Chuyên đề 1. NGÔN NGỮ LẬP TRÌNH

I. TÓM TẮT LÝ THUYẾT

1. Khái niệm

1.1. Bài toán (problem)

Trong toán học: Là vấn đề ta muốn giải quyết.

Trong tin học: Là vấn đề ta muốn **máy tính** giải quyết.

1.2. Đầu vào – đầu ra

Đầu vào của bài toán (Input): dữ liệu cần cung cấp cho máy tính

Đầu ra của bài toán (Output): là kết quả máy tính thực hiện được.

1.3. Thuật toán

Thuật toán: là một dãy các bước hữu hạn sao cho khi máy tính thực hiện tuần tự các bước đó thì từ **input** ta tìm được **output** của bài toán

2. Mô tả thuật thoán

Có 3 cách để mô tả thuật toán:

- *Liệt kê* : sử dụng ngôn ngữ tự nhiên để mô tả
- *Sơ đồ khối*: sử dụng hình vẽ
- *Ngôn ngữ lập trình*: sử dụng ngôn ngữ lập trình như Pascal, C, Java, Python,..

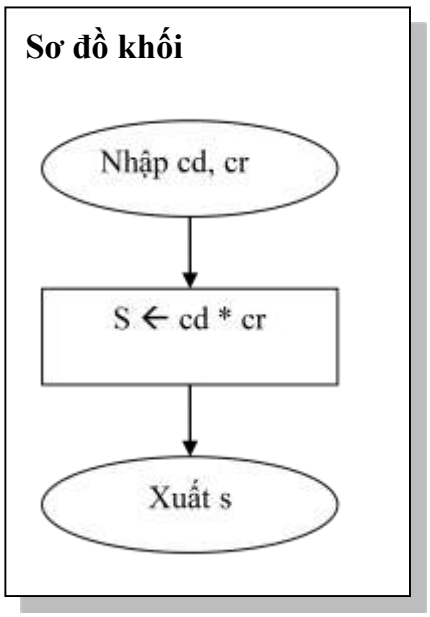
Liệt kê

B1: Nhập cd, cr

B2: s ← cd * cr

B3: Xuất s

Sơ đồ khối



Ngôn ngữ lập trình

```
Readln(cd, cr);
S:=cd*cr;
Write(S);
```

3. Sử dụng Free Pascal

Free Pascal: là môi trường để lập trình bằng **NNLT Pascal**, hỗ trợ soạn thảo, biên dịch, thông dịch chương trình.

Download bản cài trên Windows :

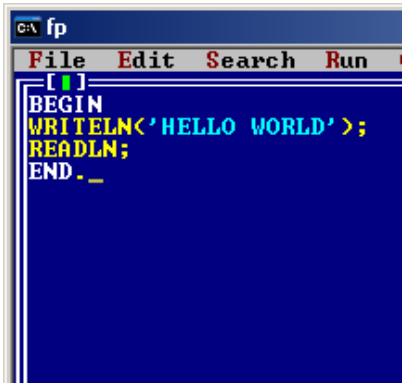
B1: Chạy chương trình Free Pascal:

B2: Mở cửa sổ soạn thảo mới: **File → New**

B3: Soạn thảo chương trình

B4: lưu lại chương trình **FILE → SAVE AS**

B5: chạy thử chương trình **RUN → RUN (CTRL + F9)**



4. Các bước cơ bản khi lập một chương trình bằng Pascal

Bước 1: Soạn thảo chương trình.

Bước 2: Dịch chương trình (nhấn phím **F9**), nếu có lỗi thì phải sửa lỗi.

Bước 3: Chạy chương trình (nhấn phím **Ctrl-F9**).

5. Các thành phần cơ bản của ngôn ngữ Pascal

- Từ khóa
- Tên (định danh)
- Dấu chấm phẩy (;)
- Lời giải thích

II. VÍ DỤ

Ví dụ 1. Xuất ra màn hình câu "Hello World!"

```
BEGIN
      Write('Hello World!');
END.
```

Ví dụ 2. Tính diện tích hình tròn

```
Program  Vidu2;
Const  PI=3.14;
Var  R,S:Real;
Begin
      R:=10;                    {Bán kính đường tròn}
      S:=R*R*PI;  {Diện tích hình tròn}
      Writeln('Dien tich hinh tron = ', S:0:2);  { In ra màn hình }
      Readln;
End.
```

III. BÀI TẬP

Bài 1. Cơ bản

1. Khởi động Pascal.

2. Nhập vào đoạn chương trình sau:

```
Uses Crt;
Begin
Writeln('*************************************************************');
Writeln('*  CHUONG TRINH PASCAL DAU TIEN CUA TOI                      *');
Writeln('*                  Oi! Tuyet voi!...                         *');
Writeln('*************************************************************');
Readln;
End.
```

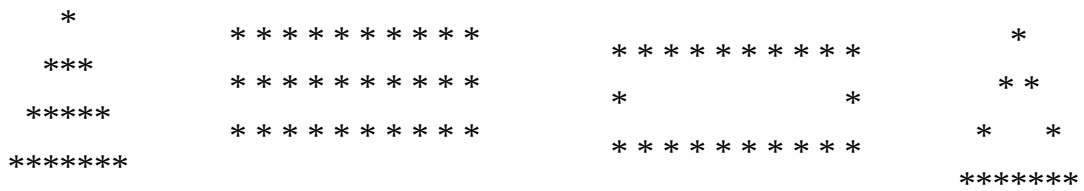
3. Dịch và chạy chương trình trên.

4. Lưu chương trình vào đĩa với tên BAI1.PAS.

5. Thoát khỏi Pascal.
6. Khởi động lại Pascal.
7. Mở file BAI1.PAS.
8. Chèn thêm vào dòng: **CLRSCR;** vào sau dòng **BEGIN**
9. Dịch và chạy thử chương trình.
10. Lưu chương trình vào đĩa.
11. Thoát khỏi Pascal.

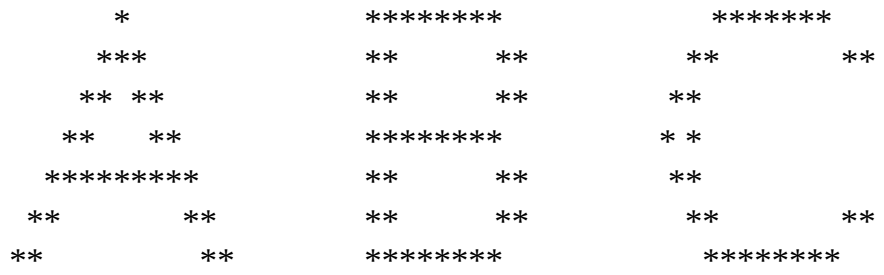
Bài 2. Xuất hình

In ra màn hình những hình sau đây



Bài 3. Xuất chữ

Viết chương trình in ra màn hình các hình sau:



Chuyên đề 2. CẤU TRÚC CHƯƠNG TRÌNH

I. TÓM TẮT LÝ THUYẾT

1. Cấu trúc của một chương trình

```
{ Phần tiêu đề }
PROGRAM Tên_chương_trình;
{ Phần khai báo }
USES      ......;
CONST  .....;
TYPE      .......;
VAR        ........;
PROCEDURE  ............;
  { Phần thân chương trình con }
End;
FUNCTION ..............;
..............
{ Phần thân chương trình con }
End;
BEGIN
{ Phần thân chương trình chính }
END.
```

2. Khai báo hằng

Hằng là một đại lượng có giá trị không thay đổi trong suốt quá trình thực hiện của chương trình chứa nó.

Cú pháp:

```
CONST <Tên hằng> = <Giá trị>;
```

hoặc:

```
CONST <Tên hằng>:= <Biểu thức hằng>;
```

3. 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>;
```

4. Các kiểu dữ liệu cơ bản

4.1. 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**.

4.2. Kiểu số nguyên

a. 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

b. Các phép toán trên kiểu số nguyên

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).

4.3. Kiểu số thực

Các kiểu số thực:

Tên kiểu	Phạm vi	Dung lượng
Single	$1.5\times10^{-45} \rightarrow 3.4\times10^{+38}$	4 byte
Real	$2.9\times10^{-39} \rightarrow 1.7\times10^{+38}$	6 byte
Double	$5.0\times10^{-324} \rightarrow 1.7\times10^{+308}$	8 byte
Extended	$3.4\times10^{-4932} \rightarrow 1.1\times10^{+4932}$	10 byte

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

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

Một số hàm số học thông dụng

SQR(x): Trả về x^2
SQRT(x): Trả về căn bậc hai của x (x≥0)
ABS(x): Trả về |x|
SIN(x): Trả về sin(x) theo radian
COS(x): Trả về cos(x) theo radian
LN(x): Trả về ln(x)
EXP(x): Trả về e^x
TRUNC(x): Trả về số nguyên gần với x nhất nhưng bé hơn x.
ROUND(x): Làm tròn số nguyên x
INC(n): Tăng n thêm 1 đơn vị (n:=n+1).
DEC(n): Giảm n đi 1 đơn vị (n:=n-1).

5. 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 +sin(y))/(5-2*x) biểu thức **số học**

(x+4)*2 = (8+y) biểu thức **quan hệ**

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

1. Lời gọi hàm.
2. Dấu ngoặc ()
3. Phép toán một ngôi (NOT, -).
4. Phép toán *, /, DIV, MOD, AND.

5. Phép toán +, -, OR, XOR
6. Phép toán (quan hệ) so sánh =, <, >, <=, >=, <>, IN

6. Câu lệnh cơ bản

6.1. Câu lệnh gán

<Tên biến>:=<Biểu thức>;

6.2. Câu 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;
```

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');
Writeln('Hello':10);
Writeln(500);
Writeln(500:5);
Writeln(123.457)
Writeln(123.45:8:2)
	Hello
	     Hello
	500
	  500
	1.2345700000E+02
	  123.46

6.3. Nhập dữ liệu

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

7. Nhập xuất từ tệp

7.1. Khai báo biến tệp

```
var TênBiếnTệp:text;
```

7.2. Gán tên tệp

Sau khi khai báo, ta phải gán tên tệp cho từng biến. Sau khi gán, biến sẽ đại diện cho tệp và ta chỉ làm việc trên biến tệp.

```
Assign(BiếnTệp, TênTệp);
```

7.3. Đọc dữ liệu từ tệp

Ta gọi thủ tục "Mở tệp để đọc":

```
Reset(BiếnTệp);
```

Sau đó "đọc từng giá trị" bằng thủ tục

```
Read(BiếnTệp, DanhSáchBiến);
```

Ngoài cách "đọc từng giá trị", ta có thể "đọc từng dòng trong tệp" bằng thủ tục

```
Cú pháp: Readln(BiếnTệp, Danh sách Biến);
```

Thủ tục này đọc tất cả các giá trị trên dòng hiện tại, sau đó con trỏ sẽ nhảy xuống dòng kế tiếp.

7.4. Ghi dữ liệu từ tệp.

Mở tệp để ghi:

```
Rewrite(BienTep);
```

Ghi dữ liệu ra tệp :

```
Write (BienTep, GiaTri1, GT2, …)
```

hoặc

```
Writeln(BienTep, GiaTri1, GT2, …)
```

7.5. Đóng tệp

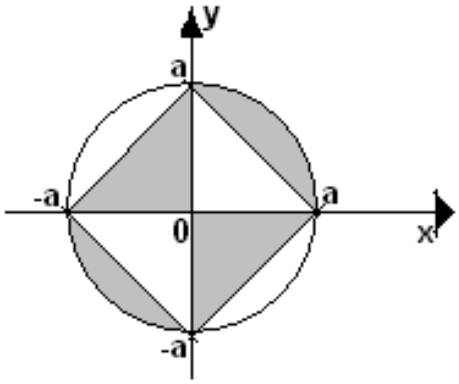
Trước khi kết thúc chương trình cần phải đóng tệp.

```
Cú pháp: Close(<BienTep>);
```

II.VÍ DỤ

Ví dụ 1. Diện tích tô đen

Tính và đưa ra màn hình diện tích phần tô đen trong hình sau.



Input: số thực a.

Output: diện tích phần tô đen S.

Ý tưởng

Diện tích phần tô đen = ½ diện tích hình tròn bán kính a.

Chương trình

```
Program To_den;
Const pi=3.14;
Var
      a:word;
      S, S1:real;
Begin
Write('Nhap a = ');
Readln(a);
S1:=pi*a*a;
S:=S1/2;
Writeln('Dien tich to den la:',S:0:2);
Readln;
```

```
End.
```

Ví dụ 2. Diện tích tam giác DTTG

Viết chương trình nhập vào độ dài hai cạnh của tam giác và góc giữa hai cạnh đó, sau đó tính và in ra màn hình diện tích của tam giác.

Ý tưởng:

Công thức tính diện tích tam giác: $S = \frac{1}{2}a.b.\sin(\theta)$ với a,b là độ dài 2 cạnh và θ là góc kẹp giữa 2 cạnh a và b.

Chương trình

```
Program Tinh_dien_tich_tam_giac;
Var a,b,goc,dientich: Real;
Begin
      Write('Nhap vao do dai canh thu nhat: '); Readln(a);
      Write('Nhap vao do dai canh thu hai: '); Readln(b);
      Write('Nhap vao goc giua hai canh: '); Readln(goc);
      Dientich:=a*b*sin(goc)/2;
      Writeln('Dien tich cua tam giac la: ',Dientich:0:2);
      Readln;
End.
```

Ví dụ 3. Hoán đổi HOANDOI

Viết chương trình nhập vào 2 số a, b. Sau đó hoán đổi giá trị của 2 số đó:

Chương trình 1: Dùng biến trung gian

```
Program Swap;
Var   a,b,tam: Integer;
Begin
      Write('Nhap vao a= '); Readln(a);
      Write('Nhap vao b= '); Readln(b);
      tam:=a;     {tam lấy giá trị của a}
      a:=b;             {a lấy giá trị của b}
      b:=tam;     {b lấy lại giá trị của tam}
      Writeln('a = ',a,' b = ',b);
      Readln;
End.
```

Chương trình 2: Không dùng biến trung gian.

```
Program Swap;
Var   a,b: Integer;
Begin
      Write('Nhap vao a= '); Readln(a);
      Write('Nhap vao b= '); Readln(b);
      a:=a+b;     {a lấy tổng giá trị của a+b}
      b:=a-b;     {b lấy giá trị của a}
      a:=a-b;           {a lấy lại giá trị của b}
      Writeln('a = ',a,' b = ',b);
      Readln;
End.
```

Ví dụ 4. Tính diện tích hình chữ nhật HCN

Tính diện tích hình chữ nhật, yêu cầu nhập xuất dữ liệu từ tệp.

Chương trình

```
Program KieuTep;
Const
      finame='hcn.inp';
      foname='hcn.out';
```

```
Var
      fi, fo:text;
      Cd, cr, S: real;
BEGIN
Assign(fi, finame); Reset(fi);
Assign(fo, foname); Rewrite(fo);
Readln(fi, cd, cr);
S:=cd*cr;
Writeln(fo,'Dien tich HCN: ',S:0:2);
Close(fi);
Close(fo);
END.
```

III. BÀI TẬP

Bài 1. Tính biểu thức BIEUTHUC

Viết chương trình nhập vào các số nguyên: a, b, x, y, ... sau đó in ra màn hình kết quả của các biểu thức sau:

a/ $\dfrac{x+y}{2+\dfrac{x}{y}}$ b/ $\dfrac{(a+4)(b-2c+3)}{\dfrac{r}{2h}-9(a-1)}$ c/ x^y , x>0 d/ $e^{\sqrt{|a+\sin^2(x)-x|}}$

Chuyên đề 3. CẤU TRÚC RẼ NHÁNH

I. TÓM TẮT LÝ THUYẾT

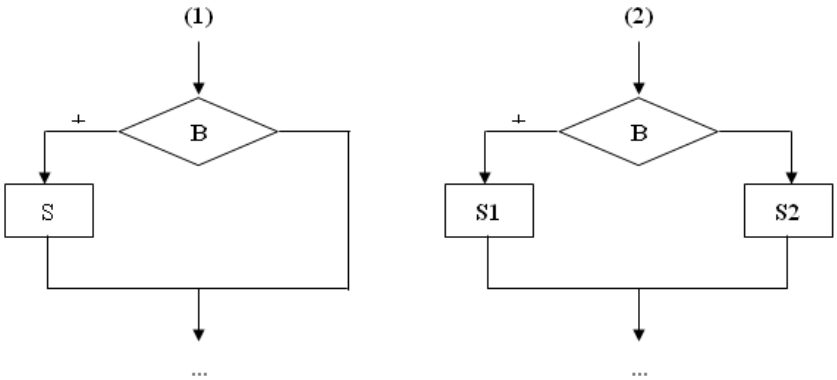
1. Lệnh IF

Cú pháp:

(1) IF B THEN S;

(2) IF B THEN S1 ELSE S2;

Sơ đồ thực hiện:



***Chú ý**: Khi sử dụng câu lệnh IF thì đứng trước từ khoá ELSE không được có dấu chấm phẩy (;).*

2. Lệnh CASE

Cú pháp:

Dạng 1	Dạng 2
CASE B OF Const 1: S_1; Const 2: S_2; ...	CASE B OF Const 1: S_1; Const 2: S_2; ...

Const n: S_n; END;	Const n: S_n; ELSE S_{n+1}; END;

Trong đó:

- **B**: Biểu thức kiểu vô hướng đếm được như kiểu nguyên, kiểu logic, kiểu ký tự, kiểu liệt kê.
- **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}.

II.VÍ DỤ

Ví dụ 1. Kiểm tra chẵn lẻ CHANLE

Viết chương trình nhập vào một số nguyên và kiểm tra xem số vừa nhập là số chẵn hay số lẻ.

```
Uses crt;
Var  x:integer;
Begin
  Write('Nhap vao mot so nguyen : ');  Readln(x);
      If x MOD 2=0 Then
            Writeln('So vua nhap vao la so chan')
      Else
            Writeln('So vua nhap vao la so le');
      Readln;
End.
```

Ví dụ 2. Giải phương trình bậc nhất PTBI

Viết chương trình giải phương trình bậc nhất ax+b=0

```
Uses Crt;
Var  a,b,x : real;
Begin
      Write('a = '); Readln(a);
      Write('b = '); Readln(b);
      If a = 0 Then      { Nếu a bằng 0 }
            If b = 0 Then   { Trường hợp a = 0 và b = 0 }
                  Writeln('Phuong trinh co vo so nghiem')
            Else    { Trường hợp a=0 và b ≠ 0 }
                  Writeln('Phuong trinh vo nghiem')
      Else  { Trường hợp a ≠ 0 }
            Begin
                  x:= -b/a;
                  Writeln('Phuong trinh co nghiem la :',x:0:2);
            End;
      Readln;
End.
```

Ví dụ 3. Độ tuổi DOTUOI

Viết chương trình nhập vào tuổi của một người và cho biết người đó là thiếu niên, thanh niên, trung niên hay lão niên. Biết rằng: nếu tuổi nhỏ hơn 18 là thiếu niên, từ 18 đến 39 là thanh niên, từ 40 đến 60 là trung niên và lớn hơn 60 là lão niên.

```
Uses crt;
Var tuoi:Byte;
Begin
      Write(Nhap vao tuoi cua mot nguoi:');     Readln(tuoi);
      Case tuoi Of
            1..17:      Writeln(Nguoi nay la thieu nien');
            18..39:     Writeln(Nguoi nay la thanh nien');
            40..60:     Writeln(Nguoi nay la trung nien');
            Else        Writeln(Nguoi nay la lao nien');
      End;
      Readln;
End.
```

Chuyên đề 4. CẤU TRÚC LẶP

I. TÓM TẮT LÝ THUYẾT

1. Vòng lặp xác định

Có hai dạng sau:

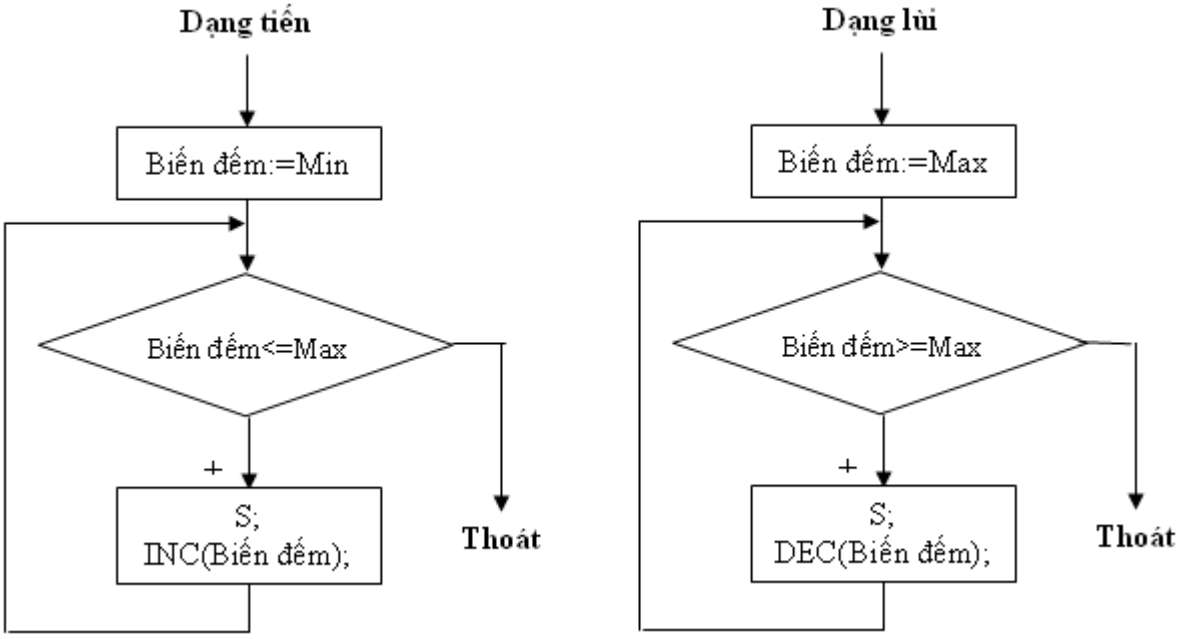
❶ ***Dạng tiến***

```
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 tuỳ 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.

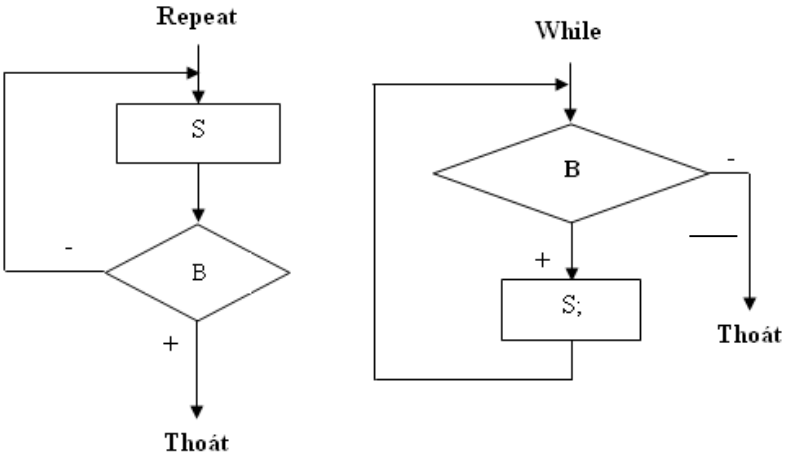
2. Vòng lặp không xác định

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.

Dạng WHILE: Trong khi biểu thức B=TRUE thì tiếp tục thực hiện công việc S.



II.VÍ DỤ

Ví dụ 1. Tính tổng TONG

Viết chương trình tính tổng S = 1+2+...+N

Chương trình 1 Dùng vòng lặp FOR.

```
Program TinhTong;
Uses crt;
Var N,i,S:integer;
Begin
      Clrscr;
      Write('Nhap vao gia tri cua N :'); Readln(N);
      S:=0;
      For i:=1 to N do S:=S+i;
      Writeln('Ket qua la :',S);
      Readln;
End.
```

Chương trình 2: Dùng vòng lặp REPEAT.

```
Program TinhTong;
Uses crt;
Var N,i,S:integer;
Begin
      Clrscr;
      Write('Nhap vao gia tri cua N :'); Readln(N);
      S:=0; i:=1;
      Repeat
            S:=S+i;
            i:=i+1;
      Until i>N;
      Writeln('Ket qua la :',S);
```

```
      Readln;
End.
```

Chương trình 3: Dùng vòng lặp WHILE.

```
Program TinhTong;
Uses crt;
Var N,i,S:integer;
Begin
      Clrscr;
      Write('Nhap vao gia tri cua N :'); Readln(N);
      S:=0; i:=1;
      While i<=N Do
            Begin
                  S:=S+i;
                  i:=i+1;
            End;
      Writeln('Ket qua la :',S);
      Readln;
End.
```

Ví dụ 2. Tổng các số TONGSO

Viết chương trình nhập vào N số nguyên từ bàn phím. Hãy tính và in ra màn hình tổng của các số vừa được nhập vào.

Ý tưởng

Dùng phương pháp cộng dồn. Cho vòng lặp FOR chạy từ 1 tới N, ứng với lần lặp thứ i, ta nhập vào số nguyên X và đồng thời cộng dồn X vào biến S.

Chương trình

```
Program  Tong;
Uses crt;
Var N,S,i,X : Integer;
Begin
      Clrscr; S:=0;
      For i:=1 To n Do
            Begin
                  Write('Nhap so nguyen X= '); Readln(X);
                  S:=S+X;
            End;
      Writeln('Tong cac so duoc nhap vao la: ',S);
      Readln;
End.
```

Ví dụ 3. Đếm số chẵn SOCHAN

Viết chương trình nhập vào các số nguyên cho đến khi nào gặp số 0 thì kết thúc. Hãy đếm xem có bao nhiêu số chẵn vừa được nhập vào.

Ý tưởng:

Bài toán này không biết chính xác số lần lặp nên ta không thể dùng vòng lặp FOR. Vì phải nhập vào số nguyên N trước, sau đó mới kiểm tra xem N=0? Do đó ta nên dùng vòng lặp REPEAT.

Chương trình

```
Program  Nhapso;
Uses crt;
Var N,dem : Integer;
Begin
      Clrscr; dem:=0;
      Repeat
            Write('Nhap vao mot so nguyen N= '); Readln(N);
            If N MOD 2 = 0 Then dem:=dem+1;
```

```
      Until N=0;
      Writeln('Cac so chan duoc nhap vao la: ',dem);
      Readln;
End.
```

Ví dụ 4. Tính số Pi SOPI

Viết chương trình tính số Pi với độ chính xác Epsilon, biết:

$$Pi/4 = 1-1/3+1/5-1/7+...$$

Ý tưởng:

Ta thấy rằng, mẫu số là các số lẻ có qui luật: 2*i+1 với i=1,...,n. Do đó ta dùng i làm biến chạy.

Vì tính số Pi với độ chính xác Epsilon nên không biết trước được cụ thể số lần lặp, do đó ta phải dùng vòng lặp WHILE hoặc REPEAT. Có nghĩa là phải lặp cho tới khi t=4/(2*i+1) ≤ Epsilon thì dừng.

Chương trình

```
Uses Crt;
Const Epsilon=1E-4;
Var Pi,t:real;
    i,s:Integer;
Begin
      Pi:=4;  i:=1; s:=-1;
      t:=4/(2*i+1);
      While t>Epsilon Do
            Begin
                  Pi:=Pi+s*t;
                  s:=-s; i:=i+1;
                  t:=4/(2*i+1);
            End;
      Writeln('So Pi = ',Pi:0:4);
      Readln;
End.
```

Ví dụ 5. Ước số UOCSO

Viết chương trình nhập vào số nguyên N. In ra màn hình tất cả các ước số của N.

Ý tưởng: Cho biến i chạy từ 1 tới N. Nếu N MOD i=0 thì viết i ra màn hình.

Chương trình

```
Uses Crt;
Var N,i : Integer;
Begin
      Clrscr;
      Write('Nhap so nguyen N= '); Readln(N);
      For i:=1 To N Do
            If N MOD i=0 Then Write(i:5);
      Readln;
End.
```

Ví dụ 6. Ước chung lớn nhất và bội chung nhỏ nhất UOCBOI

Viết chương trình tìm USCLN và BSCNN của 2 số a, b được nhập vào từ bàn phím.

Ý tưởng:

- Tìm USCLN: Lấy số lớn trừ số nhỏ cho đến khi a=b thì dừng. Lúc đó: USCLN=a.
- BSCNN(a,b) = a*b DIV USCLN(a,b).

Chương trình

```
Uses crt;
Var a,b,aa,bb:integer;
Begin
      Write('Nhap a : '); Readln(a);
      Write('Nhap b : '); Readln(b);
      aa:=a; bb:=b;
      While aa<>bb Do
            Begin
                  If aa>bb Then aa:=aa-bb Else  bb:=bb-aa;
            End;
      Writeln('USCLN= ',aa);
      Writeln('BSCNN= ',a*b DIV aa);
      Readln;
End.
```

Ví dụ 7. Tìm số TIMSO

Viết chương trình tìm các số có 3 chữ số $\overline{abc}$ sao cho: $\overline{abc} = a^3 + b^3 + c^3$.

Ý tưởng:

Dùng phương pháp vét cạn. Ta biết rằng: a có thể có giá trị từ 1→9 (vì a là số hàng trăm), b,c có thể có giá trị từ 0→9. Ta sẽ dùng 3 vòng lặp FOR lồng nhau để duyệt qua tất cả các trường hợp của a,b,c.

Ứng với mỗi bộ abc, ta sẽ kiểm tra: Nếu $100.a + 10.b + c = a^3 + b^3 + c^3$ thì in ra bộ abc đó.

Chương trình

```
Uses crt;
Var a,b,c : Word;
Begin
      For a:=1 To 9 Do
            For b:=0 To 9 Do
                  For c:=0 To 9 Do
                        If  (100*a  +  10*b  +  c)=(a*a*a  +  b*b*b  +  c*c*c)  Then
      Writeln(a,b,c);
      Readln;
End.
```

Ví dụ 8. Kiểm tra nguyên tố NGUYENTO

Viết chương trình nhập vào số tự nhiên N rồi thông báo lên màn hình số đó có phải là số nguyên tố hay không.

Ý tưởng

N là số nguyên tố nếu N không có ước số nào từ 2 → N div 2. Từ định nghĩa này ta đưa ra giải thuật:

- Đếm số ước số của N từ 2 → N div 2 lưu vào biến d.
- Nếu d=0 thì N là số nguyên tố.

Chương trình

```
Uses crt;
Var N,i,d : Word;
Begin
      If N<2 Then Writeln(N,' khong phai la so nguyen to')
      Else
      Begin
            {Đếm số ước số}
            d:=0;
            For i:=2 To N div 2 Do
                  If N MOD i=0 Then d:=d+1;
```

```
            {Kiểm tra}
            If d=0 Then Writeln(N,' la so nguyen to')
            Else Writeln(N,' khong phai la so nguyen to');
      End;
      Readln;
End.
```

Ví dụ 9. Ghép 2 tệp GHEPTEP

Cho 2 tệp văn bản NGUYEN1.INP và NGUYEN2.INP, mỗi dòng của 2 tệp chứa một số số nguyên. Hãy lập trình tạo tệp văn bản NGUYEN12.OUT, những dòng đầu tiên là các dòng của tệp NGUYEN1.INP, những dòng còn lại là của tệp NGUYEN2.INP

Ví dụ

NGUYEN1.INP	NGUYEN2.INP	NGUYEN12.OUT
1 3 5	2 4	1 3 5 2 4

Ý tưởng

Vì không biết tệp có bao nhiêu giá trị, ta Sử dụng hàm eof kết hợp với câu lệnh while để đọc từng giá trị. Mỗi lần đọc lưu vào biến temp và ghi ngay temp ra tệp NGUYEN12.OUT. Sau khi đọc xong tệp NGUYEN1.INP ta đọc tiếp NGUYEN2.INP.

Chương trình

```
Var
        fi1,fo,fi2:text;
        x:longint;
Begin
Assign(fo,'nguyen12.out');rewrite(fo);
assign(fi1,'nguyen1.inp');reset(fi1); //doc file 1 va xuat ra
while not eof(fi1) do
        begin
        read(fi1,x);
        writeln(fo,x);
        end;
close(fi1);
assign(fi2,'nguyen2.inp');reset(fi2); //doc file 1 va xuat ra
while not eof(fi2) do
        begin
        read(fi2,x);
        writeln(fo,x);
        end;
close(fi2);
close(fo);
end.
```

Chuyên đề 5. MẢNG MỘT CHIỀU

I. TÓM TẮT LÝ THUYẾT

1. Khái niệm

Mảng: là tập hợp các biến cùng kiểu dữ liệu. Mỗi biến là một phần tử và được gắn chỉ số để xác định.

Mảng 1 chiều: kích thước n là mảng có n phần tử được xếp thành một dãy liên tục.

Ví dụ: Mảng 1 chiều có 6 phần tử kiểu số nguyên.

Giá trị	37	29	13	-6	5	4
Chỉ số	**1**	**2**	**3**	**4**	**5**	**6**

2. Khai báo

Cú pháp khai báo trực tiếp:

```
VAR TenBienMang : ARRAY[CSDau..CSCuoi] OF KieuPhanTu;
```

CSDau..CSCuoi dùng để xác định miền chỉ số của mảng, cũng là để xác định số phần tử của mảng. Csdau và cscuoi thường là kiểu số nguyên, csdau thường bắt đầu từ 1.

Ví dụ: khai báo mảng có 7 phần tử để lưu trữ 7 số nguyên

```
VAR t:ARRAY[1..7] OF LongInt;
```

Ngoài cách trên ta còn có cách khai báo gián tiếp như sau:

```
TYPE TenKieuMang = ARRAY[CSDau..CSCuoi] OF KieuPhanTu;
VAR TenBienMang:TenKieuMang;
```

Ví dụ: khai báo mảng trên theo cách gián tiếp

```
TYPE M1C = ARRAY[1..7] OF Longint;
VAR t:M1C;
```

3. Truy xuất phần tử

Muốn truy xuất đến phần tử (biến) nào của mảng, ta cung cấp chỉ số của phần tử đó theo cú pháp

```
BienMang[ChiSo]
```

Ví dụ: Gán và xuất giá trị phần tử thứ ba của mảng trên ra màn hình

```
t[3]:=13;
Writeln('Phan tu thu ba cua mang la: ', t[3]);
```

4. Số ngẫu nhiên

Việc tạo ra những giá trị ngẫu nhiên giúp xây dựng bộ giá trị đầu vào (input) kiểm thử chương trình 1 cách nhanh chóng và khách quan.

- **Thủ tục xáo trộn bộ số ngẫu nhiên** : Randomize;
- **Hàm tạo ra số ngẫu nhiên trong khoảng [0,k)**: Random(k);

Ví dụ

Khởi tạo mảng gồm các số nguyên dương có giá trị ngẫu nhiên nhỏ hơn 50

```
Randomize;
FOR i:=1 TO n DO a[i]:=Random(50);
```

II. VÍ DỤ

Ví dụ 1. Nhiệt độ trong tuần NHIETDO

Nhập nhiệt độ mỗi ngày trong tuần, tính nhiệt độ trung bình của tuần và số lượng ngày trong tuần có nhiệt độ cao hơn nhiệt độ trung bình.

Input

- Gồm 7 dòng, mỗi dòng là nhiệt độ t_i một ngày ($0 \le t_i \le 100$)

Output

- Dòng 1 là nhiệt độ trung bình
- Dòng 2 là số ngày lớn có nhiệt độ lớn hơn trung bình

Test

```
Input
30
31
32
33
34
35
36
Output
33
3
```

Thuật toán

Ta khai báo mảng t[]gồm 7 phần tử kiểu số thực để lưu trữ nhiệt độ cho 7 ngày.

```
t: ARRAY[1..7] OF REAL;
```

Ta đọc dữ liệu 7 lần từ tệp input, mỗi lần đọc một số và lưu vào phần tử tương ứng trong mảng.

```
FOR i:=1 TO 7 DO read(fi, t[i]);
```

Để tính trung bình, ta tính tổng nhiệt độ 7 ngày trước. Ta dùng vòng lặp duyệt qua các phần tử trong mảng và cộng dồn vào biến S.

```
FOR i:=1 TO 7 DO s:=s+t[i];
```

Để điếm số ngày thỏa đề bài, ta cũng dùng vòng lặp duyệt qua các phần tử trong mảng. Phần tử nào thỏa điều kiện thì tăng dem

```
FOR i:=1 TO 7 DO
      IF nhietdo[i] > tb THEN inc(d);
```

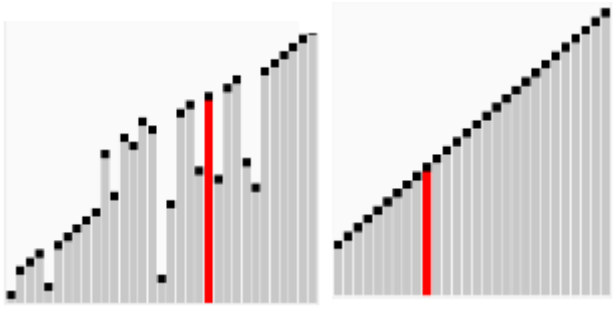
Chương trình

```
PROGRAM nhietdotrungbinh;
CONST
      fin='nhietdo.inp';
      fon='nhietdo.out';

VAR
      t: ARRAY[1..7] OF REAL;
      s, tb: REAL;
      i,d: INTEGER;
      fi, fo: text;
Begin
{Nhap du lieu}
Assign(fi,fin);Reset(fi);
```

```
FOR i:=1 TO 7 DO read(fi, t[i]);
Close(fi);
{Tinh trung binh}
s:=0;
FOR i:=1 TO 7 DO s:=s+t[i];
tb:=s/7;
{Dem so ngay lon hon trung binh}
d:=0;
FOR i:=1 TO 7 DO
      IF nhietdo[i] > tb THEN inc(d);
{Xuat ket qua}
Assign(fo,'Nhietdo.inp');Rewrite(fo);
writeln(fo,tb:0:2);
writeln(fo,d);
end.
```

Ví dụ 2. Sắp xếp mảng SAPXEP



Tạo mảng a ngẫu nhiên có n phần tử số nguyên ($-1000<a_i<1000$). Sắp xếp mảng không giảm.

Input

- Số n ($n <= 10^3$)

Output

- Dòng đầu là mảng a khi chưa sắp xếp
- Dòng 2 là mảng a sau khi đã sắp xếp.

Thuật toán

Khai báo

Dễ thấy phải khai báo a[] có n phần tử, tuy nhiên khi khai báo ta chưa có thông tin về giá trị của n nên không thể biết mảng có bao nhiêu phần tử. Trong trường hợp này, thông thường ta sẽ khai báo số phần tử của mảng là tối đa (theo giới hạn của n). Trong chương trình ta chỉ sử dụng n phần tử đầu tiên.

```
Const
      Fin='SAPXEP.INP';
      Fon='SAPXEP.OUT';
      Maxn = 1000;
Var
      N:longint;
      A:array[1..Maxn] of Longint;
```

Tạo mảng Ngẫu nhiên

Đề bài yêu cầu tạo mảng số ngẫu nhiên chứ không phải đọc từ tệp cho trước. Các số ngẫu nhiên này do máy sinh ra và ta không biết trước giá trị. Việc tạo số ngẫu nhiên giống như trò chơi "Lô tô". Đầu tiên ta phải làm thao tác "xóc túi" để xáo trộn các con số bằng thủ tục xáo trộn bộ số ngẫu nhiên : **Randomize;** . Thao tác này chỉ cần làm một lần trong chương trình.

Sau khi xáo trộn, khi nào cần tạo ra một giá trị ngẫu nhiên ta sử dụng hàm tạo giá trị ngẫu nhiên **Random(k:word): word.** Hàm này sẽ cho ta một giá trị ngẫu nhiên thuộc [0, k).

Để tạo ra mảng gồm n giá trị ngẫu nhiên, ta sẽ thực hiện n lần, mỗi lần tạo một giá trị và lưu vào một phần tử của mảng a.

Để tạo một giá trị ngẫu nhiên từ -1000 .. 1000, ta có nhiều cách. Ví dụ: -1000 + random(2000); hoặc random(1000) – random(1000);

```
Read(fi, n);
Randomize;
For i:=1 to n do a[i]:=-1000+random(2000);
```

Sau khi tạo xong ta có thể xuất mảng a để xem giá trị :

```
For i:=1 to n do write(fo,a[i],' ');
```

Sắp xếp mảng: có nhiều thuật toán khác nhau để sắp xếp mảng tăng dần, ta có thể sử dụng thuật toán "nổi bọt" như sau:

Ý tưởng: so sánh từng cặp 2 phần tử kề nhau, nếu sai vị trí thì đảo lại.

Lần 1: duyệt từ phần từ 1 đến phần tử n-1, nếu a[i]>a[i+1] thì đảo lại. Sau lần này ta được phần tử lớn nhất "nổi" về đúng vị trí (cuối cùng) n.

Lần 2: duyệt từ phần tử 1 đến phần tử n-2, nếu a[i]>a[i+1] thì đảo lại. Sau lần này ta được phần tử lớn nhì"nổi" về đúng vị trí (kế cuối) n-1.

…

Lần n-1: duyệt từ phần tử 1 đến phần tử n - (n-1) nếu a[i]>a[i+1] thì đảo lại. Sau lần này ta được phần tử lớn thứ n-1 "nổi" về đúng vị trí (thứ 2).

```
for i:=1 to n-1 do
      for j:=1 to n-i do
         if a[j]>a[j+1] then
            begin
            tam:=a[j];
            a[j]:=a[j+1];
            a[j+1]:=tam;
            end;
```

Chương trình

```
Program SapXep;
Const
      Fin='SAPXEP.INP';
      Fon='SAPXEP.OUT';
      Maxn = 1000;
Var
N,i,j,tam:longint;
fi,fo:text;
A:array[1..Maxn] of Longint;
Begin
Assign(fi,fin);Reset(fi);
Assign(fo,fon);Rewrite(fo);
{Doc n}
Read(fi, n);
{Tao mang ngau nhien}
Randomize;
For i:=1 to n do a[i]:=-1000+random(2000);
{Xuat mang chua sap xep}
For i:=1 to n do write(fo,a[i],' ');
{Sap xep mang}
for i:=1 to n-1 do
      for j:=1 to n-i do
         if a[j]>a[j+1] then
            begin
            tam:=a[j];
            a[j]:=a[j+1];
```

```
            a[j+1]:=tam;
            end;
{Xuat mang da sap xep}
writeln(fo);
For i:=1 to n do write(fo,a[i],' ');
Close(fo);
Close(fi);
End.
```

Kết quả chạy thử:

Lần 1

```
Input
5
Output
-73 18 399 -788 -2
-788 -73 -2 18 399
```

Lần 2

```
Input
10
Output
731 708 -271 468 105 -512 -875 426 59 -726
-875 -726 -512 -271 59 105 426 468 708 731
```

Ví dụ 3. Lỗ hổng chữ số LOHONG

Lớp của Bờm tuy không phải là lớp chuyên nhưng phong trào học Toán – Tin rất sôi động. Thầy giáo của Bờm rất tâm huyết, luôn tìm các bài toán hay, mới, lạ, đặc biệt là bài toán quy luật để dạy trò. Một hôm, thầy giáo đến lớp thật sớm trước tiết toán, viết lên bảng một dãy có quy luật sau:

42 → 1, 1337 → 0, 669 → 3, 1882 → 4, 688 → 5, 12345 → 1, 123 → 0, 456 → 2, 789 → 3. Và thầy đố cả lớp rằng: 45678 → ?

Bằng một cái đầu rất nhạy bén Toán Tin, nhất là những bài toán Tin logic, Bờm đã kiếm được lời giải trên … Google ngay khi về đến nhà. Chật vật một hồi, cậu đã tìm ra quy luật của bài toán: chuyển số "lỗ hổng" trong các chữ số của số đã cho và biểu diễn chúng (không có chữ số 0 ở đầu). Chữ số 1, 2, 3, 5 và 7 không có lỗ hổng nào; các chữ số 0, 4, 6, 9 có một "lỗ hổng" và đặc biệt chữ số 8 có đến 2 lỗ hổng. Hôm sau Bờm rất tự tin mở rộng bài toán bằng cách thêm vào một vài số nữa đến đố bạn các lớp bên cạnh. Bạn đọc hãy giúp các bạn ấy giải bài này để Bờm không được dịp "kiêu" nhé.

Dữ liệu vào

- Dữ liệu vào được nhập từ bàn phím, gồm một số nguyên duy nhất n (1 <=n <=1000000)

Dữ liệu ra

- In ra màn hình một số nguyên duy nhất là số lỗ hổng của số đã cho.

Ví dụ

Input	Output
42 669 456 45678	1 3 2 4

Thuật toán

Ta khai báo một mảng hằng có 10 phần tử để lưu số lỗ hổng của các chữ số tương ứng từ 0 → 9.

```
Const
lohong:array[0..9] of longint = (1,0,0,0,1,0,1,0,2,1);
```

Dùng các phép toán div, mod để tách từng chữ số của n. Với mỗi chữ số ta tìm số lỗ hổng được lưu trữ trong mảng để tăng biến dem.

```
While n>0 do
      Begin
      dem:=dem+lohong[n mod 10];
      n:=n div 10;
      End;
```

Chương trình

```
Program Bt;
Const
lohong:array[0..9] of longint = (1,0,0,0,1,0,1,0,2,1);
Var
dem,n:longint;
Begin
Write('Nhap n: ');Readln(n);
Write(n,' ');
dem:=0;
While n>0 do
Begin
dem:=dem+lohong[n mod 10];
n:=n div 10;
End;
Write('-> ',dem);
Readln;
End.
```

Chuyên đề 6. MẢNG HAI CHIỀU

I. TÓM TẮT LÝ THUYẾT

1. Khái niệm

Mảng 2 chiều: kích thước m*n là bảng có m hàng, n cột, m* n phần tử.

Ví dụ: Mảng 2 chiều A kích thước 3*5

	1	2	3	4	5
1	2	5	1	3	6
2	5	4	3	6	7
3	6	5	4	6	7

Mỗi phần tử được xác định bởi chỉ số hàng và chỉ số cột. Ví dụ A[2, 3] là phần tử hàng 2 cột 3 có giá trị 3.

2. Khai báo

Cú pháp:

```
VAR TenBien:ARRAY[CSDau..CSCuoi,CSDdau..CSCcuoi] OF KieuDLPhanTu;
```

Ví dụ 1: Khai báo bảng 3 hàng, 5 cột cho lưu trữ các giá trị như bảng trên

```
VAR t:ARRAY[1..3,1..5] OF Byte;
```

Vi du 2: Khai báo mảng m hàng, n cột kiểu số nguyên

Vì chưa biết cụ thể số hàng, số cột nên ta khai báo dư ra.

```
Const
      mmax = 1000;
      Nmax=1000;
Var a:array[1..Mmax,1..Nmax] of longint;
```

3. Truy xuất phần tử

Cú pháp

```
TenBien[CSHang,CSCcot]
```

Ví dụ 1: Tính tổng các phần tử ở cột 2

```
S:=t[1,2]+t[2,2]+t[3,2];
```

Ví dụ 2: Nhập các phần tử của mảng từ bàn phím

```
For i:=1 to 3 do
      For j:=1 to 5 do
            Begin
            Write('a',i, j,' = ');Readln(a[i, j]);
            End;
```

Ví dụ 3: Xuất các phần tử theo dạng bảng

```
For i:=1 to 3 do
      Begin
      For j:=1 to 5 do Write(a[i, j]:4);
      Writeln;
      End;
```

II.VÍ DỤ

Ví dụ 1. Phần tử yên ngựa YENNGUA

Cho bảng A kích thước MxN. Phần tử A_{ij} được gọi là phần tử yên ngựa nếu nó là phần tử nhỏ nhất trong hàng của nó đồng thời là phần tử lớn nhất trong cột của nó. Ví dụ trong bảng số sau đây:

15 3 9

55 4 6

76 1 2

thì phần tử A22 chính là phần tử yên ngựa.

Bạn hãy lập chương trình nhập từ bàn phím một bảng số kích thước MxN và kiểm tra xem nó có phần tử yên ngựa hay không? Nếu có không xuất ra NO nếu có xuất ra vị trí phần tử đó.

Ví dụ

```
YENNGUA.INP
3 3
15    3     9
55    4     6
76    1     2
YENNGUA.OUT
A22
```

Thuật toán

Ta có thuật toán đơn giản để giải bài toán trên, tuy không tốt lắm. Bạn đọc có thể cải tiến thêm.

Đầu tiên, ta dùng một mảng 2 chiều A để lưu trữ bảng số. Lưu ý khi khai báo ta chưa biết cụ thể giá trị của m, n nên phải cấp phát chỉ số hàng và cột dư ra.

```
Cont
      maxm=1000;
      Maxn=1000;
Var
      A:array[1..maxm,1..maxn] of longint;
```

Để đọc dữ liệu ta dùng 2 vòng for lồng nhau, vòng lặp i:1→m duyệt qua các hàng, với mỗi hàng i ta lặp j:1→n để duyệt qua các phần tử trên hàng i đó. Lần lượt đọc từng số vào cho mỗi phần tử

```
Read(fi,m,n);
For i:=1 to m do
      For j:=1 to n do
            Read(fi,a[i,j]);
```

Sau đó duyệt qua các phần tử của mảng, phần tử nào thỏa mãn yêu cầu đề bài (nhỏ nhất trong hàng, lớn nhất trong cột) thì xuất ra. Ta có thể dùng một biến **kt: boolean** để đánh dấu. Với mỗi phần tử a[i, j] ta phải xem trên hàng i có phần tử nào nhỏ hơn nó không? Trên cột j có phần tử nào lớn hơn nó không?

```
For i:=1 to m do
      For j:=1 to n do
            Begin
            Kt:=true;
            For k:=1 to n do //kiem tra nho nhat tren hang
                  If a[i,k]<a[i,j] then kt:=false;
            For k:=1 to m do //kiem tra lon nhat tren cot
                  If a[k,j]>a[i,j] then kt:=false;
            If kt then write(fo,'a',i,j,' ');//neu thoa thi xuat ra
            End;
```

Chương trình

```
Const
      maxm=1000;
      Maxn=1000;
      Finame='yenngua.inp';
      Foname='yenngua.out';
var
   fi,fo:text;
   i,j,k,m,n:longint;
    A:array[1..maxm,1..maxn] of longint;
   kt:boolean;
BEGIN
assign(fi,finame);reset(fi);
Read(fi,m,n);
For i:=1 to m do
      For j:=1 to n do
            Read(fi,a[i,j]);
close(fi);
assign(fo,foname);rewrite(fo);
For i:=1 to m do
      For j:=1 to n do
            Begin
            Kt:=true;
            For k:=1 to n do //kiem tra nho nhat tren hang
                  If a[i,k]<a[i,j] then kt:=false;
            For k:=1 to m do //kiem tra lon nhat tren cot
                  If a[k,j]>a[i,j] then kt:=false;
            If kt then write(fo,'a',i,j,' ');//neu thoa thi xuat ra
            End;
close(fo);
END.
```

Ví dụ 2. Rectangle RECT

Trên giấy kẻ ô khổ N x N có vẽ một số hình chữ nhật. Mỗi hình chữ nhật được tạo ra từ các ô nguyên vẹn, các hình chữ nhật khác nhau không chồng lên nhau và không tiếp xúc nhau

Cho mảng A có kích thước N x N, trong đó A[i,j]=0 nếu ô [i,j] thuộc một hình chữ nhật nào đó, còn A[i,j] =1 trong trường hợp ngược lại.

Hãy viết chương trình xác định số các hình chữ nhật có trong bảng.

Dữ liệu vào: Từ File văn bản RECT.INP có cấu trúc như sau:

- Dòng đầu tiên ghi số nguyên dương N (N<=250).
- N dòng tiếp theo mỗi dòng ghi N số 0 hoặc 1 là các phần tử của mảng, mỗi số viết cách nhau ít nhất một dấu cách

Kết quả: xuất ra file RECT.OUT

- Số lượng hình chữ nhật

Ví dụ

```
RECT.INP
7
0 1 1 1 1 1 1
1 1 0 0 0 1 1
0 1 0 0 0 1 1
0 1 0 0 0 1 1
1 1 1 1 1 1 1
1 1 1 1 0 0 0
1 1 1 1 0 0 0
RECT.OUT
4
```

Thuật toán

Mỗi hình chữ nhật có một ô góc dưới bên phải a[i, j]=0 có tính chất là ô bên phải nó a[i, j+1] và ô bên dưới nó a[i+1,j] mang giá trị 1. Vậy ta duyệt bảng và đếm các phần tử có tính chất này sẽ được số lượng hình chữ nhật.

```
For i:=1 to m do
      For j:=1 to n do
            If a[i, j]=0 and a[i, j+1]=1 and a[i+1,j]=1 then inc(dem);
```

Tuy nhiên đối với các ô nằm trên hàng n và cột n, khi xét các ô bên phải (cột n+1) và bên dưới (hàng n+1) sẽ vượt khỏi kích thước bảng. Khi đó có thể gây lỗi cài đặt. Một cách đơn giản là ta "tạo viền" cho bảng bằng cách bổ sung thêm cột n+1 và hàng n+1 gồm các phần tử có giá trị 1.

```
{tao vien}
For i:=1 to n+1 do
   begin
   a[i, n+1]:=1;//tao cot n+1
   a[n+1,i]:=1;//tao hang n+1
   end;
```

Khi đó bảng trở thành

```
0 1 1 1 1 1 1 1
1 1 0 0 0 1 1 1
0 1 0 0 0 1 1 1
0 1 0 0 0 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 0 0 0 1
1 1 1 1 0 0 0 1
1 1 1 1 1 1 1 1
```

Chương trình

```
Const
   Maxn=1000;
   Finame='rec.inp';
```

```
   Foname='rec.out';
var
   fi,fo:text;
   i,j,dem,m,n:longint;
   A:array[1..maxn,1..maxn] of longint;
BEGIN
assign(fi,finame);reset(fi);
Read(fi,n);
For i:=1 to n do
      For j:=1 to n do
            Read(fi,a[i,j]);
close(fi);
{tao vien}
For i:=1 to n+1 do
   begin
   a[i, n+1]:=1;//tao cot n+1
   a[n+1,i]:=1;//tao hang n+1
   end;
{dem phan tu dac biet}
dem:=0;
For i:=1 to n do
      For j:=1 to n do
            If (a[i,j]=0) and (a[i,j+1]=1) and (a[i+1,j]=1) then inc(dem);

assign(fo,foname);rewrite(fo);
write(fo,dem);
close(fo);
END.
```

Ví dụ 3. Tổng lớn nhất SUM

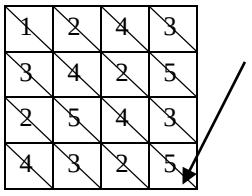
Đề thi tuyển sinh phổ thông năng khiếu 2003

Cho một bảng A gồm N x N số nguyên (N ≤ 100), các dòng được đánh số từ trên xuống dưới bắt đầu từ 1, các cột được đánh số từ trái qua phải cũng bắt đầu từ 1. Mỗi số trong bảng có giá trị tuyệt đối không vượt quá 10000. Đường chéo chính của bảng là đường thẳng nối hai ô (1,1) và (N,N). Như vậy trên bảng có 2N-1 đường chéo song **Yêu cầu:** Tìm đường chéo song song với đường chéo chính có tổng các phần tử trên đường chéo đó là lớn nhất.

Dữ liệu vào cho trong file văn bản SUM.INP trong đó:

- Dòng đầu chứa số N.
- Dòng thứ i trong N dòng tiếp theo chứa N số nguyên lần lượt ứng với các phần tử nằm trên dòng thứ i của bảng A.

Kết quả ghi ra trong file văn bản SUM.OUT trong đó chứa một số nguyên duy nhất là tổng các phần tử trên đường chéo mà bạn tìm được.



Đường chéo chính

Ví dụ: với bảng A như hình vẽ, đường chéo chính chính là đường chéo có tổng lớn nhất (bằng 14), các file dữ liệu vào/ra lần lượt có nội dung như sau:

SUM.INP
4 1 2 4 3 3 4 2 5

SUM.OUT
14

2 5 4 3 4 3 2 5

Thuật toán

Để giải bài toán này ta cần biết cách duyệt qua các đường chéo song song với đường chéo chính của bảng số một cách nhanh gọn và hiệu quả. Có rất nhiều cách để thực hiện điều này. Sau đây là một cách: để ý rằng trên một đường chéo, hiệu giữa chỉ số hàng và chỉ số cột là không đổi. Hiệu này nhận giá trị từ 1-N (tương ứng với đường chéo gồm 1 ô góc trên phải (1,N)) cho đến N-1 (tương ứng với đường chéo gồm 1 ô góc dưới trái (N,1)). Do đó ta lần lượt xét các giá trị hiệu từ 1-N đến N-1. Ký hiệu giá trị này là T. Với mỗi giá trị T, ta duyệt qua các cột trên đường chéo tương ứng. Nếu T≥0 (tương ứng với các đường chéo ở nửa dưới của bảng) thì cột bắt đầu là 1 còn nếu T<0 thì cột bắt đầu là 1-T.

Đoạn lệnh sau thể hiện cách làm này:

```
for t:=1-n to n-1 do
      if (t>=0) then j:=1 else j:=1-t;{j là biến lưu cột bắt đầu của đường chéo t}
         s:=0;                       {tổng các phần tử trên đường chéo}
         repeat
               s:=s+a[j+t,j];        {cộng giá trị ô hiện thời vào tổng}
               inc(j);                     {sang cột mới}
         until (j+t>n) or (j>n);           {vượt quá phạm vi của bảng}
         if (s>kq) then kq:=s;             {cập nhật kết quả}
     end;
```

Chương trình

```
const
     MAXN=105;
     finp='sum.inp';
     fout='sum.out';

var
   n, i, j, t, s, kq: longint;
   a: array[1..MAXN, 1..MAXN] of longint;
begin
     assign(input, finp);
     reset(input);
     assign(output, fout);
     rewrite(output);

     readln(n);
     for i:=1 to n do
         for j:=1 to n do
             read(a[i,j]);
     kq:=-maxlongint;
     for t:=1-n to n-1 do begin
         if (t>=0) then j:=1 else j:=1-t;
         s:=0;
         repeat
               s:=s+a[j+t,j];
               inc(j);
         until (j+t>n) or (j>n);
         if (s>kq) then kq:=s;
     end;
     writeln(kq);
     close(input);
     close(output);
end.
```

Chuyên đề 7. XÂU KÍ TỰ

I. TÓM TẮT LÝ THUYẾT

1. Khái niệm

- Xâu là mảng một chiều mà mỗi phần tử là 1 kí tự.
- **Độ dài xâu**: là số các phần tử = số các kí tự.
- **Xâu rỗng**: là xâu có độ dài bằng 0
- **Giá trị xâu** phải đặt giữa 2 dấu nháy đơn ‘ (trong pascal)

Ví dụ:

```
s:='thieu nhi';
writeln(s[4]);→ e
writeln(length(s));→ 9 //s có độ dài 9
S1:='';//xâu rỗng
```

2. Khai báo

```
VAR <TenBien>:STRING[<DoDaiXau>];
```

<Độ Dài Xâu Tối Đa> có thể không khai báo, khi đó biến có độ dài tối đa 255 kí tự.

Ví dụ: dùng biến **hs** để lưu trữ họ tên một học sinh ta có thể khai báo:

```
VAR hs:STRING[50];
Hoặc
VAR hs:String;
```

3. Phép toán trên xâu

3.1. Phép ghép xâu:+

Ví dụ:

```
S : = 'Ha' + 'Noi' + ' - Viet Nam' ;
Writeln(s) ;  →Cho kết quả là :'Ha Noi – Viet Nam' ;
S1 : = '2012';
S : = s + S1 ;
Writeln(s) ; → 'Ha Noi – Viet Nam2012' ;
```

3.2. Các phép so sánh =, <>, <, >,<=, >=

a. Mã ASCII : để dễ dàng cho việc xử lý, máy tính mã hóa mỗi kí tự thành các con số (vì máy tính chỉ tính toán trên số), có tổng cộng 256 kí tự nên máy tính sử dụng các con số từ 0 → 255. Mỗi kí tự được gán một mã số. Ví dụ : ‘A’ → 65, ‘Z’ → 90, ‘a’ → 97, ‘0’ → 48

Để xem mã ASCII của các kí tự, trong Free Pascal ta chọn menu **Tool → Ascii table**

b. Phép so sánh: Có thứ tự ưu tiên thấp hơn phép ghép xâu và thực hiện so sánh theo quy tắc sau:

- *Xâu A > Xâu B nếu kí tự đầu tiên khác nhau giữa chúng kể từ trái sang trong xâu A có mã ASCII lớn hơn*
- *Nếu A, B có độ dài khác nhau và A là đoạn đầu của B thì A<B*
- *A=B khi A,*B giống nhau hoàn toàn

Ví dụ 1:

```
'May tinh'>'May cay'
```

Ví dụ 2:

```
S1:='May tinh';
S2:='May tinh de ban'
If s1>s2 then writeln('s1 lon hon s2') else writeln('s2 lon hon s1');
```

4. Hàm và thủ tục

Length(S: String): Byte → Tính độ dài của xâu S, trả về 1 số nguyên kiểu Byte

Upcase(s:String) : String → Trả về xâu viết hoa của xâu s

Delete(S: String, VT: Byte, N: Byte)→ Xóa xâu S bắt dầu từ vị trí VT, xóa N kí tự

Insert(S1: String, S2:String, VT:Byte)→ Chèn xâu S1 vào xâu S2, bắt đầu từ vị trí VT

Pos(S1: String, S2:String) : Byte→ Tìm vị trí xuất hiện đầu tiên của xâu S1 trong xâu S2. Nếu không có trả về 0.

Str(n:longint,S:Tring);→ chuyển số thành kí tự

5. Kiểu kí tự Char

Dùng để lưu trữ 1 kí tự

Khai báo:

```
VAR <TenBien>:CHAR;
```

5.1. Các hàm và thủ tục liên quan đến kí tự

Ord(Ch: Char): Byte→ Trả về mã ASCII của kí tự Ch

```
Write('Mã ASCII của A là: ', Ord('A'));//65
```

Chr(N: Byte):Char → Trả về kí tự có mã ASCII bằng N

```
Write('Kí tự có ma 65 là', Chr(65));//A
```

II. VÍ DỤ

Ví dụ 1. Kiểm tra Ngoặc đơn đúng NGOAC

Xét xâu S chỉ bao gồm các kí tự ngoặc mở '(' và ngoặc đóng ')'. Xâu S xác định một cách đặt ngoặc đúng, nếu thỏa mãn các điều kiện:

- Tổng số ngoặc đóng = tổng số ngoặc mở
- Đi từ trái qua phải, ở bất cứ vị trí nào số ngoặc đóng phải nhỏ hơn hoặc bằng số ngoặc mở

Ví dụ:

Input	Output
3 (()) ()() ((())	YES YES NO

Thuật toán

Ta lưu lại số ngoặc đóng và số ngoặc mở tại từng thời điểm. Duyệt qua từng kí tự của xâu s từ trái sang phải, ở mỗi kí tự s[i] ta cập nhật lại số ngoặc đóng và số ngoặc mở, sau đó kiểm tra theo yêu cầu của đề bài

```
Kt:=true;Dong:=0;Mo:=0;
For i:=1 to length(s) do
      If s[i]=')' then inc(Dong) else inc(Mo);
      If Dong>Mo then kt:=false;
If Dong>Mo then kt:=false;
```

Cần lưu ý khi đọc dữ liệu từ file, mỗi lần ta sẽ đọc một hàng vào xâu s nên phải dùng câu lệnh **readln(fi,...)**

```
For k:=1 to n do
      Readln(fi,s);
      {xu li xau s}
```

Chương trình

```
Const
   Finame='ngoac.inp';
   Foname='ngoac.out';
var
   fi,fo:text;
   k,i,n,Mo,Dong:longint;
   s:string;
   kt:boolean;
BEGIN
assign(fi,finame);reset(fi);
Assign(fo,foname);rewrite(fo);
Readln(fi,n);
For k:=1 to n do
      Begin
Readln(fi,s);
      Kt:=true;Dong:=0;Mo:=0;
      For i:=1 to length(s) do
            begin
            If s[i]=')' then inc(Dong) else inc(Mo);
            If Mo<Dong then kt:=false;
            end;
      If Mo<>Dong then kt:=false;
      if kt then writeln(fo,'YES')
      else writeln(fo,'NO');
      End;
Close(fi);
close(fo);
END.
```

Ví dụ 2. Thay thế từ THAYTHE

Nhập xâu S, thay thế tất cả các cụm từ “anh” bằng từ “em”

Ví dụ

Input

anh hoc tieng anh

Output

em hoc tieng em

Thuật toán

Lặp khi có từ ‘anh’ trong xâu

- X:= vị trí từ ‘anh’
- Xóa từ anh tại vị trí x
- Chèn từ ‘em’ tại vị trí x

Chương trình

```
Program ReplaceWord;
Uses crt;
Var
Begin
Writeln('Nhap xau s: ');
Readln(s);
While Pos('anh',s)>0 do
            Begin
      X:=pos('anh',s);
            Delete(s,x,3);
            Insert('em',s,x);
            End;
```

```
Writeln('Xau da thay: ',s);
Readln;
End.
```

Ví dụ 3. Đếm chữ cái CHUCAI

Nhập xâu S, xuất ra màn hình số lần xuất hiện các chữ cái tiếng anh viết hoa trong S.

Ví dụ:

S=’AABCAEEF12AbA’

So chu A la 4, so chu B la 1, so chu C la 1, so chu E la 2, so chu F la 1

Thuật toán

Ta dùng một mảng Dem[] 26 phần tử, mỗi phần tử lưu số lần xuất hiện một kí tự, ở đây ta có thể dùng các kí tự ‘A’ ...’Z’ để đánh chỉ số cho mảng. Dem[‘A’] lưu số lần xuất hiện kí tự ‘A’

```
Var Dem:array['A'..'Z'] of byte;
```

Duyệt các kí tự của s, nếu gặp kí tự là chữ cái viết hoa thì cập nhật Dem[]. Vì mã ASCII của các kí tự ‘A’, …, ‘Z’ liên tục từ 65 … 91, ta có thể dùng phép so sánh trực tiếp các kí tự

```
For i:=1 to length(s) do
   if ('A'<=s[i]) and (s[i]<='Z') then inc(Dem[s[i]]);
```

Sau khi cập nhật, ta duyệt Dem[] và xuất ra các phần tử lớn hơn 0. Để duyệt Dem[] ta dùng một biến **c:char** (vì chỉ số mảng là kí tự)

```
for c:='A' to 'Z' do
   if Dem[c]>0 then writeln(fo,c,':',dem[c]);
```

Chương trình

```
Const
   Finame='demkt.inp';
   Foname='demkt.out';
var
   fi,fo:text;
   k,i,n,Mo,Dong:longint;
   s:string;
   Dem:array['A'..'Z'] of byte;
   c:char;
BEGIN
assign(fi,finame);reset(fi);
Readln(fi,s);
close(fi);
For i:=1 to length(s) do
   if ('A'<=s[i]) and (s[i]<='Z') then inc(Dem[s[i]]);
Assign(fo,foname);rewrite(fo);
for c:='A' to 'Z' do
   if Dem[c]>0 then writeln(fo,c,':',dem[c]);
close(fo);
END.
```

Ví dụ 4. Nén xâu

Một xâu ký tự có thể nén lại thành một xâu mới bằng cách nén các ký tự giống nhau đứng cạnh nhau. Ví dụ trong xâu có 4 ký tự **“A”** đứng cạnh nhau, sẽ được nén thành 4A. Hãy lập trình để nén một xâu ký tự in hoa theo cách trên.

Dữ liệu vào cho từ file văn bản NENXAU.INP là một xâu ký tự chữ cái in HOA.

Kết quả: Ghi vào file NENXAU.OUT là xâu ký tự sau khi nén.

Ví du:

NENXAU.INP	NENXAU.OUT

MMAABBBEEEEZH	2M2A3B4E1Z1H

Thuật toán

Gọi s1 là xâu nén của xâu s; ban đầu s1 rỗng;

Gọi d là số lượng của từng dãy kí tự giống nhau. Duyệt từng kí tự của xâu s từ trái qua phải, nếu s[i]=s[i-1] ta tăng d thêm 1 vì dãy kí tự vẫn còn giống nhau, **ngược lại** ta nối d và kí tự s[i-1] vào xâu s1 đồng thời gán dem = 1 vì đã sang dãy kí tự khác.

```
for i:=2 to length(s) do
   if s[i]<>s[i-1] then
      begin
      str(d,t);
      s1:=s1+t+s[i-1];
      d:=1;
      end
    else inc(d);
```

Lưu ý, để có thể nối d vào xâu s1 ta dùng **thủ tục str(d,t) chuyển số d thành xâu t**

Với vòn lặp trên, dãy kí tự s[n] sẽ chưa được nối vào xâu s1 vì khi i=n và s[i] <> s[i-1] thì ta mới chỉ nối s[i-1].Ta thêm một kí tự khoảng trắng vào sau xâu s, lúc này xâu s có n+1 kí tự. Việc lập trình sẽ đơn giản hơn.

Chương trình

```
program NEN_XAU;
const
  finame='NENXAU.INP';
  foname='NENXAU.OUT';
var
  s,s1,t:string;
  i,d:longint;
begin
assign(input,finame);reset(input);
assign(output,foname);rewrite(output);
readln(input,s);
d:=1;
s:=s+' ';//them mot ki tu vao sau s
for i:=2 to length(s) do
   if s[i]<>s[i-1] then
      begin
      str(d,t);
      s1:=s1+t+s[i-1];
      d:=1;
      end
   else inc(d);

writeln(output,s1);
close(input);close(output);
end.
```

Chuyên đề 8. CHƯƠNG TRÌNH CON

I. TÓM TẮT LÝ THUYẾT

1. Khái niệm

1.1. Chương trình con

Là những đoạn chương trình nhỏ được tạo ra để giải quyết một vấn đề hoặc một lớp vấn đề nào đó và được lưu trữ lại để sử dụng ở nhiều nơi khác nhau khi lập trình.

Chương trình con là do người dùng hoặc nhà thiết kế ngôn ngữ lập trình tự định nghĩa ra để sử dụng lại trong nhiều chương trình hoặc modules khác nhau.

Có 2 dạng chương trình con: Hàm (**Procedure**) và thủ tục (**Function**)

Tham số là thông tin được sử dụng trong chương trình con, cách sử dụng tham số quyết định tính đơn giản và hiệu quả của thủ tục hoặc hàm.

1.2. Một số hàm và thủ tục chuẩn trong Pascal:

Hàm:

Hàm	Công dụng
Upcase(x:char):char	Trả về kết quả là một kí tự hoa
Cos(x:real):real	Trả về giá trị cosin của góc x
Sin(x:real):real	Trả về giá trị sin của góc x
Sqrt(x:real):real	Trả về giá trị căn bậc 2 của x
Sqr(x:real):real	Trả về giá trị bình phương của một số
Round(x:real):real;	Trả về giá trị làm tròn số thực x
Length(s:string):byte	Trả về giá trị là chiều dài của xâu kí tự
Ord(x:char):byte	Trả về kết quả là số thứ tự của kí tự x trong bảng mã ACSII

Thủ tục:

Tên thủ tục	Chức năng
Read(<v1, v2,...>); Readln(<v1, v2,...>);	Đọc dữ liệu từ bàn phím hoặc từ file và gán cho biến v1, v2….
Write(v1, v2,…); Writeln(v1, v2,…);	Xuất giá trị các biến v1, v2, v3 ra màn hình
gotoXY(x,y:byte);	Định vị vị trí con trỏ màn hình đến tọa độ x, y trên màn hình làm việc
Delay(time:word);	Tạm dừng xử lý trong một khoảng thời gian là time
Clrscr;	Xóa màn hình làm việc và đưa con trỏ về vị trí có tọa độ 1,1
Exit;	Rời khỏi chương trình

2. Hàm – Thủ tục

2.1. Hàm – Function

a. Chức năng

Là chương trình con được dùng để thực hiện một tác động xác định vào cấu trúc dữ liệu của chương trình và trả về một kết quả nhất định và được sử dụng trong chương trình chính.

b. Cấu trúc

```
Function   <Tên hàm>(<danh sách tham số>):<kiểu dữ liệu trả về của hàm>;
[Khai báo các hằng số sử dụng trong hàm – nếu có]
[Khai báo các kiểu dữ liệu người dùng sử dụng trong hàm – nếu có]
[Khai báo chương trình con sử dụng trong hàm – nếu có]
[Khai báo các biến sử dụng trong hàm – nếu có]
      //phần thân hàm
      Begin
            // các câu lệnh;
            // lưu ý: trong phần thân của hàm phải có câu lệnh trả về giá trị cho hàm;
     End;
```

2.2. Thủ tục (Procedure)

a. Chức năng

Là chương trình con được dùng để thực hiện một tác động xác định vào cấu trúc dữ liệu của chương trình và không cung cấp giá trị trả về.

b. Cấu trúc

```
Procedure  <Tên thủ tục>(<danh sách tham số>);
[Khai báo các hằng số sử dụng trong thủ tục – nếu có]
[Khai báo các kiểu dữ liệu người dùng sử dụng trong thủ tục – nếu có]
[Khai báo chương trình con sử dụng trong hàm – nếu có]
[Khai báo các biến sử dụng trong thủ tục]
//phần thân thủ tục
Begin
// các câu lệnh;
End;
```

Ví dụ:

Thủ tục in ra chuỗi đảo ngược của một chuỗi cho trước

```
Procedure InNguoc(s: string);
      Var i: byte;
      Begin
            For i:= length(s) downto 1 do Write(s[i]);
  End;
```

Ví dụ: Hàm tính tổng hai số a, b

```
Function Tong(a,b:integer): longint;
Var c: longint;
Begin
      c:= a+b;
      Tong:=c;
//co the dung cau lenh exit(c);
End;
```

2.3. So sánh giữa hàm và thủ tục

Giống nhau:

- Đều là chương trình con.
- Có tên gọi xác định và duy nhất trong chương trình chính.

- Có thể chứa hoặc không chứa tham số.

Khác nhau:

- Hàm trả về một giá trị cụ thể, do đó hàm được sử dụng trong các biểu thức hoặc được gán cho một biến nào đó. Do đó cần lưu ý kiểu dữ liệu trả về của hàm mà sử dụng cho thích hợp.
- Thủ tục không trả về giá trị nên được dùng để thực hiện độc lập một nhóm lệnh tiến hành một thao tác đặc trưng trong chương trình..

3. Tham số trong chương trình con:

3.1. Phân loại tham số:

a. Tham số hình thức:

- Là các biến được khai báo trong phần khai báo tên của chương trình con.
- Cách khai báo tham số cho chương trình con tương tự như khai báo biến cho chương trình.

Ví dụ:

```
Procedure vidu1(i,j,k: integer; Var x,y: real);
Function vidu2(h:char; var z:Boolean):real;
```

Có hai loại tham số hình thức:

- **Tham số biến**: là tham số đứng sau từ khóa **var**, và các thay đổi giá trị biến trong chương trình con sẽ được cập nhật cho tham số truyền vào.
- **Tham số trị**: là tham số không đi kèm với từ khóa **var,** những thay đổi giá trị của biến trong chương trình con sẽ không làm thay đổi giá trị của tham số truyền vào.

b. Tham số thực:

- Là các tham số dùng chung không cần phải khai báo trong chương trình con. Khi dùng loại tham số này ta phải chú ý đến tác động của các chương trình con đến giá trị của tham số.

3.2. Truyền tham số:

a. Định nghĩa

- Việc truyền giá trị từ bên ngoài vào chương trình con thông qua tham số (tham số hình thức) được gọi là truyền tham số. Giá trị của tham số sẽ được sử dụng trong chương trình con.

b. Các hình thức truyền tham số

Truyền tham trị:

- Không dùng từ khóa var trong khai báo tham số cho chương trình con.
- Những thay đổi của tham số trong chương trình con không ảnh hưởng đến giá trị của tham số trong chương trình gọi nó.

Truyền tham biến

- Các tham số truyền theo hình thức này bắt buộc phải đứng sau từ khóa var trong phần khai báo tham số cho chương trình con.
- Những thay đổi của tham số trong chương trình con sẽ được lưu lại trong chương trình đã gọi nó.

3.3. Biến toàn cục và biến địa phương:

Biến toàn cục: là các biến được khai báo trong chương trình chính. Các biến này có tác dụng ở mọi nơi trong toàn bộ chương trình.

Biến địa phương: là các biến được khai báo trong các chương trình con. Các biến này chỉ có tác dụng trong phạm vi chương trình con đó mà thôi.

Chú ý: Trong một chương trình con, nếu biến toàn cục trùng tên với biến địa phương thì biến địa phương được ưu tiên hơn.

```
Program  Test;
Var a,b: Integer;   {biến toàn cục}
Procedure  ThuBien;
Var  a: Integer; {biến địa phương}
Begin
    a:=10;
    Writeln('A=',a,'B=',b);
End;
Begin
    a:=50;
    b:=200;
    ThuBien;                          {A=10  B=200}
    Writeln('A=',a,'B=',b);       {A=50  B=200}
End.
```

4. Đệ quy

4.1. Khái niệm

Trong lập trình, có khái niệm: một chương trình con (hàm, thủ tục) được gọi là đệ quy nếu trong quá trình thực hiện nó có phần phải gọi đến chính nó.

4.2. Cấu trúc chính

Một chương trình con đệ qui căn bản gồm hai phần.

- **Phần cơ sở**: chứa các tác động của hàm hoặc thủ tục với một số giá trị cụ thể ban đầu của tham số.
- **Phần đệ quy**: định nghĩa tác động cần được thực hiện cho giá trị hiện thời của các tham số bằng các tác động đã được định nghĩa trước đây với kích thước tham số nhỏ hơn.

Ví dụ: hàm Gt(n) để tính n! có thể được tính như sau:

```
      Gt(n):=n*Gt(n-1);
// trong đó Gt(n-1) chính là hàm Gt(n) được gọi lại với tham số nhỏ hơn
```

4.3. Ưu điểm của đệ quy:

Gọi đệ quy là một kỹ thuật lập trình rất quan trọng vì nó thường ngắn gọn và phù hợp với suy nghĩ tự nhiên về nhiều cách giải bài toán. Thậm chí nhiều bài toán hầu như chỉ có thể dùng đệ quy.

II. VÍ DỤ

Ví dụ 1. Chẵn lẻ CHANLE

Viết chương trình nhập vào một số nguyên và kiểm tra xem số vừa nhập là số chẵn hay số lẻ.

Hướng dẫn:

Khai báo hàm kiểm tra chẵn lẻ với tham số truyền vào là số cần kiểm tra và kiểu dữ liệu trả về là kiểu boolean:

- True nếu tham số truyền vào là số chẵn.
- False nếu tham số truyền vào là số lẻ.

Dựa vào kết quả trả về mà xuất câu kết luận phù hợp.

Chương trình:

```
Program ktra_chan_le;
Var  x:integer;
Function kiemtra(a: integer): boolean;
```

```
      Begin
            If x MOD 2=0 then
                  Kiemtra:= true
            Else
                  Kiemtra:=false;
End;
Begin
  Write('Nhap vao mot so nguyen : ');  Readln(x);
      If (kiemtra(x)=true) Then
            Writeln('So vua nhap vao la so chan')
      Else
            Writeln('So vua nhap vao la so le');
      Readln;
End.
```

Ví dụ 2. Giải phương trình GIAIPT

Viết chương trình nhập vào 3 số a, b, c và giải phương trình $ax^2 + bx + c = 0$

Hướng dẫn:

Chia làm hai trường hợp:

- Nếu a=0 thì phương trình suy biến về phương trình bậc nhất bx + c = 0
- Ngược lại giải phương trình bậc hai $ax^2 + bx + c = 0$

Thiết kế thủ tục giải phương trình bậc nhất và phương trình bậc 2 tương ứng:

- Procedure PTBI(a,b: integer);
- Procedure PTBII(a,b,c: integer);

Chương trình:

```
Program Giaiphuongtrinh;
Uses crt;
Var a,b,c: integer;
Procedure PTBI(a,b: integer);
var x: real;
Begin
      If a = 0 Then     // N?u a b?ng 0
            If b = 0 Then  // Tru?ng h?p a = 0 và b = 0
                  Writeln('Phuong trinh co vo so nghiem')
            Else   // Tru?ng h?p a=0 và b khác 0
                  Writeln('Phuong trinh vo nghiem')
      Else  // Tru?ng h?p a khác 0
            Begin
                  x:= -b/a;
                  Writeln('Phuong trinh co nghiem la :',x:0:2);
            End;
      Readln;
End;

Procedure PTBII(a,b,c: integer);
var d, x1, x2: real;
Begin
      d:=sqr(b)-4*a*c;
    IF d<0 THEN
        write('Phuong trinh vo nghiem!')
    ELSE
    BEGIN
         x1:=(-b-sqrt(d))/(2*a);
         x2:=(-b+sqrt(d))/(2*a);
         IF d=0 THEN
                writeln('Phuong trinh co nghiem kep x = ',x1:5:1)
         ELSE
                writeln('Phuong trinh co 2 nghiem phan biet:',x1,' ',x2);
```

```
    END;
End;
Begin
            Writeln('Nhap vao ba so a,b,c');
            Writeln('a = '); Readln(a);
            Writeln('b = '); Readln(b);
            Writeln('c = '); Readln(c);
            If(a=0)    then
                  PTBI(b,c)
            Else
                  PTBII(a,b,c);
                readln;
End.
```

The end.