



CHUYÊN ĐỀ TURBO PASCAL

LỚP 8+9

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

GIỚI THIỆU



TÓM TẮT TẬP TÀI LIỆU

- 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
- VI. Chương trình con.
- VII. Xâu ký tự
- VIII. Chuyên đề: Tính chia hết- Số nguyên tố.

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**).

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 : $34 \text{ DIV } 5 = 6$).
- Phép chia lấy số dư: **MOD** (Ví dụ: $34 \text{ MOD } 5 = 4$).

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 HẲNG

- 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. 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**.
- Lệnh 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**;

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**: Xóa màn hình và đưa con trỏ về góc trên bên trái màn hình.
- Thủ tục **CLREOL**: Xóa các ký tự từ vị trí con trỏ đến hết dòng.
- Thủ tục **DELLINE**: Xóa 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.

CHƯƠNG II

CÂU LỆNH CÓ CẤU TRÚC Rẽ NHÁNH

A. LÝ THUYẾT

I. CÂU LỆNH Rẽ NHÁNH

1.1. Lệnh IF

Cú pháp:

- (1) **IF B THEN S;**
- (2) **IF B THEN S1 ELSE S2;**

1.2. Lệnh CASE

Cú pháp:

Dạng 1	Dạng 2
CASE B OF Const 1: S₁; Const 2: S₂; ... Const n: S_n; END;	CASE B OF Const 1: S₁; Const 2: S₂; ... Const n: S_n; ELSE S_{n+1}; END;

B. BÀI TẬP:

Bài tập 1:

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 dịch;
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');
    end;
end;

```

```

8: Write('--> Sunday');
else
Write(' Không có thu nay');
end;
readln
end.

```

Bài tập 2

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

I. VÒNG LẶP CÓ SỐ LẦN XÁC ĐỊNH

A. LÝ THUYẾT:

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;

B. BÀI TẬP

Bài tập 1:

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.
```

Bài tập 2:

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:

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.
```


Bài tập 4

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.
```

Bài tập 5

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
end.
```

II. VÒNG LẶP CÓ SỐ LẦN 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 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.
```

Bài tập 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('Nhap so n: ');readln(n);
    until n>0;
    gt:=1;
    while n>0 do
    begin
        gt:=gt*n;
        n:=n-1;
    end;
    writeln('Giai thua cua n la: ',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 thua cua n la:”

Bài tập 3

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 F_1 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.
```

CHƯƠNG IV

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
- Kết quả của bài toán trả về 1 giá trị duy nhất (kiểu vô hướng, kiểu string) . - Lỗi gọi CTC cần nằm trong các biểu thức tính toán.	- 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) . - Lỗi gọi CTC không nằm trong các biểu thức tính toán.

B. BÀI TẬP

Bài 7: Viết các hàm USCLN(a, b) và BSCNN(a, b).

Hướng dẫn: Khai báo hàm USCLN có:

Tên hàm: USCLN

Kiểu dữ liệu trả về: integer

Tham số: 2 tham trị là int a, int b

Giá trị trả về: chính là giá trị ước số chung lớn nhất của a và b.

Xử lý tìm USCLN, BSCNN: Dựa trên ví dụ đã có ở phần trước.

Nội dung hàm:

Function USCLN(a, b : Integer) : Integer;

Var Sodu:integer;

Begin

While (b <> 0) do

Begin

Sodu:= a mod b;

a := b;

b := Sodu;

End;

USCLN := a;

End;

{Tuong tu ham BSCNN(a, b : Integer)}

Function BSCNN(a, b : Integer) : Integer;

Var k :integer;

Begin

For k := a to a*b do { Lap voi moi i co gia tri tu a den 1 }

If ((k mod a = 0) and (k mod b = 0)) then { Kiem tra a, b cung chia het cho k? }

Break; { break de thoat, luu giu lai gia tri cua k. }

BSCNN:= k;

End;

Bài 8: Viết hàm kiểm tra số N có phải là số nguyên tố hay không?

Hướng dẫn: Khai báo hàm kiểm tra số nguyên tố có:

Tên hàm: KiemtraSNT

Dữ liệu trả về: int, (0 nếu không phải là số nguyên tố, là 1 nếu là số nguyên tố)

Tham số: tham trị là int N, số cần kiểm tra.

Xử lý kiểm tra: Dựa trên ví dụ đã có ở phần trước.

Nội dung hàm:

Function KiemtraSNT(N : Integer) : Boolean;

Var flag : Boolean;

i : Integer;

Begin

flag := True; { Khoi gan gia tri dung cho ktrSNT }

For i := 2 to N-1 do { Lap voi moi i co gia tri tu 2 den N-1 }

If (N mod i = 0) then { Kiem tra xem N co' chia het cho i hay khong }

flag:=false; { Neu dung thi khong con la so nguyen to nua }

KiemtraSNT:=flag;

End;

Cải tiến, không sử dụng biến phụ:

Function KiemtraSN(N : Integer) : Integer;

Var i :integer;

Begin

For i := 2 to N-1 do { Lap voi moi i co gia tri tu 2 den N-1 }

If (N mod i = 0) then { Kiem tra xem N co' chia het cho i hay khong }

```

        Writeln('N không là SNT')    { Neu dung thi khong con la so nguyen to nua}
    Else
        Writeln('N là SNT');        { N là SNT }
End;

```

Bài toán số 2.4: Viết chương trình thực hiện lần lượt các công việc sau:

- Lập thủ tục nhập ba số thực dương a , b , c từ bàn phím .
- Lập thủ tục kiểm tra xem ba số trên có lập thành ba cạnh của tam giác hay không ?
- Viết hàm tính diện tích của tam giác .
- Viết hoàn thiện chương trình chính.

Chương trình

Program Tam_Giac;

Uses crt;

Var a,b,c : Integer;

{ 1. Thu tục nhap }

Procedure NhapABC (Var a, b, c : Integer);

Begin

Write('Nhap a:'); Readln(a);

Write('Nhap b:'); Readln(b);

Write('Nhap c:'); Readln(c);

End;

{ 2. Ham tinh dien tich }

Function Dientich(a, b, c:Integer) : Real;

Var dt,p : Real;

Begin

p := (a+b+c)/2;

dt := sqrt(p*(p-a)*(p-b)*(p-c));

Dientich := dt;

End;

{ 3. Thu tục kiểm tra và In qua tính diện tích (nếu là tam giác) }

Procedure KiemTra_InDienTich (a,b,c:Integer);

Begin

If ((a+b>c) and (b+c>a) and (a+c>b)) then

Writeln(' Ba cạnh trên tạo thành tam giác diện tích', Dientich(a,b,c))

Else

Writeln('Ba cạnh trên không tạo thành tam giác');

End;

{ 3. Thân chương trình chính }

BEGIN

NhapABC(a,b,c);

KiemTra_InDienTich (a,b,c);

END.

Bài 9: Viết chương trình hoàn chỉnh thực hiện phân tích ra N ra thành các thừa số nguyên tố.

Ví dụ: $N = 1260 = 2 \cdot 2 \cdot 3 \cdot 3 \cdot 5 \cdot 7$

Hướng dẫn:

- Viết hàm nhập số N. Lưu ý, hàm nhập cần truyền tham số N theo dạng tham biến để lưu lại giá trị đã nhập vào.
- Viết hàm phân tích số N ra các thừa số nguyên tố và in các thừa số đó ra màn hình với ý tưởng thuật toán:
 - o Chia N cho số nguyên tố u (nhỏ nhất là 2).
 - o Trong khi N còn chia hết cho u thì tiến hành phân tích N với u là thừa số. Giảm N đi u lần.
 - o Nếu N không chia hết cho u, thì tăng u lên 1.
 - o Quá trình lặp lại với
 - u từng bước tăng lên 1 nếu N không chia hết cho u.

- N từng bước giảm xuống u lần nếu N chia hết cho u.
- o Quá trình lặp lại đến một mức u tăng lên và N giảm xuống để $N = u$, khi đó giá trị mới của N sẽ là 1.

Chương trình:

Program Phan_Tich_Thua_So;

Uses crt;

Var N : integer;

{ 1. Thu tục nhập số N. }

Procedure NhapSoN(var NN : Integer);

Begin

 NN := 0; {Gán khởi đầu bằng 0 để vào vòng lặp, vòng lặp dừng khi nhập khác 0}

 While (N = 0) do

 Begin

 Write('Nhập N='); Readln(NN);

 End;

End;

{ 2. Thu tục phân tích N ra thành các thừa số nguyên tố }

Procedure PhantichSoN (Var N1 : Integer);

Var u,dem:Integer;

Begin

 If (N1 > 1) Then

 Begin

 u := 2; dem := 0;

 While (N1 > 1) do

 If (N1 mod u = 0) Then

 Begin

 dem:=dem+1;

 Writeln(u);

 N1 := N1 div u;

 End

 Else

 u:=u+1;

 End

 Else

 Writeln(' Không thể phân tích được ');

End;

{ 3. Thân chương trình chính }

BEGIN

 Clrscr;

 Writeln('Phân tích số N thành tích của các số nguyên tố :');

 NhapSoN(N);

 PhantichSoN(N);

 Writeln ('Nhấn Enter để kết thúc ...');

 Readln;

END.

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.

$$n! = 1 \text{ nếu } n = 0;$$

$$n! = 1.2.3.4.5...n \text{ (Tích của } n \text{ thừa số).}$$

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;
```

```
        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.
```

CHƯƠNG V

DỮ LIỆU KIỂU MẢNG

PHẦN 1: 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.

III. CÁC CÂU LỆNH LÀM VIỆC VỚI DỮ LIỆU KIỂU MẢNG:

1. Câu lệnh nhập, xuất:

a. Nhập mảng:

Write('Nhập n: '); Readln(n);

For i:=1 to n do

 Begin

 Write('a[', i, ']=');

 Readln(a[i]);

 End;

b. Xuất mảng:

For i:=1 to n do

 Write(a[i]: 5);

2. Sắp xếp mảng

For i:=1 to n-1 do

For j:= i+1 to n do

If a[i] > a[j] then

 Begin

 Tam:=a[i];

 A[i]:=a[j];

 A[j]:=tam;

 End;

3. Xác định phần tử lớn nhất, bé nhất:

Min:=a[1]; max:=a[1];

```

For i:=1 to n do
Begin
    If max<a[i] then max:=a[i];
    If min >a[i] then min: =a[i];
End;

```

4. Thuật toán xóa, chèn phần tử trong dãy:

a. Xóa phần tử trong dãy:

```

Nhập vị trí cần xóa (vitri)
For i:= vitri To cuoi-1 Do
    A[i]:=a[i+1];
Cuoi:=cuoi-1;
Xuất mảng sau khi xóa

```

b. Chèn phần tử vào trong dãy:

```

Nhập vitri, so cần xóa
For i:= cuoi+1 Downto vitri + 1 Do
    A[i]:=a[i-1];
A[vitri]:=so;
Cuoi:=cuoi+1;
Xuất mảng sau khi chèn

```

5.Phương pháp tìm kiếm nhị phân:

a.Xác định bài toán:

Input: Dãy A là dãy tăng gồm N số nguyên khác nhau a_1, a_2, \dots, a_n và một số nguyên k.

Output: Chỉ số i mà $a_i = k$ hoặc thông báo không có số hạng nào của dãy A có giá trị bằng k.

b.Ý tưởng:

Sử dụng tính chất dãy A là dãy tăng, ta tìm cách thu hẹp nhanh phạm vi tìm kiếm sau mỗi lần so sánh khóa với số hạng được chọn. Để làm điều đó ta chọn số hạng a_{giua} ở giữa dãy để so sánh với k, trong đó $Giua = (n+1)/2$. Khi đó chỉ xảy ra 1 trong 3 trường hợp:

-Nếu $a_{giua} = k$ thì Giua là số cần tìm. Việc tìm kiếm kết thúc.

-Nếu $a_{giua} > k$ thì do dãy A là dãy được sắp xếp nên việc tìm kiếm tiếp theo chỉ xét trên $a_1, a_2, \dots, a_{giua}$

-Nếu $a_{giua} < k$ thì thực hiện tìm kiếm trên dãy $a_{giua+1}, a_{giua+2}, \dots, a_N$.

Quá trình trên sẽ được lặp lại một số lần cho đến khi hoặc đã tìm thấy khóa k trong dãy hoặc phạm vi tìm kiếm bằng rỗng.

c.Thuật toán:

B1: Nhập N, các số hạng a_1, a_2, \dots, a_n và khóa k;

B2: Đầu $\leftarrow 1$; Cuoi $\leftarrow 1$

B3: Giua $\leftarrow (\text{Đầu} + \text{Cuoi})/2$;

B4: Nếu $a_{giua} = k$ thì thông báo chỉ số Giua rồi kết thúc;

B5: Nếu $a_{giua} > k$ thì đặt Cuoi = Giua – 1, rồi chuyển đến B7;

B6: Đầu \leftarrow Giua + 1;

B7: Nếu Đầu > Cuoi thì thông báo dãy A không có số hạng có giá trị bằng k, rồi kết thúc;

B8: Quay lại B3

B. BÀI TẬP

Bài tập 1

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.
```

Bài tập 2

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.
```

Bài tập 3

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 4

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: 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 Fibonacy_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 6: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 về mảng 1 chiều trong Pascal

Bài 1: Cho 1 dãy số gồm n phần tử $n \leq 15$. Giá trị từng phần tử là số thực. Thực hiện các chương trình sau:

- Nhập từ bàn phím số phần tử và giá trị từng phần tử
- Tính tổng các phần tử âm của dãy
- Tính tổng và trung bình cộng của các phần tử chẵn của dãy
- Tìm phần tử có giá trị lớn nhất của dãy
- Tìm phần tử có giá trị nhỏ nhất
- Sắp xếp dãy theo chiều giảm dần về giá trị của từng phần tử
- In các kết quả trên ra màn hình

```

program vd1;
var   i,n :Byte;
      ta, tb, tam, c, d :real;
      a: array[1..15] of real ;
begin
  write(' nhập số lượng phần tử n ='); readln(n);
  Writeln('Nhap mang a: ');
  for i:=1 to n do
    begin
      write('a[' ,i,']= ');
      readln(a[i])
    end;
  ta:=0;
  for i:=1 to n do if a[i] <0 then ta= ta + a[i] ; { tính tổng âm}
  tc:= 0; t:=1;
  for i:=1 to n do if i mod 2 =0 then begin
      t= t+1 ;
      tc= tc + a[i] ; {tính tổng phần tử chẵn}
    end;
  ttc = tc/t ; { trung bình phần tử chẵn}
  max:= a[1];
  for i=2 to n do if max < a[i] then max =a[i] ; { tìm phần tử lớn nhất}
  min:= a[1];
  for i=2 to n do if min > a[i] then min =a[i] ; { tìm phần tử nhỏ nhất}
  for i=1 to n-1 do
    for j=i+1 to n do
      if (a[i]>a[j]) then
        begin
          tam:=a[i];
          a[i]:=a[j];
          a[j]:=tam;
        end;
    {sắp xếp các phần tử giảm dần về giá trị}
  for i=1 to n then write( a[i]:3);
  readln;
end.

```

Bài 2: Nhập xuất và tính tổng các phần tử trong mảng.

Ví dụ: A: 1 5 6 7 4

Kết quả: Tong S = 23

Chương trình:

Program Mang1C;

```
Uses crt;
Type  Mang20 = array[1..20] of Integer;
Var   N:Integer;
      A:Mang20;
Procedure NhapMang1C(Var A : Mang20 ;Var N:Integer);
Var i: Integer;
Begin
  Write( 'So luong phan tu:' );      Readln( N);
  For i:=0 to N do
    Begin
      Write( 'Nhap phan tu thu ', i," );    Readln( A[i] );
    End;
  End;
Procedure XuatMang1C( Var A : Mang20;Var N :Integer );
Var i:Integer;
Begin
  For i :=0 to N do
    Write( A[i]:2 );
  Writeln;
End;
Function TongMang1C( A : Mang20; N:Integer):longint;
Var i :Integer;
    S :longint;
Begin
  S := 0;
  For i:=0 to N do
    S := S + A[i] ;
  TongMang1C:=S;
End;
BEGIN
  NhapMang1C( A, N );
  XuatMang1C( A, N );
  Writeln(' Tong cac phan tu trong mang ', TongMang1C(A, N) );
  Readln;
END.
```

Bài toán tương tự:

(1) Tính tổng các số nguyên dương chia hết cho 5.

Function TongSoChiaHet5(A:Mang20, N:Integer):Integer;

```
Var S,i :Integer;
Begin
  S:=0;
  For i:=0 to N do
    If(A[i] mod 5=0)
      S := S+A[i];
  TongSoChiaHet5:= S;
End;
```

(2) Tính tổng các số nguyên tố trong mảng

Function LaSoNT(Var N:Integer) :Integer;

```
Var i:Integer;
Begin
  For i:=2 to N-1 do
    If(N mod i = 0) then
      return 0
```

```

Else
    return 1;
End;
Function TongSoNT(Var A : Mang20, Var N:Integer):Integer;
Var S,i:Integer;
Begin
    S:=0;
    For i:=0 to Ndo
        If ( LaSoNT( A[i] ) ) then
            S :=S+ A[i];
    TongSoNT :=S;
End;

```

Bài 3: Đếm số lần xuất hiện của giá trị X trong mảng A. Đếm số lần xuất hiện của các phần tử trong mảng.

Ví dụ: A: 1 5 6 7 4 1 5 5 1 1

X: 6

Kết quả: So lan xuất hiện X là 1

So lan xuất hiện của các phần tử:

1 ==> 4	5 ==> 3
6 ==> 1	7 ==> 1
4 ==> 1	1 ==> 4

Hướng dẫn:

+ Viết hàm đếm số lần xuất hiện của một giá trị X nào đó được nhập vào, và xem như X nhà là tham số cho việc đếm số lần xuất hiện của nó trong A

+ Viết hàm in ra số lần xuất hiện của tất cả các phần tử trong mảng, sử dụng lại hàm đã xây dựng ở trước.

+ Xây dựng chương trình giải quyết bài toán trên gồm:

- Khai báo mảng A, N phần tử.
- Nhập / Xuất mảng A với N phần tử (lưu ý, phải có định nghĩa hàm nhập /xuất mảng).

Hai hàm này được sử dụng kết quả của bài toán 3.1

- Nhập giá trị X cần đếm số là xuất hiện.
- In số lần xuất hiện của X trong A. Ý tưởng:
 - o Khởi tạo biến đếm ban đầu là 0.
 - o Sử dụng vòng lặp i, lặp từ 1 đến N.
 - o Đối với mỗi phần tử A[i], nếu A[i] = X thì tăng biến đếm lên 1
 - o Kết thúc, giá trị biến đếm là số lần xuất hiện của X trong A.
- In số lần xuất hiện của các phần tử trong A.

{ 1. Dem so phan tu A[i] trong mang bang gia tri X }

Function DemPtuX(Var A : Mang20; N, X : Integer) : Integer;

Var i , Count : Integer;

Begin

Count := 0;

For i:=0 to N do

If (A[i] = X) then

Count := Count + 1;

DemPtuX := Count;

End;

{ 2. Dem so lan xuất hiện của tất cả các phần tử trong mảng }

Procedure InSoLanXHcuaPTu(A:Mang20; N: Integer);

Var i :Integer;

Begin

For i:=0 to N do

Writeln(A[i] , ' ==> ', DemPtuX(A, N, A[i]));

End;

BEGIN

```
Clrscr;
NhapMang1C( A, N );
XuatMang1C( A, N );
Write( 'Gia tri X:' );   Readln( X );
Writeln( 'So lan xuat hien trong A la:', DemPtux(A, N, X) );
InSoLanXHcuaPTu ( A, N );
Readln;
```

END .

Cải tiến: Không in ra các phần tử được lặp lại.

Hướng dẫn: Đối với mỗi phần tử, trước khi in, kiểm tra xem nó xuất hiện trước nó hay không.

- Nếu $A[i]$ chưa xuất hiện trước nó, thì in ra số lần xuất hiện của $A[i]$
- Nếu $A[i]$ có xuất hiện trước nó, thì không in ra số lần xuất hiện của $A[i]$ nữa, vì đã in ra số lần xuất hiện của phần tử có giá trị bằng $A[i]$ rồi.

Mở rộng: In ra phần tử xuất hiện ít nhất và nhiều nhất trong mảng.

Bài 4: Tìm kiếm và thay thế. Tìm kiếm vị trí xuất hiện của x trên mảng A. Thay thế những giá trị A_i là x thành y.

Ví dụ: A: 1 5 6 7 4 1 5 5 1 1
 X=5 Y=15

Kết quả: Vị trí xuất hiện X là 1

Kết quả thay thế: 1 15 6 7 4 1 15 15 1 1

Hướng dẫn:

- Xây dựng hàm tìm kiếm giá trị X trong mảng A, N phần tử. Sử dụng vòng lặp từ 0 đến N-1 để kiểm tra tất cả các giá trị A_i , nếu bằng x thì trả về vị trí i tìm thấy. Nếu thoát vòng lặp mà không tìm thấy thì trả về là -1.
- Xây dựng hàm thay thế giá trị x bằng y tại vị trí tìm thấy đầu tiên. Tương tự như tìm kiếm, nhưng khi tìm thấy thì tiến hành gán giá trị mới cho A_i là y.
- Xây dựng hàm thay thế tất cả các giá trị x bằng y tại mỗi vị trí tìm thấy. Sử dụng vòng lặp duyệt qua tất cả các giá trị của A_i , nếu A_i bằng x thì tiến hành gán thành y.

Các hàm xây dựng:

{ 1. Hàm tìm kiếm giá trị X trong mảng A với N phần tử }

Function TimKiem (A:Mang20; N:Integer; X:Integer):Boolean;

Var Flag : Boolean;

i:Integer;

Begin

Flag:=False;

For i:=0 to N do

Begin

If (A[i] = x) then

Begin

Flag:=True;

Break;

{Tim thay ==> Tra ve vi tri tim thay }

End;

End;

TimKiem:=Flag;

End;

{ 2. Thay thế phần tử X đầu tiên tìm thấy trong mảng bằng giá trị Y }

Function ThayThe(Var A:Mang20;Var N, x, y:Integer):Integer;

Var i:Integer;

Begin

For i:=0 to N do

If (A[i] = x) then

Begin

A[i] := y;

{ Tim thay x ==> thay the thanh y }

```

        Break;           { Cham dut qua trinh thay the }
    End;
    ThayThe := i;
End;
{ 3. Thay the tat ca cac phan tu co gia tri X tim thay bang gia tri Y }
Procedure ThayTheTatCa (Var A:Mang20; Var N, x,y:Integer);
Var i:Integer;
Begin
    For i:=0 to N do
        If(A[i] = x) then           { Tim thay x ==> thay the thanh y }
            A[i] := y;
End;
BEGIN
    NhapMang1C(A, N);           { Ham nhap xuat khong lam lai nua }
    XuatMang1C(A, N);           { Su du let qua o truoc }
    Write('Gia tri x:'); Readln(x);
    If (TimKiem(A,N,x)) then
        Writeln( 'Tim thay tai vi tri trong mang A.', x, k )
    Else
        Writeln( 'Khong tim thay trong mang A', x );
    Write('gia tri y:'); Readln(y);
    ThayThe(A, N, x, y);
    Writeln('Ket qua thay the ',x, y);
    XuatMang1C(A, N);
    ThayTheTatCa(A, N, x, y);
    Writeln('Ket qua thay the tat ca la:',x, y);
    XuatMang1C(A, N);
    Readln;

```

END.

Mở rộng: + Tìm kiếm các cặp 2 phần tử gần nhau có tổng chia hết cho 10. Thay thế các phần tử đó bằng tổng của chúng.

Ví dụ: A: 1 19 62 7 8 32 12

Ket qua: 20 20 62 7 40 40 12

Procedure ThayTheBangTong(Var A:Mang20; N:Integer; X, Y:Integer);

```

Var i,k:Integer;
Begin
    For i:=0 to N do
        If( (A[i-1]+A[i]) mod 10 = 0) then
            Begin
                k := (A[i-1]+A[i]);
                A[i-1] := k;
                A[i] := k;
            End;

```

End;

Bài 54: Kiểm tra mảng có đối xứng hay không? Kiểm tra mảng có tăng dần hay không?

Mảng đối xứng là mảng có phần tử $A_i = A_{N-i-1}$

Nếu mảng không phải là mảng tăng dần, hãy sắp xếp nó thành mảng tăng dần.

Ví dụ: Mảng A: 1 15 6 7 4 7 6 15 1

Kết quả: Mảng A đối xứng, Mảng A không phải là mảng tăng dần

Mảng A: 2 5 6 7 14 17 26 26 31

Kết quả: Mảng A không đối xứng, Mảng A không phải là mảng tăng dần

Hướng dẫn:

+ Xây dựng hàm **int KtraDoiXung(A, N)** để kiểm tra tính đối xứng của mảng. Ý tưởng: Giả sử mảng A là mảng đối xứng, sử dụng vòng lặp để tìm kiếm một cặp đối xứng bất kỳ nhưng lại có giá trị không bằng nhau, khi đó trả về là mảng không đối xứng. Ngược lại là không tìm thấy nên mảng là mảng đối xứng.

+ Xây dựng hàm **int KtraMangTang(A, N)** để kiểm tra xem mảng A có phải là mảng tăng hay không. Mảng tăng là mảng có các phần tử đứng sau không nhỏ hơn phần tử đứng trước nó. Ý tưởng: Giả sử mảng A là mảng tăng, sử dụng vòng lặp để kiểm tra có tồn tại phần tử nào nhỏ hơn phần tử đứng trước nó hay không, nếu có thì trả về là mảng không phải là mảng tăng (return 0). Ngược lại là không tìm thấy nên mảng là mảng tăng (return 1).

+ Xây dựng hàm **Function SxepMangTang(A, N)** để sắp xếp mảng A thành mảng tăng dần. Ý tưởng: Sử dụng 2 vòng lặp lồng nhau để kiểm tra hai phần tử tại vị trí i, j nếu $i < j$ mà $A[i] > A[j]$ thì hoán đổi giá trị của chúng.

+ Xây dựng chương trình để thể hiện kết quả đánh giá trên.

{ 1. Ham kiểm tra mang doi xung }

Function KtraDoiXung (A:Mang20; N:Integer) : Boolean;

Var Flag:Boolean;

i:Integer;

Begin

Flag:=True;

For i :=1 to N do

If(A[i] <> A[N-i +1]) Then

Flag :=False; { Cham dut kiểm tra, ket qua qua trình : khong doi xung }

KtraDoiXung :=Flag;

End;

{ 2. Ham kiểm tra mang tang }

Function KtraMangTang (A:Mang20; N :Integer) : Boolean;

Var Flag : Boolean;

i:Integer;

Begin

Flag := True;

For i :=1 to N do

If(A[i] < A[i-1]) Then

Flag :=False; { Cham dut kiểm tra, ket qua qua trình : khong tang }

KtraMangTang :=Flag;

End;

{ 3. Thu tục sap xep mang tang }

Function SxepMangTang (A:Mang20; N:Integer):Integer;

Var i ,j,k :Integer;

Begin

For i :=1 to N do

For j :=1 to N do

If ((i<j) and (A[i] > A[j])) then

Begin

k := A[i]; { Tien hanh hoan doi gia tri A[i], A[j] }

A[i] := A[j]; { cho nhau thong qua bien tam k }

A[j] := k;

End;

End;

BEGIN

Clrscr;

NhapMang1C(A, N); { Ham nhap xuất không nhắc lại nữa }

XuatMang1C(A, N); { Su dụng kết quả ở trước }

If (KtraDoiXung (A, N)) then

Writeln(' Mang A doi xung.')

```

Else
    Writeln(' Mang A khong doi xung. ');
If ( KtraMangTang (A, N ) ) then
    Writeln( 'Mang A la mang tang ')
Else
Begin
    Writeln( 'Mang A khong phai la mang tang. ');
    SxepMangTang( A, N );
    Writeln( 'Ket qua sap sep:');
    XuatMang1C(A, N);
End;
Readln;
END .

```

Mở rộng:

- + Kiểm tra mảng A chỉ chứa toàn những số nguyên tố?
- + Kiểm tra mảng giảm dần, Sắp xếp mảng giảm dần.
- + Sắp xếp mảng A có các số dương tăng dần, các số âm giảm dần.

Function SxepDuongTangAmGiam (A[]:Mang20, N:Integer);

```

Var i ,j ,k:Integer;
Begin
    For i:=1 to N do
        For j:=1 to N do
            If ( ( (i<j)and (A[i] > A[j]) and (A[i]>0) and (A[j]>0)) or
                ((i<j) and ( A[i] < A[j] ) and ( A[i]<0) and ( A[j]<0))) then
                Begin
                    k := A[i];    { Tien hanh hoan doi gia tri A[i], A[j]}
                    A[i] := A[j];    { thong qua bien tam k }
                    A[j] := k;
                End;
        End;
End;

```

- + Kiểm tra mảng A là một chuỗi cấp số cộng có công sai k = 5?

Ví dụ: 1 6 11 16 21 26 31

Function KtraMangCapSoCong (A:Mang20; N:Integer; k:Integer):Boolean;

```

Var flag :boolean;
    i :Integer;
Begin
    for i:=1 to N do
        if(A[i] <> A[i-1] + k) then
            flag:=false;    { Cham dut, ket qua: khong phai}
        KtraMangCapSoCong:=flag; {Ket quakiem tra la mang cap so cong}
    End;

```

Bài 6: Viết thủ tục và chương trình chèn phần tử X vào vị trí k trong mảng A, N phần tử. Xóa phần tử ở vị trí h trong mảng A.

Ví dụ: A : 12 2 3 6 5 17
 X = 20 , k = 3 h = 2
 Kết quả chèn: 12 2 3 20 6 5 17
 Kết quả xóa: 12 2 20 6 5 17

Hướng dẫn:

- Viết thủ tục chèn một phần tử X vào vị trí k nào đó cho mảng A (có N phần tử). Ý tưởng thuật toán:
- + Dịch chuyển các phần tử từ vị trí k đến N-1 lùi một vị trí, trở thành các phần tử từ vị trí k+1 đến N. Lưu ý, để tránh trường hợp các phần tử đè lên nhau, giải thuật phải tiến hành di dời các phần tử sau trước....đến các phần tử k sau.

- + Gán giá trị cho $A[k]$ là x .
- + Tăng số lượng phần tử của A lên 1, như thế N phải được truyền theo dạng tham biến
- Viết thủ tục xoá một phần tử ở vị trí k trên mảng A (có N phần tử). Ý tưởng thuật toán:
- + Dịch chuyển các phần tử từ vị trí k đến $N-1$ tiến về trước một vị trí, trở thành các phần tử từ vị trí $k-1$ đến $N-2$.
- + Giảm số lượng phần tử của A xuống 1, như thế N phải được truyền theo dạng tham biến.

Nội dung các thủ tục chính xây dựng:

Procedure ChenPhanTu(A:Mang20;Var N : Integer; k, X:Integer);

Var i :Integer;

Begin

For i:=N downto k+ 1 do

A[i] := A[i-1];

A[k] := X;

N:=N+1;

End;

Procedure XoaPhanTu(A:Mang20; Var N ,k:Integer);

Var i :Integer;

Begin

For i:=k to N-1 do

A[i] := A[i+1];

N:=N-1;

End;

BEGIN

NhapMang1C(A, N); { Ham nhap xuat khong lam lai nua }

XuatMang1C(A, N); { Su du let qua o truooc }

Write('Gia tri x:'); Readln(x);

Write('Vi tri k,h:'); Readln(k,h);

Writeln('Ket qua chen vao la:');

ChenPhanTu(A, N, k, x);

XuatMang1C(A, N);

Writeln('Ket qua xoa phan tu o vi tri la:', h);

XoaPhanTu(A, N, h);

XuatMang1C(A, N);

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:

$A = a_1, a_2 \dots a_k$

$B = b_1, b_2 \dots b_k$

Được $C = a_1, b_1, a_2, b_2 \dots a_k, b_k$.

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.
```

PHẦN 2: MẢNG HAI 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, 1..100] of Integer;
 C: Array['A'..'Z', 'a'..'z'] of byte;

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

II. MỘT SỐ THUẬT TOÁN LÀM VIỆC VỚI MẢNG HAI CHIỀU:

1. Câu lệnh nhập, xuất:

a. Nhập mảng:

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

b. Xuất mảng:

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

2. Sắp xếp mảng hai chiều thông qua mảng một chiều:

Ý tưởng:

- Đưa mảng hai chiều về mảng 1 chiều bằng công thức $B[(i-1)*cột+j]:=A[i,j]$;
- Sắp xếp trên mảng 1 chiều
- Đưa mảng 1 chiều về mảng hai chiều bằng công thức $A[i,j]:= B[(i-1)*cột+j]$;
- Xuất mảng hai chiều sau khi đã sắp xếp

Thuật giải:

{ Đưa mảng hai chiều về mảng 1 chiều}

```
For i:=1 to n do  
  For j:= 1 to m do
```

```

        B[(i-1)*m+j]:=A[i,j];
{Sắp xếp trên mảng 1 chiều B}
    For i:=1 to n*m-1 do
    For j:= i+1 to n*m do
    If B[i] > B[j] then
        Begin
            Tam:=B[i];
            B[i]:=B[j];
            B[j]:=tam;
        End;
{Đưa mảng 1 chiều về mảng hai chiều A}
    For i:=1 to n do
    For j:= 1 to m do
        A[i,j]:= B[(i-1)*m+j];
{Xuất mảng hai chiều sau khi đã sắp xếp}
    For i:=1 to n do
    For j:=1 to m do
        Write(a[ i,j]: 5);

```

3. Xác định phần tử lớn nhất, bé nhất:

```

    Min:=a[1,1]; max:=a[1,1];
    For i:=1 to n do
    For j:=1 to m do
    Begin
        If max<a[i,j] then max:=a[i,j];
        If min >a[i,j] then min:=a[i,j];
    End;

```

4. Các thuật toán biến đổi ma trận:

a. Xoay ma trận vuông một góc 90^0

```

    For i:=1 to n do
    For j:=1 to n do
        B[i,j]:=A[n-j+1,i];

```

b. Xây dựng ma trận chuyển vị:

```

    For i:=1 to n do
    For j:=1 to n do
        B[i,j]:=A[j,i];

```

5. Tìm phần tử Yên ngựa: Phần tử $A[i,j]$ được gọi là điểm Yên ngựa của ma trận nếu nó đồng thời vừa là phần tử lớn nhất của cột j vừa là phần tử bé nhất của dòng i .

```

uses crt;
var a:array[1..100,1..100] of integer;
    i,j,n,m:integer;
function minh(h:integer):integer;
var min:integer;
begin

```

```
min:=a[h,1];
for i:=1 to n do
  if min>a[h,i] then min:=a[h,i];
minh:=min;
end;
function maxh(h:integer):integer;
var max:integer;
begin
  max:=a[h,1];
  for i:=1 to n do
    if max<a[h,i] then max:=a[h,i];
  maxh:=max;
end;
function minc(h:integer):integer;
var min:integer;
begin
  min:=a[1,h];
  for i:=1 to n do
    if min>a[i,h] then min:=a[i,h];
  minc:=min;
end;
function maxc(h:integer):integer;
var max:integer;
begin
  max:=a[1,h];
  for i:=1 to n do
    if max<a[i,h] then max:=a[i,h];
  maxc:=max;
end;
begin
  clrscr;
  readln(n,m);
  for i:=1 to n do
    for j:=1 to m do readln(a[i,j]);
  for i:=1 to n do
    begin
      for j:=1 to m do write(a[i,j]:4);
      writeln;
    end;
  writeln;
  writeln('Cac phan tu yen ngua la:');
  for i:=1 to n do
    for j:=1 to m do
      if ((a[i,j]=minh(i)) and (a[i,j]=maxc(j)))
        or ((a[i,j]=maxh(i)) and (a[i,j]=minc(j))) then
        writeln('a[' ,i ,',' ,j ,']= ',a[i,j]);
    readln;
  end.
```

6. Tìm các điểm cực tiểu: Một phần tử trên ma trận $A[n,m]$ được gọi là phần tử cực tiểu nếu nó không lớn hơn các phần tử kề nó.

- Mỗi phần tử của ma trận có 8 phần tử kề nó;
- Các phần tử nằm ở 4 góc có 3 phần tử kề nó;
- Các phần tử trên biên có 5 phần tử kề nó.

Muốn biết $A[i,j]$ có phải là phần tử cực tiểu hay không ta so sánh nó với các phần tử kề. Vậy ta phải chia ra 3 trường hợp như trên, làm như thế rất phức tạp. Để vượt qua yếu điểm này, ta áp dụng thủ thuật nhỏ để đưa 3 trường hợp về một trường hợp tổng quát. Thủ thuật này đơn giản là ta nói rộng ma trận ở bốn phía của nó. Các phần tử nằm trong phần nói rộng này sẽ được gán trị là số lớn nhất trong miền giá trị của A. Thủ thuật này được gọi là **“Phương pháp lính canh”**.

```

program cuctieu;
uses crt;
type mang=array[0..50,0..50]of integer;
var i,j,m,n,dem:integer;
a:mang;
begin
  clrscr;
  write('Nhap so hang, so cot:');readln(n,m);
  for i:=1 to n do
    for j:=1 to m do
      begin
        write('A[' ,i ,',' ,j ,']=');readln(a[i,j]);
      end;
  for i:=0 to n+1 do
    for j:=0 to m+1 do
      begin
        a[i,0]:=32767;
        a[0,j]:=32767;
        a[n+1,i]:=32767;
        a[j,m+1]:=32767;
      end;
  dem:=0;
  writeln('Mang sau khi nhap:');
  for i:=1 to n do
    begin
      for j:=1 to m do
        write(a[i,j], ' ');
      writeln;
    end;
  writeln('Cac diem cuc tieu va vi tri:');
  for i:=1 to n do
    for j:=1 to m do
      if (a[i,j]<a[i,j-1])and(a[i,j]<a[i+1,j-1])and(a[i,j]<a[i+1,j])
        and(a[i,j]<a[i+1,j+1])and(a[i,j]<a[i,j+1])and(a[i,j]<a[i+1,j-1])
        and(a[i,j]<a[i-1,j])and(a[i,j]<a[i-1,j-1]) then
        begin
          inc(dem);
          write('a[' ,i ,',' ,j ,']= ',a[i,j]);
          writeln;
        end;
    end;
end;

```

```
if dem<>0 then write('Co tong cong ',dem,' cuc tieu')
else write('Khong co diem cuc tieu nao');
readln;
end.
```

XÂU KÝ TỰ (STRING)

I. KHAI BÁO KIỂU STRING

TYPE TênKiểu = STRING[Max];

VAR Tên biến : TênKiểu;

hoặc khai báo biến trực tiếp:

VAR Tên biến : STRING[Max];

Trong đó Max là số ký tự tối đa có thể chứa trong chuỗi (Max ∈ [0,255]). Nếu không có khai báo [Max] thì số ký tự mặc định trong chuỗi là 255.

Ví dụ:

Type Hoten = String[30];

St80 = String[80];

Var Name : Hoten;

Line : St80;

St : String; {St có tối đa là 255 ký tự}

II. TRUY XUẤT DỮ LIỆU KIỂU STRING

- Có thể sử dụng các thủ tục xuất nhập Write, Writeln, Readln để truy xuất các biến kiểu String.
- Để truy xuất đến ký tự thứ k của chuỗi ký tự, ta sử dụng cú pháp sau: Tênbiến[k].

III. CÁC PHÉP TOÁN TRÊN XÂU KÝ TỰ

3.1. Phép nối chuỗi: +

3.2. Các phép toán quan hệ: =, <>, <, <=, >, >=.

Chú ý: Các phép toán quan hệ được so sánh theo thứ tự từ điển.

IV. CÁC THỦ TỤC VÀ HÀM VỀ XÂU KÝ TỰ

4.1. Hàm lấy chiều dài của chuỗi ký tự

LENGTH(St : String):Integer;

4.2. Hàm COPY(St : String; Pos, Num: Byte): String;

Lấy ra một chuỗi con từ trong chuỗi St có độ dài Num ký tự bắt đầu từ vị trí Pos.

4.3. Hàm POS(SubSt, St :String):Byte;

Kiểm tra chuỗi con SubSt có nằm trong chuỗi St hay không? Nếu chuỗi SubSt nằm trong chuỗi St thì hàm trả về vị trí đầu tiên của chuỗi con SubSt trong chuỗi St, ngược lại hàm trả về giá trị 0.

4.4. Thủ tục DELETE(Var St:String; Pos, Num: Byte);

Xoá trong chuỗi St Num ký tự bắt đầu từ vị trí Pos.

4.5. Thủ tục INSERT(SubSt: String; Var St: String; Pos: Byte);

Chèn chuỗi SubSt vào chuỗi St bắt đầu tại vị trí Pos.

4.6. Thủ tục STR(Num; Var St:String);

Đổi số nguyên hay thực Num thành dạng chuỗi ký tự, kết quả lưu vào biến St.

4.7. Thủ tục VAL(St:String; Var Num; Var Code:Integer);

Đổi chuỗi St thành số và gán kết quả lưu vào biến Num. Nếu việc chuyển đổi thành công thì biến Code có giá trị là 0, ngược lại biến Code có giá trị khác 0 (vị trí của lỗi).

V. XÓA KÍ TỰ TRẮNG THỪA TRONG XÂU:

{Xóa các ký tự trắng ở đầu chuỗi} While St[1]=#32 Do Delete(St,1,1);

{Xóa các ký tự trắng ở cuối chuỗi} While St[Length(St)]=#32 Do Delete(St,Length(St),1);

{Xóa các ký tự trắng ở giữa chuỗi} While POS(#32#32,St)<>0 Do Delete(St,POS(#32#32,St),1);

BÀI TẬP MẪU

Bài tập 1: Viết chương trình nhập vào một chuỗi ký tự từ bàn phím. Đổi chuỗi ký tự đó sang chữ in hoa rồi in kết quả ra màn hình.

Ví dụ :Chuỗi abcdAbcd sẽ cho ra chuỗi ABCDABCD.

Uses Crt;

```
Var St:String;    i:Byte;
Begin
  Write('Nhap xau St: '); Readln(St);
  For i:=1 to length(St) do St[i]:=Ucase(St[i]);
  Write('Xau ket qua: ', St);
  Readln;
End.
```

Bài tập 2: Viết chương trình nhập vào một xâu ký tự từ bàn phím. Đổi xâu ký tự đó sang chữ thường rồi in kết quả ra màn hình.

Ví dụ :Xâu abCdAbcD sẽ cho ra xâu abcdabcd.

```
Uses Crt;
Var St:String;    i:Byte;
Begin
  Write('Nhap xau St: '); Readln(St);
  For i:=1 to length(St) do
    If St[i] IN ['A'..'Z'] Then St[i]:=CHR(ORD(St[i])+32);
  Write('Xau ket qua: ', St);
  Readln;
End.
```

Bài tập 3: Viết chương trình đếm số ký tự chữ số trong một xâu ký tự được nhập vào từ bàn phím.

```
Uses Crt;
Var St:String;    i,d:Byte;
Begin
  Write('Nhap xau St: '); Readln(St);
  For i:=1 to length(St) do
    If St[i] IN ['0'..'9'] Then d:=d+1;
  Write('So ky tu chu so trong xau: ', d);
  Readln;
End.
```

Bài tập 4: Viết chương trình nhập một xâu từ bàn phím. In ra xâu đó sau khi xóa hết các ký tự trắng thừa trong xâu. (Ký tự trắng thừa là các ký tự trắng đầu xâu, cuối xâu và nếu ở giữa xâu có 2 ký tự trắng liên tiếp nhau thì có 1 ký tự trắng thừa).

```
Uses Crt;
Var St:String;
Procedure XoaTrangThua(Var St:String);
Begin
  {Xóa các ký tự trắng ở đầu xâu}
  While St[1]=#32 Do Delete(St,1,1);
  {Xóa các ký tự trắng ở cuối xâu}
  While St[Length(St)]=#32 Do Delete(St,Length(St),1);
  {Xóa các ký tự trắng ở giữa xâu}
  While POS(#32#32,St)<>0 Do Delete(St,POS(#32#32,St),1);
End;
Begin
  Write('Nhap xau St: '); Readln(St);
  XoaTrangThua(St);
  Write('Xau sau khi xoa cac ky tu trang thua: ', St);
  Readln;
End.
```

Bài tập 5: Viết chương trình liệt kê các từ của một xâu ký tự được nhập vào từ bàn phím, mỗi từ phải được viết trên một dòng.

```
Uses Crt;
Var St:String;
Procedure XoaTrangThua(Var St:String);
```


Begin

```
{Xóa các ký tự trắng ở đầu xâu}
While St[1]=#32 Do Delete(St,1,1);
{Xóa các ký tự trắng ở cuối xâu}
While St[Length(St)]=#32 Do Delete(St,Length(St),1);
{Xóa các ký tự trắng ở giữa xâu}
While POS(#32#32,St)<>0 Do Delete(St,POS(#32#32,St),1);
```

End;

Begin

```
Write('Nhap xau St: '); Readln(St);
XoaTrangThua(St);
St:=St+#32;
Writeln('Liet ke cac tu trong xau: ');
While POS(#32,St)<>0 Do
    Begin
        Writeln(Copy(St,1,POS(#32,St)));
        Delete(St,1,POS(#32,St));
```

End;

Readln;

End.

Bài tập 6: Viết chương trình nhập vào một xâu ký tự từ bàn phím. Tìm xâu đảo ngược của xâu đó rồi in kết quả ra màn hình theo 2 cách: Đệ qui và không đệ qui.

Ý tưởng:

- Nếu xâu St có 1 ký tự thì xâu đảo = St.
- Ngược lại: Xâu đảo = Ký tự cuối + Đệ qui(Phần còn lại của xâu St).

Uses Crt;

Var St:String;

{Giải thuật không đệ qui}

Function XauDao(St:String):String;

Var S:String; i:Byte;

Begin

S:='';

For i:=Length(St) Downto 1 Do S:=S+St[i];

XauDao:=S;

End;

{Giải thuật đệ qui}

Function DeQui(St:String):String;

Begin

If Length(St)<=1 Then DeQui:=St

Else DeQui:=St[Length(St)] + DeQui(Copy(St,1,Length(St)-1));

End;

Begin

Write('Nhap xau St: '); Readln(St);

Write('Xau dao nguoc: ', XauDao(St));

Readln;

End.

Bài tập 7: Viết chương trình nhập vào một xâu ký tự từ bàn phím. Thông báo lên màn hình các chữ cái có trong xâu và số lượng của chúng (Không phân biệt chữ hoa hay chữ thường).

Ý tưởng:

- Dùng một mảng dem với chỉ số là các chữ cái để lưu trữ số lượng của các chữ cái trong xâu.
- Duyệt qua tất cả các ký tự của xâu St: Nếu ký tự đó là chữ cái thì tăng ô biến mảng dem[St[i]] lên 1 đơn vị.

Uses Crt;

Var St:String;

```

    dem: Array['A'..'Z'] Of Byte;
    i:Byte;    ch:Char;
Begin
    Write('Nhap xau St: '); Readln(St);
    {Khởi tạo mảng}
    For ch:='A' To 'Z' Do dem[ch]:=0;
    {Duyệt xâu}
    For i:=1 To Length(St) Do
        If Ucase(St[i]) IN ['A'..'Z'] Then Inc(dem[Ucase(St[i])]);
    {Liệt kê các ký tự ra màn hình}
    For ch:='A' To 'Z' Do
        If dem[ch]>0 Then Writeln(ch, ' : ',dem[ch]);
    Readln;
End.

```

Bài tập 8: Viết chương trình xóa các ký tự chữ số trong một xâu ký tự được nhập vào từ bàn phím.

```

Uses Crt;
Var St:String;
    {Hàm POSNUM kiểm tra xem trong xâu St có ký tự chữ số hay không? Nếu có, hàm trả về vị trí đầu tiên của ký tự chữ số, ngược lại hàm trả về giá trị 0}
Function POSNUM(St:String):Byte;
Var OK:Boolean;    i:Byte;
Begin
    OK:=False;
    i:=1;
    While (i<=Length(St)) AND (Not OK) Do
        If St[i] IN ['0'..'9'] Then OK:=True
        Else i:=i+1;
    If OK Then POSNUM:=i Else POSNUM:=0;
End;
Begin
    Write('Nhap xau St: '); Readln(St);
    While POSNUM(St)<>0 Do Delete(St,POSNUM(St),1);
    Write('Xau sau khi xoa: ',St);
    Readln;
End.

```

Bài tập 9: Viết chương trình để mã hoá và giải mã một xâu ký tự bằng cách đảo ngược các bit của từng ký tự trong xâu.

```

Uses crt;
Var st:string;
    {Hàm đảo bit ký tự c}
Function DaoBit(c:char):char;
Var n,i,s,bitcuoi,Mask:byte;
Begin
    {Đổi ký tự sang số}
    n:=ORD(c);
    {s: kết quả đảo bit, Mask: mặt nạ dùng để bật bit thứ i}
    s:=0;
    Mask:=128;
    For i:=1 To 8 Do {duyet qua 8 bit của n}
    Begin
        {Lấy bit cuối cùng của n: bit cực phải}
        bitcuoi:=n AND 1;
        n:=n shr 1; {loại bỏ bit cuối cùng: n:=n DIV 2}
        {Bật bit thứ i lên: từ trái sang phải}
    End

```

```

    if bitcuoi=1 then s:=s OR Mask;
    Mask:=Mask shr 1; { Mask:= Mask DIV 2}
End;
DaoBit:=CHR(s);
End;
Function MaHoa(st:string):string;
Var i:Byte;
Begin
    {Đảo bit từng ký tự trong xâu st}
    For i:=1 To Length(st) Do st[i]:=DaoBit(st[i]);
    Mahoa:=st;
End;
Begin
    Write('Nhập xâu: '); Readln(st);
    st:=MaHoa(st);
    Writeln('Xâu sau khi ma hoa: ',st);
    Readln;
    st:=MaHoa(st);
    Writeln('Xâu sau khi giai ma: ',st);
    Readln;
End.

```

Bài tập 10: Viết chương trình thực hiện phép cộng 2 số tự nhiên lớn (không quá 255 chữ số).

```

Uses crt;
Var so1,so2,kqua:string;
Procedure LamDayXau(Var st1,st2:string);
{Them so 0 vào trước xâu ngắn}
var i:Byte;
Begin
    If Length(st1)>Length(st2) Then
        For i:=1 To Length(st1)-Length(st2) Do st2:='0'+st2
    Else
        For i:=1 To Length(st2)-Length(st1) Do st1:='0'+st1;
End;
Function Cong(st1,st2:string):string;
Var i,a,b,c,sodu:Byte;
    code:integer;
    st,ch:string;
Begin
    st:=""; sodu:=0;
    LamDayXau(st1,st2);
    {Lấy từng số của 2 xâu: từ phải sang trái}
    For i:=Length(st1) DownTo 1 Do
        Begin
            {Đổi ký tự sang số nguyên}
            Val(st1[i],a,code);
            Val(st2[i],b,code);
            {Tính tổng của 2 số a,b vừa lấy ra cho vào biến c}
            c:=(a+b+sodu) MOD 10;
            {Lấy phần dư của tổng a+b}    sodu:=(a+b+sodu) DIV 10;
            {Đổi số nguyên c sang xâu ký tự ch}    str(c,ch);
            {Cộng xâu ch vào bên trái xâu kết quả st}    st:=ch+st;
        End;
    {Xử lý trường hợp số dư cuối cùng >0}
    If sodu>0 Then

```

```
Begin
  str(sodu,ch);  st:=ch+st;
End;
Cong:=st;
End;
Begin
  Write('Nhap so thu nhat: '); Readln(so1);
  Write('Nhap so thu hai: '); Readln(so2);
  kqua:=Cong(so1,so2);
  Writeln('Tong= ',kqua);
  Readln;
End.
```

CHƯƠNG VI CHUYÊN ĐỀ CHIA HẾT – SỐ NGUYÊN TỐ.

A. LÝ THUYẾT:

- Xem lại chuyên đề toán nâng cao.

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 tố nhỏ nhất lớn hơn ',n, ' là: ',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 nhatt,',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: Giữa bạn bè với nhau có tình **hữu nghị** và các con số cũng có "sự thân thiết". Một nhà toán học cổ **Hy Lạp** nói: "Ai là bạn tốt của tôi thì chúng tôi sẽ giống như hai con số **“220 và 284”**". Vậy tại sao 220 và 284 lại tượng trưng cho những người bạn thân thiết?

Thì ra, 220 ngoài bản thân nó ra, nó còn có 11 ước số là 1, 2, 4, 5, 10, 11, 20, 44, 55 và 110. Tổng của 11 ước số này vừa đúng bằng 284. Cũng vậy, 284 ngoài bản thân nó, nó còn 5 ước số khác là: 1, 2, 4, 71, 142, tổng của chúng cũng vừa đúng bằng 220. Cụ thể, $1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$ và $1 + 2 + 4 + 71 + 142 = 220$

Một số bài tập tham khảo

1/ Nhập 3 cạnh vào và cho biết là:

- Tam giác hay không?

- Tam giác thường, vuông, đều, cân!

HD: Sử dụng cấu trúc rẽ nhánh là đã hoàn thành được bài này.

```
PROGRAM tamgiac;
USES crt;
VAR a,b,c:REAL;
    vuong,can,deu:BOOLEAN;
BEGIN
    clrscr;
    REPEAT
        clrscr;
        write('Nhap 3 canh tam giac: ');readln(a,b,c);
        UNTIL ((a+b)>c) AND ((a+c)>b) AND ((b+c)>a);
        IF (a*a+b*b=c*c) OR (a*a + c*c=b*b) OR (b*b + c*c=a*a) THEN vuong:=TRUE;
        IF ((a=b) AND (b<>c)) OR ((a=c)AND (c<>b)) OR ((c=b)AND(b<>a)) THEN can:=TRUE;
        IF (a=b) AND (a=c) THEN deu:=TRUE;
        IF (vuong AND NOT can) THEN write('Tam giac vuong ');
        IF (NOT deu AND can) THEN write('Tam giac can');
        IF (vuong AND can) THEN write('Tam giac vuong can');
        IF deu THEN write('Tam giac deu');
        IF (NOT vuong) AND (NOT can) AND (NOT deu) THEN
            write('Tam giac thuong');
        readln;
    END.
```

2/ Đảo ngược chuỗi ký tự bất kì nhập vào

```
PROGRAM daoxtukitu;
USES crt;
VAR a,b:STRING;
    i,x:INTEGER;
BEGIN
    clrscr;
    write('Nhap bat ki 1 xau ki tu vao: ');
    readln(a);
    b:="";
    x:=length(a);
    FOR i:=x DOWNTO 1 DO
        b:=b+a[i];
    writeln('Xau ki tu duoc dao lai la: ',b);
    readln;
END.
```

3/ Giải PT bậc 2 (bài tập cổ điển)


```
PROGRAM ptb2;
USES crt;
VAR a,b,c,x1,x2,d:REAL;
BEGIN
  clrscr;
  REPEAT
    write('Nhap cac he so a, b, c: ');
    readln(a,b,c);
  UNTIL a<>0;
  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)
      ELSE writeln('Phuong trinh co 2 nghiem phan biet: ',x1,x2);
    END;
  readln;
END.
```

4/ Tìm số lớn nhất trong 1 dãy n phần tử:

```
PROGRAM maxdayso;
USES crt;
VAR a:ARRAY[1..100] OF REAL;
    n,i:BYTE;
    t:REAL;
BEGIN
  clrscr;
  write('Nhap so phan tu cua day vao: ');readln(n);
  FOR i:=1 TO n DO
    BEGIN
      write('Nhap so thu ',i,' vao: ');readln(a[i]);
    END;
  t:=a[1];
  FOR i:=1 TO n DO
    BEGIN
      IF a[i]>t THEN t:=a[i];
    END;
  writeln('Phan tu *** nhat cua day la: ',t:5:1);
  readln;
END.
```

5/ Sắp xếp 1 dãy số nhập vào theo thứ tự tăng dần

```
PROGRAM tangdan;
USES crt;
VAR a:ARRAY[1..100] OF REAL;
    i,j,n:BYTE;
    t:REAL;
```

```
BEGIN
  clrscr;
  write('Nhap so phan tu cua day vao: '); readln(n);
  FOR i:=1 TO n DO
    BEGIN write('Nhap phan tu thu: ',i,' vao: ');
           readln(a[i]);
    END;
  FOR j:=n DOWNTO 2 DO
  FOR i:=1 TO n-1 DO
    IF a[i+1]<a[i] THEN
      BEGIN t:=a[i];
            a[i]:=a[i+1];
            a[i+1]:=t;
      END;
    writeln('Day so sap xep tang dan la: ');
  FOR i:=1 TO n DO
    write(a[i]:5:1,' ');
  readln;
END.
```

6/ Nhập 1 ma trận vào, in ra màn hình theo hàng cột và tìm phần tử nhỏ nhất trong ma trận đó:

```
PROGRAM in_ma_tran_va_tim_phan_tu_min;
USES crt;
VAR a:ARRAY[1..100,1..100] OF REAL;
    i,j,m,n:BYTE;
    min:REAL;
BEGIN
  clrscr;
  write('Nhap so hang va so cot ma tran vao: ');
  readln(m,n);
  FOR i:=1 TO m DO
  FOR j:=1 TO n DO
    BEGIN
      write('Nhap phan tu A['',i','',j,''] vao: ');
      readln(a[i,j]);
    END;
  writeln;
  writeln('Ma tran duoc in lai la: ');
  writeln;
  FOR i:=1 TO m DO
    BEGIN
      FOR j:=1 TO n DO
        write(a[i,j]:5:1);
        writeln;
      END;
    min:=a[1,1];
  FOR i:=1 TO m DO
```

```
FOR j:=1 TO n DO
    IF a[i,j]<a[1,1] THEN min:=a[i,j];
writeln;
write('Phan tu nho nhat trong ma tran la: ',min:5:1);
readln;
END.
```

7/ Tính giá trị đa thức:

```
PROGRAM tinh_da_thuc;
USES crt;
VAR
    a:ARRAY[1..10] OF REAL;
    i,n:BYTE;
    t,x:REAL;
BEGIN
    clrscr;
    write('Nhap bac cua da thuc vao: ');
    readln(n);
    write('Nhap X vao: ');
    readln(x);
    FOR i:=n DOWNTO 0 DO
        BEGIN
            write('Nhap he so da thuc vao: ',a[i,i]: ');
            readln(a[i]);
        END;
    t:=a[i];
    FOR i:=n DOWNTO 1 DO
        t:=t*x + a[i-1];
    write('Gia tri cua da thuc la: ',t:5:1);
    readln;
END.
```

8/ Nhập vào 1 dãy n số, sau đó in số dương 1 dòng, số âm 1 dòng:

```
PROGRAM am_duong_cach_biet;
USES crt;
VAR
    a:ARRAY[1..100] OF INTEGER;
    i,n:BYTE;
BEGIN
    clrscr;
    write('Nhap so phan tu cua day vao: ');
    readln(n);
    FOR i:=1 TO n DO
        BEGIN
            write('Nhap phan tu thu: ',i,' vao: ');
            readln(a[i]);
        END;
    write('Nhưng số dương là: ');
```

```
FOR i:=1 TO n DO
  IF a[i]>0 THEN write(a[i],' ');
  writeln;
  write('Nhưng số âm là  ');
  FOR i:=1 TO n DO
    IF a[i]<0 THEN write(a[i],' ');
  readln;
END.
```

9/ Tìm ước chung lớn nhất 2 số dương

```
PROGRAM uocchung***nhat;
USES crt;
VAR x,y:WORD;
BEGIN
  clrscr;
  write('Nhập 2 số vào: ');readln(x,y);
  REPEAT
    IF x>y THEN x:=x-y ELSE y:=y-x
  UNTIL x=y;
  writeln;
  write('Ước chung *** nhất của 2 số là: ',x);
  readln;
END.
```

10/ Tính trung bình cộng riêng các số dương, âm nhập vào từ 1 mảng

```
PROGRAM tbc_am_duong_mang;
USES crt;
VAR a:ARRAY[1..100] OF REAL;
    i,n,x,y:INTEGER;
    tongam,tongduong,tba,tbd:REAL;
BEGIN
  clrscr;
  write('Số phần tử của dãy n= ');readln(n);
  FOR i:=1 TO n DO
    BEGIN
      write('Nhập phần tử a[' ,i, ']: ');
      readln(a[i]);
    END;
  tongam:=0; x:=0;
  FOR i:=1 TO n DO
    IF a[i]<0 THEN
      BEGIN tongam:=tongam+a[i];
        x:=x+1;
      END;
  tba:=tongam/x;
  write('TB cộng các số âm là: ',tba:5:1);
  writeln;
  tongduong:=0; y:=0;
```

```

FOR i:=1 TO n DO
  IF a[i]>0 THEN
    BEGIN tongduong:=tongduong+a[i];
      y:=y+1;
    END;
  tbd:=tongduong/y;
  write('TB cong cac so duong la: ',tbd:5:1);
  readln;
END.

```

11/ Tính tích vô hướng 2 Véc tơ cùng chiều với số chiều n nhập từ bàn phím:

```

PROGRAM tichvohuong2vecto;
USES crt;
VAR a:ARRAY[1..20] OF REAL;
    b:ARRAY[1..20] OF REAL;
    i,n:BYTE;
    tich:REAL;
BEGIN
  clrscr;
  write('Nhap so chieu 2 Vecto: ');readln(n);
  writeln;
  writeln('Nhap toa do Vecto thu nhât:');
  FOR i:=1 TO n DO
    BEGIN write('Nhap gia tri chieu ',i,' ');
      readln(a[i]);
    END;
  writeln;
  writeln('Nhap toa do Vecto thu 2:');
  FOR i:=1 TO n DO
    BEGIN write('Nhap gia tri chieu ',i,' ');
      readln(b[i]);
    END;
  tich:=0;
  FOR i:=1 TO n DO tich:=tich + a[i]*b[i];
  writeln;
  write('Tich vo huong 2 Vecto la: ',tich:5:1);
  readln;
END.

```

12/ Tính e^x

$\text{Exp}(x) = 1 + (x^1)/(1!) + (x^2)/(2!) + \dots (x^n)/(n!)$

Với độ chính xác $\text{ABS}((x^n)/(n!)) < 0.0001$

```

PROGRAM tinh_exp(x);
USES crt;
VAR x,s,r:REAL;
    n:INTEGER;
BEGIN
  write('x='); readln(x);

```

```
s:=1;
r:=x;
n:=1;
WHILE abs(r)>=0.0001 DO
  BEGIN  s:=s+r;
         n:=n+1;
         r:=r*x/n;
  END;
write('Exp(x)=',s:5:1);
readln;
END.
```

13/ Chọn hoa: Nhân ngày 8/3 mình đến cửa hàng bán hoa để mua hoa tặng mẹ. Mẹ mình rất thích hoa lan, cửa hàng hiện đang có n ($n \leq 20$) giò phong lan rất đẹp được đánh số từ 1 đến n , các giò phong lan đều đẹp như nhau nên mình phân vân không biết chọn giò phong lan nào. Mình muốn mua tặng mẹ k ($k \leq n$) giò phong lan. Em hãy tìm giúp mình tất cả các cách có thể chọn.

input: Tập chonhoa.inp gồm một dòng chứa 2 số nguyên n, k

output: đưa ra màn hình tất cả các cách chọn có thể và tổng số cách chọn.

Ví dụ: Input: $N = 4; k = 2$

output: 1. 1 2

2. 1 3

3. 1 4

4. 2 3

5. 2 4

6. 3 4

Tổng số cách chọn: 6

CI:

```
PROGRAM chonhoa;
uses crt;
var a: array [1..100] of integer;
n,i,j,k,kt,d: integer;
Procedure xuất;
begin
  for i:=1 to k do
    write(a[i], ' ');
  writeln;
end;
begin
  clrscr;
  write('Nhập n: ');readln(n);
  write('Nhập k: ');readln(k);
  d:=0;
  for i:=1 to k do
    a[i]:=i;
  repeat
    kt:=0;
    xuất;
```

```

    d:=d+1;
    if a[k]<n then
        begin
            a[k]:=a[k]+1;
            kt:=1;
        end
    else
        begin
            for i:=k-1 downto 1 do
                if a[i]<a[i+1]-1 then
                    begin
                        a[i]:=a[i]+1;
                        for j:=i+1 to k do
                            a[j]:=a[j-1]+1;
                        kt:=1;
                        break;
                    end;
                end;
            until kt=0;
            write('Tong so cach chon: ',d);
            readln;

```

end.

C2: Đề quy quay lùi

```

PROGRAM chonhoa;
uses crt;
Var A:array[0..100] of integer;
    n,k,i:integer;
Procedure printresult;
Var i:integer;
Begin
    For i:=1 to k do
        write(A[i]:3);
        writeln;
    End;
Procedure attempt(i:integer);
Var j:integer;
Begin
    for j:=A[i-1]+1 to n-k+i do
        Begin
            A[i]:=j;
            if i=k then printresult
            else
                attempt(i+1);
            End;
        End;
    End;
Begin
    write('n=');readln(n);
    for i:=1 to n do
        a[i]:=i;

```

```

write('k=');
readln(k);
a[0]:=0;
attempt(1);
readln;

```

End.

14/ Phân tích số

Cho số tự nhiên n ($n \leq 400$). Hãy cho biết có bao nhiêu cách phân tích số n thành tổng của dãy các số nguyên dương, các cách phân tích là hoán vị của nhau chỉ tính là một cách.

Ví dụ: Với $n=5$, có 7 cách phân tích:

1. $5 = 1 + 1 + 1 + 1 + 1$

2. $5 = 1 + 1 + 1 + 2$

3. $5 = 1 + 1 + 3$

4. $5 = 1 + 2 + 2$

5. $5 = 1 + 4$

6. $5 = 2 + 3$

7. $5 = 5$

Dùng quy hoạch động.

```

Program dem_cach_phan_tich_so;
var
  n,i,j:integer;
  f:array[0..100,0..100] of longint;
begin
  readln(n);
  for i:=1 to n do begin f[0,i]:=0; f[i,0]:=1; end;
  for i:=1 to n do
    for j:=1 to n do
      if (i>j) then f[i,j]:=f[i-1,j]
      else f[i,j]:=f[i-1,j]+f[i,j-i];
  writeln(f[n,n]);
  readln;
end.

```

15/ Có n thành phố được đánh số thứ tự 1, 2, ..., n . Giữa hai thành phố bất kì có thể có một đường đi trực tiếp hoặc không có. Cho trước ma trận kề $A = (a_{ij})$ chứa dữ liệu về các đường đi trực tiếp, trong đó a_{ij} là độ dài đường đi trực tiếp từ thành phố i đến thành phố j , $a_{ij} = 0$ nếu giữa hai thành phố không có đường đi trực tiếp. Tìm độ dài của đường đi ngắn nhất từ thành phố 1 đến thành phố n , và cho biết đường đi đó qua những thành phố nào.

```

Program Timduong;
Const Fi='graph.inp';
      Fo='graph.out';
Type mang=array[1..100,1..100] of integer;
      mang1=array[1..100] of integer;
      mang2=array[1..100] of boolean;
Var A:mang;
    D,trace:mang1;
    F:mang2;
    n:integer;

```



```

g:Text;
Procedure Load;
Var i,j:integer;
Begin
    Assign(g,Fi);
    Reset(g);
    Readln(g,n);
    For i:=1 to n do
        For j:=1 to n do
            if j=n then readln(g,A[i,j])
            else read(g,A[i,j]);
        Close(g);
    End;
Procedure init;
Var i:integer;
Begin
    d[1]:=0;
    For i:=2 to n do d[i]:=maxint;
    Fillchar(F,sizeof(F),true);
End;
Procedure process;
Var u,v,min,i:integer;
Begin
    Repeat
        u:=0;min:=maxint;
        For i:=1 to n do
            if F[i] and (d[i]<min) then
                Begin
                    min:=d[i];
                    u:=i;
                End;
        if (u=n) or (d[u]=maxint) then break;
        F[u]:=false;
        For v:=1 to n do
            if F[v] and (d[v]>d[u]+a[u,v]) then
                Begin
                    d[v]:=d[u]+a[u,v];
                    trace[v]:=u;
                End;
    Until False;
End;
procedure print;
Begin
    assign(g,Fo);
    Rewrite(g);
    if d[n]=maxint then
        writeln(g,'Khong tim duoc duong di')
    else
        Begin

```

```

        writeln(g,d[n]);
        while n<>1 do
            Begin
                write(g,n,'<-');
                n:=trace[n];
            End;
        writeln(g,1);
        End;
        Close(g);
    End;
Begin
    Load;
    Init;
    Process;
    print;
End.

```

16/ Nhập vào một số n ($5 \leq n \leq 10$) và n phần tử của dãy a, $1 < a_i < 100$ (có kiểm tra dữ liệu khi nhập).

- In ra các phần tử là số nguyên tố của dãy.
- Tính ước chung lớn nhất của tất cả các phần tử của dãy.
- Tính biểu thức sau:

$$S = a_1^1 + a_2^2 + \dots + a_n^n$$

- Sắp xếp dãy tăng dần và in ra dãy sau sắp xếp.

HƯỚNG DẪN

Ta nên chia chương trình thành các chương trình con, mỗi chương trình thực hiện một yêu cầu. Ngoài ra ta cũng viết thêm các hàm kiểm tra nguyên tố, hàm mũ, hàm UCLN để thực hiện các yêu cầu đó.

Chương trình như sau:

Khai báo dữ liệu:

```

uses crt;
var n : integer;
    a : array[1..10] of integer; {n<=10 nên mảng có tối đa 10 phần tử}

```

Thủ tục nhập dữ liệu, có kiểm tra khi nhập.

```

procedure nhap;
var i : integer;
begin
    clrscr;
    write('NHAP VAO SO PHAN TU N = ');
    repeat
        readln(n);
        if (5<=n) and (n<=10) then break; {nếu thỏa mãn thì dừng vòng lặp}
        writeln('Khong hop le (5<=n<=10). Nhap lai!!!'); {ngược lại thì báo lỗi}
    until false;

    writeln('NHAP VAO N PHAN TU (1<ai<100)');

```

```

    for i := 1 to n do begin
        write('a',i,'=');
        repeat
            readln(a[i]);
            if (1<a[i]) and (a[i]<100) then break;
            writeln('Khong hop le. Nhap lai!!!');
        until false;
    end;
end;

function ngto(n : integer): boolean; {hàm kiểm tra nguyên tố, xem giải thích ở phần trên}
var i : integer;
begin
    ngto := false;
    if n < 2 then exit;
    for i := 2 to round(sqrt(n)) do
        if n mod i = 0 then exit;
    ngto := true;
end;

```

Thủ tục in các số nguyên tố của một mảng

```

procedure inngto;
var i :integer;
begin
    writeln('CAC PHAN TU NGUYEN TO TRONG DAY:');
    for i := 1 to n do
        if ngto(a[i]) then writeln(a[i]);
        {duyet qua mọi phần tử từ 1 đến n}
        {nếu ai là nguyên tố thì in ra}
end;

function UCLN(a,b: integer): integer;
var r : integer;
begin
    while b<>0 do begin
        r := a mod b;
        a := b;
        b := r;
    end;
    UCLN := a;
end;

```

Thủ tục tính UCLN của các phần tử của một mảng

```

procedure TinhUC;
var i,u : integer;
begin
    u := a[1];
    {u là UCLN của các phần tử từ 1 đến i}
    for i := 2 to n do u := UCLN(u,a[i]);
    {là UCLN của các phần tử từ 1 đến i-1 và ai}
    writeln('UCLN của ca day la:',u);
end;

```

```
function hammu(a : real; n : integer): real; {hàm mũ tính  $a^n$ }
var s : real; i : integer;
begin
    s := 1;
    for i := 1 to n do s := s * a;
    hammu := s;
end;
```

Thủ tục tính tổng các phần tử có lũy mũ:

```
procedure tong;
var s : real; i : integer; {s phải khai báo là số thực để tránh tràn số}
begin
    s := 0;
    for i := 1 to n do s := s + hammu(a[i],i); {s := s +  $(a_i)^i$ }
    writeln('Tong can tinh:',s:10:0);
end;
```

Thủ tục sắp xếp tăng dần các phần tử của một mảng:

```
procedure sxep;
var i,j,tg : integer;
begin
    for i := 1 to n-1 do
        for j := i + 1 to n do
            if a[i] > a[j] then begin
                tg := a[i]; a[i] := a[j]; a[j] := tg;
            end;
        writeln('DAY SAU KHI SAP XEP TANG DAN:');
        for i := 1 to n do writeln(a[i]);
    end;
```

Chương trình chính: lần lượt gọi từng thủ tục

```
BEGIN
    nhap;
    inngto;
    tinhuc;
    tong;
    sxep;
END.
```

17/ Tìm phần tử nhỏ nhất, lớn nhất của một mảng (cần chỉ ra cả vị trí của phần tử).

HƯỚNG DẪN

Giả sử phần tử min cần tìm là phần tử k. Ban đầu ta cho $k=1$. Sau đó cho i chạy từ 2 đến n, nếu $a[k] > a[i]$ thì rõ ràng $a[i]$ bé hơn, ta gán k bằng i. Sau khi duyệt toàn bộ dãy thì k sẽ là chỉ số của phần tử min. (Cách tìm min này đơn giản vì từ vị trí ta cũng suy ra được giá trị).

```
procedure timmin;
var i, k : integer;
begin
    k := 1;
    for i := 2 to n do
        if a[k] > a[i] then k := i;
```

```
writeln('Phan tu nho nhat la a['k,']=',a[k]);
end;
```

Tìm max cũng tương tự, chỉ thay dấu so sánh.

```
procedure timmax;
var i, k : integer;
begin
    k := 1;
    for i := 2 to n do
        if a[k] < a[i] then k := i;
    writeln('Phan tu lon nhat la a['k,']=',a[k]);
end;
```

Chú ý:

1. Nếu áp dụng với mảng 2 chiều thì cũng tương tự, chỉ khác là để duyệt qua mọi phần tử của mảng 2 chiều thì ta phải dùng 2 vòng **for**. Và vị trí một phần tử cũng gồm cả dòng và cột.

Ví dụ 1. Tìm phần tử nhỏ nhất và lớn nhất của mảng 2 chiều và đổi chỗ chúng cho nhau:

```
procedure exchange;
var i,j,i1,j1,i2,j2,tg : integer;
begin
    i1 := 1; j1 := 1; {i1,j1 là vị trí phần tử min}
    i2 := 1; j2 := 1; {i2,j2 là vị trí phần tử max}
    for i := 1 to m do
        for j := 1 to n do begin
            if a[i1,j1] > a[i,j] then begin {so sánh tìm min}
                i1 := i; j1 := j; {ghi nhận vị trí min mới}
            end;
            if a[i2,j2] < a[i,j] then begin {so sánh tìm max}
                i2 := i; j2 := j; {ghi nhận vị trí max mới}
            end;
        end;
    end;
    tg := a[i1,j1]; a[i1,j1] := a[i2,j2]; a[i2,j2] := tg; {đổi chỗ}
end;
```

2. Nếu cần tìm phần tử lớn nhất / nhỏ nhất hoặc sắp xếp 1 dòng (1 cột) của mảng 2 chiều thì ta cũng coi dòng (cột) đó như 1 mảng 1 chiều. Chẳng hạn tất cả các phần tử trên dòng k đều có dạng chỉ số là $a[k,i]$ với i chạy từ 1 đến n (n là số cột).

Ví dụ 2. Tìm phần tử lớn nhất của dòng k và đổi chỗ nó về phần tử đầu dòng.

```
procedure timmax(k : integer);
var i, vt, tg : integer;
begin
    vt := 1; {vt là vị trí của phần tử min dòng k}
    for i := 1 to n do
        if a[k,i] > a[k,vt] then vt := i; {các phần tử dòng k có dạng a[k,i]}
    tg := a[k,1]; a[k,1] := a[k,vt]; a[k,vt] := tg;
end;
```

Ví dụ 3. Sắp xếp giảm dần cột thứ k.

```

procedure sapxep(k: integer);
var i,j,tg : integer;
begin
    for i := 1 to m-1 do {mỗi cột có m phần tử, vì bảng có m dòng}
        for j := i+1 to m do
            if a[i,k] > a[j,k] then begin {các phần tử cột k có dạng a[i,k]}
                tg := a[i,k]; a[i,k] := a[j,k]; a[j,k] := tg;
            end;
        end;
end;

```

18/ Tìm các phần tử thoả mãn 1 tính chất gì đó.**HƯỚNG DẪN**

Nếu tính chất cần thoả mãn là cần kiểm tra phức tạp (chẳng hạn: nguyên tố, hoàn thiện, có tổng chữ số bằng 1 giá trị cho trước...) thì ta nên viết một hàm để kiểm tra 1 phần tử có tính chất đó không. Còn tính chất cần kiểm tra đơn giản (chẵn / lẻ, dương / âm, chia hết, chính phương...) thì không cần.

Sau đó ta duyệt qua các phần tử từ đầu đến cuối, phần tử nào thoả mãn tính chất đó thì in ra.

Ví dụ 1. In ra các số chính phương của một mảng:

Để kiểm tra n có chính phương không, ta lấy căn n, làm tròn rồi bình phương và so sánh với n. Nếu biểu thức $\text{sqr}(\text{round}(\text{sqrt}(n))) = n$ là true thì n là chính phương.

Vậy để in các phần tử chính phương ta viết:

```

for i := 1 to n do begin
    if sqr(round(sqrt(a[i]))) = a[i] then writeln(a[i]);

```

Ví dụ 2. In ra các số hoàn thiện từ 1 đến n:

Để kiểm tra số có hoàn thiện ta dùng hàm tổng ước (đã có ở phần đầu).

```

for i := 1 to n do begin
    if tongus(i) = i then writeln(i);

```

Ví dụ 3. In ra các phần tử của mảng chia 3 dư 1, chia 7 dư 2:

```

for i := 1 to n do begin
    if (a[i] mod 3=1) and (a[i] mod 7=2) then writeln(a[i]);

```

Ví dụ 4. In ra các số có 3 chữ số, tổng chữ số bằng 20, chia 7 dư 2.

Ta dùng hàm tổng chữ số đã có ở trên:

```

for i := 100 to 999 do begin {duyệt qua mọi số có 3 chữ số}
    if (tongcs(i)=20) and (i mod 7=2) then writeln(i);

```

Chú ý: Nếu áp dụng với mảng 2 chiều thì cũng tương tự, chỉ khác là để duyệt qua mọi phần tử của mảng 2 chiều thì ta phải dùng 2 vòng **for**.

Ví dụ, để in các phần tử nguyên tố của 1 mảng 2 chiều:

```

for i := 1 to m do begin
    for j := 1 to n do begin
        if ngto(a[i,j]) then writeln(a[i,j]);

```

19/ Nhập và in mảng 2 chiều dạng ma trận (m dòng, n cột).**HƯỚNG DẪN**

Để nhập các phần tử của mảng 2 chiều dạng ma trận, ta cần dùng các lệnh sau của unit CRT (nhớ phải có khai báo `user crt` ở đầu chương trình).

`GotoXY(a,b)`: di chuyển con trỏ màn hình đến vị trí (a,b) trên màn hình (cột a, dòng b). Màn hình có 80 cột và 25 dòng.

whereX: hàm cho giá trị là vị trí cột của con trỏ màn hình.

whereY: hàm cho giá trị là vị trí dòng của con trỏ màn hình.

Khi nhập 1 phần tử ta dùng lệnh readln nên con trỏ màn hình sẽ xuống dòng, do đó cần quay lại dòng của bằng lệnh GotoXY(j * 10, whereY - 1), nếu ta muốn mỗi phần tử của ma trận ứng với 10 cột màn hình.

```
procedure nhap;
var i,j : integer;
begin
    clrscr;
    write('Nhap m,n = '); readln(m,n);
    for i := 1 to m do begin
        for j := 1 to n do begin
            write('A[' ,i ,',' ,j ,']='); readln(a[i,j]); {nhập xong thì xuống dòng}
            gotoXY(j*10,whereY-1); {di chuyển về dòng trước, vị trí tiếp theo}
        end;
        writeln; {nhập xong 1 hàng thì xuống dòng}
    end;
end;
```

Để in bảng dạng ma trận thì đơn giản hơn, với mỗi dòng ta sẽ in các phần tử trên 1 hàng rồi xuống dòng:

```
procedure inbang;
var i,j : integer;
begin
    for i := 1 to m do begin
        for j := 1 to n do write(a[i,j]:6); {viết các phần tử của hàng i }
        {mỗi phần tử chiếm 6 ô để căn phải cho thẳng cột và không sát nhau}
        writeln; {hết 1 hàng thì xuống dòng}
    end;
end;
```

20/ Nhập vào một chuỗi s khác rỗng và thực hiện chuẩn hoá chuỗi, tức là:

a) Xoá các dấu cách thừa

b) Chuyển những ký tự đầu từ thành chữ hoa, những ký tự khác thành chữ thường.

HƯỚNG DẪN

Chương trình như sau:

```
var s : string;
procedure chuanhoa(var s : string); {s là tham biến để có thể thay đổi trong chương trình con}
var i : integer;
begin
    while s[1]=' ' do delete(s,1,1); {xoá các ký tự cách thừa ở đầu chuỗi}
    while s[length(s)]=' ' do delete(s,length(s),1); {xoá các ký tự cách thừa ở cuối chuỗi}
    {xoá các ký tự cách thừa ở giữa các từ: nếu s[i-1] là cách thì s[i] là dấu cách là thừa. Phải dùng vòng lặp for downto vì nếu trong quá trình xoá ta làm giảm chiều dài của chuỗi, nếu for to sẽ không dừng được.}
    for i := length(s) downto 2 do
        if (s[i]=' ') and (s[i-1]=' ') then delete(s,i,1);
    {Chuyển ký tự đầu chuỗi thành chữ hoa}
    s[1] := Uppercase(s[1]);
    for i := 2 to length(s) do
```

```

if s[i-1]=' ' then s[i] := Upcase(s[i]) {Chuyển s[i] là kí tự đầu từ thành chữ hoa.}
else
  if s[i] in ['A'..'Z'] then      {s[i] là kí tự chữ hoa không ở đầu một từ}
    s[i] := chr(ord(s[i]) + 32); {thì phải chuyển thành chữ thường}
end;

BEGIN
  write('Nhập vào 1 xâu s:');
  readln(s);
  chuanhoa(s);
  writeln('Xâu s sau khi chuan hoa:',s);
  readln;
END.

```

21/ Nhập vào một xâu x khác rỗng và thông báo xâu đó có phải là xâu đối xứng hay không?

HƯỚNG DẪN

Xâu đối xứng nếu nó bằng chính xâu đảo của nó. Vậy cách đơn giản nhất là ta sẽ xây dựng xâu đảo của x và kiểm tra xem nó có bằng x không. Để xây dựng xâu đảo của x , cách đơn giản nhất là cộng các kí tự của x theo thứ tự ngược (từ cuối về đầu).

Chương trình:

```

var x : string;
(*****)
function doixung(x : string) : boolean; {hàm kiểm tra xâu đối xứng}
var y : string;
    i : integer;
begin
  y := '';
  {xây dựng y là xâu đảo của x, bằng cách cộng dần các kí tự của x vào y theo thứ tự ngược}
  for i := length(x) downto 1 do y := y + x[i];
  {so sánh x và xâu đảo của nó}
  if x=y then doixung := true else doixung := false;
end;
BEGIN
  write('Nhập vào 1 xâu:');
  readln(x);
  if doixung(x) then
    writeln('Xâu doi xung!')
  else
    writeln('Xau khong doi xung!');
  readln;
END.

```


22/ Nhập vào một chuỗi s và đếm xem nó có bao nhiêu từ. Từ là một dãy các ký tự, cách nhau bởi dấu cách?

HƯỚNG DẪN

Cách đếm từ đơn giản nhất là đếm dấu cách: nếu $s[i]$ là ký tự khác cách và $s[i-1]$ là ký tự cách thì chúng ta $s[i]$ là vị trí bắt đầu của một từ. Chú ý là từ đầu tiên của chuỗi không có dấu cách đứng trước.

Chương trình:

```
var s : string;
{Hàm đếm số từ của một chuỗi}
function sotu(s : string) : integer;
var i, dem : integer;
begin
{cộng thêm dấu cách phía trước chuỗi để đếm cả từ đầu tiên}
  s := ' ' + s; dem := 0;
  for i := 2 to length(s) do {s[i] là vị trí bắt đầu 1 từ}
    if (s[i-1]=' ') and (s[i]<>' ') then dem := dem + 1;
  sotu := dem;
end;
BEGIN
  write('Nhập vào 1 chuỗi:');
  readln(s);
  writeln('Số từ trong chuỗi là:',sotu(s));
  readln;
END.
```

23/ Nhập vào một chuỗi s và in ra các từ của nó (Từ là một dãy các ký tự, cách nhau bởi dấu cách). Chuỗi có bao nhiêu từ là đối xứng?

HƯỚNG DẪN

Có nhiều cách để tách một chuỗi thành các từ. Cách đơn giản nhất tiến hành như sau:

- 1) Bỏ qua các dấu cách cho đến khi gặp một ký tự khác cách (hoặc hết chuỗi).
- 2) Ghi các ký tự tiếp theo vào chuỗi tạm cho đến khi gặp dấu cách hoặc hết chuỗi, khi đó ta được 1 từ.
- 3) Nếu chưa hết chuỗi thì quay lại bước 1.

Mỗi khi tìm được một từ, ta ghi luôn nó ra màn hình, nếu từ đó là đối xứng thì tăng biến đếm. Ta cũng có thể lưu các từ tách được vào một mảng nếu bài tập yêu cầu dùng đến những từ đó trong các câu sau.

Chương trình:

```
var s : string;
    dem : integer;
{Hàm kiểm tra từ đối xứng}
function doixung(x : string) : boolean;
var y : string;
    i : integer;
begin
  y := '';
  for i := length(x) downto 1 do y := y + x[i];
  if x=y then doixung := true else doixung := false;
end;
```

```

end;
{Thủ tục thực hiện tách từ}
procedure tach;
var i, len : integer;
    t : string;
begin
    writeln('Cac tu trong xau:');
    i := 1; len := length(s);
    repeat
        {B1: bỏ qua các dấu cách cho đến khi hết xâu hoặc gặp 1 kí tự khác cách;}
        while (s[i]=' ') and (i<=len) do inc(i);
        if i>=len then break; {nếu hết xâu thì dừng}
        t := ''; {t là biến tạm lưu từ đang tách}
        {B2: lấy các kí tự khác cách đưa vào biến tạm cho đến khi hết xâu hoặc gặp 1 kí tự cách;}
        while (s[i]<>' ') and (i<=len) do begin
            t := t + s[i];
            inc(i);
        end;
        {in ra từ vừa tách được và kiểm tra đối xứng}
        writeln(t);
        if doixung(t) then inc(dem);
    until i >= len;
    writeln('So tu doi xung trong xau:',dem);
end;
(*****)
BEGIN
    write('Nhap vao 1 xau:');
    readln(s);
    tach;
END.

```

24/ Một số nguyên gọi là palindrom nếu nó đọc từ trái sang cũng bằng đọc từ phải sang. Ví dụ 121 là một số palindrom. Nhập một dãy n phần tử nguyên dương từ bàn phím, $5 \leq n \leq 20$ và các phần tử có 2 đến 4 chữ số. In ra các số là palindrom trong dãy.

Các số nguyên tố Palindrome dưới 20000 gồm:

2, 3, 5, 7, 11, 101, 131, 151, 181, 191, 313, 353, 373, 383, 727, 757, 787, 797, 919, 929, 10301, 10501, 10601, 11311, 11411, 12421, 12721, 12821, 13331, 13831, 13931, 14341, 14741, 15451, 15551, 16061, 16361, 16561, 16661, 17471, 17971, 18181, 18481, 19391, 19891, 19991.

HƯỚNG DẪN

Một số là palindrom thì xâu tương ứng của nó là xâu đối xứng. Ta sẽ xây dựng một hàm kiểm tra một số có phải là palindrom không bằng cách chuyển số đó thành xâu và kiểm tra xâu đó có đối xứng không?

Chương trình:

```

uses crt;
var n : integer;
    a : array[1..20] of integer;
{Thủ tục nhập dữ liệu}

```

```

procedure nhap;
var i : integer;
begin
  clrscr;
  repeat
    write('n= '); readln(n);
    if (n<=20) and (n>=5) then break; {nếu đã thoả mãn thì thoát khỏi vòng lặp}
    writeln('Yeu cau 5<=n<=20. Nhap lai!');
  until false;
  for i := 1 to n do
    repeat
      write('A[' ,i,']='); readln(a[i]);
      if (a[i]<=9999) and (a[i]>=10) then break; {a[i] có 2 đến 4 chữ số}
      writeln('Yeu cau cac phan tu co 2 den 4 chu so. Nhap lai!');
    until false;
  end;
  {Hàm kiểm tra bằng các kiểm tra chuỗi đối xứng}
  function palindrom(k : integer): boolean;
  var x,y : string;
      i : integer;
  begin
    str(k,x); {chuyển k thành chuỗi x}
    y := '';
    for i := length(x) downto 1 do y := y + x[i];
    {nếu x là chuỗi đối xứng thì k là palindrom}
    if x=y then palindrom := true else palindrom := false;
  end;
  {In kết quả:}
  procedure palin;
  var i : integer;
  begin
    writeln('Cac so la palindrom trong day:');
    for i := 1 to n do
      if palindrom(a[i]) then writeln(a[i]);
    readln;
  end;
  (* Chương trình chính *)
  BEGIN
    nhap;
    palin;
  END.

```

25/ Nhập một mảng 2 chiều m dòng, n cột từ file BANGSO.TXT. Cấu trúc file như sau: dòng đầu là 2 số m và n, cách nhau bằng dấu cách, m dòng sau, mỗi dòng n số nguyên.

- a) Hãy in ra những số là số nguyên tố của mảng.**
- b) Tìm vị trí phần tử lớn nhất trong mảng.**
- c) Sắp xếp mỗi dòng của mảng tăng dần và in ra mảng dạng ma trận.**

HƯỚNG DẪN

Ta khai báo một mảng 2 chiều và nhập dữ liệu từ file vào mảng. Quá trình nhập từ file văn bản giống như nhập từ bàn phím, không cần thực hiện kiểm tra dữ liệu.

Để sắp xếp mảng theo yêu cầu, ta thực hiện sắp xếp từng dòng của mảng bằng cách viết một thủ tục sắp xếp (kiểu đổi chỗ cho đơn giản) coi mỗi dòng của mảng như 1 mảng 1 chiều.

Chương trình:

```
var m,n : integer;
    a : array[1..100,1..100] of integer;
(* Nhập dữ liệu *)
procedure nhap;
var f : text;
    i,j : integer;
begin
    assign(f,'BANGSO.TXT'); reset(f);
    readln(f,m,n);
    for i := 1 to m do
        for j := 1 to n do read(f,a[i,j]);
    close(f);
end;

function ngto(k : integer): boolean;
var i : integer;
begin
    ngto := false;
    if k < 2 then exit;
    for i := 2 to round(sqrt(k)) do
        if k mod i = 0 then exit;
    ngto := true;
end;

procedure inngto;
var i,j : integer;
begin
    writeln('Cac phan tu nguyen to cua mang:');
    for i := 1 to m do
        for j := 1 to n do
            if ngto(a[i,j]) then write(a[i,j], ' ');
    writeln;
end;

procedure timmax;
var max,i,j,im,jm : integer;
begin
    max := a[1,1]; im := 1; jm := 1; {im, jm lưu tọa độ phần tử đạt max}
    for i := 1 to m do
        for j := 1 to n do
            if max < a[i,j] then begin
                max := a[i,j];    {mỗi lần gán max thì gán tọa độ luôn}
                im := i; jm := j;
            end;
end;
```

```

    end;
    writeln('Phan tu lon nhat bang la A['im','jm,']=',max);
end;
{Thủ tục thực hiện sắp xếp tăng dần dòng thứ k. Các phần tử dòng k có dạng a[k,i]}
procedure xepdong(k: integer);
var i,j, tg : integer;
begin
    for i := 1 to n do
        for j := i+1 to n do
            if a[k,i] > a[k,j] then begin
                tg := a[k,i]; a[k,i] := a[k,j]; a[k,j] := tg;
            end;
        end;
    end;

procedure sapxep;
var i,j : integer;
begin
    for i := 1 to m do xepdong(i); {sắp xếp từng dòng}
    writeln('Mang sau khi sap xep:');
    for i := 1 to m do begin {in dạng ma trận}
        for j := 1 to n do write(a[i,j] : 5); {in các phần tử trên 1 dòng}
        writeln; {in hết 1 dòng thì xuống dòng}
    end;
end;
END.
nhap;
innngto;
timmax;
sapxep;
END.

```

26/ Nhập 2 số m, n từ bàn phím, sau đó sinh ngẫu nhiên $m \times n$ số nguyên ngẫu nhiên có giá trị từ 15 đến 300 để ghi vào file BANG.TXT. Sau đó thực hiện các yêu cầu sau:

a) In $m \times n$ số đã sinh dạng ma trận m dòng, n cột.

b) In ra các số chính phương.

Yêu cầu: không được dùng mảng 2 chiều để lưu trữ dữ liệu.

HƯỚNG DẪN

Do yêu cầu không được dùng mảng 2 chiều để lưu trữ dữ liệu nên ta sẽ đọc file đến đâu, xử lý đến đấy.

- Để sinh các số ngẫu nhiên từ a đến b, ta dùng biểu thức **$a + \text{random}(b-a+1)$** .
- Để kiểm tra số k có phải là số chính phương không, ta lấy căn bậc 2 của k, làm tròn rồi bình phương. Nếu kết quả bằng k thì k là số chính phương. Tức là kiểm tra **$\text{sqr}(\text{round}(\text{sqr}(k))) = k$** .

Chương trình:

```

var m,n : integer;
    f : text;
procedure sinh;

```

```
var
  i,j : integer;
begin
  write('Nhap vao 2 so m,n: '); readln(m,n);
  assign(f,'BANG.TXT'); rewrite(f);
  writeln(f,m,' ',n);
  for i := 1 to m do begin
    for j := 1 to n do
      write(f,15 + random(300-15+1) : 6); {sinh số ngẫu nhiên từ 15 đến 300}
    writeln(f);
  end;
  close(f);
end;
{Hàm chính phương}
function cp(k : integer) : boolean;
begin
  if sqr(round(sqrt(k))) = k then cp := true
  else cp := false;
end;

procedure chinhphuong;
var
  i,j,k : integer;
begin
  assign(f,'BANG.TXT'); reset(f);
  readln(f,m,n);
  writeln('CAC SO CHINH PHUONG CUA BANG:');
  for i := 1 to m do begin
    for j := 1 to n do begin
      read(f,k);
      if cp(k) then write(k, ' '); {vừa đọc vừa xử lí}
    end;
    end;
  close(f);
end;

procedure inbang;
var
  i,j,k : integer;
begin
  assign(f,'BANG.TXT'); reset(f); {mở lại để in dạng ma trận}
  readln(f,m,n);
  writeln(#10,'IN BANG DANG MA TRAN:');
  for i := 1 to m do begin
    for j := 1 to n do begin
      read(f,k);
      write(k : 6); {đọc đến đâu in đến đó}
    end;
    writeln;
```

```
end;  
close(f);  
end;  
  
BEGIN  
  sinh;  
  chinhphuong;  
  inbang;  
END.
```

27/ Viết chương trình quản lí sách. Mỗi cuốn sách gồm tên sách, tên nhà xuất bản, năm xuất bản, giá tiền, số lượng:

- a) Đưa ra danh sách các cuốn sách của nhà xuất bản Giáo dục.
- b) Tính tổng số tiền sách.
- c) Sắp xếp danh sách theo năm xuất bản giảm dần và ghi kết quả ra màn hình.
- d) In ra màn hình các cuốn sách có giá tiền $\leq 10.000đ$ và xuất bản sau năm 2000.

HƯỚNG DẪN

Mô tả mỗi cuốn sách là một bản ghi, các thông tin về nó (tên sách, tên tác giả,...) là các trường. Danh sách cuốn sách sẽ là một mảng các bản ghi.

Khai báo kiểu dữ liệu mô tả sách như sau:

```
type  
  sach = record  
    ten : string[30];           {tên sách}  
    nxb : string[20];           {tên Nhà xuất bản}  
    namxb : integer;           {năm xuất bản}  
    soluong : integer;         {số lượng}  
    gia : real;                 {giá tiền}  
  end;
```

Thông tin của tất cả các cuốn sách ta lưu trong một mảng các bản ghi kiểu sach:

```
var  
  ds : array[1..100] of sach;  
  n : integer;
```

Nhập dữ liệu: ta nhập tên sách trước. Nếu tên sách là chuỗi rỗng thì dừng nhập, ngược lại lần lượt nhập các thông tin khác:

```
procedure nhap;  
var t : string;  
begin  
  ClrScr;  
  writeln('NHAP THONG TIN VE CAC CUON SACH');  
  writeln('(nhap ten sach la xau rong neu muon dung)');  
  repeat  
    write('Ten sach: ');  
    readln(t);  
    if t="" then break;  
    n := n + 1;  
    with ds[n] do begin
```

```
ten := t;
write('NXB: ');readln(nxb);
write('Nam xuất bản: ');readln(namxb);
write('Số lương: ');readln(soluong);
write('Giá tiền: ');readln(gia);
end;
until false;
end;
```

Câu a: ta sẽ duyệt qua toàn bộ danh sách các cuốn sách, kiểm tra nếu tên nhà xuất bản là **Giáo dục** thì in ra tất cả các thông tin của cuốn sách tương ứng:

```
procedure insach;
var
  i : integer;
begin
  Clrscr;
  writeln('CAC CUON SACH CUA NXB GIAO DUC:');
  for i:=1 to n do
    with ds[i] do
      if nxb='Giao duc' then begin
        writeln('Ten:',ten);
        writeln('Nam xuất bản:',namxb);
        writeln('Số lương:',soluong);
        writeln('Giá tiền:',gia);
      end;
    readln;
  end;
```

Câu b: ta cũng duyệt qua toàn bộ các cuốn sách, nhân số lượng và giá tiền rồi cộng dồn vào một biến tổng. Sau đó in ra biến tổng đó:

```
procedure tinh;
var i : integer;
    tong : real;
begin
  tong := 0;
  for i := 1 to n do
    with ds[i] do tong := tong + gia * soluong;
  writeln('TONG GIA TRI CUA TAT CA CAC CUON SACH:', tong:0:3);
end;
```

Câu c: Sắp xếp danh sách giảm dần theo năm xuất bản bằng phương pháp nổi bọt (2 vòng **for**). *Chú ý biến trung gian trong đối chiếu phải có kiểu **sach** thì mới gán được.*

```
procedure sxep;
var i,j : integer;
    tg : sach;
begin
  for i := 1 to n do
    for j := i + 1 to n do
      if ds[i].namxb < ds[j].namxb then begin
```



```
        tg := ds[i]; ds[i] := ds[j]; ds[j] := tg;
    end;
for i:=1 to n do
    with ds[i] do begin
        writeln('Ten:',ten);
        writeln('Nam xuất bản:',namxb);
        writeln('Số lượng:',soluong);
        writeln('Giá tiền:',gia);
    end;
    readln;
end;
```

Câu d: ta làm tương tự việc in danh sách các sách của NXB Giáo dục:

```
procedure inds;
var i : integer;
begin
    writeln('CAC CUON SACH GIA RE HON 10000 VA XUAT BAN TU NAM 2000:');
    for i := 1 to n do
        with ds[i] do
            if (gia <= 10000) and (namxb >= 2000) then writeln(ten);
    end;
```

Chương trình chính: Lần lượt gọi các chương trình con theo thứ tự:

```
BEGIN
    nhap;
    insach;
    tinh;
    sxep;
    inds;
    readln;
END.
```

28/ Viết chương trình quản lý cán bộ. Thông tin về cán bộ gồm tên, tuổi, hệ số lương, phụ cấp, thu nhập.

a) Nhập thông tin cán bộ từ file văn bản CANBO.TXT. Các thông tin gồm tên, tuổi, hệ số lương, phụ cấp, mỗi thông tin trên một dòng.

Tính thu nhập = hệ số lương × 350000đ + phụ cấp

b) Đưa ra danh sách các bộ trẻ (tuổi ≤ 30), in đầy đủ các thông tin

c) Sắp xếp tên cán bộ theo abc và ghi lên file truy cập trực tiếp SAPXEP.DAT.

d) Đọc danh sách từ file SAPXEP.DAT, in ra màn hình các cán bộ có thu nhập từ 3 triệu trở lên.

HƯỚNG DẪN

Làm tương tự bài 1, chú ý là nhập dữ liệu từ file chứ không phải từ bàn phím. Do đó không cần ghi các thông tin yêu cầu nhập ra màn hình. Hơn nữa, phải tạo trước một file văn bản là CANBO.TXT để chương trình có thể chạy mà không báo lỗi.

Toàn văn chương trình:

```
uses crt;
type
```

```
canbo = record
    ten : string[20];
    tuoi : byte;
    hsl, phucap, thunhap: real;
end;
var
    ds : array[1..100] of canbo;
    n : integer;
(*****)
procedure nhap;
var f : text;
begin
    assign(f,'CANBO.TXT'); reset(f);
    n := 0;
    while not eof(f) do begin
        n := n + 1;
        with ds[n] do begin
            readln(f,ten);
            readln(f,tuoi);
            readln(f,hsl);
            readln(f,phucap);
            thunhap := hsl * 350000 + phucap;
        end;
    end;
    close(f);
end;
(*****)
procedure in30;
var i : integer;
begin
    writeln('DANH SACH CAC CAN BO TRE:');
    for i := 1 to n do
        with ds[i] do
            if tuoi <= 30 then begin
                writeln('Ten:',ten);
                writeln('Tuoi:',tuoi);
                writeln('He so luong:',hsl :0 :3);
                writeln('Phu cap:',phucap :0 :3);
                writeln('Thu nhap:',thunhap :0 :3);
            end;
    end;
(*****)
procedure sxep;
var i,j : integer;
    tg : canbo;
begin
    for i := 1 to n do
        for j := i + 1 to n do
            if ds[i].ten > ds[j].ten then begin
```

```
        tg := ds[i]; ds[i] := ds[j]; ds[j] := tg;
    end;
end;
(*****)
procedure ghitep;
var f : file of canbo;
    i : integer;
begin
    assign(f,'SAPXEP.DAT'); rewrite(f);
    for i := 1 to n do write(f,ds[i]);
    close(f);
end;
procedure doctep;
var f : file of canbo;
    i : integer;
begin
    assign(f,'SAPXEP.DAT'); reset(f);
    i := 0;
    while not eof(f) do begin
        i := i + 1;
        read(f,ds[i]);
    end;
    n := i;
    close(f);
end;
(*****)
procedure in3M;
var i : integer;
begin
    writeln('DANH SACH CAC CAN BO CO THU NHAP CAO:');
    for i := 1 to n do
        with ds[i] do
            if thunhap >= 3000000 then begin
                writeln('Ten:',ten);
                writeln('Tuoi:',tuoi);
                writeln('Thu nhap:',thunhap :0 :3);
            end;
    end;
end;
(*****)
BEGIN
    nhap;
    in30;
    sxep;
    in3M;
    readln;
END.
```

Bài 1: Cho một xâu X có chiều dài N ký tự. Xâu chỉ gồm hai loại ký tự 'A' và 'B'. Một xâu Y được gọi là xâu con của X nếu thỏa mãn các tính chất sau:

- +Y là một đoạn ký tự liên tiếp của xâu X.
- + Có chiều dài $L \leq M$ (M cho trước)
- +Có K ký tự 'A'

Ví dụ: X là: BABABAABBB

Với M=5 và K=3 thì xâu ABAA là một xâu con

Yêu cầu: Hãy tìm số lượng xâu con của X

Dữ liệu vào: cho trong file văn bản XAUCON.INP có cấu trúc: gồm nhiều dòng

Dòng 1: chứa 3 số N M K ($0 < N < 1000000$; $0 < L \leq M \leq 14$; $0 < k \leq M$)

Trong đó: N là độ dài của xâu X, M là chiều dài lớn nhất của các xâu con Y, K là số lượng ký tự 'A'. Các số

ghi cách nhau ít nhất một dấu cách.

Dòng 2: Chứa n ký tự của xâu X. Các ký tự viết liền nhau

Dòng 3: chứa các dấu *****

Dữ liệu ra: Ghi ra file văn bản có tên: XAUCON.OUT theo cấu trúc

Dòng đầu ghi các xâu con thỏa mãn điều kiện

Dòng tiếp theo ghi tổng số xâu con

Ví dụ:

XAUCON.INP	XAUCON.OUT
5 3 2	AA AAB AAB AAB AAB
AABAA	tong so xau la: 5
*****	AABA ABAA AABAA AABAA AABAA AABAA
10 5 3	AABAA AABAA
AABAAAABAA	tong so xau la: 88

PROGRAM CT;

USES CRT;

const mm=14; kk=7; nn=100000;

fi='xaucon.inp';

fo='xaucon.ou';

var f,f1:text;

st:string;

n,m,k,d,i,soxau:longint; dem:array[1..14] of integer;

function soluong(s:string):integer; var i,dem:integer;

begin dEM:='';

for i:=1 to length(s) do

if s[i]='A' then dEM:=dEM+1;

soluong:=dEM;

end;

procedure xuly;

var i,j:integer;

c:char;

tam,s1: string;

begin

soxau:=0;

for i:=1 to 14 do dem[i]:=0;

st:='';

```
readln(f,n,m,k);
for i:=1 to m do
begin
    read(f,c);
    st:=st+c;
end;
for i:=k to m do
for j:=1 to m-i+1 do
begin
    tam:=copy(st,j,i);
    if soluong(tam)=k then
    begin
        write(f1, tam, ' ');
        soxau:=soxau+1;
    end;
end;
d:=m;
repeat read(f,c);
    st:=st+c;
    d:=d+1;
    delete(st,1,1);
    for i:=k to m do
    begin
        s1:=copy(st,m-i+1,i);
        if soluong(s1)=k then
        begin
            soxau:=soxau+1;
            write(f1, tam, ' ');
        end;
    end;
until d=n;
READLN(F);
end;
BEGIN
    assign(f1,fo); rewrite(f1);
    assign(f,fi); reset(f); while not eof(f) DO
    begin
        xuly;
        READLN(F,ST);
        d:=0;
        writeln(f1,'tong so xau la: ', soxau);
    end;
close(f);
close(f1);
readln;
end.
```

Bai 2: KÉN CHỜNG:

Vua nước Baitotxia muốn gả chồng cho con gái là công chúa Ada, công chúa muốn người chồng tương lai phải thông minh, không bủn xỉn nhưng cũng không hoang phí. Sau một thời gian dài suy nghĩ ông quyết định xây một cung điện gồm N phòng trưng bày đánh số từ 1 đến N, có M hành lang hai chiều nối trực tiếp một số cặp phòng. Trong số các phòng, phòng W là phòng vào, phòng K là phòng công chúa.

Mỗi chàng trai được phát S đồng và phải chọn một hành trình qua một dãy phòng (không nhất thiết khác

nhau), bắt đầu từ phòng vào, kết thúc phòng công chúa. Mỗi lần đến một phòng trên hành trình, cần phải trả một số tiền qui định đối với phòng đó. Chàng trai nào đó được hành trình mà khi đến phòng công chúa thì tiêu hết đúng S đồng sẽ được kén làm phò mã.

Đã nhiều năm qua, chưa có chàng trai nào trở thành phò mã và công chúa Ada vẫn còn đang chờ đợi người chồng lý tưởng của mình. Liệu bạn có thể giúp một bạn trai thân thiết đang sống tại Baitotxia không?

INPUT: file ZAM.INP

-Dòng đầu ghi 5 số nguyên dương N, M, W, K, S , $1 \leq N \leq 100$, $1 \leq M \leq 1000$, $1 \leq K, W \leq N$, $1 \leq S \leq 1000$

-Dòng thứ 2 ghi N số nguyên dương O1, O2, ..., On , $1 \leq Oi \leq 1000$, Oi là số tiền phải trả khi đến phòng i

-M dòng tiếp theo mỗi dòng ghi hai số nguyên dương X,Y với $x < y$, $1 \leq X, Y \leq N$ là 2 phòng đầu mút

của một hành lang

OUTPUT: file ZAM.OUT

-Một dòng viết dãy các phòng theo trình tự phòng vào đến phòng công chúa

ZAM.INP	ZAM.OUT
5 6 3 4 9	3 2 4
1 2 3 4 5	
2 4	
5 4	
1 5	
1 2	
2 3	
3 1	

Program kenchong;

const fi='c:\dauvao.inp';

fo='c:\daura.out';

var k, n, w: byte;

m, s: integer;

c: array[1..100, 1..100] of byte;

t: array[1..100] of integer;

x: array[1..1000] of byte;

procedure khoitao;

var f: text; i: integer; u, v: byte;

begin

fillchar(C, sizeof(C), 0);

assign(f, fi);

reset(f);

readln(f, n, m, w, k, s);

for i:=1 to n do

read (f, t[i]);

```

    for i:=1 to m do
    begin
        read(f, u, v);
        c[u,v]:=1; c[v,u]:=1;
    end;
    close(f);
end;
procedure inkq(tt:integer);
var f:text; i:integer;
begin
    assign(f, fo); rewrite(f);
    for i:= 1 to tt do write(f, x[i], ' '); close(f);
end;
procedure try(i:byte; s, tt:integer);
var j:byte;
begin
    x[tt]:=i; s:=s-t[i];
    if i=k then
        if s= 0 then inkq(tt) else exit;
    for j:= 1 to n do
        if (c[i,j]=1) and (s>=t[j]) then try(j,s, tt+1);
end;
begin
    khoitao;
    try(w,s,1);
end.

```

Bài 3: (5 điểm) Dãy con

Một dãy N các số nguyên $a_1, a_2, a_3, \dots, a_n$ được gọi là dãy chia hết hoàn toàn nếu a_j chia hết cho a_i với mọi i, j . Ví dụ: 3, 15, 60, 720 là một dãy chia hết hoàn toàn.

Một dãy con của một dãy cho trước được thiết lập bằng cách xóa đi một số phần tử nào đó của dãy. Ví dụ: Với dãy gồm 9 phần tử: 2, 3, 7, 8, 14, 39, 145, 76, 320 thì dãy 3, 7, 14, 76 là một dãy con nhưng 3, 14, 7 không phải là dãy con.

Yêu cầu: Với một dãy N số nguyên đã cho, hãy đưa ra một dãy con chia hết hoàn toàn có độ dài lớn nhất

ứng các test dữ liệu vào sau đây (mỗi test gồm 2 dòng; dòng 1: ghi số N các phần tử, dòng 2: ghi N phần tử của dãy, các phần tử cách nhau ít nhất một dấu cách)

Ví dụ: Với bộ test sau: 9

2 3 7 8 14 39 145 76 320 Thì dãy kết quả có thể là: 2 8 320

Program daychiahethtoan;

uses crt;

var x,a,kq:array[1..20] of integer;

f:text;

chuaxet: array[1..20] of boolean;

n,spt,max,i:integer;

procedure try(I:integer);

var j:integer;

begin

```
for j:=1 to n do
if chuaxet[j] then if i=1 then
begin
    x[i]:=j;
    chuaxet[j]:=false;
    spt:=spt+1;
    if spt>max then
    begin
        max:=spt;
        kq:=x;
    end;
    if i<n then
    begin
        try(i+1);
        chuaxet[j]:=true;
        spt:=spt-1;
    end;
end else if (j>x[i-1]) and (a[j] mod a[x[i-1]]=0) then
begin
    x[i]:=j;
    chuaxet[j]:=false;
    spt:=spt+1;
    if spt>max then
    begin
        max:=spt;
        kq:=x;
    end;
    if i<n then
    begin try(i+1);
        chuaxet[j]:=true;
        spt:=spt-1;
    end;
end
end;
BEGIN
    clrscr;
    spt:=0; max:=0;
    assign(f,'c:\dauvao.txt');
    reset(f);
    readln(f,n);
    for i:=1 to n do begin read(f,a[i]);
    chuaxet[i]:=true;
    write(a[i]:5);
    end; try(1);
    close(f);
    writeln;
    writeln('chieu dai cua cap so cong la',max:5);
    writeln;
    for i:=1 to max do write(a[kq[i]]:5);
```



```
readln;  
END.
```

Bài 4: Bài toán chia kẹo

Người ta chia N viên kẹo ($1 < n < 100$). Bỏ vào M túi ($1 < m < N$) cho trước sao cho số kẹo trong mỗi túi là không bằng nhau và không rỗng.

Viết chương trình tìm xem có bao nhiêu cách chia như vậy và liệt kê các cách đã chia, hoặc thông báo không chia được

Chú ý: Không tính số hoán vị, tức (1)(3)(2) và (3)(2)(1) được xem là như nhau

Ví dụ: 1) Dữ liệu vào: $N=10$ $M=4$

Dữ liệu ra 10 \rightarrow (1)(2)(3)(4) (Có 1 cách)

2) Dữ liệu vào: $N=10$, $M=3$

Dữ liệu ra: 10 \rightarrow (1) (2) (7)

10 \rightarrow (1) (3) (6)

10 \rightarrow (1) (4) (5)

10 \rightarrow (2) (3) (5) (có 4 cách)

Program baitoanchiakeo;

uses crt;

var t,x:array[0..20] of integer;

 n,m,i:integer;

 f:text;

procedure inkq(i:integer);

var j:integer;

begin

 write(n,'->');

 for j:=1 to i-1 do write('(',x[j],');');

 writeln('(',x[i], ');');

end;

procedure try(i:integer);

var j:integer;

begin

 for j:= 1 to n do if j>x[i-1] then

 begin

 x[i]:=j;

 t[i]:=t[i-1]+ j;

 if (t[i]=n) and (i=m) then inkq(i) else if t[i]<n then

 try(i+1);

 t[i]:=t[i]-x[i-1];

 end;

end;

begin

clrscr;

 assign(f,'c:\dauvao.inp');

 reset(f); readln(f,n,m);

 x[0]:=0; t[0]:=0;

 try(1);

 readln;

end.

Bài 5: Cặp số bạn bè

Cặp số (a,b) được gọi là cặp số bạn bè cùng nhóm nếu nó cùng thỏa mãn các tính chất sau

-UCLN(a,b)=1

-Số đảo của a là số b và ngược lại

-Số a và b là các số có thể gồm từ 2 đến k chữ số ($2 \leq k \leq 4$) và được tạo thành từ n ($4 \leq n \leq 9$) các chữ số của hệ thập phân cho trước khác nhau.

Input: File văn bản có tên dauvao.txt

-dòng 1: ghi số n và k

-dòng 2: chứa n chữ số thập phân cho trước

Output: File văn bản có tên là daura.txt

-Từ dòng thứ nhất và các dòng kế tiếp: liệt kê các cặp số đôi bạn (theo từng trang màn hình nếu nhiều số)

-dòng cuối: ghi tổng số đôi bạn

Ví dụ:

Dauvao.txt	Daura.txt
9	9 = 2 + 3 + 4
8	9 = 4 + 5

	8 = không phân tích được

Program csbb;

Uses crt;

var T: set of byte;

f:text;

m,n,k,i:integer;

function dao(a:integer):integer;

var tong,du:integer;

begin tong:=0;

while a<>0 do

begin

du:=a mod 10;

a:=a div 10;

tong:=tong*10+du;

end;

dao:=tong;

end;

function ucln(a,b:integer):integer;

begin

while a<>b do

if a>b then a:=a-b

else b:=b-a;

ucln:=a;

end;

function timthay(a:integer):boolean;

{Kiểm tra xem a có thuộc tập hợp T hay không }

var co: boolean; du:integer;

begin co:=true;

```

while a<>0 do begin
    du:=a mod 10;
    if not (du in T) then co:=false;
    a:=a div 10;
end;
timthay:=co;
end;
begin
    assign(f,'c:\dauvao.inp');
    reset(f);
    readln(f,n,k);
    t:=[];
    for i:=1 to n do
        begin
            read(f,m);
            t:= t + [m]; { kiểu tập hợp, thêm phần tử m vào tập hợp T }
        end;
    m:=1;
    for i:=1 to k do m:=m*10;
    for i:=1 to m-1 div 2 do
        if timthay(i) and (i<dao(i)) and (ucln(i, dao(i))=1) then
            writeln(i, ' ', dao(i));
        readln;
    end.

```

Bài 6: Viết chương trình tính $2n$, với $n \geq 300$,

Dữ liệu vào: từ tệp dauvao.inp gồm nhiều đoạn mỗi đoạn gồm 2 dòng:

Dòng đầu: số nguyên

Dòng sau: dấu ****

DAUVAO.INP	DAURA.OUT
16 ***	65536
8 ***	256
3	8

```

Program mu2n;
var i,n:integer; tich,st:string; f:text;
function nhan(s:string):string;
var i,nho,x,tich, code:integer; tam,kq:string;
begin
    nho:=0; kq:="";
    for i:=length(s) downto 1 do
        begin
            val(s[i],x,code);
            tich:=x*2+nho; nho:=tich div 10;
            tich:=tich mod 10;
            str(tich,tam); kq:=tam+kq;
        end;
    if nho=1 then kq:='1'+kq;

```

```

    nhan:=kq;
end;
begin
    assign(f,'c:\dauvao.inp');
    reset(f);
    while not eof(f) do begin
        tich:='1';
        readln(f,n);
        for i:=1 to n do tich:=nhan(tich);
        writeln(2 , 'mu ', n , ' = ' , tich);
        readln(f,st);
    end;
    close(f);
    readln;
end.

```

Bài 7: Chuỗi con thực sự:

Một chuỗi S1 có độ dài K được gọi là chuỗi con thực sự của S nếu:

- Nó gồm k kí tự và được lập nên từ các kí tự của chuỗi s bằng cách rút bớt tất cả các ký tự giống nhau trong chuỗi và không thêm một số kí tự nào khác.

- Trong S1 các kí tự là tuân theo trật tự đã có trogn S

Hãy viết chương trình tìm dãy con thực sự độ dài K của chuỗi S cho trước.

Yêu cầu:

Dữ liệu vào: S,K được nhập từ bàn phím

Dữ liệu ra: Các chuỗi con thực sự độ dài K chuỗi s đó, mỗi kết quả viết trên 1 dòng

Ví dụ: Chuỗi S: HOC SINH NAM TRUONG TRUNG HOC

K-2

Cho kết quả là: SI

SA SM

IA

IM AM

Số chuỗi con là :6

Program chuoiconthatsv;

uses crt;

var i,n,sl,max:integer; x:array[1..100] of byte;

s:string;k:integer;

c:array[1..100] of boolean;

procedure inkq;

var j:integer;

begin

for j:=1 to k do write(s[x[j]]);

writeln;

end;

procedure try(i:integer);

var j:integer;

begin

for j:=1 to n do

if c[j]=false then

if i=1 then

```
begin
    x[i]:=j;
    if i=k then inkq
    else
        begin
            c[j]:=true;
            try(i+1);
            c[j]:=false;
        end;
    end else if J>x[i-1] then
        begin x[i]:=j;
            if i=k then inkq else
                begin c[j]:=true;
                    try(i+1);
                    c[j]:=false;
                end;
            end;
        end;
end;
function xaucon(s:string):string;
var trung:boolean;
    i,j:integer;
begin
    i:=1;
    while i<=length(s) do
        begin
            j:=i+1;trung:=false;
            while j<=length(s) do
                begin
                    if s[i]=s[j] then
                        begin
                            trung:=true;
                            delete(s,j,1);
                        end
                    else j:=j+1;
                end;
            end;
            if trung then delete(s,i,1) else i:=i+1;
        end;
    i:=1; while i<=length(s) do if s[i]=' ' then delete(s,i,1) else i:=i+1;
    xaucon:=s;
end;
begin
    s:='hoc sinh nam truong trung hoc';
    writeln('xau con thuc su ', xaucon(s)); k:=3;
    s:=xaucon(s); n:=length(s);
    for i:=1 to length(s) do c[i]:=false;
    try(1);
    readln;
end.
```

Bài 8: Có n công việc đánh số từ 1 đến N. Việc thứ i cần thực hiện trong T_i đơn vị thời gian và thu được hiệu quả G_i ($1 \leq i \leq n$). Giả sử tại mỗi thời điểm ta chỉ có thể thực hiện một công việc và thời gian để chuyển từ một công sang thực hiện công việc tiếp theo là không đáng kể.

Yêu cầu: Với S đơn vị thời gian làm việc, hãy chọn ra một số công việc để thực hiện sao cho hiệu quả đạt được là lớn nhất.

Dữ liệu vào: Được cho trong tập tin văn bản dauvao.txt trong đó:

- Dòng đầu ghi hai số N và S
- Dòng thứ hai ghi N số t_1, t_2, \dots, t_n
- Dòng thứ ba ghi N số g_1, g_2, \dots, g_n

Kết quả xuất ra tập văn bản Daura.txt

- Dòng thứ nhất ghi hiệu quả đạt được
- Dòng thứ hai ghi số thứ tự các công việc được chọn

Ví dụ:

Dauvao.txt	Daura.txt
7 20	
4 6 10 5 6 10 7	
50 100 150 100 40 80 60	

Program daytangcapsong;

uses crt;

var tt,g,gt,t,a,kq,x:array[0..20] of integer;

f:text;

chuaxet: array[1..20] of boolean;

n,spt,max,i,sovat:integer;

procedure try(I:integer);

var j:integer;

begin

for j:=1 to n do

if chuaxet[j] and (j>x[i-1])then

begin

x[i]:=j;

if tt[i-1]+t[j]<=tmax then

begin

tt[i]:=tt[i-1]+ t[j];

gt[i]:=gt[i-1]+g[j];

chuaxet[j]:=false;

spt:=spt+1;

if gt[i]>max then

begin

max:=gt[i];

kq:=x;

sovat:=spt;

end;

if i<n then

begin

try(i+1);

chuaxet[j]:=true;

spt:=spt-1;

tt[i]:=tt[i-1]- t[j];

```

                gt[i]:=gt[i-1] -g[j];
            end;
        end ;
    end;
end;
begin
    clrscr;
    spt:=0; max:=0;tt[0]:=0;
    g[0]:=0; x[0]:=0;
    assign(f,'c:\dauvao.txt');
    reset(f);
    readln(f,n,tmax);
    for i:=1 to n do
        begin
            read(f,t[i]);
            chuaxet[i]:=true;
            write(t[i]:5);

        end;
        readln(f);
        for i:=1 to n do
            begin
                read(f,g[i]);
                chuaxet[i]:=true;
                write(g[i]:5);

            end;
        try(1);
        close(f);
        writeln;
        writeln(' tong gia tri la ',max:5);
        writeln;
        for i:=1 to sovat do write(kq[i]:5);
        readln;
    end.

```

Bài 9: Tô màu

Trong một bản đồ biểu diễn mạng lưới giao thông của một thành phố gồm N địa điểm ($N < 20$), một số điểm trong chúng được nối với nhau bởi các con đường. Để thuận tiện trong việc quản lý đường xá người ta thực hiện tô màu trên bản đồ. Hãy dùng số màu ít nhất để tô màu các địa điểm trên theo quy luật: 2 điểm có chung đoạn thẳng nối với nhau thì được tô bằng hai màu khác nhau

Dữ liệu vào: Trong file c:\Dauvao.inp gồm :

Dòng đầu tiên là số N (số điểm)

Các dòng tiếp theo cho biết các cặp điểm được nối với nhau

Dữ liệu ra: Trong file TOMAU.OUT gồm

N dòng: dòng I cho biết màu cần tô cho địa điểm i

Dauvao.inp	Tomau.out
------------	-----------

5	Dia diem 1 to mau 1
1 2	Dia diem 2 to mau 2
1 3	Dia diem 3 to mau 2
1 5	Dia diem 3 to mau 1
2 4	Dia diem 5 to mau 3
2 5	
3 4	
3 5	

```

Program tomau;
uses crt;
const max=200;
var a:array[1..max] of byte;
    dato,daxet:array[1..max] of boolean;
    b:array[1..max, 1..max] of byte;
    n,i,d,j:byte;
procedure input; {đọc file ra mảng B}
var i,j,t1,t2: byte;
    f:text;
begin
    assign(f, 'c:\dauvao.inp');
    reset(f);
    readln(f,n); writeln(n);
    while not eof(f) do
    begin
        readln(f, t1,t2);
        b[t1,t2]:=1;
        b[t2,t1]:=1;
    end;
    close(f);
end;
function kiemtra(dinh,mau:integer):boolean;
{ Kiểm tra xem có đỉnh nào kề với đỉnh mà được tô màu bằng mau chưa}
var i:integer;
begin
    kiemtra:=false;
    for i:= 1 to n do if (b[dinh,i]=1) and (a[i]=mau) then kiemtra:=true ;
end;
procedure xuly;
var i,mau,t,h:byte;
begin
    fillchar(a, sizeof(char),0); input;
    i:=1; { số đỉnh của đồ thị là từ 1 đến n nên ta bắt đầu tô đỉnh số 1 trước}
    { không được dùng câu lệnh For vì nếu dùng nó khi dùng câu lệnh break thì chương trình sẽ
    thoát mất}
    while i<=n do
    begin
        for mau:=1 to n do { số màu có thể tô từ 1 đến n}
            if not kiemtra(i,mau) then
                begin

```



```

                a[i]:=mau;
                dato[i]:=true;
                break;
            end;
            inc(i);
        end;
    end;
begin
    clrscr; xuly;
    for i:=1 to n do writeln(' diem ', i , 'can to mau ', a[i]);
    readln;
end.

```

Bài 10: Cho trước một xâu kí tự gồm toàn các chữ số. Hãy loại bỏ một số kí tự khỏi xâu sao cho các kí tự còn lại là một dãy giảm dần và theo đúng thứ tự đó tạo nên một số lớn nhất.

Yêu cầu:

- Tên file bài làm BAI2B.PAS
- Dữ liệu vào: Đọc từ file BAI2B.INP, gồm nhiều dòng. Mỗi dòng là một xâu kí tự .
- Dữ liệu ra: Ghi vào file BAI2B.OUT, gồm nhiều dòng. Mỗi dòng là xâu kết quả ứng với các xâu trong file dữ liệu vào.

Ví dụ:

DAUVAO.INP	DAURA.OUT
865258964125278632545	865432
4546788998254525123231	9854321

```

Program bai2b;
uses crt;
var t,x:array[0..20] of integer;
    chuaxet :array[1..30] of boolean;
    n,i:integer;  kq,s: string; max:longint;  f:text;
procedure inkq(i:integer);
var j:integer;
begin
    write(n , '=');
    for j:=1 to i-1 do write(x[j],'+');
    writeln(x[i]);
end;
function kiểmtra(i,j:integer): boolean;
begin
    kiểmtra:=true;
    if (i>=2) and (s[x[i]]<=s[j]) then kiểmtra:=false;
end;
procedure try(i:integer);
var j,code:integer; tam:longint;
begin
    for j:= 1 to n do
        if chuaxet[j] then
            begin
                if (j>x[i-1]) and (i>=2) and (s[j]<s[x[i-1]]) then
                    begin

```

```

        x[i]:=j; kq:=kq+s[j];
        val(kq,tam,code);
        if tam> max then max:=tam;
        chuaxet[j]:=false;
        if i<n then
        begin try(i+1);
            chuaxet[j]:=true;
            delete(kq, length(kq),1)
        end;
    end;
    if i= 1 then
    begin
        begin
            x[i]:=j;
            kq:=kq+ s[j];
            val(kq,tam,code);
            if tam> max then max:=tam;
            chuaxet[j]:=false;
            if i<n then
            begin try(i+1);
                chuaxet[j]:=true;
                delete(kq,length(kq),1);
            end;
        end;
    end;
end;
end;
begin
    clrscr;
    x[0]:=0; n:=10;
    t[0]:=0;max:=0;
    fillchar(chuaxet, sizeof(chuaxet), true);
    assign(f,'c:\dauvao.inp');
    reset(f);
    while not eof(f) do
    begin
        readln(f,s); writeln(s);n:=length(s); kq:="";
        try(1);
        writeln('so lon nhat sau khi xoa bot la ', max);
    end;
    close(f);
    readln;
end.

```

Bài 11: Có một cái cân đĩa (thăng bằng) và N quả cân có trọng lượng khác nhau đánh số thứ tự từ 1 đến N; quả cân thứ j có khối lượng C_j ($n \leq 30$). Người ta muốn cân một vật có trọng lượng P. Hãy tìm cách đặt vật và các quả cân lên đĩa sao cho thăng bằng và số quả cân phải dùng ít nhất.

Ví dụ:

DAUVAO.INP	DAURA.OUT
10 131	2 2 3
67 27 10 73 95 60 55 87 77	2 4 5

```
Program can;l
uses crt;
var x,a,b, truoc, sau:array[0..20] of integer;
    n,m,i,s,spt,p,traí,phai,dem,t:integer;
    f:text;
    tong:integer;
    done:boolean;
procedure xuly;
var i,j,k:integer;
begin
    if (s=p ) then
    begin
        dem:=0;
        for i:=1 to n do
            if (b[i]=-1) or (b[i]=1) then dem:=dem+1;
            if dem<t then
            begin
                traí:=0; phai:=0;
                for i:=1 to n do
                begin
                    if (b[i]=-1) then
                    begin
                        inc(traí);
                        truoc[traí]:=i;
                    end;
                    if (b[i]=1) then
                    begin
                        inc(phai);
                        sau[phai]:=i;
                    end;
                end;
            end;
            t:=dem;
        end;
        done:=true;
    end;
end;
procedure try(i:integer);
var j:integer;
begin
    if i>n then xuly else
    begin
        for j:=-1 to 1 do
        begin
            s:=s+j*a[i];
            b[i]:=j;
            try(i+1);
            s:=s-j*a[i];
        end;
    end;
end;
```

```

end;
begin
    clrscr;
    spt:=0;
    done:=false;
    assign(f,'c:\dauvao.inp');
    reset(f);
    readln(f,n,p);
    writeln('so can ben phai ', n , ' ', p);
    x[0]:=0;
    s:=0;
    fillchar(b, sizeof(b),0);
    t:=100;sau[0]:=0;
    trai:=0;
    phai:=0;
    for i:=1 to n do
    begin
        read(f,a[i]);
        write(a[i]:5);
    end;
    try(1);
    writeln;
    writeln('tong so qua can can dung la ', t);
    write('so qua can cung voi vat ', trai, ' ');
    for i:= 1 to trai do write( truoc[i], ' ');
    writeln;
    write('so qua can nguoc voi vat ', phai, ' ');
    for i:=1 to phai do write( sau[i], ' ');
    writeln;
    tong:=0;
    for i:=1 to phai do tong:=tong + a[sau[i]];
    writeln('tong trong luong cua moi ben la ', tong); readln;
end.

```

Bài 12: N con bọ được bố trí rải rác ngẫu nhiên trên các nút của một lưới ô vuông mà mỗi cạnh ô vuông bằng đơn vị. Mỗi nút của lưới ô vuông được xác định bởi cặp tọa độ nguyên (x,y). Các con bọ có thể di chuyển lên, xuống trái, phải mỗi lần một đơn vị (tương ứng với việc thay đổi các hoành độ hay trung độ 1 hay – 1 đơn vị. Các con bọ di chuyển sao cho cuối cùng chúng đứng thành đường thẳng nằm ngang, con bọ nọ cạnh con bọ kia: lúc đó các vị trí của các con bọ là (x,y), (x+1,y),..., (x+n-1,y với x,y nào đó. Giá trị nguyên của x, y cũng như thứ tự các con bọ là tùy ý.

Yêu cầu:

Tìm số lần di chuyển ít nhất để đạt được thỏa mãn yêu cầu trên.
tại mỗi nút của lưới ô vuông không thể có hơn một con bọ tại cùng một thời điểm.

Dữ liệu vào:

Ghi trên file MOLE.INP, GỒM N+1 dòng:

Dòng đầu ghi số nguyên dương N ($1 \leq n \leq 10000$), chỉ số con bọ.

Trên dòng thứ i+1 ($1 \leq i \leq n$), chứa hai số nguyên x[i], y[i] ($-10000 \leq x[i], y[i] \leq 10000$), cách nhau một khoảng trắng, chỉ hoành độ và trung độ của con bọ thứ i.

Dữ liệu ra:

Ghi trên file MOLE.OUT, gồm 1 dòng, chỉ số nguyên chỉ số bước di chuyển ít nhất cần thực hiện.

Ví dụ:

MOLE.INP	MOLE.OUT
3	4
1 0	
2 4	
3 2	

Program bo;

Uses crt;

const fi='mole.inp'; fo='mole.out';

max_mole=10000;

vocuc=maxlongint;

type maxxy=-10000..10000;

xytype= record

 x,y:maxxy;

end;

molexy=array[1..max_mole] of xytype;

var n,minpos:0..max_mole;

t: molexy;

minxy:xytype;

step:longint;

procedure readdata(filename:string);

var f:text; i:integer;

begin

 assign(f,filename);

 reset(f);readln(f,n);

 for i:=1 to n do readln(f,t[i].x,t[i].y);

 close(f);

end;

procedure swap (var a,b:xytype);

var tmp:xytype;

begin

 tmp:=a;

 a:=b;

 b:=tmp;

end;

procedure qsorty(var t:molexy; lo,hi:integer);

var i,j,mid:integer;

begin

 i:=lo;

 j:=hi;

 mid:=t[(lo+hi)div 2].y;

 repeat

 while t[i].y> mid do inc(i);

 while t[j].y <mid do dec(j);

 if i<=j then

 begin

 swap(t[i],t[j]);

 inc(i);

```
        dec(j);
    end;
until i>j;
if lo<j then qsorty(t,lo,j);
if hi>i then qsorty(t,i,hi);
end;
procedure qsortx(var t:molexy; lo,hi:integer);
var i,j,mid:integer;
begin
    i:=lo;j:=hi;mid:=t[(lo+hi)div 2].x;
    repeat
        while t[i].x> mid do inc(i);
        while t[j].y >mid do dec(j);
        if i<=j then
            begin
                swap(t[i],t[j]);
                inc(i);      dec(j);
            end;
        until i>j;
        if lo<j then qsortx(t,lo,j);
        if hi>i then qsorty(t,i,hi);
    end;
procedure findminxy(var minxy:xytype);
var i:0..max_mole;
begin
    qsorty(t,1,n); minxy.y:=t[(n+1)div 2].y;
    qsortx(t,1,n);
    for i:=0 to n-1 do dec(t[i+1].x,i);
    qsortx(t,1,n); minxy.x:=t[(n+1)div 2].x;
end;
procedure solve;
var i:1..max_mole;
begin
    findminxy(minxy);
    for i:=1 to n do inc(step, abs(t[i].x-minxy.x )+abs(t[i].y-minxy.y));
end;
procedure output;
var f: text;
begin
    assign(f,fo);
    rewrite(f);
    write(f,step);
    close(f);
end;
BEGIN
    readdata(fi);
    solve;
    output;
END.
```