

## **Bài tập dành cho bồi dưỡng học sinh Giỏi cấp Trung học cơ sở:**

### **◆ Bài 1:**

Nhập vào một mảng có n số nguyên dương khác nhau. Hãy in ra tất cả các phần tử trong mảng có giá trị nhỏ hơn giá trị lớn nhất và lớn hơn giá trị nhỏ nhất của mảng.

Dữ liệu: Cho trong file BT.inp, gồm 2 dòng:

- Dòng 1: Số n là số phần tử của mảng
- Dòng 2: Giá trị các phần tử của mảng, mỗi phần tử cách nhau 1 khoảng trắng

Kết quả: Xuất ra file BT.out, là các phần tử tìm được, mỗi phần tử viết trên 1 dòng

### **Chương trình:**

```
Type mang=array[1..100] of longint;
const fi='bt.inp';
      fo='bt.out';
var f:text;  n:integer;  A:mang;
procedure nhap;
var i:integer;
begin
    assign(f,fi);
    reset(f);
    read(f,n);
    for i:=1 to n do
        read(f,A[i]);
    close(f);
end;
procedure xuli;
var i,min,max:longint;
begin
    assign(f,fo);
    rewrite(f);
    max:=0;
    min:=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;
    for i:=1 to n do
        if (A[i]<>max) and (A[i]<>min) then
            writeln(f,A[i]);
    close(f);
end;
begin
    nhap;
```

xuli;  
end.

## ◆ Bài 2:

### Tổng lập phương

Tìm tất cả các số có 3 chữ số sao cho tổng lập phương của các chữ số bằng chính số đó

Ví dụ:  $153=1^3+5^3+3^3$

Dữ liệu xuất: Xuất ra file LP.OUT, mỗi dòng là một số tìm được

### Chương trình:

```
program bt;  
var f:text;n:longint;  
const fo='lp.out';  
procedure xuli;  
var i,j,k:integer;  
begin  
    assign(f,fo);  
    rewrite(f);  
    for i:=1 to 9 do  
        for j:=0 to 9 do  
            for k:=0 to 9 do  
                if i*i*i+j*j*j+k*k*k=i*100+j*10+k then writeln(f,i,j,k,'=',i,'^3+',j,'^3+',k,'^3');  
            close(f);  
end;  
begin  
    xuli;  
end.
```

## ◆ Bài 3:

Hè đến rồi, Nam được bố dẫn đi chơi công viên. Trong công viên có nhiều gian hàng trò chơi, trong đó Nam thích nhất là gian trò chơi chọn số. Người ta viết sẵn một số dãy số số nguyên dương không được sắp xếp (dãy này có N phần tử, các phần tử có giá trị khác nhau đôi một,  $N \leq 10000$ , các số của dãy có giá trị từ 1 đến 30000), sau đó họ đưa ra 1 số M ( $M \leq N$ ) và yêu cầu người chơi tìm số lớn thứ M trong dãy số này. Nam vốn tính toán nhanh lẹ, nhưng đứng trước câu đố này Nam cũng nói: “Phải nhờ lập trình ta mới biết nhanh được!”. Bạn hãy giúp Nam thực hiện công việc này.

Dữ liệu vào: từ file văn bản CHONSO.INP bao gồm:

- Dòng đầu tiên là số N và M
- N dòng tiếp theo, mỗi dòng một số là giá trị của các số trong dãy số

Dữ liệu ra: file văn bản CHONSO.OUT, gồm 1 dòng chứa 2 số K, L trong đó K chỉ số thứ tự của số cần tìm trong dãy và L là giá trị của số cần tìm.

Ví dụ:

CHONSO.INP
------------

CHONSO.OUT
------------

10 3	5 38
16	
57	
20	
19	
38	
41	
6	
13	
25	
32	

### **Chương trình:**

```

program bt;
type mang=array[1..10000] of integer;
var n,m:integer;a,b:mang;f:text;
const fi='chonso.inp';
      fo='chonso.out';
procedure nhap;
var i:integer;
begin
    assign(f,fi);
    reset(f);
    read(f,n,m);
    for i:=1 to n do
        read(f,A[i]);
    close(f);
end;
procedure sx;
var i,j,tam:integer;
begin
    for i:=1 to n do
        b[i]:=a[i];
    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;
end;
end;
procedure xuli;
var i:integer;
begin
    assign(f,fo);
    rewrite(f);
    for i:=1 to n do

```

```

        if b[i]=a[m] then
        begin
            write(f,i,' ',A[m]);
            close(f);
            exit;
        end;
    end;
end;
begin
    nhap;
    sx;
    xuli;
end.

```

#### ◆ **Bài 4: Dãy tuyến tính**

Một dãy tuyến tính là 1 bộ 3 có thứ tự  $(s_1, s_2, s_3)$  thỏa mãn  $s_2 - s_1 = s_3 - s_2$

Ví dụ: (1,2,3), (2,4,6), (14,21,28) là các dãy tuyến tính

Cho trước S ( $3 \leq S \leq 30$ ) số nguyên tăng thuộc đoạn  $[1, 100]$ , đếm số các dãy tuyến tính

Dữ liệu: Cho trong tập tin văn bản LSEQ.INP, dòng đầu là số nguyên S, dòng tiếp theo là S số nguyên, cách nhau ít nhất 1 khoảng trắng

Kết quả: Cho trong tập tin văn bản LSEQ.OUT, chứa số nguyên duy nhất chỉ số dãy tuyến tính tìm được.

Ví dụ:

LSEQ.INP	LSEQ.OUT
7 1 2 3 4 6 8 9	5

Giải thích:

5 dãy tuyến tính tìm được là:

1 2 3

2 3 4

2 4 6

3 6 9

4 6 8

#### **Chương trình:**

```

program b2;
type mang=array[1..30] of integer;
var n:integer;a:mang;f:text;
const fi='lseq.inp';
      fo='lseq.out';
procedure nhap;
var i:integer;
begin
    assign(f,fi);
    reset(f);
    read(f,n);
    for i:=1 to n do read(f,A[i]);
    close(f);

```

```

end;
function kt(a,b,c:integer):boolean;
begin
    if b-a=c-b then kt:=true else kt:=false;
end;
procedure xuli;
var i,j,k,bao:integer;
begin
    assign(f,fo);
    rewrite(f);
    bao:=0;
    for i:=1 to n-2 do
        for j:=i+1 to n-1 do
            for k:=j+1 to n do
                if kt(A[i],A[j],A[k]) then inc(bao);
            write(f,bao);
        close(f);
    end;
begin
    nhap;
    xuli;
end.

```

#### ◆ **Bài5:**

Nhập một dãy A có N số tự nhiên ( $N < 40$ ) và 1 số K. Hãy xuất ra các phần tử có số lần xuất hiện trong dãy A từ K lần trở lên ( Mỗi số chỉ xuất 1 lần)

Dữ liệu nhập: file DAYSO.INP:

- Dòng 1: 2 số N, K giữa 2 số cách nhau 1 khoảng trắng
- Dòng 2: Dãy A

Kết quả: file DAYSO.OUT: xuất các số thỏa điều kiện trên, trường hợp không có số nào thỏa thì xuất số -1

DAYSO.INP	DAYSO.OUT
6 2 1 2 2 3 6 6	2 6

#### **Chương trình:**

```

program bt;
type mang=array[1..50] of integer;
var n,k:integer;a:mang;f:text;
const fi='dayso.inp';
      fo='dayso.out';
procedure nhap;
var i:integer;
begin
    assign(f,fi);
    reset(f);

```

```

        read(f,n,k);
        for i:=1 to n do
            read(f,A[i]);
        close(f);
end;
procedure xuli;
var i,j,bao,t:integer;
begin
    assign(f,fo);
    rewrite(f);
    t:=0;
    for i:=1 to n do
        begin
            bao:=0;
            for j:=i to n do
                if A[j]=A[i] then inc(bao);
            if bao=k then
                begin
                    write(f,A[i],' ');
                    t:=1;
                end;
            end;
        if t=0 then write(f,'-1');
        close(f);
    end;
begin
    nhap;
    xuli;
end.

```

### ◆ **Bài 6 :**

Cho 5 số. Trong 5 số đó:

Nếu cả 5 số bằng nhau thì in ra số 1, ngược lại

Nếu có 4 số bằng nhau thì in ra số 2, ngược lại

Nếu có 3 số bằng nhau và 2 số còn lại cũng bằng nhau thì in ra số 3, ngược lại

Nếu có 3 số bằng nhau thì in ra số 4, ngược lại

Nếu có 2 cặp số bằng nhau thì in ra 5, ngược lại

Nếu có 2 số bằng nhau thì in ra 6, ngược lại in ra 7.

### **Chương trình:**

```

program bai;
type mang=array[1..5] of integer;
var f:text;a,b:mang;
const fo='test.out';

procedure nhap;
var i:integer;

```

```

begin
    randomize;
    for i:=1 to 5 do A[i]:=random(5);
end;
procedure xuli;
var i,j,k:integer;
begin
    assign(f,fo);
    rewrite(f);
    for i:=1 to 5 do write(f,A[i],' ');
    k:=0;
    for i:=1 to 4 do
        for j:=i+1 to 5 do
            if A[i]=A[j] then inc(k);
        writeln(f);
        if k=10 then write(f,1) else
            if k=6 then write(f,2) else write(f,7-k);
        close(f);
    end;
begin
    nhap;
    xuli;
end.

```

### ◆ **Bài 7:**

Nhập dãy số thực a và số k. Xét xem trong dãy có k số dương đứng cạnh nhau hay không?

Dữ liệu nhập: Nhập từ bàn phím: dãy a và số k

Dữ liệu xuất: Trả lời có hay không.

Ví dụ:

-1 2 3 4 -2 -4 4 5 -3 4 5 6

3

=> Có

### **Chương trình:**

```

uses crt;
type mang=array[1..20] of real;
var a:mang; n,k:integer;
procedure nhap( var a:mang; var n, k:integer);
var i:integer;
begin
    write('Nhập số pt của dãy: ');
    readln(n);
    for i:=1 to n do
        begin
            write('Nhập pt thu A['i,']');
            readln(a[i]);
        end;
end;

```

```

    write('Nhap vao so k can kiem tra: ');
    readln(k);
end;
function xuly(a:mang;n,k:integer): boolean;
var i,d,d1:integer;f:boolean;
begin
    d:=0;
    d1:=0;
    f:=false;
    for i:=1 to n do
        begin
            if a[i]>0 then d1:=d1+1
            else d1:=0;
            if d1=k then d:=d+1;
        end;
        if d > 0 then f:=true;
        xuly:=f;
    end;
begin
    clrscr;
    nhap(a,n,k);
    if xuly(a,n,k) then write('Co ',k,' so duong dung canh nhau.')
    else write('Khong co ',k,' so duong dung canh nhau.');
```

readln;

end.

#### ◆ **Bài 8:**

Nhập dãy số thực a. Tìm các số của dãy bằng tổng 2 số khác trong dãy.

Ví dụ:

```

          4
        6 9 4 2
=> 6 = 4 + 2
```

#### **Chương trình:**

```

uses crt;
type mang=array[1..20] of real;
var a:mang;
    n:integer;
procedure nhap( var a:mang; var n:integer);
var i:integer;
begin
    write('Nhap so pt cua day: ');
    readln(n);
    for i:=1 to n do
        begin
            write('Nhap pt thu A['i,']:');
```

readln(a[i]);



```

    end;
end;
procedure xuly(a:mang;n:integer);
var i,j,k:integer;
begin
    for i:=1 to n do
        for j:=1 to n-1 do
            for k:=j+1 to n do
                if (i<>j) and (i<>k) and (a[i]=a[j]+a[k]) then
                    writeln(a[i]:0:1,' = ', a[j]:0:1,' + ',a[k]:0:1);
            end;
        end;
    end;
begin
    clrscr;
    nhap(a,n);
    xuly(a,n);
    readln;
end.

```

### ◆ **Bài 9:**

Có bao nhiêu số có 2 chữ số có tính chất sau: Mỗi số trừ đi số viết theo thứ tự ngược lại là 1 số chính phương.

### **Chương trình:**

```

uses crt;
function chinhphuong(n:integer):boolean;
begin
    chinhphuong:=false;
    if frac(sqrt(n))=0 then chinhphuong:=true;
end;
procedure kt;
var a,k,b,c,d,dem:integer;l:boolean;
begin
    k:=10;dem:=0;
    writeln('Cac so co 2 chu so thoa man tinh chat tren la:');
    while (k<100) do
        begin
            a:=k mod 10;
            b:= k div 10;
            c:= a*10+b;
            d:=k-c;
            if d>0 then
                begin
                    if chinhphuong(d) then
                        begin
                            writeln('So: ',k,' So nguoc lai: ',c,' So-So nguoc lai: ',d);
                            writeln('So chinh phuong la: ',d);
                            dem:=dem+1;
                        end;
                end;
        end;
    k:=k+1;
end;

```

```

        end;
        k:=k+1;
    end;
    writeln('Vay co ',dem,' so thoa tinh chat tren ');
end;
begin
    clrscr;
    kt;
    readln;
end.

```

### ◆ **Bài 10:**

Nhập số thực x. Tìm phần nguyên có mấy chữ số, in ra màn hình chữ số lớn nhất.

Ví dụ:

X=12345.78

→ Phần nguyên có 5 chữ số

Chữ số lớn nhất là 5.

### **Chương trình:**

```

Uses crt;
type sn=array[1..30] of integer;
var x:real;  d:integer;
procedure nhap( var x: real);
begin
    write('Nhap 1 so thuc bat ky: ');
    readln(x);
end;
procedure dem( x: real);
var n:longint;
begin
    n:=trunc(x);
    d:=0;
    while (n<>0) do
    begin
        n:= n div 10;
        d:=d+1;
    end;
    writeln('So ',x:0:2,' phan nguyen co ',d,' chu so.');
```

```

end;
procedure tim_so_lon(x:real);
var  l,i,b,max:integer;n:longint;ma:sn;
begin
    n:=trunc(x);
    i:=1;
    repeat
        ma[i]:=n mod 10;
        i:=i+1;
```

```

        n:=n div 10;
    until n=0;
    max:=ma[1];
    for l:=2 to i-1 do
        if ma[l]>=max then max:=ma[l];
    writeln('Số lớn nhất trong các số nguyên là: ',max);
end;
begin
    clrscr;
    nhap(x);
    dem(x);
    tim_so_lon(x);
    readln;
end.

```

### ◆ **Bài 11:**

Viết chương trình nhập vào số kg gạo. Giả sử ta có 1 số bao như sau: 20 kg, 10 kg, 5 kg, 2 kg, 1 kg.

Tìm tất cả các cách cho gạo vào bao tương ứng với số gạo đã có.

Ví dụ:

- Nhập số kg gạo: 8
- Cách 1:
  - + Bao 1kg: 8
- Cách 2:
  - + Bao 2kg: 1
  - + Bao 1kg: 6
- Cách 3:
  - + Bao 5kg: 1
  - + Bao 1: 3
- Cách 4:
  - + Bao 5: 1
  - + Bao 2: 1
  - + Bao 1: 1
- Cách 5:
  - + Bao 2 kg: 4
- Cách 6:
  - + Bao 5 kg: 1
  - + Bao 1 kg: 3
- Cách 7:
  - + Bao 2 kg: 2
  - + Bao 1 kg: 4

### **Chương trình:**

```

uses crt;
var a,b20,b10,b5,b2,b1,c:integer;
begin

```

```

clrscr;
write('Nhap vao so kg gaio: ');
readln(a);
c := 1;
for b20 := 0 to a div 20 do
  for b10 := 0 to a div 10 do
    for b5 := 0 to a div 5 do
      for b2 := 0 to a div 2 do
        for b1:=0 to a do
          begin
            if b20*20+b10*10+b5*5+b2*2+b1=a then
              begin
                writeln('==> Cach: ',c);
                c := c + 1;
                if b20>0 then writeln('Bao 20: ',b20);
                if b10>0 then writeln('Bao 10: ',b10);
                if b5>0 then writeln('Bao 5: ',b5);
                if b2>0 then writeln('Bao 2: ',b2);
                if b1>0 then writeln('Bao 1: ',b1);
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;
readln;
end.

```

### ◆ **Bài 12:**

Nhập vào 1 số  $n$  ( $n > 5$ ). Hãy phân tích  $n$  thành tổng 3 số nguyên tố

Ví dụ :

6

$\Rightarrow 6 = 2 + 2 + 2$

### **Chương trình:**

```

program tinh tong;
uses crt;
var    n1:integer;
procedure nhap(var n:integer);
begin
  write('Nhap n (n>5): ');
  readln(n);
  if n<5 then
    repeat
      write('Nhap lai: ');
      readln(n);
    until(n>5);
end;
function nto(x:integer):boolean;
var i:integer;
begin
  nto:=false;

```

```

    for i:=2 to trunc(sqrt(x)) do
        if x mod i=0 then exit;
    nto:=true;
end;
procedure xet(n:integer);
var i,t,j:integer;
begin
    for i:=2 to n do
        for t:=2 to n do
            for j:=2 to n do
                if (i+t+j=n) then
                    if (nto(i)) then
                        if (nto(t)) then
                            if (nto(j)) then
                                begin
                                    writeln(n:4,' = ':3,i:4,' + ':3,t:4,' + ':3,j:4);
                                writeln;
                                end;
            end;
end;
begin
    clrscr;
    nhap(n1);
    xet(n1);
    readln;
end.

```

### ◆ **Bài 13 :**

Viết chương trình nhập vào 1 số nguyên dương  $n$  ( $0 < n \leq 32000$ ). Hãy cho biết số nguyên này là bội của tổng tất cả các chữ số của nó hay không? Nếu không thì hãy tìm một số nguyên dương  $m$  nhỏ hơn gần  $n$  nhất để cho  $m$  thỏa mãn tính chất trên?

Dữ liệu nhập: Nhập từ bàn phím

Dữ liệu xuất: Xuất ra File BAI.out.

Ví dụ 1:

Nhập: 12

➔ File BAI.out:  
Số 12 là bội của  $1 + 2$

Ví dụ 2:

Nhập: 25

➔ File BAI.out:  
Số 25 không là bội của  $2 + 5$ , số nguyên dương nhỏ hơn gần số 25 nhất là số 24 thỏa mãn tính chất trên ( Vì số 24 là bội của  $2 + 4$  )

### **Chương trình:**

```

program nguyenduong;
uses crt;
type songuyen=array[1..6] of integer;
var A:songuyen;n1,i1:integer;

```

```

function kiemtra1(n2:integer):integer;
var h:integer;
begin
    h:=0;
    while n2<>0 do
    begin
        n2:=n2 div 10;
        h:=h+1;
    end;
    kiemtra1:=h;
end;
procedure nhap(var n:integer);
begin
    write('Nhap vao mot so nguyen duong: ');
    readln(n);
    if (n<=0) and (n>32000) then
        repeat
            write('  Nhap lai mot so khac: ');
            readln(n);
        until (n>0) and (n<=32000);
end;
procedure xet(var i:integer;n:integer);
var l,k,t:integer;
begin
    i:=kiemtra1(n);
    l:=n;
    for t:=1 to i do
    begin
        A[t]:=l mod 10;
        l:=l div 10;
    end;
end;
procedure tongso(i,n:integer;var tong:integer);
var t:integer;
begin
    tong:=0;
    for t:=1 to i do
        tong:=tong+A[t];
end;
procedure boi(i,n:integer);
var t,tong1,m:integer;
begin
    tongso(i,n,tong1);
    if (n mod tong1=0) then
        begin
            write(n,' la boi cua tong ');
            for t:=i downto 1 do
                if t<>1 then write(A[t],' + ')

```

```

        else write(A[t]);
    end
else begin
    write(n,' không la boi cua tong ');
    for t:=i downto 1 do
        if t<>1 then write(A[t],' + ') else writeln(A[t]);
        m:=n;
        repeat
            m:=m-1;
            xet(i,m);
            tongso(i,m,tong1);
        until (m mod tong1=0);
        write('So nguyen duong nho hon gan so ',n,' nhat la so ',m,' thoa man dieu kien vi ',m,' la
boi cua tong ');
        for t:=i downto 1 do
            if t<>1 then write(A[t],' + ') else writeln(A[t],' ');
        end;
    end;
begin
    clrscr;
    nhap(n1);
    xet(i1,n1);
    boi(i1,n1);
    readln;
end.

```

#### ◆ **Bài 14 :\_Ma trận đều theo hàng, cột:**

Một ma trận  $n \times n$  gọi là ma trận đều theo hàng, cột nếu tổng các số hạng theo từng hàng hoặc từng cột đều bằng nhau. Hãy lập trình xem 1 ma trận có phải là ma trận đều theo hàng, cột hay không?

- Dữ liệu nhập: Nhập từ bàn phím: số nguyên  $n$  và giá trị từng phần tử của ma trận.
- Dữ liệu xuất: Xuất ra màn hình trả lời đúng hay sai.

Ví dụ:

```

    3
    1  6  7
    2  7  5
    11 1  2
=> Đúng vì tổng hàng hay tổng cột đều bằng 14.

```

#### **Chương trình:**

```

program matranguyen;
uses crt;
type songuyen=array[1..60,1..60] of integer;
    songuyen1=array[1..60] of integer;
var A:songuyen; n1:integer; B:songuyen1;
procedure nhap(var n:integer);
var i,t:integer;

```

```

begin
    write('Nhap vao so hang cua ma tran vuong: ');
    readln(n);
    writeln('Nhap vao gia tri cua ma tran ');
    for i:=1 to n do
        for t:=1 to n do
            begin
                write('A['i,',',t,']= ');
                readln(A[i,t]);
                writeln;
            end;
        end;
    end;
    procedure tinhhtong(n:integer);
    var i,y,t,h:integer;
    begin
        y:=1;
        for i:=1 to n do
            begin
                h:=0;
                for t:=1 to n do
                    h:=h+A[i,t];
                B[y]:=h;
                y:=y+1;
            end;
        for i:=1 to n do
            begin
                h:=0;
                for t:=1 to n do
                    h:=h+A[t,i];
                B[y]:=h;
                y:=y+1;
            end;
        end;
    end;
    procedure xet(n:integer);
    var c:boolean;d,i:integer;
    begin
        c:=true;
        d:=B[1];
        for i:=2 to n*2 do
            if B[i]<>d then
                begin
                    c:=false;
                    break;
                end;
            if c=true then
                begin
                    writeln('Day dung la ma tran deu');
                    writeln('Vi no co tong tung hang cot deu la ',d);
                end;
            end;
    end;

```



```

        end
    else writeln('Day khong phai la ma tran deu');
end;
begin
    clrscr;
    nhap(n1);
    tinh tong(n1);
    xet(n1);
    readln;
end.

```

### ◆ **Bài 15:**

Ma trận vuông A gọi là ma phương nếu tổng mỗi hàng, tổng mỗi cột, tổng mỗi đường chéo bằng nhau. Nhập ma trận A, kiểm tra tính chất ma phương.

#### **Chương trình:**

```

uses crt;
type matran=array[1..100,1..100] of integer;
var a: matran;  n:integer;
procedure nhap;
var i,j:integer;
begin
    write('Nhap so hang, cot cua ma tran: ');
    readln(n);
    for i:=1 to n do
        for j:=1 to n do
            begin
                write('Nhap gt pt thu A['i,',',j,']:');
                read(a[i,j]);
            end;
        end;
end;
function tong1hang( i:integer): integer;
var k,s:integer;
begin
    s:=0;
    for k:=1 to n do
        s:=s+a[i,k];
    tong1hang:=s;
end;
function tong1cot(j:integer):integer;
var k,s:integer;
begin
    s:=0;
    for k:=1 to n do
        s:=s+a[k,j];
    tong1cot:=s;
end;

```

```

procedure xuly;
var s,s1,s2,i,j:integer;
begin
  s:=tong1hang(1);
  s1:=0;
  s2:=0;
  for i:=1 to n do
  begin
    if ((s<>tong1hang(i)) or (s<>tong1cot(i))) then
    begin
      writeln('Khong phai ma phuong!');
      exit;
    end;
    s1:=s1+a[i,i];
    s2:=s2+a[i,n-i+1];
  end;
  if (s<>s1) or (s<>s2) then writeln('Khong phai ma phuong!')
  else writeln('Do la ma phuong');
end;
procedure xuat;
var i,j:integer;
begin
  for i:=1 to n do
  begin
    for j:=1 to n do
      write(a[i,j]:4);
    writeln;
  end;
end;
begin
  clrscr;
  nhap;
  xuat;
  xuly;
  readln;
end.

```

### ◆ **Bài 16:**

Một ma trận gọi là thừa nếu số phần tử 0 chiếm quá một nửa. Cho ma trận các số thực và kiểm tra tính chất thừa của ma trận đó.

Dữ liệu nhập: cho trong file Bai.inp gồm:

- Dòng 1: Số nguyên n là số dòng và m là số cột của ma trận
- Các dòng còn lại là giá trị từng phần tử của ma trận.

Dữ liệu xuất: Xuất ra file Bai.out gồm 1 dòng duy nhất trả lời đúng hay sai.

Ví dụ: **FILE BAI.INP**

3 3

```

0  4  0
0  0  0
0  8  7

```

**FILE BAI.OUT:**

Đúng là ma trận thưa

### **Chương trình:**

```

program vd;
const fi='bai.inp';
      fo='bai.out';
type ar=array[1..15,1..15]of real;
var n,m:byte;  a:ar;  f:text;
procedure nhap;
var i,j:byte;
begin
  assign(f,fi);
  reset(f);
  read(f,n,m);
  for i:=1 to n do
    for j:=1 to m do
      read(f,a[i,j]);
  close(f);
end;
procedure xl;
var i,j,d:byte;
begin
  assign(f,fo);
  rewrite(f);
  d:=0;
  for i:=1 to n do
    for j:=1 to m do
      if a[i,j]=0 then inc(d);
  if d*2>n*m then writeln(f,'dung, la ma tran thua')
  else writeln(f,'sai, khong la ma tran thua');
  close(f);
end;
begin
  nhap;
  xl;
end.

```

### **◆ Bài 17:**

Nhập ma trận A các số nguyên. Tìm vị trí các phần tử đạt max.

Dữ liệu nhập: cho trong file Bai.inp gồm:

- Dòng 1: Số nguyên n là số dòng và m là số cột của ma trận
- Các dòng còn lại là giá trị từng phần tử của ma trận.

Dữ liệu xuất: Xuất ra file Bai.out gồm các dòng là vị trí của các phần tử đạt max.

Ví dụ:

**FILE BAI.INP:**

```
3 4
2 4 5 3
2 7 9 9
9 3 2 1
```

**FILE BAI.OUT:**

Vị trí các phần tử đạt max:

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

**Chương trình:**

```
program vd;
const fi='bai.inp';
      fo='bai.out';
type ar=array[1..50,1..50]of real;
var n,m:byte;  a:ar;  f:text;
procedure nhap;
var i,j:byte;
begin
  assign(f,fi);
  reset(f);
  read(f,n,m);
  for i:=1 to n do
    for j:=1 to m do
      read(f,a[i,j]);
  close(f);
end;
function timmax:real;
var i,j:byte;  max:real;
begin
  max:=a[1,1];
  for i:=1 to n do
    for j:=1 to m do
      if a[i,j]>max then
        max:=a[i,j];
  timmax:=max;
end;
procedure xl;
var i,j,d:byte;  max:real;
begin
  assign(f,fo);
  rewrite(f);
  max:=timmax;
  for i:=1 to n do
    for j:=1 to m do
      if a[i,j]=max then
```

```

        writeln(f,i,' ',j);
    close(f);
end;
begin
    nhap;
    xl;
end.

```

#### ◆ **Bài 18:**

Nhập ma trận A các số thực. Xét xem:

- a) Trong ma trận có số dương hay không ?
- b) Tất cả các phần tử của ma trận đều dương ?

Dữ liệu nhập: Nhập từ bàn phím.

Dữ liệu xuất: Xuất ra file BAI.OUT:

- Dòng 1: Trả lời câu a có hay không.
- Dòng 2: Trả lời câu b có hay không.

Ví dụ:

SỐ DÒNG: 3

SỐ CỘT: 3

1 -9 0

-1 0 1

-2 2 4

**FILE BAI.OUT:**

a) Có

b) Không

#### **Chương trình:**

```

program vd;
const fo='bai.out';
type ar=array[1..50,1..50]of real;
var n,m:byte;  a:ar;  f:text;
procedure nhap;
var i,j:byte;
begin
    write('moi nhap so dong:');
    readln(n);
    write('moi nhap so cot:');
    readln(m);
    for i:=1 to n do
        for j:=1 to m do
            begin
                write('a[' ,i,' ',j,'=');
                readln(a[i,j]);
            end;
    end;
end;
function kt1(a:ar;n,m:byte):boolean;
var i,j:byte;
begin

```

```

    kt1:=false;
    for i:=1 to n do
        for j:=1 to m do
            if a[i,j]>0 then kt1:=true;
        end;
    end;
function kt2(a:ar;n,m:byte):boolean;
var i,j:byte;
begin
    kt2:=true;
    for i:=1 to n do
        for j:=1 to m do
            if a[i,j]<0 then kt2:=false;
        end;
    end;
begin
    nhap;
    assign(f,fo);
    rewrite(f);
    if kt1(a,n,m) then writeln(f,'a co')else writeln(f,'a khong');
    if kt2(a,n,m) then writeln(f,'b co')else writeln(f,'b khong');
    close(f);
end.

```

### **Bài 19:**

Nhập ma trận A nguyên. Kiểm tra xem có 2 hàng giống nhau hay không?

Dữ liệu nhập: cho trong file Bai.inp gồm:

- Dòng 1: Số nguyên n là số dòng và m là số cột của ma trận
- Các dòng còn lại là giá trị từng phần tử của ma trận.

Dữ liệu xuất: Xuất ra file bai.out

Ví dụ:

FILE BAI.INP:

```

3 4
1 1 2 2
1 1 2 2
1 2 1 1

```

➔ FILE BAI.OUT: CÓ

### **Chương trình:**

```

program vd;
const fi='bai.inp';
      fo='bai.out';
type ar=array[1..50,1..50]of byte;
      arr=array[1..50]of byte;
var n,m:byte;  a:ar;  f:text;
procedure nhap;
var i,j:byte;
begin
    assign(f,fi);
    reset(f);

```

```

    read(f,n,m);
    for i:=1 to n do
        for j:=1 to m do
            read(f,a[i,j]);
        close(f);
    end;
    procedure sx(n:byte;var b:arr);
    var i,j:byte;
    begin
        for i:=1 to m do
            b[i]:=a[n,i];
        end;
    procedure xl;
    var i,j,k:byte;
        b:arr;
        kt:boolean;
    begin
        assign(f,fo);
        rewrite(f);
        for i:=1 to n do
            begin
                k:=0;
                for j:=1 to m do
                    begin
                        sx(i,b);
                        inc(k);
                        if a[i,j]=b[k] then
                            kt:=true else kt:=false;
                    end;
                end;
            if kt then writeln(f,'co')else writeln(f,'khong');
            close (f);
        end;
    begin
        nhap;
        xl;
    end.

```

#### ◆ **Bài 20:**

Nhập ma trận A.

- a) Tìm hàng có tổng lớn nhất.
- b) Tìm tất cả các hàng có tổng lớn nhất.

Dữ liệu nhập: cho trong file Bai.inp gồm:

- Dòng 1: Số nguyên n là số dòng và m là số cột của ma trận
- Các dòng còn lại là giá trị từng phần tử của ma trận.

Dữ liệu xuất: Xuất ra file Bai.out gồm :

- Dòng 1: Xuất hàng có tổng lớn nhất.

- Các dòng tiếp theo là vị trí của hàng có tổng lớn nhất.

Ví dụ:

**FILE BAI.INP:**

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

**FILE BAI.OUT:**

- Hàng 1 có tổng lớn nhất ( hoặc hàng 3)
- Các hàng có tổng lớn nhất là:

```
1
3
```

**Chương trình:**

```
program vd;
const fi='bai.inp';
      fo='bai.out';
type ar=array[1..50,1..50]of byte;
      tong=record x,y:integer;
end;
      arr=array[1..50]of tong;
var n,m:byte;  a:ar;  f:text;
procedure nhap;
var i,j:byte;
begin
  assign(f,fi);
  reset(f);
  read(f,n,m);
  for i:=1 to n do
    for j:=1 to m do
      read(f,a[i,j]);
  close(f);
end;
procedure sx(var b:arr);
var i,j:byte;  d:tong;
begin
  for i:=1 to n-1 do
    for j:=i+1 to n do
      if b[i].x<b[j].x then
        begin
          d:=b[i];
          b[i]:=b[j];
          b[j]:=d;
        end;
  end;
end;
procedure xl;
var i,j:byte;  t:integer;  b:arr;
begin
```



```

assign(f,fo);
rewrite(f);
for i:=1 to n do
  begin
    t:=0;
    for j:=1 to m do
      t:=t+a[i,j];
      b[i].x:=t;
      b[i].y:=i;
    end;
  sx(b);
  writeln(f,b[1].y);
  for i:=1 to n do
    if b[i].x=b[1].x then write(f,b[i].y,' ');
  close(f);
end;
begin
  nhap;
  xl;
end.

```

#### ◆ **Bài 21:**

Nhập dãy số thực a. Xét xem:

a/ Đây là dãy tăng ?

b/ Đây là dãy đan dấu ? ( Dãy đan dấu là dãy không có số 0, các số dương và âm xen kẽ nhau )

Dữ liệu nhập: Cho trong file BAI .INP gồm 2 dòng:

- Dòng 1: Số n là số phần tử của dãy.
- Dòng 2: Dãy số thực a.

Dữ liệu xuất: Xuất ra file BAI .out gồm 2 dòng:

- Dòng 1: Trả lời câu a
- Dòng 2: Trả lời câu b

Ví dụ:

**File BAI.inp:**

7

9 -4 3 -8 4 -1 6

**File BAI.out**

a/ Không

b/ Có

#### **Chương trình:**

```

program vd;
const fi='bai.inp';
      fo='bai.out';
type ar=array[1..100]of real;
var a:ar;  n:byte;  f:text;
procedure nhap;
var i:byte;

```

```

begin
    assign(f,fi);
    reset(f);
    read(f,n);
    for i:=1 to n do
        read(f,a[i]);
    close(f);
end;
function kt1(a:ar;n:byte):boolean;
var i:byte;
begin
    kt1:=false;
    for i:=1 to n-1 do
        if a[i]>a[i+1] then exit;
    kt1:=true;
end;
function kt2(a:ar;n:byte):boolean;
var i:byte;
begin
    kt2:=false;
    for i:=1 to n-1 do
        if a[i]*a[i+1]>0 then exit;
    kt2:=true;
end;
procedure xl1;
begin
    if kt1(a,n) then writeln(f,'co')
    else writeln(f,'khong');
end;
begin
    nhap;
    assign(f,fo);
    rewrite(f);
    xl1;
    xl2;
    close(f);
end.

```

### ◆ **Bài 22:**

Nhập dãy số thực a. Đưa tất cả số âm lên đầu, các số dương xuống cuối và các số 0 đứng giữa, thứ tự các số cùng dấu không đổi.

Nhập từ bàn phím, xuất ra màn hình

Ví dụ: 1 -3 0 3 4 -8

➔ -3 -8 0 1 3 4

### **Chương trình:**

```

program vd;
uses crt;
type ar=array[1..100]of real;

```

```

var a:ar;  n:byte;  x:real;
procedure nhap;
var i:byte;
begin
    write('moi nhap so phan tu: ');
    readln(n);
    for i:=1 to n do
        begin
            write('a[' ,i,']=');
            readln(a[i]);
        end;
    end;
procedure dc(var a,b:real);
var t:real;
begin
    t:=a;
    a:=b;
    b:=t;
end;
procedure xl;
var i:byte;
begin
    for i:=1 to n-1 do
        if ((a[i]>=0)and(a[i+1]<0))or((a[i]>0)and(a[i+1]=0)) then
            begin
                dc(a[i],a[i+1]);
                i:=0;
            end;
    for i:=1 to n do
        write(a[i]:0:1, ' ');
    end;
begin
    nhap;
    xl;
    readln;
end.

```

### ◆ **Bài 23:**Bài toán đua xe:

Cho  $n$  chiếc xe  $X_i$  ( $1 \leq i \leq n < 100$ ) cùng đua trên một đường đua. Các xe đua khởi hành ở những thời điểm  $t_i$ , cách mức đến  $D$  những khoảng bất kỳ di và với những vận tốc cũng không nhất thiết bằng nhau  $V_i$ .

Cho biết  $t_i$ ,  $d_i$ ,  $V_i$  (nguyên dương nhỏ hơn 10000). Hãy lập trình xác định những xe nào về đích sớm nhất.

Dữ liệu nhập: cho trong file Bai.inp gồm:

- Dòng 1: Số  $n$
- $N$  dòng tiếp theo, mỗi dòng là thời điểm  $t$ , khoảng cách  $d$ , vận tốc  $v$  của xe tương ứng.

Dữ liệu xuất: Xuất ra file Bai.out là số các xe về đích trước nhất ( có thể có nhiều xe về đích một lượt )

Ví dụ:

**File Bai.inp:**

```
5
1 7 100
2 8 125
3 10 135
5 12 140
6 16 200
```

**File Bai.out : 1**

**Chương trình:**

```
program duaxe;
uses crt;
const fi='bai.inp';
      fo='bai.out';
      max=10;
type mt=array[1..max,1..3] of integer;
      mang=array[1..max] of integer;
var  n,x:integer; a:mt; xe:mang;
procedure nhap;
var  i,j:integer; f:text;
begin
  assign(f,fi);
  reset(f);
  read(f,n);
  for i:=1 to n do
    for j:=1 to 3 do
      read(f,a[i,j]);
    close(f);
  end;
procedure xuly;
var  min,t:real; i:integer;
begin
  min:=a[1,1]+(a[1,2]/a[1,3]);
  xe[1]:=1;
  x:=1;
  for i:=2 to n do
    begin
      t:=a[i,1]+(a[i,2]/a[i,3]);
      if t<min then
        begin
          min:=t;
          xe[1]:=i;
        end
      else if t=min then
        begin
```

```

        x:=x+1;
        min:=t;
        xe[x]:=i;
    end;
end;
end;
procedure xuất;
var i:integer; f:text;
begin
    assign(f,fo);
    rewrite(f);
    for i:=1 to x do
        write(f,xe[i],' ');
    close(f);
end;
begin
    clrscr;
    nhap;
    xuly;
    xuất;
end.

```

#### ◆ **Bài 24:**

Viết chương trình nhập vào giờ đến và giờ đi của các xe tại một bến xe. Biết rằng các xe có thể đậu trong ngày ( từ 0 đến 23 giờ )

a/ Nhập vào giờ cần thống kê. Cho biết tại bến có bao nhiêu xe vào thời điểm đó và đó là những xe nào.

b/ Cho biết xe tập trung đông nhất tại bến là lúc nào.

Ví dụ:

- Số xe tại bến: 7

Xe		Giờ đến	Giờ đi
1	7	11	
2	10	13	
3	3	8	
4	0	7	
5	16	20	
6	11	15	
7	18	23	

- Giờ cần thống kê: 12

=> Tổng số xe đang đậu tại bến 12 giờ là 2 xe.

Là xe số: 2, 6.

- Xe tập trung đông nhất vào lúc:

7 giờ / xe số 1, 3, 4

11 giờ / xe số 1, 2, 6

---

#### **Chương trình:**

```

program benxe;
uses crt;
type soxe=array[1..60,1..60] of integer;
    lietke=array[1..60] of integer;
var A:soxe; n1:integer; B:lietke; u,u1:integer;
procedure vaomatran (var f:text; var n:integer);
var i,t:integer;
begin
    writeln('          +-----+');
    writeln('          |      LICH DI VA DEN CUA BEN XE      |');
    writeln('          +-----+');
    write('          So xe co tai ben: ');
    readln(n);
    writeln;
    writeln('          ****');
    for i:=1 to n do
        begin
            A[i,1]:=i;
            writeln(' Xe thu ',i);
            write(' =>Gio den :');
            readln(A[i,2]);
            write(' =>Gio di :');
            readln(A[i,3]);
        end;
    end;
end;
procedure thongke(var l:text;n:integer);
var i,h,t,x:integer;
begin
    write('Nhap vao gio can thong ke:');
    readln(x);
    if (not(x in[0..23])) then
        repeat
            write(' =>Nhap lai gio can thong ke: ');
            readln(x);
        until (x in[0..23]);
    h:=1;
    for i:=1 to n do
        if (A[i,2]<A[i,3]) then
            if (x in[A[i,2]..A[i,3]]) then
                begin
                    B[h]:=A[i,1];
                    h:=h+1;
                end
            else if (A[i,2]>A[i,3]) then
                if (x in[A[i,2]..23]) or (x in[0..A[i,3]]) then
                    begin
                        B[h]:=A[i,1];
                        h:=h+1;
                    end
                end;
    end;
end;

```

```

        end;
writeln('Tong so xe dang dau tai ben luc ',x,' gio la ',h-1,' xe. ');
write('La xe so ');
for i:=1 to h-1 do
    begin
        write(B[i], ' ');
    end;
writeln;
end;
procedure dongnhhat(n:integer);
var i,t,d,max:integer;
begin
    for i:=0 to 23 do
        begin
            d:=0;
            for t:=1 to n do
                if (A[t,2]<A[t,3]) then
                    if (i in[A[t,2]..A[t,3]]) then d:=d+1
                else if (A[t,2]>A[t,3]) then
                    if (i in[A[t,2]..23]) or (i in[0..A[t,3]]) then d:=d+1;
                B[i+1]:=d;
            end;
            max:=B[1];
            for i:=2 to 24 do
                if max<B[i] then max:=B[i];
            writeln('Xe tap trung dong nhhat vao luc ');
            for i:=1 to 24 do
                begin
                    If max=B[i] then
                        begin
                            write(' ',i-1,' gio / xe so ');
                            for t:=1 to n do
                                if (i-1 in[A[t,2]..A[t,3]]) then write(t, ' ');
                            writeln;
                        end;
                end;
            end;
        end;
    end;
begin
    clrscr;
    vaomatran(f1,n1);
    thongke(l1,n1);
    dongnhhat(n1);
    readln;
end.

```

#### ◆ **Bài 25:**

Cho bảng A các số nguyên  $A(n \times m)$ . Hãy dùng phép hoán vị hai phần tử trong bảng để sắp xếp đưa các phần tử chẵn về đầu bảng theo thứ tự từ trái sang phải, từ trên xuống dưới, sao cho “số lần hoán vị giữa 2 phần tử trong bảng là ít nhất”

Ví dụ :

- Ma trận  $3 \times 3$ :

10	52	12
152	11	25
16	-27	21

Kết quả

- Số lần hoán vị : 1 lần

10	52	12
152	<b>16</b>	25
<b>11</b>	-27	21

### Chương trình:

```

uses crt;
const fi='bang.inp';
      fo='bang.out';
type matran=array[1..3,1..3] of integer;
      mang=array[1..9] of integer;
      mangboolean=array[1..20] of boolean;
var a:matran; q,n,m,k:integer; b:mang;
procedure chuyenfile( var a:matran; var n, m:integer);
var i, j: integer; f: text;
begin
    assign(f,fi);
    reset(f);
    read(f,n,m);
    for i:=1 to n do
        for j:=1 to m do
            read(f,a[i,j]);
        close(f);
    end;
procedure xuatmatran(a: matran; n, m:integer);
var i,j:integer;
begin
    for i:=1 to n do
        begin
            for j:=1 to m do
                write(a[i,j]:5);
            writeln;
        end;
    end;
procedure chuyenmang( a:matran; n,m: integer; var b:mang);
var i, j: integer;
begin
    for i:=1 to n do
        for j:=1 to m do
            begin
                b[(i-1)*n+j]:=a[i,j];
            end;
end;

```



```

end;
procedure hoanvi(var b:mang; k,n,m,q:integer);
var z,i,j,t:integer; c:matran;
begin
    q:=0;
    z:=n*m;
    for i:=1 to z-1 do
        for j:=z downto i+1 do
            if (b[i] mod 2 <> 0 ) and (b[j] mod 2 = 0) then
                begin
                    q:=q+1;
                    t:=b[i];
                    b[i]:=b[j];
                    b[j]:=t;
                end;
            end;
        end;
    end;
end;
procedure chuyenmatran(b: mang; var a: matran;n,m:integer);
var i, j: integer;
begin
    for i:=1 to n do
        for j:=1 to m do
            a[i,j]:=b[(i-1)*n+j];
        end;
    end;
end;
procedure ghifile( a: matran; n, m, q: integer);
var i, j: integer; f: text;
begin
    assign(f,fo);
    rewrite(f);
    writeln(f,q);
    writeln(f,n,' ',m);
    for i:=1 to n do
        begin
            for j:=1 to m do
                write(f,a[i,j], ' ');
            writeln(f);
        end;
    close(f);
end;
begin
    clrscr;
    chuyenfile(a,n,m);
    chuyenmang(a,n,m,b);
    hoanvi(b,k,n,m,q);
    chuyenmatran(b,a,n,m);
    ghifile(a,n,m,q);
    readln;
end.

```

◆ **Bài 26:**

Cho bảng  $n * n$ , hãy sắp xếp  $n * n$  chữ cái sao cho trên mỗi dòng cũng như mỗi cột, mỗi chữ cái đó chỉ xuất hiện đúng 1 lần.

**Ví dụ:**

- Nhập  $n = 4$ :

A B C D

B C D A

C D A B

D A B C

**Chương trình:**

```
type arr=array[1..26,1..26] of char;
```

```
var a: arr; i,j,n,m: integer;
```

```
begin
```

```
  write('Nhập n: ');
```

```
  readln(n);
```

```
  for i:=1 to n do
```

```
    for j:=1 to n do
```

```
      if i=1 then a[i,j]:=chr(ord('A')+j-1)
```

```
      else
```

```
        if ord(a[i-1,j])-ord('A')+1<n then
```

```
          a[i,j]:=chr(ord(a[i-1,j])+1)
```

```
        else
```

```
          a[i,j]:=chr(ord(a[i-1,j])+1-n);
```

```
  for i:=1 to n do
```

```
    begin
```

```
      for j:=1 to n do
```

```
        write(a[i,j]:3);
```

```
      writeln;
```

```
    end;
```

```
  readln;
```

```
end.
```

◆ **Bài 27:**

Tìm cách điền 9 chữ số khác nhau (1,2,3,4,5,6,7,8,9) vào bảng vuông  $3 * 3$  như sau:

a	B	c
A'	B'	c'
A''	B''	c''

$$\text{Sao cho } \begin{cases} a'b'c' = 2\overline{abc} \\ a''b''c'' = 3\overline{abc} \end{cases}$$

**Chương trình:**

```
uses crt;
```

```
procedure kt(i,j,k: integer);
```

```
var x:array[1..10] of byte;
```

```

begin
  x[1]:=i mod 10;
  x[2]:=i div 10 mod 10;
  x[3]:=i div 100;
  x[4]:=j mod 10;
  x[5]:=j div 10 mod 10;
  x[6]:=j div 100;
  x[7]:=k mod 10;
  x[8]:=k div 10 mod 10;
  x[9]:=k div 100;
  if ([x[1],x[2],x[3],x[4],x[5],x[6],x[7],x[8],x[9]]=[1,2,3,4,5,6,7,8,9]) then
    begin
      writeln(x[3],',',x[2],',',x[1]);
      writeln(x[6],',',x[5],',',x[4]);
      writeln(x[9],',',x[8],',',x[7]);
      readln;
    end;
  end;
  procedure dienso;
  var i,j,k:integer;
  begin
    for i:=111 to 333 do
      begin
        j:=2*i;
        k:=3*i;
        kt(i,j,k);
      end;
  end;
  begin
    clrscr;
    dienso;
    readln;
  end.

```

### ◆ **Bài 27:** Trò chơi bốc kẹo

Có  $N = 100$  cái kẹo. Mỗi người đến lượt mình đi có thể bốc 1, 2, 3 hoặc 4 cái. Ai không còn kẹo để bốc là thua. Tìm chiến thuật chơi.

#### **Chương trình:**

```

program bockeo;
uses crt;
var trangthai: integer;
    maydi:boolean;

procedure maydi1( var trangthai: integer);
var sokeoboc:integer;

```

```

begin
  if trangthai mod 5 <> 0 then
    sokeoboc:=trangthai mod 5
  else { boc bua}
    sokeoboc:=random(4)+1;
  writeln('May da boc: ',sokeoboc,' cuc keo. ');
  trangthai:=trangthai-sokeoboc;
  writeln('So keo con lai: ',trangthai);
end;
procedure nguoidi1( var trangthai: integer);
var sokeoboc:integer;
begin
  repeat
    write('Moi boc keo (tu 1 -> 4): ');
    readln(sokeoboc);
  until (sokeoboc>0) and (sokeoboc<=4);
  trangthai:=trangthai-sokeoboc;
  writeln('So keo con lai: ',trangthai);
end;
function ketthuc(trangthai:integer): boolean;
begin
  if (trangthai=0) then
    ketthuc:=true
  else ketthuc:=false;
end;
begin
  clrscr;
  randomize;
  trangthai:=100;
  write('Co tat ca ',trangthai);
  writeln(' cuc keo. Bat dau cuoc choi nhe !');
  maydi:=(random(1)=0);
  readln;
  repeat
    if maydi then
      begin
        maydi1(trangthai);
        maydi:=false;
      end
    else
      begin
        nguoidi1(trangthai);
        maydi:=true;
      end;
    readln;
    clrscr;
  until ketthuc(trangthai);
  if maydi then write('Chuc mung! Ban da thang.')

```

```

        else write('Toi thang roi nhe!');
        readln;
    end.

```

### ◆ **Bài 28:**

Lập trình tìm tất cả các cách thay thế các dấu ? bởi các phép tính +, -, \*, / trong biểu thức dưới đây sao cho biểu thức có giá trị bằng 35:

$((((1 ? 2) ? 3) ? 4) ? 5) ? 6$

### **Chương trình:**

```

program vd;
const a:array[1..4]of char
    =('+','-','*','/');
var i,j,k,l,m:integer;
function f(x,y,d:integer):integer;
begin
    case d of
        1:f:=x+y;
        2:f:=x-y;
        3:f:=x*y;
        4:if x mod y=0 then
            f:=x div y else f:=0;
    end;
end;
begin
    for i:=1 to 4 do
        for j:=1 to 4 do
            for k:=1 to 4 do
                for l:=1 to 4 do
                    for m:=1 to 4 do
                        if f(f(f(f(f(1,2,i),3,j),4,k),5,l),6,m)=35then
                            writeln('(((1',a[i],')2)',a[j],')3)',a[k],')4)',a[l],')5)',a[m],')6 = 35');
                    end;
                end;
            end;
        end;
    end;
    readln;
end.

```

### ◆ **Bài 29:**

Cho số N nguyên dương. Hãy xây dựng công thức để tính các phần tử của bảng kích thước  $N \times N$ , dựa theo quy luật sau:

**N=5**

```

9 7 5 3 1
7 5 3 1 2
5 3 1 2 4
3 1 2 4 6
1 2 4 6 8

```

**N=6**

```

11 9 7 5 3 2
 9 7 5 3 1 2
 7 5 3 1 2 4
 5 3 1 2 4 6
 3 1 2 4 6 8
 1 2 4 6 8 10

```

### **Chương trình:**

```

program vd;
type ar=array[1..100]of integer;
  arr=array[1..10,1..10]of integer;
var n:byte;  a:arr;
procedure xl;
var i,j,t,t2:integer;
begin
  for i:=1 to n do
    begin
      for j:=1 to n+1-i do
        a[i,j]:=2*(n-i+1)-1-2*(j-1);
      t:=0;
      for j:=n-i+2 to n do
        begin
          t:=t+1;
          a[i,j]:=t*2;
        end;
      end;
    for i:=1 to n do
      begin
        for j:=1 to n do
          write(a[i,j]:3);
        writeln;
      end;
    end;
begin
  write('nhap n: ');
  readln(n);
  xl;
  readln;
end.

```

### **◆ Bài 30:**

Cho số nguyên dương N, hãy viết thuật toán và chương trình để tạo lập bảng N\*N phần tử nguyên dương theo quy luật được cho trong ví dụ sau:

```

N=6:
1 2 3 4 5 6
2 4 6 8 10 12

```

```

3 6 9 12 2 4
4 8 12 2 4 6
5 10 2 4 6 8
6 12 2 4 6 8

```

### **Chương trình:**

```

program vd;
type ar=array[1..20,1..20]of integer;
var a:ar;  n:byte;
procedure xl;
var i,j,t:byte;
begin
  for i:=1 to n do
    begin
      for j:=1 to n do
        a[i,j]:=i*j;
      t:=0;
      for j:=(n*2 div i)+1 to n do
        begin
          inc(t);
          a[i,j]:=t*2;
        end;
      end;
    for i:=1 to n do
      begin
        for j:=1 to n do
          write(a[i,j]:3);
        writeln;
      end;
    end;
end;
begin
  write('nhap n:');
  readln(n);
  xl;
  readln;
end.

```

### **◆ Bài 31:**

Cho mảng  $A[1..n, 1..m]$  chứa các số nguyên.

Hãy sắp xếp lại giá trị của các ô trong A sao cho:

- $A[i,1] \leq A[i,2] \leq \dots \leq A[i,m]$
- $A[1,j] \leq A[2,j] \leq \dots \leq A[N,j]$

Ví dụ:

```

5 8
1 3 9 8 3 2 4 5
5 2 4 1 6 1 7 9
4 3 3 4 1 2 3 2
5 3 8 1 6 3 5 4

```

```

      8 2 1 2 1 1 3 4
➔ 1 1 1 1 1 1 1 1
    2 2 2 2 2 2 3 3
    3 3 3 3 3 3 4 4
    4 4 4 4 5 5 5 5
    6 6 7 8 8 8 9 9

```

### **Chương trình:**

```

program vd;
const fi='bai.inp';
      fo= 'bai.out';
type ar=array[1..10,1..10]of integer;
      arr=array[1..100]of integer;
var a:Ar;  n,m:byte;  f:text;
procedure nhap;
var i,j:byte;
begin
  assign(f,fi);
  reset(f);
  read(f,n,m);
  for i:=1 to n do
    for j:=1 to m do
      read(f,a[i,j]);
  close(f);
end;
procedure xuat(a:ar;n,m:byte);
var i,j:byte;
begin
  for i:=1 to n do
    begin
      for j:=1 to m do
        write(f,a[i,j], ' ');
      writeln(f);
    end;
end;
procedure sx(var a:arr;spt:byte);
var i,j:byte;
    t:integer;
begin
  for i:=1 to spt-1 do
    for j:=i+1 to spt do
      if a[i]>a[j] then
        begin
          t:=a[i];
          a[i]:=a[j];
          a[j]:=t;
        end;
end;
end;

```



```

procedure xl;
var b:Arr;
    i,j:byte;
begin
    for i:=1 to n do
        for j:=1 to m do
            b[(i-1)*m+j]:=a[i,j];
        sx(b,m*n);
    for i:=1 to n do
        for j:=1 to m do
            a[i,j]:=b[(i-1)*m+j];
        assign(f,fo);
        rewrite(f);
        xuất(a,n,m);
        close(f);
end;
begin
    nhập;
    xl;
end.

```

### ◆ **Bài 32:**

Viết chương trình liệt kê tất cả các số tự nhiên có 3 chữ số mà tổng các chữ số của nó bằng số tự nhiên cho trước.

#### **Chương trình:**

```

program vd;
const fi='bai.inp';
    fo='bai.out';
var n:byte;  f:text;
procedure nhập;
begin
    assign(f,fi);
    reset(f);
    read(f,n);
    close(f);
end;
function kt(x:integer):boolean;
var a,b,c:byte;
begin
    kt:=true;
    a:=x div 100;
    b:=x mod 100 div 10;
    c:=x mod 10;
    if a+b+c=n then exit;
    kt:=false;
end;
procedure xl;

```

```

var i:integer;
begin
  for i:=100 to 999 do
    if kt(i) then write(f,i,' ');
  end;
begin
  nhap;
  assign(f,fo);
  rewrite(f);
  if n>27 then writeln(f,'khong co so nao ca')else
  xl;
  close(f);
end.

```

### ◆ **Bài 33:**

Có mấy cách phân tích số tự nhiên n thành:  $n = i^3 + j^3$  ( $i \leq j$ )  
 Nhập n, xuất ra các cách.

#### **Chương trình:**

```

program vd;
const fi='bai.inp';
      fo='bai.out';
var n:byte;  f:text;
procedure nhap;
begin
  assign(f,fi);
  reset(f);
  read(f,n);
  close(f);
end;
procedure xl;
var i,j,d:byte;
begin
  d:=0;
  for i:=1 to n do
    for j:=i to n do
      if i*i*i+j*j*j=n then
        begin
          inc(d);
          writeln(f,n,'=',i,'^3 + ',j,'^3');
        end;
  if d=0 then writeln(f,'khong co so nao ca')
  else write(f,'co tat ca ',d,' kq');
end;
begin
  nhap;
  assign(f,fo);
  rewrite(f);

```

```

xl;
close(f);
end.

```

#### ◆ **Bài 34:**

Viết chương trình nhân 2 số nguyên có độ dài lớn nhất đến 300 chữ số.

Ví dụ:  $123 * 789 = 97047$

$123456 * 123456 = 15241383936$

$123456789 * 123456789 = 15241578750190521$

#### **Chương trình:**

```

uses crt;
const spt=600;
type mang=array[1..spt] of integer;
var a,b,c,kq: mang; i,j, k, na, nb, tam: integer;
procedure nhap ( var a: mang; var na: integer);
var k: integer;
begin
    na:=0;
    repeat
        inc(na);
        write('Chu so thu: ',na,'=');
        readln(a[na]);
    until a[na]=-1;
    dec(na);
    for k:=0 to na-1 do
        a[spt-k]:=a[na-k];
    for k:=1 to spt-na do
        a[k]:=0;
end;
procedure cong(a: mang; var b: mang);
var tam1, tam2, k: integer;
begin
    tam1:=0;
    for k:=spt downto 1 do
    begin
        tam2:=(a[k]+b[k]+tam1) div 10;
        b[k]:=(a[k]+b[k]+tam1) mod 10;
        tam1:=tam2;
    end;
end;
begin
    clrscr;
    write('Nhap so thu 1: ');
    writeln;
    nhap(a,na);
    write('Nhap so thu 2: ');

```

```

writeln;
nhap(b,nb);
for k:=1 to spt do
  kq[k]:=0;
for j:=spt downto spt-nb do
begin
  for k:=1 to spt do
    c[k]:=0;
  tam:=0;
  for i:=spt downto spt-na do
  begin
    c[j+i-spt]:=(b[j]*a[i]+tam) mod 10;
    tam:=(b[j]*a[i]+tam) div 10;
  end;
  cong(c,kq);
end;
write('So thu nhat: ');
for k:=spt-na+1 to spt do
  write(a[k]);
writeln;
write('So thu hai: ');
for k:=spt-nb+1 to spt do
  write(b[k]);
writeln;
write('Tich bang: ');
for k:=spt -(na+nb)+1 to spt do
  write(kq[k]);
readln;
end.

```

### ◆ **Bài 35: Số gần nhất**

Cho số nguyên dương  $N$  và chữ số  $c$ , tìm số nguyên dương nhỏ nhất  $k$  thỏa  $k > N$  và trong biểu diễn của  $k$  trong hệ thập phân có chữ số  $c$ .

Dữ liệu: Cho trong tập tin văn bản Near.inp gồm 1 dòng duy nhất chứa số nguyên dương  $N$  và chữ số  $c$ ,  $N$  và  $c$  cách nhau ít nhất 1 khoảng trắng,  $0 < N < 32750$ ,  $0 \leq c \leq 9$ .

Kết quả: Cho trong tập tin văn bản Near.out, gồm 1 dòng duy nhất chứa số nguyên  $k$  tìm được.

Ví dụ: Near.inp:

11 7

Near.out:

17

### **Chương trình:**

```

program vd;
const fi='near.inp';
      fo='near.out';
var f:text;  n:integer;  c:byte;
procedure nhap;
begin

```

```

    assign(f,fi);
    reset(f);
    read(f,n,c);
    close(f);
end;
function KT(x:integer;c:byte):Boolean;
begin
    KT:=true;
    while(x mod 10<>c)and(x>0)do
        x:=x div 10;
    if(x=0)then kt:=false;
end;
procedure xl;
var x:integer;
begin
    x:=n+1;
    while(x>n)and not(KT(x,c))do
        inc(x);
        write(f,x);
end;
begin
    nhap;
    assign(f,fo);
    rewrite(f);
    xl;
    close(f);
end.

```

### ◆ **Bài 36:** Dãy 0-1:

Dãy 0-1 là dãy vô hạn phần tử chỉ gồm các số 0 và 1 được xây dựng theo quy luật sau đây: Số đầu tiên là số 0. Nối vào dãy các số đã viết “dãy số đảo” của nó ( là dãy số có cùng chiều dài với dãy đã viết, trong đó số 0 ở dãy đã viết thành số 1 ở “dãy số đảo” và số 1 ở dãy đã viết trở thành số 0 ở “dãy số đảo”). Như vậy sau mỗi lần nối, chiều dài của dãy tăng lên gấp đôi.

Dưới đây là 5 bước đầu tiên để hình thành dãy 0-1:

```

0
0 1
0 1 1 0
0 1 1 0 1 0 0 1
0 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0

```

#### Nhiệm vụ:

Viết chương trình xalxc định chữ số ở vị trí thứ N của dãy 0-1

Dữ liệu: Cho trong tập tin SEQ01.inp gồm 1 dòng duy nhất chứa 1 số nguyên dương N duy nhất (  $N \leq 30000$  )

Kết quả: Cho trong tập tin văn bản SEQ01.out, chứa 1 số nguyên duy nhất chỉ chữ số ở vị trí thứ N của dãy 0-1

Ví dụ: SEQ01.inp:

11

SEQ01.out:

0

**Chương trình:**

```
program vd;
const fi='seq.in';
      fo='seq.out';
type arr=array[1..16400]of char;
var f:text;  n:integer;  a:arr;
procedure nhap;
begin
    assign(f,fi);
    reset(f);
    readln(f,n);
    close(f);
end;
Procedure G(var a:arr;spt:integer);
var i:integer;
begin
    i:=1;
    while(i<=spt)and(spt+i<=n)do
        Begin
            if(a[i]='0')then a[spt+i]:='1'else
                a[spt+i]:='0';
            inc(i);
        end;
    end;
end;
procedure xl;
var spt:integer;
begin
    a[1]:='0';
    spt:=1;
    while(spt*2<n)do
        begin
            G(a,spt);
            spt:=spt*2;
        end;
    if(a[n-spt]='0')then write(f,'1')else write(f,'0');
end;
begin
    nhap;
    assign(f,fo);
    rewrite(f);
    xl;
    close(f);
end.
```

◆ **Bài 37: Tạo bảng:**

Cho 1 bảng A N\*N số nguyên (  $N \leq 100$ ). Mỗi số trong bảng có giá trị tuyệt đối không vượt quá 30000. Bảng B được tạo ra từ bảng A theo quy tắc sau:

Phần tử của B nằm dòng i, cột j có giá trị bằng tổng của các số nằm trong ô  $(i,j)$  và các ô kề nó trong bảng A:

$$B_{ij} = A_{ij} + A_{(i+1)j} + A_{(i-1)j} + A_{i(j+1)} + A_{i(j-1)}$$

Chú ý: Các phần tử nằm ngoài bảng coi như bằng 0.

Bài toán: Cho bảng A. Hãy tạo bảng B tương ứng.

Dữ liệu vào: Table.inp:

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

Kết quả : Table.out

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

### **Chương trình:**

```
program vd;
const fi='table.inp';
      fo='table.out';
type arr=array[1..50,1..50]of byte;
var f:text;  a:arr;  n:byte;
procedure nhap;
var i,j:byte;
begin
  assign(f,fi);
  reset(f);
  read(f,n);
  for i:=1 to n do
    for j:=1 to n do
      read(f,a[i,j]);
  close(f);
end;
procedure Tinh(var b:arr;a:arr;i,j:byte);
begin
  b[i,j]:=0;
  b[i,j]:=b[i,j]+a[i,j];
  if(i-1>0)then b[i,j]:=b[i,j]+a[i-1,j];
  if(i+1<=n)then b[i,j]:=b[i,j]+a[i+1,j];
  if(j-1>0)then b[i,j]:=b[i,j]+a[i,j-1];
  if(j+1<=n)then b[i,j]:=b[i,j]+a[i,j+1];
end;
procedure xl;
var b:arr;  i,j:byte;
begin
  for i:=1 to n do
    for j:=1 to n do
      Tinh(b,a,i,j);
```

```

for i:=1 to n do
begin
  for j:=1 to n do
    write(f,b[i,j], ' ');
  writeln(f);
end;
end;
begin
  Nhap;
  assign(f,fo);
  rewrite(f);
  XL;
  close(f);
end.

```

### ◆ **Bài 38: Khoảng cách giữa 2 số**

Với 2 chữ số  $x$  và  $y$ , khoảng cách của chúng được định nghĩa là số nguyên không âm nhỏ nhất  $d(x,y)$  mà khi cộng thêm  $d(x,y)$  vào một chữ số nào đó trong 2 chữ số  $x, y$  thì kết quả nhận được là một số nguyên có chữ số hàng đơn vị trùng với chữ số còn lại.

VD  $d(2,5)=3$  vì  $2+3=5$ ;  $d(5,1)=4$  vì  $1+4=5$ ; còn  $d(1,9)=2$  vì  $9+2=11$

Với 2 số nguyên dương  $x$  và  $y$  có cùng số lượng chữ số, khoảng cách  $d(x,y)$  giữa 2 số  $x$  và  $y$  là tổng khoảng cách giữa các cặp chữ số cùng hàng tương ứng.

VD:  $d(213,419)=d(2,4)+d(1,1)+d(3,9)=2+0+4=6$

**Bài toán:** Cho 2 chữ số  $x$  và  $y$  có cùng lượng chữ số  $N$  ( $0 < N < 100$ ), hãy tìm khoảng cách  $d(x,y)$

**Dữ liệu vào:** từ file Distance.inp trong đó dòng đầu chứa số  $x$ , dòng 2 chứa số  $y$  thỏa mãn ràng buộc của bài toán.

**Kết quả:** ghi ra file Distance.out trong đó chứa một số nguyên duy nhất là kết quả  $d(x,y)$  tìm được.

**Ví dụ:**

Distance.inp:

213

419

Distance.out:

6

### **Chương trình:**

Program BT;

Const fi='BT.inp';

fo='BT.out';

Var x,y:string[100]; n:integer;

Procedure Nhap;

Var f:text; i:integer;

Begin

Assign(f,fi);

Reset(f);

Readln(f,x);

Readln(f,y);



```

    n:=length(x);
    Close(f);
End;
Procedure Xuly;
Var F:text;  i,S,d,Code1,Code2:integer;  A,B:array [1..100] of integer;
Begin
    Assign(f,fo);
    Rewrite(f);
    For i:=1 to n do
        Begin
            Val(X[i],A[i],Code1);
            Val(Y[i],B[i],Code2);
        End;
    S:=0;
    For i:=1 to n do
        Begin
            d:=Abs(A[i]-B[i]);
            If d>5 then
                If A[i]<B[i] then d:=A[i]+10-B[i]
                else d:=B[i]+10-A[i];
            S:=S+d;
        End;
    Write(f,S);
    Close(f);
End;
Begin
    Nhap;
    Xuly;
End.

```

### ◆ **Bài 39:**

Xét một phương trình có dạng như sau:  $x+y+z=K$

Trong đó K là một số nguyên dương. Phương trình này có thể vô số nghiệm. Tuy nhiên, ở đây người ta chỉ quan tâm đến các nghiệm (x,y,z) mà trong đó các số x, y, z đều là các số nguyên tố.

**Bài toán:** Với số K cho trước (  $K < 5000$  ), hãy tìm tất cả các bộ số nguyên tố x, y, z (  $x \leq y \leq z$  ) là nghiệm của phương trình trên hoặc cho biết không có nghiệm thỏa mãn yêu cầu bài toán.

**Dữ liệu vào:** Cho trong file Equa.inp trong đó có chứa duy nhất số K

**Kết quả:** Ghi ra file Equa.out chứa n+1 dòng ( n là số nghiệm tìm được ), trong đó:

- Dòng thứ I trong N dòng đầu tiên chứa 3 số nguyên cho biết bộ nghiệm thứ I tìm được
- Dòng n+1 chứa 3 số 0 cho biết điểm kết thúc file output

### **Chương trình:**

```

program vd;
const fi='equa.inp';
      fo='equa.out';
var f:text;  n:integer;
procedure nhap;
begin

```

```

assign(f,fi);
reset(f);
{$i-}
  readln(f,n);
  if(ioresult<>0)or(n<=0)or(n>=5000)then
  begin
    assign(f,fo);
    rewrite(f);
    writeln(f,'Du lieu khong hop le');
    close(f);
    halt;
  end;
{$i+}
close(f);
end;
function NT(x:integer):Boolean;
var i:integer;
begin
  NT:=False;
  for i:=2 to trunc(sqrt(x)) do
    if(x mod i=0)then Exit;
  NT:=True;
end;
procedure xl;
var i,j,k,d:integer;
begin
  i:=2;
  d:=0;
  while(i<=n-4)do
  begin
    while(NT(i))do
    begin
      j:=i;
      while(j<=n-3)do
      begin
        while(NT(j))do
        begin
          k:=j;
          while(k<=n-2)do
          begin
            while(NT(k))do
            begin
              if(i+j+k=n)then
              begin
                inc(d);
                writeln(f,i,' ',j,' ',k);
              end;
              inc(k);
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

        end;
        inc(k);
    end;
    inc(j);
end;
inc(j);
end;
inc(i);
end;
inc(i);
end;
if(d=0)then writeln(f,'Vo nghiem');
writeln(f,'0 0 0');
end;
begin
    nhap;
    assign(f,fo);
    rewrite(f);
    xl;
    close(f);
end.

```

#### ◆ **Bài 40:**

Nhập từ bàn phím 2 xâu ký tự S, M có chiều dài tối đa.

a/ Hãy kiểm tra: có thể nhận được M từ S bằng cách xoá đi 1 số ký tự của S hay không ? Nếu được hãy hiển thị số thứ tự của các ký tự được giữ lại trong S

b/ Thực hiện câu b với điều kiện bổ sung: Hiệu của số thứ tự kí tự cuối cùng và số thứ tự của ký tự đầu tiên được giữ lại trong S không vượt quá p ( p nhập từ bàn phím)

#### **Chương trình:**

```

program vd;
uses crt;
type arr=array[1..255]of byte;
var s,m:string; a:arr; spt:byte;
procedure nhap;
begin
    write('Nhap s:');
    readln(s);
    write('Nhap m:');
    readln(m);
end;
function KT(s,m:string;var a:arr;var spt:byte):Boolean;
var vt,j,i:byte;
begin
    vt:=0;
    KT:=False;
    spt:=0;
    for i:=1 to length(m) do

```

```

begin
    j:=vt+1;
    while(j<=length(s))and(s[j]<>m[i])do
        inc(j);
    if(j>length(s))then exit;
    vt:=j;
    inc(spt);
    a[spt]:=vt;
end;
KT:=True;
end;
procedure x11;
var i:byte;
begin
    Writeln('Cau a');
    if(KT(s,m,a,spt))then
        begin
            writeln('  Co');
            writeln('  Vi tri cac ky tu duoc giu lai trong chuoai s la:');
            write(' ');
            for i:=1 to spt do
                write(a[i], ' ');
            end else write('  Khong');
        end;
end;
procedure x12;
var p:byte;
begin
    write('Nhap p:');
    readln(p);
    writeln('Cau b');
    if(a[spt]-a[1]<=p)then Writeln('  Co')else
        writeln('  Khong');
end;
begin
    clrscr;
    nhap;
    x11;
    writeln;
    if(KT(s,m,a,spt))then
        x12;
    readln;
end.

```

#### ◆ **Bài 41:**

Trong 1 khu dân cư có  $n \times n$  lô đất cạnh nhau ( $1 \leq n \leq 100$ ), giả sử lô đất  $(i,j)$  (với  $1 \leq i,j \leq 100$ ) được quy hoạch xây dựng toà nhà cao  $a_{i,j}$  tầng ( $1 \leq a_{i,j} \leq 100$ ). Giả sử lượng sơn cần dùng là tương ứng tỉ lệ với đơn vị diện tích bề mặt ( $1$  bề mặt của mỗi tầng ở mỗi toà nhà

chiếm 1 đơn vị diện tích) và ta sơn cả phần nóc của các toà nhà này. Hãy tính lượng sơn cần phải sơn toàn bộ khu nhà sau khi xây xong ( các bức tường bị khuất thì không cần sơn)  
Ví dụ: Khu dân cư với 3\*3 lô đất được quy hoạch xây dựng các toà nhà cao tầng như sau:

2	3	2
2	3	2
2	3	2

Kết quả: Lượng sơn cần dùng là 41. Trong bảng dưới đây minh hoạ lượng sơn tương ứng cho mỗi toà nhà của khu nhà trên:

5	6	5
3	3	3
5	6	5

Dữ liệu vào: Lưu trong tập tin sonnha.inp:

- Dòng đầu tiên chứa số nguyên n
- N dòng tiếp theo chứa ma trận  $n \times n$  trong đó  $a_{i,j}$  là độ cao của toà nhà được xây trên lô đất thứ (i,j)

Dữ liệu xuất:

Xuất ra tập tin sonnha.out chứa 1 số duy nhất là số lượng sơn cần dùng để sơn toàn bộ toà nhà trong khu dân cư.

### **Chương trình:**

```

program vd;
uses crt;
const fi='sonnha.inp';
      fo='sonnha.out';
type arr=array[1..100,1..100]of byte;
var f:text; n:byte; a:arr;
procedure nhap;
var i,j:byte;
begin
  assign(f,fi);
  reset(f);
  read(f,n);
  for i:=1 to n do
    for j:=1 to n do
      read(f,a[i,j]);
  close(f);
end;
procedure xl;
var t:longint; i,j:byte;
begin
  t:=n*n;
  for i:=1 to n do
    for j:=1 to n do
      begin
        if((i=1)or(i=n))and((j=n)or(j=1))then t:=t+a[i,j]*2 else

```

```

        if(i=1)or(j=1)or(i=n)or(j=n)then t:=t+a[i,j];
        if(j<>1)then t:=t+abs(a[i,j]-a[i,j-1]);
        if(i<>1)then t:=t+abs(a[i,j]-a[i-1,j]);
    end;
    writeln(f,t);
end;
begin
    nhap;
    assign(f,fo);
    rewrite(f);
    xl;
    close(f);
end.

```

#### ◆ **Bài 42:**

Nhập dãy số nguyên.

a/ Hãy đưa ra các phần tử mà xuất hiện trong dãy đúng 1 lần

b/ Hãy tìm các số trong dãy xuất hiện quá 1 lần

#### **Chương trình:**

```

program vd;
const fi='bai1.inp';
      fo='bai1.out';
type arr=array[1..100]of integer;
var f:text; a:arr; n:byte;
procedure nhap;
var i:byte;
begin
    assign(f,fi);
    reset(f);
    read(f,n);
    for i:=1 to n do
        read(f,a[i]);
    close(f);
end;
function XH(x:integer):byte;
var d,i:byte;
begin
    d:=0;
    for i:=1 to n do
        if(a[i]=x)then inc(d);
    XH:=d;
end;
function KT(x:integer;vt:byte):Boolean;
var i:byte;
begin
    KT:=false;
    for i:=1 to vt-1 do

```

```

        if(a[i]=x)then exit;
    KT:=True;
end;
procedure Xuat(a:arr;spt:byte);
var i:byte;
begin
    for i:=1 to spt do
        write(f,a[i]:5);
        writeln(f);
    end;
procedure xl;
var sptb,sptc,i:byte;  b,c:arr;
begin
    sptb:=0;
    sptc:=0;
    for i:=1 to n do
        if(XH(a[i])=1)then
            begin
                inc(sptb);
                b[sptb]:=a[i];
            end else
                if(KT(a[i],i))then
                    begin
                        inc(sptc);
                        c[sptc]:=a[i];
                    end;
    end;
    if(sptb>0)then
        begin
            writeln(f,'Cac so xuat hien 1 lan la:');
            Xuat(b,sptb);
        end else writeln(f,'Khong co so nao xuat hien 1 lan');
    if(sptc>0)then
        begin
            writeln(f,'Cac so xuat hien qua 1 lan la:');
            Xuat(c,sptc);
        end else writeln(f,'Khong co so nao xuat hien qua 1 lan');
    end;
begin
    Nhap;
    assign(f,fo);
    rewrite(f);
    xl;
    close(f);
end.

```

◆ **Bài 43:**

Cho dãy số nguyên. Gọi M và m lần lượt là các giá trị lớn nhất và bé nhất của dãy số nguyên. Hãy đưa ra các số nguyên theo thứ tự tăng dần thuộc đoạn [m.M] mà không xuất hiện trong dãy.

Dữ liệu nhập: Cho trong file MINMAX.INP gồm 2 dòng;

- Dòng 1: Số phần tử của dãy n.
- Dòng 2: Dãy số nguyên, giữa các số cách nhau 1 khoảng trắng.

Kết quả: Ghi ra file MINMAX.OUT 1 dòng duy nhất là dãy tìm được.

Ví dụ:

MINMAX.INP	MINMAX.OUT
5	2 4 7 8
9 1 5 3 6	

### **Chương trình:**

```

program vd;
const fi='minmax.inp';
      fo='minmax.out';
type arr=array[1..100]of integer;
var f:text; a:arr; n:byte; Min,Max:integer;
procedure nhap;
var i:byte;
begin
    assign(f,fi);
    reset(f);
    read(f,n);
    for i:=1 to n do
        read(f,a[i]);
    close(f);
end;
procedure MIN_MAX(var Min,Max:integer;a:arr);
var i:byte;
begin
    Min:=a[1];
    Max:=a[1];
    for i:=2 to n do
        begin
            if(Min>a[i])then Min:=a[i];
            if(Max<a[i])then Max:=a[i];
        end;
end;
Function KT(a:arr;x:integer):Boolean;
var i:byte;
begin
    KT:=False;
    for i:=1 to n do
        if(x=a[i])then exit;
    KT:=True;
end;

```



```

procedure xl;
var i:byte;
begin
    MIN_MAX(Min,Max,a);
    for i:=Min+1 to Max-1 do
        if(KT(a,i))then write(f,i:5);
end;
begin
    Nhap;
    assign(f,fo);
    rewrite(f);
    xl;
    close(f);
end.

```

#### ◆ **Bài 44:**

Cho các số nguyên  $a_1, a_2, \dots, a_n; b_1, b_2, \dots, b_n$ . Hãy kiểm tra xem có phải 2 dãy trên chỉ khác nhau thứ tự sắp xếp các số hay không?

#### **Chương trình:**

```

program vd;
const fi='bai.inp';
      fo='bai.out';
type arr=array[1..100]of integer;
var f:text;  a,b:arr;  n:byte;
procedure nhap;
var i:byte;
begin
    assign(f,fi);
    reset(f);
    read(f,n);
    for i:=1 to n do
        read(f,a[i]);
    for i:=1 to n do
        read(f,b[i]);
    close(f);
end;
procedure SX(var a:arr);
var i,j:byte;  t:integer;
begin
    for i:=1 to n-1 do
        for j:=i+1 to n do
            if(a[i]>a[j])then
                begin
                    t:=a[i];
                    a[i]:=a[j];
                    a[j]:=t;
                end;
end;

```

```

end;
Function KT(a,b:arr):boolean;
var i:byte;
begin
    KT:=False;
    for i:=1 to n do
        if(a[i]<>b[i])then exit;
    KT:=True;
end;
procedure xl;
begin
    SX(a);
    SX(b);
    if(KT(a,b))then writeln(f,'Dung')else writeln(f,'Khong');
end;
begin
    Nhap;
    assign(f,fo);
    rewrite(f);
    xl;
    close(f);
end.

```

#### ◆ **Bài 45:**

Cho ma trận thực cấp  $n \times n$ . Hãy xây dựng dãy  $a_1, a_2, \dots, a_n$  theo quy tắc: Nếu phần tử trên dòng thứ  $i$  nằm trên đường chéo chính mà âm thì ai bằng tổng các phần tử nằm trước phần tử âm đầu tiên. Trong trường hợp ngược lại ai bằng tổng các phần tử nằm sau phần tử không âm đầu tiên.

#### **Chương trình:**

```

program vd;
const fi='bai.inp';
      fo='bai.out';
type arr=array[1..10,1..10]of integer;
      arr1=array[1..10]of integer;
var f:text;  a:arr;  b:arr1;  n:byte;
procedure nhap;
var i,j:byte;
begin
    assign(f,fi);
    reset(f);
    read(f,n);
    for i:=1 to n do
        for j:=1 to n do
            read(f,a[i,j]);
    close(f);
end;

```

```

procedure xl;
var i,j,e:byte;  t:integer;
begin
  for i:=1 to n do
  begin
    t:=0;
    if(a[i,1]<0)then
    begin
      j:=1;
      while(a[i,j]>=0)do
      begin
        t:=t+a[i,j];
        inc(j);
      end;
    end else
    begin
      j:=1;
      while(a[i,j]<0)and(j<=n)do
        inc(j);
      inc(j);
      for e:=j to n do
        t:=t+a[i,e];
      end;
      b[i]:=t;
    end;
  end;
  for i:=1 to n do
    write(f,b[i]:5);
end;
begin
  Nhap;
  assign(f,fo);
  rewrite(f);
  xl;
  close(f);
end.

```

#### ◆ **Bài 46:**

Cho số tự nhiên  $n$ , số thực  $x$  và ma trận cấp  $n \times 2n$ . Hãy tìm dãy  $b_1, b_2, \dots, b_n$  trong đó  $b_i = 1$  nếu các phần tử của dòng thứ  $i$  không lớn hơn  $x$ ,  $b_i = 0$  trong trường hợp ngược lại.

#### **Chương trình:**

```

program vd;
const fi='bai.inp';
      fo='bai.out';
type arr=array[1..10,1..100]of real;
      arr1=array[1..10]of byte;
var f:text;  a:arr;  b:arr1;  n:byte;  x:real;
procedure nhap;

```

```

var i,j:byte;
begin
    assign(f,fi);
    reset(f);
    read(f,x);
    read(f,n);
    for i:=1 to n do
        for j:=1 to n*2 do
            read(f,a[i,j]);
        close(f);
    end;
function KT(i:byte;x:real):Boolean;
var j:byte;
begin
    KT:=False;
    for j:=1 to n*2 do
        if(a[i,j]>x)then exit;
    KT:=True;
end;
procedure xl;
var i,j:byte;
begin
    for i:=1 to n do
        if(KT(i,x))then b[i]:=1 else
            b[i]:=0;
    for i:=1 to n do
        write(f,b[i]:5);
end;
begin
    Nhap;
    assign(f,fo);
    rewrite(f);
    xl;
    close(f);
end.

```

#### ◆ **Bài 47: BÀI TOÁN DIỆN TÍCH (DIENTICH.PAS)**

Một bản đồ hình chữ nhật gồm một số lãnh thổ khác nhau. Người ta chia lưới các ô vuông đủ nhỏ trên bản đồ để có thể xem như mỗi vùng lãnh thổ được bao gồm một số ô vuông kề cạnh. Các ô vuông thuộc cùng một vùng được ghi nhận bởi cùng một giá trị nguyên dương (gọi là số hiệu vùng). Các số hiệu vùng giống nhau không nhất thiết liên tục.

##### ***Yêu cầu:***

Viết chương trình xác định diện tích của tất cả các vùng trên một bản đồ cho trước (tính theo đơn vị ô vuông).

##### ***Dữ liệu vào: (DIENTICH.INP)***

Dữ liệu đọc vào từ file văn bản, dòng đầu ghi kích thước  $m, n$  ( $1 \leq n, m \leq 100$ ) của bản đồ. Dòng thứ  $i$  trong  $m$  dòng tiếp theo ghi lần lượt các số  $A(i,j)$ ,  $j = 1, 2, \dots, n$  là giá trị các ô

vuông tương ứng trên dòng. Các số trên cùng một dòng được ghi cách nhau bởi một khoảng trắng.

**Dữ liệu ra:** (*DIENTICH.OUT*)

Kết quả được ghi trên file gồm các hàng. Mỗi hàng gồm hai phần tử là số hiệu vùng và diện tích của vùng đó. Hai phần tử cách nhau bởi một khoảng trắng.

**Ví dụ:**

DIENTICH.INP	DIENTICH.OUT
5 5	1 8
1 1 2 3 3	2 6
1 2 2 2 3	3 3
1 1 1 2 2	6 5
1 1 6 6 6	7 3
6 6 7 7 7	

**Chương trình:**

Uses CRT;

Const MAX = 100;

Type

DAY = array[1..MAX\*MAX] of integer;

MANG = array[1..MAX,1..MAX] of integer;

Procedure ThemPhanTu(x: integer; var d,c: DAY; var s: integer);

var i : integer;

Begin

for i := 1 to s do

if d[i] = x then

begin

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

exit;

end;

s := s + 1;

d[s] := x;

c[s] := 1;

End;

Procedure InKQ(var d,c: DAY; s: integer);

var i: integer;

Begin

for i := 1 to s do

writeln(d[i], ' : ', c[i]);

End;

Procedure DienTich(var a: MANG; m,n: integer);

var

d,c: DAY;

s, i, j: integer;

Begin

s := 0;

for i := 1 to n do

for j := 1 to m do

ThemPhanTu(a[i,j],d,c,s);

```

    InKQ(d,c,s);
End;
Var
    a: MANG;
    m,n: integer;
BEGIN
    m := 4;
    n := 4;
    a[1,1] := 1; a[1,2] := 2; a[1,3] := 3; a[1,4] := 4;
    a[2,1] := 1; a[2,2] := 2; a[2,3] := 3; a[2,4] := 4;
    a[3,1] := 1; a[3,2] := 2; a[3,3] := 3; a[3,4] := 4;
    a[4,1] := 1; a[4,2] := 2; a[4,3] := 3; a[4,4] := 4;
    DienTich(a, n, m);
    readln;
END.

```

#### ◆ **Bài 48 : MATCHING (MATCHING.PAS)**

Cho hai chuỗi ký tự. Viết chương trình tìm kiếm chuỗi con dài nhất có trong hai chuỗi trên. Ta không quan tâm đến chữ hoa và chữ thường, tức là chữ hoa và chữ thường là giống nhau.

##### **Dữ liệu:**

Dữ liệu chứa trong file MATCHING.INP gồm nhiều test. Mỗi test gồm hai chuỗi trên hai dòng. Dòng đầu tiên cho biết số test.

MATCHING.INP

2

Helloworld

World

What is the biggest thing in the world?

The biggest thing in the world is universal.

##### **Kết quả:**

Kết quả được ghi ra file MATCHING.OUT. Tùy theo số test mà ta có bấy nhiêu kết quả. Mỗi dòng là một kết quả tương ứng với mỗi test.

MATCHING.OUT

world

the biggest thing in the world

##### **Chương trình:**

Program Matching;

Uses Strings;

Function SubStr(str: string; pos1, pos2: integer): string;

Var s: string;

Begin

    s := "";

    while pos1 <= pos2 do

    begin

        s := s + str[pos1];

        pos1 := pos1 + 1;

    end;

    SubStr := s;

End;

Procedure MaxMatching(str1, str2: string);



**Chương trình:**

```
const fi='base.inp';
      fo='base.out';
type mang=array[1..100] of integer;
var a:mang;  n,m,c:integer;
procedure nhap;
var f:text;
begin
    assign(f,fi);
    reset(f);
    read(f,n,c);
    close(f);
end;
procedure doi(var a:mang;n,j:integer;var m:integer);
var i:integer;
begin
    i:=0;
    repeat
        inc(i);
        a[i]:=n mod j;
        n:=n div j;
    until n=0;
    m:=i;
end;
procedure xuat;
var f:text;
    i,j,ln,d,t:integer;
begin
    assign(f,fo);
    rewrite(f);
    m:=n;
    ln:=0;
    repeat
        if m mod 10 = c then inc(ln);
        m:=m div 10;
    until m=0;
    t:=ln;
    for i:=2 to 9 do
        begin
            d:=0;
            doi(a,n,i,m);
            for j:=1 to m do
                if a[j]=c then inc(d);
            if d>ln then ln:=d;
        end;
    for i:=2 to 9 do
        begin
            d:=0;
```



```

        doi(a,n,i,m);
        for j:=1 to m do
            if a[j]=c then inc(d);
            if d=ln then write(f,i,' ');
        end;
        if ln=t then write(f,10);
        close(f);
    end;
begin
    nhap;
    xuất;
end.

```

#### ◆ **Bài 50: Số dư thừa**

Số tự nhiên  $N$  gọi là dư thừa nếu tổng các ước số của  $N$  ( bao gồm cả 1 nhưng không có  $N$  ) lớn hơn  $N$

##### Bài toán:

Tìm số  $M$  là số dư thừa nhỏ nhất không nhỏ hơn  $N$

Dữ liệu: Cho trong file văn bản ANUM.INP, gồm số tự nhiên  $N$

Kết quả: Cho trong tập tin văn bản ANUM.OUT, gồm số  $M$  là số dư thừa nhỏ nhất không nhỏ hơn  $N$ . Giả sử rằng kết quả là số tự nhiên không quá 10000.

Ví dụ:

ANUM.INP	ANUM.OUT
6	12

Giải thích:

$$3+2+1=6$$

$$1 < 7$$

$$4+2+1 < 8$$

$$3+1 < 9$$

$$5+2+1 < 10$$

$$1 < 11$$

$$6+4+3+2+1 > 12$$

##### **Chương trình:**

```

const fi='anum.inp';
      fo='anum.out';
var n:integer;
procedure nhap;
var f:text;
begin
    assign(f,fi);
    reset(f);
    read(f,n);
    close(f);
end;
function kt(n:integer):boolean;
var i,t:integer;

```

```

begin
    t:=0;
    for i:=1 to n-1 do
        if n mod i=0 then t:=t+i;
    if t>n then kt:=true
    else kt:=false;
end;
procedure xuất;
var f:text;
begin
    assign(f,fo);
    rewrite(f);
    repeat
        if kt(n) then
            begin
                writeln(f,n);
                close(f);
                exit;
            end;
        inc(n);
    until n=10000;
    writeln(f,'qua gioi han');
    close(f);
end;
begin
    nhap;
    xuất;
end.

```

#### ◆ **Bài 51: MAX**

Trong 1 dãy số nguyên, số lần mà 1 số nguyên xuất hiện được gọi là tần số của số nguyên đó. Tần số lớn nhất của dãy số nguyên là tần số lớn nhất trong các tần số của các số nguyên của dãy.

Bài toán: Cho 1 dãy các số nguyên, tìm tần số lớn nhất của dãy số nguyên đó.

Dữ liệu: Cho trong tập tin MAX.INP gồm nhiều dòng. Dòng đầu chứa số nguyên duy nhất N (  $1 \leq N \leq 10000$  ) chỉ số phần tử của dãy số nguyên. Mỗi dòng trong N dòng tiếp theo chứa 1 số nguyên n (  $1 \leq n \leq 1000$  ) của dãy đó.

Kết quả: Cho trong tập tin văn bản MAX.OUT chứa 1 số nguyên duy nhất chỉ tần số lớn nhất của dãy số nguyên tương ứng trong tập tin dữ liệu.

Ví dụ:

MAX.INP	MAX.OUT
12	3
1	
2	
5	
6	
3	

7	
11	
345	
754	
2	
5	
2	

### **Chương trình:**

```

const fi='max.inp';
      fo='max.out';
type mang=array[1..10000] of integer;
var a:mang;  n:integer;
procedure nhap;
var f:text;
      i:integer;
begin
  assign(f,fi);
  reset(f);
  read(f,n);
  for i:=1 to n do
    read(f,a[i]);
  close(f);
end;
procedure xuất;
var f:text;  i,j,d,ln:integer;
begin
  ln:=0;
  assign(f,fo);
  rewrite(f);
  for i:=1 to n do
    begin
      d:=0;
      for j:=1 to n do
        if a[j]=a[i] then inc(d);
      if d>ln then ln:=d;
    end;
  writeln(f,d);
  close(f);
end;
begin
  nhap;
  xuất;
end.

```

### **◆ Bài 52: Số đơn điệu**

Các số nguyên dương 3748, 58, 859, 324354, 65768 được gọi là các số đơn điệu do nếu ta quan sát chữ số của các số này ta thấy chúng luân phiên tăng giảm hoặc giảm tăng. Chẳng hạn:  $3 < 7 > 4 < 8$

Số chỉ có 1 chữ số là số đơn điệu chiều dài 1

Nhiệm vụ: Xác định chỉ số chữ số đầu tiên lớn nhất tạo thành số đơn điệu của 1 số cho trước

Dữ liệu vào: Cho trong file DONDIEU.INP gồm 1 dòng chứa 1 số nguyên duy nhất không quá 75 chữ số

Dữ liệu ra: Cho trong file DONDIEU.OUT, gồm 1 số nguyên dương duy nhất là chỉ số chữ số đầu tiên lớn nhất tạo thành số đơn điệu của số tương ứng trong tập **tin dữ liệu**

Ví dụ:

DONDIEU.INP	DONDIEU.OUT
3748	4

### **Chương trình:**

```
const fi='dondieu.inp';
      fo='dondieu.out';
type mang=array[1..75] of char;
var a:mang;  n:integer;
procedure nhap;
var f:text;
begin
  assign(f,fi);
  reset(f);
  n:=0;
  repeat
    inc(n);
    read(f,a[n]);
  until eof(f);
  close(f);
end;
function kt:boolean;
var i,kiem:integer;
begin
  kt:=true;
  if a[1]>a[2] then kiem:=1
  else kiem:=-1;
  for i:=1 to n-1 do
    begin
      if (kiem=1) and (a[i]<=a[i+1]) then kt:=false;
      else if (kiem=-1) and (a[i]>=a[i+1]) then kt:=false;
      kiem:=kiem*-1;
    end;
  end;
end;
procedure xuat;
var f:text;  i,j:integer;  ln:char;
begin
  assign(f,fo);
```

```

rewrite(f);
if (kt)or(n=1) then
begin
  ln:=a[1];
  for i:=2 to n do
    if a[i]>ln then ln:=a[i];
  for i:=1 to n do
    if a[i]=ln then
      begin
        writeln(f,i);
        break;
      end;
  end
else writeln(f,0);
close(f);
end;
begin
  nhap;
  xuất;
end.

```

### ◆ **Bài 53: Số đối xứng**

Một số mà đọc từ trái sang phải giống hệt như đọc từ phải sang trái gọi là số đối xứng.

Ví dụ: 14541 là số đối xứng, 0330 không là số đối xứng. Số 21 ( biểu diễn trong cơ số 10) không là số đối xứng nhưng số 21 là số đối xứng nếu biểu diễn trong cơ số 2 ( 10101)

Nhiệm vụ: Viết chương trình đọc 2 số ( cơ số 10)

- N (  $1 \leq N \leq 15$  )

- S (  $0 < S < 10000$  )

Và xuất ra ( trong cơ số 10 )

- N số đầu tiên > S và là số đối xứng khi biểu diễn trong ít nhất 2 cơ số c khác nhau (  $2 \leq c \leq 10$  )

- Số số nguyên tố trong N số trên.

Dữ liệu vào: Gồm 1 dòng duy nhất chứa 2 số nguyên N và S cách nhau 1 khoảng trắng

Dữ liệu ra: Gồm N dòng chứa N số đầu tiên đối xứng khi biểu diễn ít nhất trong 2 hệ số số c (  $2 \leq c \leq 10$  ), sắp xếp tăng dần.

Dòng N+1 chứa 1 số nguyên duy nhất chỉ số số nguyên tố trong N dòng trên

Ví dụ:

DOIXUNG.INP	DOIXUNG.OUT
4	33
32	36
	40
	45
	0

### **Chương trình:**

```

const fi='doixung.inp';
      fo='doixung.out';
type mang=array[1..100] of integer;

```

```

var a:mang;  n,m,s,k:integer;
procedure nhap;
var f:text;
begin
    assign(f,fi);
    reset(f);
    read(f,n,s);
    close(f);
end;
procedure doi(var a:mang;k,j:integer;var m:integer);
var i:integer;
begin
    i:=0;
    fillchar(a,sizeof(a),0);
    repeat
        inc(i);
        a[i]:=k mod j;
        k:=k div j;
    until k=0;
    m:=i;
end;
function doixung(a:mang; m: integer):boolean;
var  i:integer;
begin
    doixung:=true;
    for i:=1 to m div 2 do
        if a[i]<>a[m-i+1] then
            begin
                doixung:=false;
                exit;
            end;
    end;
end;
procedure tach(k:integer; var a: mang; var m: integer);
var  i: integer;
begin
    i:=0;
    fillchar(a,sizeof(a),0);
    repeat
        inc(i);
        a[i]:= k mod 10;
        k:=k div 10;
    until k=0;
    m:=i;
end;
function ktnt(t: integer): boolean;
var i: integer;
begin
    ktnt:=true;

```

```

for i:= 2 to trunc(sqrt(t)) do
  if t mod i = 0 then
    begin
      ktnt:=false;
      exit;
    end;
end;
procedure xuly;
var f:text;  i,j,dem,d,d1,t:integer;b:mang;
begin
  assign(f,fo);
  rewrite(f);
  dem:=0;
  k:=s+1;
  repeat
    d:=0;
    for i:=2 to 9 do
      begin
        doi(a,k,i,m);
        if doixung(a,m) then inc(d);
      end;
    tach(k,a,m);
    if doixung(a,m) then
      inc(d);
    if d>=2 then
      begin
        inc(dem);
        b[dem]:=k;
      end;
    inc(k);
  until dem=n;

  d1:=0;
  for i:= 1 to dem do
    begin
      writeln(f,b[i]);
      if ktnt(b[i]) then inc(d1);
    end;
  writeln(f,d1);
  close(f);
end;
begin
  nhap;
  xuly;
end.

```

#### ◆ **Bài 54: Dãy tăng giảm tuần hoàn**

Cho trước 1 dãy số nguyên không âm  $(a_n)$ , ( $a_i < 1000$  với mọi  $i$ ,  $n < 1000$ )

Dãy ( $a_n$ ) gọi là 1 dãy tăng giảm tuần hoàn nếu tồn tại số  $k$  dương ( $k < n$ ) sao cho bắt đầu từ số hạng đầu tiên đến số hạng cuối cùng cứ  $k-1$  lần tăng ( giảm ) dần thì tiếp theo có  $k-1$  lần giảm ( tăng ) dần.... ít nhất có 1 dãy con tăng và một dãy con giảm, dãy con tăng ( giảm ) cuối cùng không nhất thiết kéo dài tới  $k-1$  lần.

**Ví dụ các dãy tăng giảm tuần hoàn:**

1 2 3 4 7 4 3 2 1 5 6 7 8 4 3 2 1 4 9

1 2 3 9 3

**Các dãy không tăng giảm tuần hoàn:**

1 2 3 4 5 6 7 8 9

1 2 3 4 5 6 9 8 7 6 9

**Lưu ý:** tăng, giảm tuyệt đối ( không bằng )

**Nhiệm vụ:**

Hãy lập trình để nhận biết dãy số cho có phải là tăng giảm tuần hoàn ( TRUE ) hay không ( FALSE ) ?

**Dữ liệu vào:**

Dữ liệu vào được ghi trên tập tin văn bản DAY.INP. Các phần tử của dãy cách nhau 1 khoảng trắng.

**Dữ liệu ra:**

Dữ liệu ra được ghi trên tập tin DAY.OUT gồm 2 dòng:

- Dòng đầu ghi TRUE hoặc FALSE
- Dòng thứ 2 ghi số nguyên  $k$  cho trường hợp TRUE và 0 cho trường hợp FALSE

**Ví dụ:**

DAY.INP

1 2 3 4 7 4 3 2 1 5 6 7 8 4 3 2 1 4 9

DAY.OUT

TRUE

5

**Chương trình:**

```
const fi='day.inp';
      fo='day.out';
type mang=array [1..1000] of integer;
var a:mang;  n:integer;
procedure Nhap(var n:integer);
var f:text;
begin
  assign(f,fi);
  reset(f);
  n:=0;
  repeat
    inc(n);
    read(f,a[n]);
  until eoln(f);
  close(f);
end;
procedure XuLi(n:integer);
var f:text;  dem,dem1,tam,i:integer;  kt:boolean;
begin
```



```

assign(f,fo);
rewrite(f);
if a[1] > a[2] then tam:=1
else if a[1] < a[2] then tam:=-1
else
begin
    writeln(f,'False');
    writeln(f,0);
    close(f);
    exit;
end;
dem:=1;
repeat
    inc(dem);
    if (tam = 1) and (a[dem] < a[dem+1]) then break
    else if (tam = -1) and (a[dem] > a[dem+1]) then break;
until dem=n;
dec(dem);
dem1:=1;
kt:=true;
for i:=2 to n-1 do
begin
    inc(dem1);
    if dem1 = dem+1 then
    begin
        dem1:=1;
        tam:=tam*(-1);
    end;
    if (tam = 1) and (a[i] <= a[i+1]) then
    begin
        kt:=false;
        break;
    end
    else if (tam = -1) and (a[i] >= a[i+1]) then
    begin
        kt:=false;
        break;
    end;
end;
if (kt=true) and (dem+1 <> n) then
begin
    writeln(f,'True');
    writeln(f,dem+1);
end
else
begin
    writeln(f,'False');
    writeln(f,0);

```

```

end;
close(f);
end;
begin
    Nhap(n);
    XuLi(n);
end.

```

#### ◆ Bài 55:

Xét bảng số nguyên dương A, kích thước  $N \times N$  ( $N \leq 40$ ),  $A[i,j] \leq 100$ . Thay mỗi phần tử  $A[i,j]$  của bảng bằng giá trị  $f(A[i,j])$ , trong đó  $y=f(x)$  là hàm cho số lượng lớn nhất các số nguyên tố có tổng bằng x, mỗi số sử dụng 1 lần, trừ 1 số nguyên tố nào đó có thể sử dụng đúng 2 lần nếu cần.

Ví dụ:

$x=5 \Rightarrow f(x) = 2$  ( $5 = 3 + 2$ )

$x=7 \Rightarrow f(x) = 3$  ( $7 = 3 + 2 + 2$ )

1 không phải là số nguyên tố. Quy ước  $f(1) = 1$

Dữ liệu: Cho trong file INP.BL1 gồm N+1 dòng:

- Dòng đầu là số nguyên dương N
- Dòng i+1 ( $1 \leq i \leq N$ ) gồm N số  $A[i,1], A[i,2], \dots, A[i,N]$

Kết quả: Xuất ra file OUT.BL1 gồm N dòng:

- Dòng i ( $1 \leq i \leq N$ ) gồm N số  $A[i,1] \dots A[i,N]$

Các số ghi trên cùng 1 dòng, ghi cách nhau ít nhất một dấu cách

Ví dụ:

INP.BL1	OUT.BL1
5 5	1 1 7 3 4
1 2 44 11 14	6 4 3 5 5
34 16 9 19 22	2 5 2 7 1
5 24 4 43 3	4 7 5 6 4
15 46 19 39 17	6 7 6 4 3
35 43 36 16 9	

#### Chương trình:

```

uses crt;
const maxN = 40;
      maxVal = 100;
      fi='inp.bl4';
      nt:array[1..25]
byte=(2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97);
      dt:array[1..8,1..2] of shortint=((-1,-1),(-1,0),(-1,1),(0,-1),(0,1),(1,-1),(1,0),(1,1));
var n,i,j,k: byte;   a:array[1..maxN,1..maxN] of byte;   max:integer;
      spt,tam:array[1..maxN] of integer;
procedure nhap;
var f: text;
begin
    clrscr;
    assign(f,fi);
    reset(f);

```

of

```

readln(f,n);
for i:=1 to n do
begin
    for j:=1 to n do
        read(f,a[i,j]);
    readln(f);
end;
end;
procedure F;
begin
    fillchar(spt,sizeof(spt),0);
    tam:=spt;
    spt[1]:=1 ; spt[2]:=1; spt[3]:=1;
    tam[2]:=1 ; tam[3]:=2;
    for i:=4 to max do
    begin
        for k:=1 to 25 do
        begin
            if i < nt[k] then
                k:=25
            else
            begin
                if i > nt[k]+1 then
                    if (spt[i-nt[k]]+1 > spt[i]) and (tam[i-nt[k]]<k) then
                    begin
                        tam[i]:=k;
                        spt[i]:=spt[i-nt[k]]+1;
                    end;
                if (i=nt[k]) and ( spt[i]=0) then
                begin
                    tam[i]:=k;
                    spt[i]:=1;
                end;
            end;
        end;
    end;
end;
tam:=spt;
for i:=4 to max do
begin
    for k:=1 to 25 do
        if i<=nt[k] then k:=25
    else
    begin
        if tam[i-nt[k]]+1 > spt[i] then
        begin
            spt[i]:=tam[i-nt[k]]+1;
        end;
    end;
end;
end;

```

```

    end;
end;
procedure thuchien;
begin
    max:=0;
    for i:=1 to n do
        for j:=1 to n do
            if max<a[i,j] then max:=a[i,j];
        F;
    for i:=1 to n do
        for j:=1 to n do
            a[i,j]:=spt[a[i,j]];
    for i:=1 to n do
    begin
        for j:=1 to n do
            write(a[i,j]:5);
        writeln;
    end;
end;
begin
    nhap;
    thuchien;
end.

```

#### ◆ **Bài 56:**

Cho số tự nhiên  $n$  và ma trận nguyên cấp  $n \times n$ .

Hãy tính  $b_1, b_2, \dots, b_n$  trong đó  $b_k$  là:

- Giá trị bé nhất của các phần tử trên dòng thứ  $k$ , bắt đầu từ phần tử thứ nhất đến phần tử nằm trên đường chéo chính của dòng đó.
- Phần tử đầu tiên của dòng thứ  $k$  là dương. Nếu phần tử đó không tồn tại thì  $b_k = 1$ .
- Tổng các phần tử đứng ngay sau phần tử âm đầu tiên trong dòng thứ  $k$ . Nếu dòng thứ  $k$  chứa toàn số không âm hoặc không tồn tại phần tử nào sau phần tử âm đầu tiên thì  $b_k = 100$ .
- Tổng các phần tử nằm trước phần tử âm cuối cùng của dòng thứ  $k$ . Nếu dòng thứ  $k$  toàn số không âm hoặc không tồn tại phần tử nào trước phần tử âm cuối cùng thì  $b_k = -1$ .

Dữ liệu: Ma trận nguyên được nhập từ file

Kết quả: Thông báo ra file kết quả  $b_1, b_2, \dots, b_n$  theo từng câu.

Ví dụ:

Mảng nhập:

1	-1	2	9
5	2	-5	6
8	4	3	-2
-4	-8	-6	-4

Kết quả:

a.	1	2	3	-8
b.	1	5	8	1
c.	11	6	100	-18
d.	1	7	15	-18

**Chương trình:**

```
const fi='b01.inp';
      fo='b01.out';
type mt=array[1..100,1..100] of integer;
var a:mt;  n:integer;
procedure nhap;
var f:text;  i,j:integer;
begin
  assign(f,fi);
  reset(f);
  read(f,n);
  for i:=1 to n do
    for j:=1 to n do
      read(f,a[i][j]);
  close(f);
end;
procedure caua;
var f:text;  i,j,k:integer;
begin
  assign(f,fo);
  rewrite(f);
  for i:=1 to n do
    begin
      k:=a[i][1];
      for j:=1 to i do
        if a[i][j]<k then k:=a[i][j];
      write(f,k,' ');
    end;
  writeln(f);
  close(f);
end;
procedure caub;
var f:text;  i,j,k:integer;
begin
  assign(f,fo);
  append(f);
  for i:=1 to n do
    if a[i][1]>0 then write(f,a[i][1],' ')
    else write(f,1,' ');
  writeln(f);
  close(f);
end;
procedure cauc;
var f:text;  i,j,k,t:integer;  kt:boolean;
begin
  assign(f,fo);
  append(f);
```

```

for i:=1 to n do
begin
    kt:=false;
    t:=0;
    for j:=1 to n do
    if (a[i][j]<0)and(j<n) then
    begin
        for k:=j+1 to n do
            t:=t+a[i][k];
        kt:=true;
        break;
    end;
    if kt=true then write(f,t,' ')
    else write(f,100,' ');
end;
writeln(f);
close(f);
end;
procedure caud;
var f:text; i,j,k,t:integer; kt:boolean;
begin
    assign(f,fo);
    append(f);
    for i:=1 to n do
    begin
        kt:=false;
        t:=0;
        for j:=n downto 1 do
        if (a[i][j]<0)and(j>1) then
        begin
            for k:=j-1 downto 1 do
                t:=t+a[i][k];
            kt:=true;
            break;
        end;
        if kt=true then write(f,t,' ')
        else write(f,-1,' ');
    end;
    writeln(f);
    close(f);
end;
begin
    nhap;
    caua;
    caub;
    cauc;
    caud;
end.

```

### ◆ **BÀI 57: PHÉP "NHÂN"**

Bờm là một học sinh thông minh. Bờm nghĩ ra cách muốn thực hiện phép "nhân" hai số nguyên dương theo cách khác với cách đã biết. Theo đó, số A "nhân" với số B cho kết quả là số được ký hiệu bởi  $A \otimes B$ , có giá trị là tổng các tích của các cặp chữ số của A và B.

Ví dụ,  $123 \otimes 45 = 1 \times 4 + 1 \times 5 + 2 \times 4 + 2 \times 5 + 3 \times 4 + 3 \times 5 = 54$ .

**Yêu cầu:** Cho hai số nguyên dương A và B ( $A \leq 10^9$ ,  $B \leq 10^9$ ), hãy tìm  $A \otimes B$  theo cách của Bờm.

**Dữ liệu:** Vào từ tập tin văn bản PHEPNHAN.INP, gồm một dòng duy nhất chứa hai số nguyên A, B cách nhau một khoảng trắng.

**Kết quả:** Ghi ra tập tin văn bản PHEPNHAN.OUT một số nguyên duy nhất là kết quả tìm được.

**Ví dụ:**

PHEPNHAN . INP	PHEPNHAN . OUT
123 45	54

### **Chương trình:**

```
const fi = 'phepnhan.inp';
      fo = 'phepnhan.out';
var s1,s2,tcs1,tcs2,kq: longint;{tcs1: tong cac chu so cua so 1}
procedure doc;
var f1: text;
begin
  assign(f1,fi); reset(f1); read(f1,s1,s2); close(f1);
end;
procedure xuly;
begin
  tcs1:=0;tcs2:=0;
  (*tinh tong cac chu so cua so thu nhât*)
  tcs1:=tcs1+ s1 mod 10;
  s1:=s1 div 10;
  while (s1<>0) do
  begin
    tcs1:=tcs1+ s1 mod 10;
    s1:=s1 div 10;
  end;
  (*tinh tong cac chu so cua so thu hai*)
  tcs2:=tcs2+ s2 mod 10;
  s2:=s2 div 10;
  while (s2<>0) do
  begin
```

```

tcs2:=tcs2+ s2 mod 10;
s2:=s2 div 10;
end;
end;
procedure ghi;
var f2: text;
begin
    assign(f2,fo); rewrite(f2); write(f2,tcs1*tcs2); close(f2);
end;
begin
    doc;
    xuly;
    ghi;
end.

```

### ◆ **BÀI 58: GIAI THỪA**

Với mỗi số nguyên dương  $N$ , ta hiểu  $N!$  là tích của  $N$  số nguyên dương đầu tiên.

Cho trước số nguyên dương  $N$  ( $1 < N < 31$ ).

Yêu cầu: Tìm số nguyên dương  $K$  nhỏ nhất sao cho tích của  $K$  và  $N!$  là số chính phương.

Dữ liệu: Vào từ tập tin văn bản GIAITHUA.INP chứa duy nhất số  $N$ .

Kết quả: Ghi ra tập tin văn bản GIAITHUA.OUT số nguyên  $K$  tìm được.

Ví dụ:

GIAITHUA . INP	GIAITHUA . OUT	GIAITHUA . INP	GIAITHUA . OUT
4	6	25	676039

#### **Chương trình:**

```

const p:array[1..15] of byte=(2,3,5,7,11,13,17,19,23,29,31,37,41,43,47);
    fi = 'giaithua.inp';
    fo = 'giaithua.out';
var n:byte;  a:array[1..15] of byte;  t,z:longint;  i,j,k,ii:byte;  f1,f2: text;
procedure result(n:byte);
begin
    for k:=2 to n do
        begin
            i:=k;
            for j:=1 to 15 do
                while i mod p[j]=0 do
                    begin
                        i:=i div p[j];
                        a[j]:=1-a[j];
                    end;
            end;
        end;
    z:=1;

```



```

for j:=1 to 15 do if a[j]=1 then z:=z*p[j];
end;
Begin
  fillchar(a,sizeof(a),0);
  assign(f1,fi);reset(f1);
  read(f1,n);
  close(f1);
  result(n);
  assign(f2,fo);rewrite(f2);
  write(f2,z);
  close(f2);
  result(n);
end.

```

### ◆ **BÀI 59: THAM QUAN**

Có N di tích lịch sử nằm rải rác trên một tuyến đường. Tuyến đường được mô hình hóa như một trục số. Theo một trật tự ngẫu nhiên nào đó, di tích lịch sử thứ i (gọi tắt là điểm i) có tọa độ là  $x_i$ ,  $i = 1, 2, \dots, N$ .

Bờm muốn đi tham quan càng nhiều càng tốt các di tích lịch sử nói trên (do Bờm chưa từng đến). Theo kế hoạch, cuộc tham quan sẽ được bắt đầu từ *điểm khởi hành* có tọa độ bằng 0, Bờm liên tiếp đi từ điểm này đến điểm khác theo quy tắc: luôn đến điểm nào mà nó chưa tham quan và gần với điểm khởi hành nhất. Để di chuyển một đơn vị độ dài, tức là một đơn vị trên trục số, Bờm tốn một đơn vị thời gian và Bờm muốn rằng tổng thời gian di chuyển giữa các điểm không vượt quá T đơn vị.

Biết rằng không có hai di tích lịch sử nào cách đều điểm khởi hành.

**Yêu cầu:** Xác định số Q là số lớn nhất các di tích lịch sử mà Bờm có thể tham quan.

**Dữ liệu:** Vào từ tập tin văn bản THAMQUAN.INP với cấu trúc:

- Dòng đầu là hai số nguyên T, N cách nhau một khoảng trắng ( $1 \leq T \leq 10^9$ ,  $1 \leq N \leq 15000$ ).
- Dòng thứ i trong N dòng tiếp theo, chứa một số nguyên  $x_i$  ( $-100000 \leq x_i \leq 100000$ ,  $x_i \neq 0$ ),  $i = 1, 2, \dots, N$ .

**Kết quả:** Ghi ra tập tin văn bản THAMQUAN.OUT số duy nhất là số Q tìm được.

**Ví dụ:**

THAMQUAN . INP	THAMQUAN . OUT
25 5 10 -3 8 -7 1	4

### **Chương trình:**

```
CONST FI='thamquan.INP';
```

```
Fo='thamquan.out';
```

```

type arr=array[0..15000] of longint;

var a:arr;  n,count:word;  t,s:longint;  f1,f2:text;

procedure data;

var i:word;

    tmp:longint;

begin

    assign(f1,fi);reset(f1);

    readln(f1,t,n);

    for i:=1 to n do readln(f1,a[i]);

    close(f1);

end;

procedure sort;

var i,j,k:word;

    tmp:longint;

begin

    for i:=1 to n-1 do

        begin

            k:=i;

            for j:=i+1 to n do if abs(a[j])<abs(a[k]) then k:=j;

            tmp:=a[i];a[i]:=a[k];a[k]:=tmp;

        end;

    end;

procedure result;

var i: word;

begin

    s:=0;

    count:=0;

```

```

a[0]:=0;

for i:=1 to n do

begin

    s:=s+abs(a[i]-a[i-1]);

    if s>=t then break

    else inc(count);

end;

end;

procedure xuất;

begin

    assign(f2,fo);rewrite(f2); write(f2,count); close(f2);

end;

Begin

data;

sort;

result;

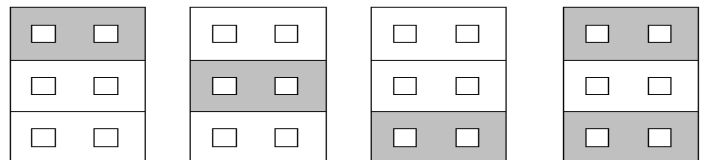
xuất;

end.

```

### ◆ **BÀI 60: SƠN CAO ỐC**

Một tòa cao ốc có N tầng đã được sơn toàn bộ bởi màu trắng. Sau một thời gian sử dụng, người ta nghĩ ra cách "làm mới" nó bằng việc sơn trọn vẹn một số tầng bởi màu xám (các tầng còn lại giữ nguyên màu sơn cũ) sao cho không có hai tầng liên tiếp nào có cùng màu xám.



Chẳng hạn, với  $N = 3$ , số cách sơn sẽ là 4 (xem hình).

Yêu cầu: Viết chương trình tính số cách sơn tòa cao ốc nói trên.

Dữ liệu: Vào từ tập tin văn bản SONCAOOC.INP, gồm duy nhất một số nguyên dương  $N$  ( $N \leq 40$ ), là số tầng của cao ốc.

Kết quả: Ghi ra tập tin văn bản SONCAOOC.OUT một số nguyên duy nhất là số cách sơn tìm được (số này không vượt quá  $10^9$ ).

Ví dụ:

SONCAOOC . INP	SONCAOOC . OUT
3	4

**Chương trình:**

```
const fi = 'soncaooc.inp';
      fo = 'soncaooc.out';
var s1,s2,s3: longint; i,n: byte; f1,f2: text;
begin
  assign(f1,fi);reset(f1); read(f1,n); close(f1);
  s1:=1;s2:=1;
  for i:=1 to n do
  begin
    s3:=s2+s1;
    s1:=s2;s2:=s3;
  end;
  assign(f2,fo);rewrite(f2);write(f2,s3-1);close(f2);
end.
```

◆ **Bài 61:**

Cho số tự nhiên a. Hãy cho biết a có dạng:  $3^m.5^n$  không?

Nếu không trả lời “Không”

Nếu có trả lời “Có” và số m, n.

Dữ liệu nhập: Cho trong file BAI.INP gồm 1 số tự nhiên duy nhất a.

Kết quả: Xuất ra file BAI.OUT gồm 1 dòng trả lời.

Ví dụ :

BAI.INP	BAI.OUT
10	KHONG

**Chương trình:**

```
const fi='bai.inp';
      fo='bai.out';
var a:integer;
procedure nhap;
var f:text;
begin
  assign(f,fi);
  reset(f);
  read(f,a);
  close(f);
end;
procedure xuly;
var m,n:integer;
begin
  assign(f,fo);
  rewrite(f);
  m:=0;
  n:=0;
  while a mod 3=0 do
```

```

begin
    inc(m);
    a:=a div 3;
end;
while a mod 5=0 do
begin
    inc(n);
    a:=a div 5;
end;
if a=1 then write(f,'Co',m,n)
else write(f,'Khong');
close(f);
end;
begin
    nhap;
    xuly;
end.

```

### ◆ **Bài 62:**

Dãy H là dãy tăng các số tự nhiên, mà mỗi số của dãy biểu diễn được dưới dạng  $2^i \cdot 3^j$  (  $i, j$  là các số nguyên không âm).

Nhiệm vụ: Viết chương trình sinh ra dãy nói trên.

Dữ liệu: Cho trong file H.inp: 1 dòng duy nhất là số phần tử cần phải sinh ra ( $1 < n < 32000$ )

Kết quả: Ghi vào tập tin văn bản H.out, trên mỗi dòng của tập tin kết quả là số nguyên, lần lượt là chỉ số của phần tử và phần tử tương ứng của dãy. Các phần tử được viết theo thứ tự của chỉ số.

Ví dụ:

H.inp	H.out
10	1 1 2 2 3 3 4 4 5 6 6 8 7 9 8 12 9 16 10 18

### **Chương trình:**

```

const fi='h2.inp';
      fo='h2.out';
var f:text;k:integer;
procedure nhap;
begin
    assign(f,fi);

```

```

    reset(f);
    read(f,k);
    close(f);
end;
function kt(a:qword):boolean;
begin
    while a mod 2=0 do a:=a div 2;
    while a mod 3=0 do a:=a div 3;
    if a=1 then kt:=true
    else kt:=false;
end;
procedure xuly;
var dem:integer;n:qword;
begin
    assign(f,fo);
    rewrite(f);
    dem:=0;
    n:=1;
    while dem<k do
    begin
        if kt(n) then
        begin
            inc(dem);
            writeln(f,dem,' ',n);
        end;
        inc(n);
    end;
    close(f);
end;
begin
    nhap;
    xuly;
end.

```

### ◆ **Bài 63:**

Cho trước số nguyên dương N, tìm số nguyên dương nhỏ nhất có được bằng cách đổi vị trí của không quá hai chữ số của N.

Chú ý: N là số nguyên dương có tối đa 7 chữ số.

Dữ liệu:

Vào từ file văn bản SWAP.INP, gồm 1 dòng duy nhất chứa duy nhất số nguyên dương N

Kết quả:

Ghi ra file văn bản SWAP.OUT, gồm 1 dòng duy nhất chứa số nguyên nhỏ nhất tương ứng tìm được.

Ví dụ:

SWAP.INP	SWAP.OUT
21435	12435

**Chương trình:**

```
const fi='swap.inp';
      fo='swap.out';
type arr=array[1..7] of integer;
var a,b:arr;      n:longint;spt:integer;
procedure nhap;
var f: text;
begin
  assign(f,fi);
  reset(f);
  read(f,n);
  close(f);
end;
procedure tach;
var i:integer;
begin
  spt:=0;
  while n<>0 do
  begin
    inc(spt);
    a[spt]:=n mod 10;
    n:=n div 10;
  end;
  for i:= 1 to spt do
    b[i]:=a[spt+1-i];
end;
procedure xuly;
var min,k,i,tam,j,min2:integer;f:text;
begin
  for i:=1 to spt-1 do
    for j:=i+1 to spt do
      if a[i]>a[j] then
        begin
          tam:=a[i];
          a[i]:=a[j];
          a[j]:=tam;
        end;
  for i:=1 to spt do
    if a[i]<>0 then
      begin
        min2:=a[i];
        break;
      end;
  min:=b[1];
  k:=1;
  for i:=2 to spt do
    if min>b[i] then
      begin
```

```

        min:=b[i];
        k:=i;
    end;
if (min<>0) and (k<>1) then
begin
    tam:=b[1];
    b[1]:=b[k];
    b[k]:=tam;
end
else
begin
    k:=2;
    for i:=2 to spt do
        if min2=b[i] then
            k:=i;
            tam:=b[1];
            b[1]:=b[k];
            b[k]:=tam;
        end;
    assign(f,fo);
    rewrite(f);
    for i:=1 to spt do write(f,b[i]);
    close(f);
end;
begin
    nhap;
    tach;
    xuly;
end.

```

#### ◆ **Bài 64:**

Cho trước 1 xâu ký tự, trong đó có ít nhất 4 chữ số. Hãy loại bỏ 1 số ký tự khỏi xâu sao cho 4 ký tự cuối cùng còn lại theo đúng thứ tự đó tạo nên 1 số lớn nhất  
 Ví dụ nếu cho xâu “24d5n4r05f704n652z393” thì số lớn nhất còn lại là 7693.  
 Hãy lập trình để giải bài toán trên.

#### **Chương trình:**

```

Var s:string;    C:char;    n,i,j,k:integer;
Begin
    Write('Nhập xâu:');
    Readln(s);
    i:=1;
    Repeat
        If s[i] in ['0'..'9'] then inc(i) else delete(s,i,1);
    Until i>length(s);
    For i:=1 to 4 do
        Begin
            k:=i;

```



```

        For j:=i to length(s)+i-4 do
            If s[k]<s[j] then k:=j;
        If k>i then delete(s,i,k-i);
    End;
    If length(s)>4 then delete(s,5,length(s)-4);
    Write(s);
    Readln;
End.

```

#### ◆ **Bài 65:**

Trong một bức thư mà Christian Goldbach gửi cho Euler, ông đã đề cập đến phỏng đoán của mình: Mọi số tự nhiên chẵn lớn hơn 2 đều là tổng của 2 số nguyên tố. Hãy lập chương trình để kiểm chứng phỏng đoán của Goldbach.

Yêu cầu:

- Dữ liệu vào từ file GB.INP gồm nhiều dòng, dòng đầu là số test (<10), các dòng tiếp theo mỗi dòng ghi 1 số tự nhiên chẵn lớn hơn 2 (<32000)
- Dữ liệu ra là file GB.OUT gồm các dòng (mỗi dòng ứng với 1 test) - mỗi dòng gồm 2 số nguyên tố cách nhau ít nhất 1 dấu cách có tổng bằng số đã cho (hoặc không tìm được – ghi là “khong”).

**GB.INP**

```

3
8
12
6

```

**GB.OUT**

```

3 5
5 7
2 3

```

#### **Chương trình:**

```

program bt;
type mang=array[1..50] of integer;
var f:text;n:integer;a:mang;
const fi='gb.inp';
      fo='gb.out';
procedure nhap;
var i:integer;
begin
    assign(f,fi);
    reset(f);
    read(f,n);
    for i:=1 to n do
        read(f,A[i]);
    close(f);
end;
function kt(a:integer):boolean;
var i:integer;
begin

```

```

kt:=true;
for i:=2 to trunc(sqrt(a)) do
  if a mod i=0 then
    begin
      kt:=false;
      exit;
    end;
end;
procedure xuli;
var i,j,b:integer;
begin
  assign(f,fo);
  rewrite(f);
  for i:=1 to n do
    begin
      b:=0;
      for j:=A[i]-2 downto 2 do
        if (kt(j)) and (kt(A[i]-j)) then
          begin
            writeln(f,a[i]-j,' ',j);
            b:=1;
            break;
          end;
      if b=0 then write(f,'Khong');
    end;
  close(f);
end;
begin
  nhap;
  xuli;
end.

```

#### ◆ **Bài 66:**

Cho trước hai số nguyên dương  $n$  và  $k$

**Yêu cầu 1:** Hãy cho biết có bao nhiêu số có số chữ số nhỏ hơn hoặc bằng  $n$  mà tổng các chữ số đúng bằng  $k$

**Yêu cầu 2:** Cho số nguyên dương  $p$ , hỏi nếu đem các số tìm được sắp xếp theo thứ tự tăng dần thì số thứ  $p$  là số nào ( $p$  không lớn hơn số lượng các số tìm được).

**Dữ liệu vào:** Đọc từ file văn bản **DIGITSUM.INP** gồm một dòng chứa ba số  $n, k, p$  tương ứng, mỗi số cách nhau một dấu cách.

**Kết quả:** Ghi ra file văn bản **DIGITSUM.OUT** gồm 2 dòng:

- Dòng 1: Ghi số lượng các số tìm được theo yêu cầu 1.
- Dòng 2: Ghi số thứ  $p$  tìm được theo yêu cầu 2.

**Ví dụ:**

DIGITSUM.INP	DIGITSUM.OUT
2 3 2	4
	12

**Chương trình:**

```
program kt;
type mang=array[1..10000000] of longint;
var n,p,k,spt,t:longint;f:text;a,b:mang;
const fi='digitsum.inp';
      fo='digitsum.out';
procedure nhap;
begin
    assign(f,fi);
    reset(f);
    read(f,n,k,p);
    close(f);
end;
procedure dem;
var i:longint;
begin
    i:=1;
    t:=10;
    repeat
        inc(i);
        t:=t*10;
    until i=n;
end;
procedure xulia;
var j,tong,c,i,d:longint;
begin
    spt:=0;
    for i:=1 to t-1 do
    begin
        c:=0;
        d:=i;
        repeat
            inc(c);
            b[c]:=d mod 10;
            d:=d div 10;
        until d=0;
        tong:=0;
        for j:=1 to c do
            tong:=tong+b[j];
        if tong=k then
        begin
            inc(spt);
            a[spt]:=i;
        end;
    end;
end;
procedure xuất;
```

```

begin
    assign(f,fo);
    rewrite(f);
    writeln(f,spt);
    write(f,a[p]);
    close(f);
end;
begin
    nhap;
    dem;
    xulia;
    xuất;
end.

```

### ◆ **BÀI 67: Y3K**

Cho số nguyên N trong phạm vi từ 1000 đến 999999. Cần xác định số này có phải là thông tin về một ngày tháng có trong thế kỷ 21 không. (Thế kỷ 21 bắt đầu từ 1 tháng 1 năm 2001 và kết thúc vào ngày 31 tháng 12 năm 3000. Biết rằng 2 chữ số cuối của N là chỉ hai chữ số cuối của năm, các chữ số còn lại (ở đầu) xác định ngày và tháng.

Ví dụ:

1111     tương ứng với 1 tháng 1 năm 2011;  
 21290   tương ứng với 2 tháng 12 năm 2090 hoặc 21 tháng 2 năm 2090;  
 131192   tương ứng với 13 tháng 11 năm 2092;  
 32392   Không phải là thông tin về một ngày tháng nào cả;  
 311198   Không phải là thông tin về một ngày tháng nào cả;  
 29205   Không phải là thông tin về một ngày tháng nào cả;

Dữ liệu: Nhập vào số N từ bàn phím.

Kết quả: Đưa ra màn hình các ngày tháng năm tương ứng với N hoặc thông báo là KHONG nếu N không phải là thông tin về một ngày tháng nào cả.

Ví dụ:

Giá trị của N	Thông báo ra màn hình tương ứng
1111	1-1-2011
21290	2-12-2090 HOAC 21-2-2090
29205	KHONG

### **Chương trình:**

```

Var n,m,nam,thang,ngay:longint; fl:boolean;
Procedure nhap;
Begin
    Write('Nhap n:');
    Readln(n);
End;
Procedure kt(n:longint);
Begin
    Case n of
        1,3,5,7,8,10,12:fl:=31;
        4,6,9,11:fl:=30;
    
```

```

    End;
    If n =2 then
        Nếu năm nhuận fl:=29
        Else fl:=28;
    If ngay<=fl then write(f,ngay,'-','thang','-',nam);
End.
Procedure xuly;
Begin
    If n mod 100 then nam:=3000
    Else nam:=2000+n mod 100;
    N:=n div 100;
    M:=n;
    If (n mod 100<13) and ( n mod 100<>0) then
    Begin
        Ngay:=n div 100;
        Kt(thang);
    End;
    If (m div 10<32) and(m div 10>0) then
    Begin
        Thang:=m mod 10;
        Ngay:=m div 10;
        Kt(thang);
    End;
End;
Begin
    Nhap;
    Xuly;
End.

```