xét: Lệnh $n:=n-1$ làm thay đổi giá trị của n nhưng khi ra khỏi chương trình con n có giá trị không đổi so với trước khi gọi chương trình con.

Bài tập 6.4:

Viết chương trình tính $n!$ với yêu cầu sử dụng hàm để tính giai thừa.

a. Hướng dẫn:

b. Mã chương trình:

```
Program Giai_thua_Ham;
uses crt;
var n: longint;
Function Giai_Thua(n:longint):longint;
Var GT:Longint;
begin
  GT:=1;
  while n > 0 do
  begin
    GT:=GT * n;
    n:=n-1;
  end;
  Giai_thua:=GT;
end;
begin
  clrscr;
  write('Nhap n: '); readln(n);
  write(n, '!=',Giai_thua(n));
  readln
end.
```

Nhận xét: Hãy so sánh sự khác nhau khi Giai_thua được viết dưới hai dạng Function và Procedure.

- Khi dùng Procedure cần một biến (toàn cục) để lưu giữ giá trị của $n!$. Biến này được truyền cho tham biến trong Procedure. Sau khi gọi nó cần lệnh để in $n!$
- Khi dùng Function, có thể sử dụng nó như là một biểu thức.

Bài tập 6.5:

Viết chương trình cho phép thực hiện rút gọn phân số.

a. Hướng dẫn:

- Tìm UCLN của tử số và mẫu số.
- Chia tử và mẫu của phân số cho UCLN vừa tìm được.

b. Mã chương trình:

```
Program Rut_gon_phan_so;
uses crt;
var tu, mau: integer;
Function UCLN(a, b: integer): integer;
var r: integer;
begin
    r:= a mod b;
    while r <> 0 do
    begin
        a:= b;
        b:= r;
        r:=a mod b;
    end;
    UCLN:=b;
end;
begin
    clrscr;
    write('Nhap tu: '); readln(tu);
    write('Nhap mau: '); readln(mau);
    write('Ket qua rut gon: ', tu, '/', mau, '=', tu div
UCLN(tu, mau), '/', mau div UCLN(tu, mau));
    readln
end.
```

Bài tập 6.6:

Viết chương trình cho phép trộn hai dãy số A và B cùng có số phần tử là k để được dãy số C theo yêu cầu sau:

A = a₁, a₂ ... a_k

B = b₁, b₂ ... b_k

Được C = a₁, b₁, a₂, b₂ ... a_k, b_k.

a. Hướng dẫn:

b. Mã chương trình:

```
Program Tron_day;
uses crt;
type kieu_mang = array[1..100] of integer;
var A, B, C: Kieu_mang;
    n, i, j: integer;
```

```

Procedure Nhap_Mang(Var X: Kieu_mang; n:byte);
var i:integer;
begin
  for i:=1 to n do
  begin
    write('M[' ,i, ']=');readln(X[i]);
  end;
end;
Procedure In_Mang(X:kieu_mang; n:byte);
var i: byte;
begin
  for i:=1 to n do write(x[i],', ');
end;
{-----}
begin
  clrscr;
  write('Nhap so phan tu cua day: ');read(n);
  Nhap_mang(A,n);
  Nhap_mang(B,n);
  i:=1;
  j:=1;
  while i<= n do
  begin
    C[j]:= A[i];
    C[j+1]:=B[i];
    j:=j+2;
    i:=i+1;
  end;
  in_mang(C,2*n);
  readln
end.

```

Bài tập 6.7:

Viết chương trình in ra các số nguyên tố nhỏ hơn n với yêu cầu dùng hàm để kiểm tra một số có phải là số nguyên tố hay không.

a. Hướng dẫn:**b. Mã chương trình:**

```

Program In_so_nguyen_to;
uses crt;
var n, i: integer;
{-----Ham kiem tra-----}
Function kiem_tra(n:integer):boolean;
var i: integer;
begin

```



```
    kiem_tra:=true;
    for i:=2 to n-1 do if (n mod i) = 0 then
kiem_tra:=false;
end;
{-----CTC-----}
begin
    clrscr;
    write('Nhap n: '); readln(n);
    for i:=2 to n do if kiem_tra(i) then write(i:3,
', ');
    readln;
end.
```

Bài tập 6.8:

Viết chương trình cho phép sắp xếp một dãy số với yêu cầu sử dụng các chương trình con: Nhập mảng, in mảng, đổi giá trị của hai số.

a. Hướng dẫn:**b. Mã chương trình:**

```
Program Sap_xep_day;
uses crt;
type kieu_mang =array[1..100] of integer;
Var A: kieu_mang;
    n,i,j:byte;
{-----}
Procedure Nhap_mang(var M:Kieu_Mang;n:byte);
Var i:byte;
begin
    for i:=1 to n do
    begin
        write('M[' ,i, ']=');readln(M[i]);
    end;
end;
{-----}
Procedure In_mang(var M:Kieu_Mang;n:byte);
Var i:byte;
begin
    for i:=1 to n do write(M[i]:3, ', ');
end;

{-----}
Procedure swap(var a,b:integer);
var tam:integer;
begin
    tam:=b;
    b:=a;
```

```
        a:=tam;
end;
{-----}

Begin
  Clrscr;
  write('Cho biet so phan tu cua day:
');readln(n);
  nhap_mang(A,n);
  for i:=1 to n-1 do
    for j:= i to n do if A[i]>A[j] then
swap(A[i],A[j]);
  In_mang(A,n);
  readln
end.
```

CHƯƠNG VII CHUYÊN ĐỀ CHIA HẾT – SỐ NGUYÊN TỐ.

A. LÝ THUYẾT:

- :
- :
.
- :

B. BÀI TOÁN:

Bài tập 7.1 :

Nhập vào một số nguyên dương n . Hãy in ra số nguyên tố nhỏ nhất lớn hơn n .
VD: Nhập $n = 10$. Kết quả in ra số 11.

Giải thuật :

- Gán $i := n$;
- Thực hiện cho đến khi i là nguyên tố việc tăng i lên 1.

```
Program Nguyen_to;
Var n,i:integer;
Function NT(n:integer):Boolean;
Var ok: Boolean;
    i: integer;
Begin
    ok:=true;
    for i:=2 to n-1 do
        if (n mod i)= 0 then ok:=ok and false;
        if n < 2 then NT:=false else NT:=ok;
    End;
Begin
    Write('Nhap n: ');Readln(n);
    i:=n;
    Repeat i:=i+1;
    Until NT(i);
    Write('So nguyên to nho nhat lon hon ',n, 'la: ',i);
    Readln
End.
```

Bài tập 7.2 :

Nhập vào từ bàn phím số tự nhiên n ($n < 1000$). Hãy phân tích n thành tích các thừa số nguyên tố.

VD: Nhập vào $n = 9$ được $9 = 3.3$

Thuật toán:

- Gán $i := 2$;
- Khi $n > 1$ thì lặp:
Nếu n chia hết cho i thì in ra i và gán lại $n := n \text{ div } i$. Ngược lại tăng i lên 1.

```
Program Phan_tich;
var n,i: integer;
Begin
  Write('Nhap so can phan tich: ');Readln(n);
  i:=2;
  Write('Ket qua phan tich:');
  Write(n,'=');
  While n>1 do
  Begin
    if n mod i = 0 then Begin Write(i,'); n:= n div i End
    else i:=i+1;
  End;
  Readln
End.
```

Nhận xét: Cài đặt trên in dư một dấu nhân ở cuối. Hãy chỉnh sửa để bỏ dấu nhân thừa này.

Bài tập 7.3:

Tìm các số tự nhiên nhỏ hơn hoặc bằng n mà sau khi làm phép phân tích ra thừa số nguyên tố có nhiều nhân tử nhất.

Ví dụ $n=9$. Các số có nhiều nhân tử nhất sau khi làm phép phân tích là: $8 = 2.2.2$

Thuật toán:

Cài đặt:

```
Program Phan_tich_nguyen_to_2;
Var n, Max, so, i:byte;
Function PTNT(n:integer):byte;
Var i,p:byte;
Begin
  i:=2;
  p:=0;
  While n>1 do if (n mod i)=0 then Begin p:=p+1; n:=n div
i end else i:=i+1;
  PTNT:=p;
End;
Procedure PT(n:integer);
Var i:byte;
Begin
  i:=2;
  While n>1 do
  if (n mod i)=0 then
  Begin Write(i,'); n:=n div i end else i:=i+1;
End;

Begin
  Write('Nhap so n: ');Readln(n);
```

```
Max:=0;
For i:= 1 to n do if PTNT(i)>=Max then Begin
Max:=PTNT(i); So:=i End;
Write('So ',So,' co nhieu uoc nhat,' ,so,' = ');
PT(So);
Readln
End.
```

Bài tập 7.4:

Viết chương trình cho phép phân tích một số ra thừa số nguyên tố và ghi kết quả dưới dạng tích các lũy thừa. Ví dụ: $300 = 2^2 \cdot 3 \cdot 5^2$

Thuật toán:

Dùng một mảng để lưu lũy thừa. Mảng này có giá trị các phần tử ban đầu đều bằng 0.

Nếu n chia hết cho i thì tăng M[i] lên 1.

Khi in kiểm tra: Nếu $M[i] > 0$ thì in $i^{M[i]}$.

Cài đặt:

```
Program Phan_tich;
Var M: array[1..1000] of byte;
    i: byte;
    n: integer;
Begin
For i:=1 to 1000 do M[i]:=0;
Write('Nhap so n: ');Readln(n);
i:=2;
While n>1 do if (n mod i = 0) then begin M[i]:=M[i]+1;
n:=n div i End
else i:=i+1;
For i:=1 to 1000 do if M[i]>0 then Begin If M[i]>1 then
Write(i,'^',M[i],'.') else Write(i,'.') End;
Readln;
End.
```

Bài tập 7.5

Mọi số tự nhiên đều có thể viết được dưới dạng tổng của hai số nguyên tố. Viết chương trình thực hiện tách một số tự nhiên thành tổng của hai số nguyên tố.

Thuật toán:

Cài đặt:

```
Program Tong_nguyen_to;
Var i,n:integer;
Function NT(n:integer):Boolean;
Var ok: Boolean;
    i:integer;
Begin
ok:=true;
```

```
For i:=2 to n-1 do if (n mod i) = 0 then ok:=ok and
false;
if n>=2 then NT:=ok else NT:=false;
End;
Begin
Write('Nhap so n: ');Readln(n);
For i:=2 to n div 2 do if (NT(i) and NT(n-i)) then
Writeln(n, ' = ', i, ' + ', n-i);
Readln
End.
```

Nhận xét: Hãy mở rộng bài toán theo hướng

- Xét xem trong đoạn $[n1...n2]$ số nào cho phép tách thành tổng hai số nguyên tố nhiều trường hợp nhất.

- Tách một số thành tổng ba số nguyên tố.

Bài tập 7.6: (Tin học trẻ toàn quốc lần I - 1995)

Hai số tự nhiên A, B được coi là hữu nghị nếu như số này bằng tổng các ước số của số kia và ngược lại. Lập trình tìm và chiếu lên màn hình các cặp số hữu nghị trong phạm vi từ 1 đến 10000. (Lưu ý: số 1 được coi là ước số của mọi số còn mỗi số không được coi là ước số của chính nó).

Thuật toán:

-

Cài đặt:

```
Program Cap_so_huu_nghi;
Var a,b,n,i:integer;
Function TU(a:integer):integer;
Var Tg,i:integer;
Begin
Tg:=0;
For i:=1 to a-1 do if (a mod i = 0) then Tg:=Tg + i;
TU:=Tg;
End;
Begin
Write('Nhap so n: ');Readln(n);
For a:=1 to n do Begin b:=TU(a); if TU(b)=a then
Writeln(a, '-', b) end;
Readln;
End.
```

Nhận xét: Các số hoàn chỉnh luôn là hữu nghị của chính nó.

CHƯƠNG VIII CHUYÊN ĐỀ DÃY CON.

A. LÝ THUYẾT:

- Dãy con là dãy các phần tử liên tục thuộc một dãy có trước (dãy mẹ) thỏa mãn một tính chất nào đó.
- Để quản lý một dãy con cần một chỉ số (nơi bắt đầu dãy con) và độ dài của dãy.
- Một cách quản lý khác là chỉ số đầu và chỉ số cuối.
- Để xây dựng một dãy con cần:
 - Xây dựng giá trị ban đầu.
 - Duyệt qua các phần tử của dãy, Nếu:
 - Thỏa điều kiện, tăng độ dài thêm 1 ngược lại:
 - Nếu dãy con đang xét cần lưu thì: Lưu lại độ dài, chỉ số đầu dãy, Xác định lại độ dài, chỉ số đầu của dãy mới.
 - Nếu dãy con đang xét không cần lưu thì: Xác định lại độ dài, chỉ số đầu của dãy mới.
- Để duyệt qua tất cả các dãy con của một dãy gồm n số ta dùng thuật toán vét cạn sau:
For i:= 1 to n
 For j:= 1 to n-i+1 Xét dãy con bắt đầu từ vị trí thứ i có độ dài j.

B. BÀI TẬP:

Bài tập 1: Cho dãy số gồm n số. Tìm dãy con lớn nhất các phần tử tăng (giảm) dần.

Giải thuật:

Sử dụng kỹ thuật xây dựng dãy con.

Cài đặt:

```
Program Day_con1;  
Var M: array[1..100] of integer;  
    i,n, dau,ldau, dai,Max: integer;  
Begin  
  Write('Nhap so n: '); Readln(n);  
  For i:=1 to n do  
    Begin Write('[',i,']='); Readln(M[i]); End;  
  {Khoi tao gia tri dau}  
  i:=0;  
  Max:=1;  
  dau:=1;  
  dai:=1;  
  ldau:=1;  
  While i<=n do  
    Begin  
      i:=i+1;  
      if M[i+1]>=M[i] then dai:=dai+1 else  
      if dai> Max then Begin Max:=dai; ldau:=dau; dai:=0  
End  
      else Begin dau:=i+1; dai:=1 End;  
    End;
```

```
Write('Xau con dai:',max,' bat dau tu: ',ldau);  
Readln  
End.
```

Nhận xét: Bài toán trên có thể sử dụng giải thuật vét cạn dãy con để giải. Sau đây là cài đặt:

```
Program Day_con1b;  
Type KM= array[1..100] of integer;  
Var M:KM;  
i,j,n, dau,ldau, dai,Max: integer;  
Function KT(A:KM;m,l:byte):boolean;  
Var ok:Boolean;  
i:byte;  
Begin  
ok:=True;  
For i:=m to m+l-1 do if A[i]>A[i+1] then ok:=ok and  
false;  
KT:=ok;  
End;  
Begin  
Write('Nhap so nc: '); Readln(n); Max:=0;  
For i:=1 to n do Begin Write('[' ,i, '=');  
Readln(M[i]); End;  
For i:= 1 to n-1 do  
For j:=1 to n-i+1 do  
if KT(M,i,j) then  
if j+1> Max then Begin ldau:=i; Max:=j+1  
End;  
Write('Xau con dai:',max,' bat dau tu: ',ldau);  
Readln  
End.
```

Bài tập 2: Cho dãy số gồm n số. Tìm dãy con lớn nhất các phần tử có cùng dấu, (đơn dấu).

Giải thuật:

Thực hiện giống như bài 1, chỉ thay điều kiện là $M[i+1]*M[i] > 0$

Cài đặt:

```
Program Day_con2;  
Var M: array[1..100] of integer;  
i,n, dau,ldau, dai,Max: integer;  
Begin  
Write('Nhap so nc: '); Readln(n);  
For i:=1 to n do Begin Write('[' ,i, '=');  
Readln(M[i]); End;
```



```
    i:=0;
    Max:=1;
    dau:=1;
    dai:=1;
    ldau:=1;
    While i<=n do
    Begin
        i:=i+1;
        if M[i+1]*M[i]>0 then dai:=dai+1 else
        if dai> Max then Begin Max:=dai; ldau:=dau; dai:=0
End
        else Begin dau:=i+1; dai:=1 End;
    End;
    Write('Xau con dai:',max,' bat dau tu: ',ldau);
    Readln
End.
```

Nhận xét: Hãy thực hiện bài tập trên bằng kỹ thuật vét cạn dãy con.

Bài tập 3: Cho dãy gồm n số. Tìm dãy con lớn nhất đơn điệu (liên tục tăng, giảm hoặc giảm, tăng).

Giải thuật:

- Dãy đang dấu nếu $M[i]*M[i+1] < 0$.

Cài đặt:

Giống bài tập 2

Nhận xét:

Bài tập 4: Cho dãy số gồm n số nguyên. Tìm dãy con có tổng lớn nhất

Giải thuật:

- Sử dụng kỹ thuật vét cạn các dãy con, dùng hàm tính tổng dãy con để kiểm tra.

Cài đặt:

```
Program Day_con1b;
Type KM= array[1..100] of integer;
  Var M:KM;
      i,j,n,ldau, dai,Max: integer;
Function TONG(A:KM;m,l:byte):Integer;
Var Tam,i:integer;
Begin
  Tam:=0;
  For i:=m to m+l do Tam:=Tam + A[i];
  TONG:=Tam;
End;
Begin
  Write('Nhập số n: '); Readln(n);
```

```
For i:=1 to n do Begin Write(['i,']=); Readln(M[i]); End;  
Max:=M[1];dai:=1;ldau:=1;  
For i:= 1 to n do  
  For j:=0 to n-i+1 do  
    if TONG(M,i,j)> Max then  
      Begin ldau:=i; Max:=Tong(M,i,j) ; dai:=j+1 End;  
  Write('Xau con co tong:',max,' bat dau tu: ',ldau, ' dai: ',dai);  
  Readln  
End.
```

Nhận xét:

CHƯƠNG IX CHUYÊN ĐỀ CHỮ SỐ, HỆ CƠ SỐ.

A. LÝ THUYẾT:

- Trong hệ cơ số 10: Số $A = a_n \dots a_2 a_1 a_0 = a_0 + 10a_1 + \dots + 10^n a_n$ nên:

Để lấy số $a_0 = A \bmod 10$; Để xóa chữ số a_0 ta dùng $A := A \text{ div } 10$.

(Tương tự: Để lấy hai số tận cùng $a_1 a_0 = A \bmod 100$; Để xóa hai chữ số $a_1 a_0$ ta dùng $A := A \text{ div } 100$.)

- Thuật toán vét cạn: Để xét tất cả các trường hợp của số A ta xét $a_n = 1..9$; $a_{n-1} \dots a_2 a_1 a_0 = 0..9$

- Hệ cơ số 2: Nếu như hệ thập phân dùng 10 chữ số để ghi số thì hệ cơ số 2 chỉ dùng hai chữ số là 0 và 1 để ghi số.

- Đổi một số từ cơ số 2 sang cơ số 10:

- Đổi một số từ cơ số 10 sang cơ số 2:

- Hệ cơ số bất kỳ:

- Vấn đề cộng, trừ, nhân, lũy thừa số lớn (hoặc kết quả được số lớn) được xem xét riêng ở một chuyên đề (sau khi được trang bị dữ liệu kiểu string).

B. BÀI TOÁN:

Bài tập 8.1:

Gọi $abcd$ là một số có 4 chữ số. Hãy lập chương trình tìm tất cả các số có 4 chữ số thỏa mãn biểu thức: $abcd = (ab + cd)^2$

Ví dụ: số $2025 = (20 + 25)^2$.

Thuật toán:

Kiểm tra tất cả các số có bốn chữ số theo các cách sau;

- Tách lấy hai số đầu, hai số sau của số có bốn chữ số để kiểm tra.

- Kiểm tra các trường hợp có thể của mỗi chữ số.

Cách 1:

```

Program Tach_so;
Var haisodau, haisocuoi, i : integer;
Begin
  Writeln( ' cac so thoa man dieu kien gom co' );
  For i:=1000 to 9999 do
    Begin
      haisodau:=i Div 100; {lay 2 so dau tien ab}
      haisocuoi:=i mod 100; {lay 2 so cuoi cd}
      If i=SQR(haisodau + haisocuoi) then write(i:5);
    End;
  Readln;
End.
```

Cách 2:

```

Program Xet_so;
Var a,b,c,d : integer;
Begin
  Writeln( ' cac so thoa man dieu kien gom co' );
```

```
For a:=1 to 9 do
  For b:=0 to 9 do
    For c:=0 to 9 do
      For d:=0 to 9 do
        If i=(1000*a + 100*b + 10*c + d) = SQRT(10*a+b + 10*c+d)
          then write(i:5);
      Readln;
End.
```

Bài tập 8.2:

Viết chương trình cho nhập hai số tự nhiên N và k. Hãy cho biết chữ số thứ k tính từ trái sang phải trong số N là số mấy? nếu k lớn hơn độ dài của N hoặc k bằng 0 thì thông báo không tìm được.

Ví dụ 1: Với N và k được nhập: N = 65932, k = 4

Kết quả tìm được là 3.

Ví dụ 2: Với N và k được nhập: N = 65932, k = 10

Kết quả tìm được là -1 (k lớn hơn độ dài số N).

```
Program Chu_so_thu_k;
Var M: array[1..10] of integer;
    so: Longint;
    i,k:integer;
Begin
  Write('Nhap so: ');Readln(so); so:=abs(so);
  Write('Nhap k: ');Readln(k);
  i:=0;
  While so>0 do
  begin
    i:=i+1;
    M[i]:=so mod 10;
    so:=so div 10;
  end;
  Write('Chu so thu ',k,'la: ',M[i-k+1]);
  Readln
End.
```

Nhận xét : Nếu bài toán yêu cầu tìm chữ số thứ k tính từ phải sang trái thì đơn giản hơn nhiều. Lúc đó ta chỉ cần xóa k-1 chữ số cuối. Rồi lấy chữ số cuối.

Bài tập 8.2:

Một số được gọi là số bậc thang nếu biểu diễn thập phân của nó có nhiều hơn một chữ số đồng thời theo chiều từ trái qua phải, chữ số đứng sau không nhỏ hơn chữ số đứng trước. Viết chương trình in ra các số bậc thang trong đoạn [n1, n2] với n1, n2 được nhập từ bàn phím.

```
Program So_bac_thang;
Var i,n1,n2: integer;
```

```
Function BT(n:integer):Boolean;
Var ok: boolean;
    so:byte;
Begin
    ok:=true;
    While n>=10 do
    Begin
        so:=n mod 10;
        n:=n div 10;
        if so < (n mod 10) then ok:=ok and false;
    End;

    BT:=ok;
End;
Begin
    Write('Nhap so n1: ');Readln(n1);
    Write('Nhap so n2: ');Readln(n2);
    For i:= n1 to n2 do if BT(i) then Write(i:4);
    Readln
End.
```

Bài tập 8.3:

Viết chương trình cho phép đổi một số từ cơ số 10 sang cơ số bất kỳ.

Thuật toán:

- Dùng mảng CS để lưu các chữ số.
- Lặp khi $n \neq 0$ việc: Chia n cho s lấy phần dư. Lấy phần dư làm chỉ số để lấy và lưu chữ số. Gán $n = n \text{ div } s$.
- Chú ý chữ số lấy sau sẽ nằm trước.

```
Program Doi_co_so;
Var n,s: longint;
Function D10_CS(n:longint;s:byte):string;
Var CS: array[0..100] of char;
    i: integer;
    ch:Char;
    Tam:string;
Begin
    {Khoi tao cac chu so 0 den 9}
    i:=0;
    ch:='0';
    while i<=9 do
    Begin
        CS[i]:=Ch;
        inc(i);
        inc(ch);
    End;
```

```
End;
{Khoi tao cac chu so A den Z}
i:=10;
ch:='A';
While ch<'Z' do
Begin
  CS[i]:=ch;
  inc(i);
  inc(ch);
End;
tam:="";
While n<>0 do
Begin
  tam:= CS[n mod s]+ Tam;
  n:=n div s;
End;
D10_CS:=Tam;
End;

Begin
  Write('Nhap n:');Readln(n);
  Write('Doi sang co so: ');Readln(s);
  Write(D10_CS(n,s));
  Readln
End.
```

Bài tập 8.4:

Viết chương trình cho phép đổi một số từ cơ số bất kỳ sang cơ số 10.

Thuật toán:

- Dùng mảng CS1 để lưu giá trị của các chữ số <10 và mảng CS2 để lưu giá trị của các chữ số lớn hơn 10.
- Lặp lại cho đến khi n='' việc: Lấy chữ số đầu tiên của n làm chỉ số để xác định giá trị của chữ số, Gán Tong:=Tong*Cơ số + Giá trị tương ứng của chữ số; Xóa chữ số đầu tiên của n.

```
Program Doi_co_so;
Var n: String;
    s:byte;
Function DCS_10(st:String;s:byte):longint;
Var CS1: array['0'..'9'] of byte;
    CS2: array['A'..'Z'] of byte;
    ch:Char;
    i:byte;
    Tam:longint;
Begin
```

```

i:=0;
ch:='0';
while i<=9 do
Begin
    CS1[ch]:=i;
    inc(i);
    inc(ch);
End;
i:=10;
ch:='A';
While ch<'Z' do
Begin
    CS2[ch]:=i;
    inc(i);
    inc(ch);
End;
Tam:=0;
While st<>' ' do
Begin
ch:=st[1];
if (ch>='0') and (ch<='9') then Tam:=Tam*s+CS1[ch]
else Tam:=Tam*s+CS2[ch];
Delete(st,1,1);
End;
DCS_10:=Tam;
End;

Begin
Write('Nhap n:');Readln(n);
Write('Co so cua so vua nhap: ');Readln(s);
Write(DCS_10(n,s));
Readln
End.

```

Bài tập 8.5: (Vĩnh Phúc 2009-2010)

Năm 1973, nhà Toán học Neil Sloan đưa ra khái niệm độ bền của một số nguyên không âm N như sau:

- Nếu N có một chữ số thì độ bền của N bằng 0.
- Nếu N có từ 2 chữ số trở lên thì độ bền của N bằng độ bền của số nguyên là tích các chữ số của N cộng 1.

Cho N , tìm số bé hơn N có độ bền lớn nhất ($0 \leq N \leq 2.000.000.000$).

Ví dụ

Persist	persist	Giải thích
.inp	.out	

100	77	Doben (77) = Doben (49) + 1 = Doben (36) + 1 + 1 = Doben (18) + 1 + 1 + 1 = Doben (8) + 1 + 1 + 1 + 1 = 0 + 1 + 1 + 1 + 1 = 4
-----	----	---

Giải thuật:

- Để tìm độ bền một số cần một hàm tính tích các chữ số của n TICH(n).
- Cho d = 0. Lặp lại điều kiện n > 9 việc: Tăng d lên 1 thay n = TICH(n).

```

Program Do_ben;
uses crt;
Var n,d,i,j,max:longint;
Function TICH(n:Longint):Longint;
Var tam:integer;
Begin
    if n=0 then tam:=0 else tam:=1;
    While n<> 0 do
        Begin
            Tam:=tam*(n mod 10);
            n:=n div 10;
        End;
    TICH:=Tam;
End;
Begin
    clrscr;

    Write('Nhap n: ');Readln(n);
    Max:=0;
    For i:=1 to n do
        Begin
            d:=0;
            j:=i;
            gotoxy(1,2);
            Write('Dang duyet den so: ',i);
            While j>9 do
                Begin
                    d:=d+1;
                    j:=TICH(j);
                End;
            if d>=Max then
                Begin
                    max:=d;
                    gotoxy(1,3);
                    Writeln('So co do ben lon nhat dang la ',i,' do
ben la: ',d);
                End;
            End;
            Gotoxy(1,4);

```



```
Writeln('Da duyet xong');  
Readln  
End.
```

CHƯƠNG X CHUYÊN ĐỀ ĐA THỨC (Một biến).

A. LÝ THUYẾT:

- Để lưu trữ một đa thức ta chỉ cần lưu các hệ số (bằng một mảng) và bậc của đa thức.
- Các phép toán đối với một đa thức gồm cộng, trừ, nhân, chia hai đa thức.
- Việc tính giá trị đa thức tại một giá trị của biến được thực hiện thông qua lược đồ hooc nơ hoặc tính trực tiếp.

B. BÀI TẬP:

Bài tập 10.1:

Viết chương trình cho phép cộng hai đa thức. Kết quả in ra dạng:

$$a_n x^n + \dots + a_1 x + a_0$$

Giải thuật:

- Dùng hai mảng A, B để lưu hệ số của hai đa thức. Có thể dùng mảng thứ ba C hoặc dùng lại một trong hai mảng A, B để lưu hệ số của đa thức tổng.
- Khi in kết quả cần kiểm tra hệ số để in dấu cho đúng.

Cài đặt:

```
Program Cong_da_thuc;
uses crt;
Type KM = array[0..10] of integer;
Var A,B: KM;
    n: byte;
Procedure NhapDT(Var A: KM; n:byte);
Var i: byte;
Begin
    For i:=n downto 0 do
        Begin Write('M[' ,i, ']='); Read(A[i]); End;
End;
Procedure CONG(Var A:KM;B:KM;n:byte);
Var i: byte;
Begin
    For i:= 0 to n do A[i]:=A[i]+B[i];
End;
Procedure INDT(A:KM;n:byte);
Var i: byte;
Begin
    if A[n] <> 0 then Write(A[n], 'x^', n);
    For i:=n-1 downto 0 do
        if A[i] <> 0 then
            if A[i] < 0 then write(A[i], 'x^', i) else
Write('+', A[i], 'x^', i);
End;
Begin
    clrscr;
    Write('Nhap bac n: ');Readln(n);
```

```
Writeln('Nhap da thuc A: '); NhapDT(A,n);
Writeln('Nhap da thuc B: '); NhapDT(B,n);
clrscr;
Indt(A,n); Writeln;
Indt(B,n); Writeln;
Cong(A,B,n);
Writeln('Da thuc tong: ');
Indt(A,n);
Readln;
Readln
```

End.

Nhận xét:

Bài tập 10.2:

Viết chương trình cho phép tính giá trị của đa thức $A(x)$ tại giá trị $x = x_0$.

Giải thuật:

- Dùng chương trình con để tính x^n .

Cài đặt:

```
Program Gia_tri_cua_da_thuc;
Var n,i: integer;
    x,a,s:real;
Function XMU(x:real; n:integer):real;
Var i: integer;
    Mu: real;
Begin
    Mu:=1;
    For i:=1 to n do Mu:=Mu*x;
    XMU:=Mu;
End;
Begin
    Write('Nhap bac cua da thuc n = '); Readln(n);
    Write('Tinh f(x) tai x = ');Readln(x);
    S:=0;
    For i:=n downto 0 do
    Begin
        Write('a',i,'= '); Readln(a);
        S:=S+a*XMU(x,i);
    End;
    Writeln('F(',x:5:2,')= ',S:5:2);
    Readln;
End.
```

Nhận xét:**Bài tập 10.3:**

Viết chương trình tính giá trị của đa thức $A(x)$ với điều kiện không dùng biến mảng.

Viết chương trình tính giá trị của đa thức $f(x) = a_n x^n + \dots + a_1 x + a_0$ tại $x = x_0$.

Thuật toán: $f(x) = a_n x^n + \dots + a_1 x + a_0$

$$= a_0 + x(a_1 + x(a_2 + \dots x(a_{n-1} + x(a_n)) \dots))$$

Lặp lại việc: Đọc và cộng hệ số rồi nhân với x từ hệ số a_n đến hệ số a_0 .

Chương trình:

```
Program Gia_tri_da_thuc;
Var S,a,x: real;
    n,i: byte;
Begin
  Write('Nhap bac cua da thuc. n = '); Readln(n);
  Write('Tinh f(x) tai x = '); Readln(x);
  S:=0;
  For i:= n downto 0 do
  Begin
    Write('a',i,'='); Readln(a);
    S:=(S+a)*x;
  End;
  Write('f(',x:3:1,')=',S:5:1);
  Readln
End.
```

Nhận xét:**Bài tập 10.4:**

Viết chương trình cho phép nhân hai đa thức đã sắp xếp.

Thuật toán: Tích đa thức A bậc m với đa thức B bậc n là đa thức C bậc $m + n$. Trong đó

: $C[m] = \text{Tổng của các tích } A[i] * A[j] \text{ sao cho } i+j = m$.

```
Program Nhan_da_thuc;
Var m,n,i,j:byte;
    A, B, C: array[0..10] of real;
Begin
  Write(' Nhap bac cua da thuc A: ');
  Readln(m);
  For i:= m downto 0 do
  Begin
    Write('A[' ,i,']= '); Readln(A[i]);
  End;
  Write(' Nhap bac cua da thuc B: ');
  Readln(n);
  For i:= n downto 0 do
  Begin
    Write('B[' ,i,']= '); Readln(B[i]);
```

```
End;  
For i:=0 to m+n do C[i]:=0;  
For i:=0 to m do  
  For j:=0 to n do C[i+j]:=C[i+j]+A[i]*B[j];  
  For i:= m+n downto 0 do if C[i]<>0 then  
Write(C[i]:3:1,'x^',i,' + ');  
  Readln  
End.
```

Nhận xét:

Bài tập 10.5:

Viết chương trình cho phép chia hai đa thức để tìm đa thức thương và đa thức dư.

Giải thuật:

Cài đặt:

```
Program Chia_da_thuc;  
uses crt;  
Type KM = array[0..10] of real;  
Var A,B,C,AB,D: KM;  
    i,n,cs,bc: byte;  
Procedure NhapDT(Var A: KM; n:byte);  
Var i: byte;  
Begin  
  For i:=n downto 0 do  
    Begin Write('M[',i,']='); Read(A[i]); End;  
End;  
Function BAC(A:KM;n:integer):Byte;  
Var i:Byte;  
Begin  
  i:=n;  
  While (A[i]=0) and (i> 0) do i:=i-1;  
  BAC:=i;  
End;  
  
Procedure TRU(Var A:KM;B:KM;n:byte);  
Var i: byte;  
Begin  
  For i:= 0 to n do A[i]:=A[i]-B[i];  
End;  
Procedure NHAN(Var C:KM;A,B:KM; Var n:byte);  
Var i,j:byte;  
    TAM:KM;  
Begin  
  n:=Bac(A,n)+Bac(B,n);  
  For i:= 0 to 2*n do TAM[i]:=0;  
  For i:= 0 to n do  
    For j:=0 to n do TAM[i+j]:=TAM[i+j]+A[i]*B[j];  
  For i:=0 to 2*n do C[i]:=TAM[i];
```

```
End;
Procedure INDT(A:KM;n:byte);
Var i: byte;
Begin
  if A[n] <> 0 then Write(A[n]:3:1,'x^',n);
  For i:=n-1 downto 0 do
    if A[i] <> 0 then
      if A[i] < 0 then write(A[i]:3:1,'x^',i) else
Write('+',A[i]:3:1,'x^',i);
End;
Procedure DT_BAC(Var A:KM;Bac:byte;gt:real);
Var i:byte;
Begin
  For i:=1 to n do A[i]:=0;
  A[Bac]:=gt;
End;
Begin
  clrscr;
  Write('Nhap bac cua da thuc bi chia: ');Readln(n);
  Writeln('Nhap da thuc A: '); NhapDT(A,n);
  Writeln('Nhap da thuc B: '); NhapDT(B,n);
  clrscr;
  Write('Da thuc bi chia: ');Indt(A,n); Writeln;
  Write('Da thuc chia: ');Indt(B,n); Writeln;
  bc:=Bac(A,n)-Bac(B,n); {Luu bac cua da thuc C}
  For i:=1 to bc do C[i]:=0;
  While BAC(A,n) >= BAC(B,n) do
  Begin
    cs:=BAC(A,n)-BAC(B,n);
    C[cs]:=A[Bac(A,n)]/B[Bac(B,n)];
    Writeln('C[' ,cs, ']=' ,C[cs]:3:1);
    DT_BAC(D,cs,C[cs]);Writeln;
    Write('Da thuc D:');
    Indt(D,n);
    NHAN(AB,B,D,n);Writeln;
    Write('Da thuc AB:');
    Indt(AB,n);Writeln;
    TRU(A,AB,n);
    Write('Da thuc A moi:');
    Indt(A,n);
  End;
  Writeln('Da thuc thuong: ');
  Indt(C,bc);
  Readln;
  Readln
End.
```

Nhận xét:
Bài tập 10.7:

Giải thuật:
Cài đặt:

Nhận xét:
Bài tập 10.8:

Giải thuật:
Cài đặt:

Nhận xét:
Bài tập 10.9:

Giải thuật:
Cài đặt:

Nhận xét:
Bài tập 10.10:

Giải thuật:
Cài đặt:

Nhận xét: