****BÀI 1****

**Trò chơi với băng số**(8 điểm) File bài làm **DIV.PAS**

Cho một băng số gồm n số nguyên dương, mỗi số được viết trên một ô. Hãy cắt băng số này thành nhiều đoạn nhất sao cho tổng các phần tử trong các đoạn là bằng nhau.

***Dữ liệu vào:* DIV.INP** + Dòng đầu ghi *n* (*n* ≤ 1000)

+ Dòng tiếp theo ghi n số nguyên dương *a1, a2, ..., an*

(các số nằm trên một dòng cách nhau bởi một dấu cách ai ≤ 1000)

***Dữ liệu ra:* DIV.OUT** Ghi K là số đoạn cần chia.

***Ví dụ:***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **10** | **2** | **6** | **2** | **5** | **2** | **1** | **2** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **10** | | |  |  |  | | --- | --- | --- | | **2** | **6** | **2** | | |  |  |  |  | | --- | --- | --- | --- | | **5** | **2** | **1** | **2** | |

|  |  |  |
| --- | --- | --- |
| **DIV.INP** | **DIV.OUT** | **Giải thích** |
| 8  10 2 ; 6 2 5 2 1 2 | 3 | Đoạn 1: 10  Đoạn 2: 2 + 6 + 2 =10  Đoạn 3: 5 + 2 + 1 + 2 = 10 |

**Cách 1**

const fi='DIV.INP';

fo='DIV.OUT';

var s,a:array[0..10000000] of longint;

n,i,j,tong,tam,dem,max:longint;

f:text;

BEGIN

assign(f,fi);

reset(f);

readln(f,n);

for i:=1 to n do

begin

read(f,a[i]);

s[i]:=s[i-1]+a[i];

end;

assign(f,fo);

rewrite(f);

max:=1;

for i:=1 to n-1 do

if s[i]=0 then

begin

dem:=1;

tam:=i;

j:=tam;

repeat

j:=j+1;

if s[j]=0 then

begin

tam:=j;

inc(dem);

end;

if (s[j]=0) and (j=n) then

if max<dem then

begin

max:=dem;

dem:=0;

end;

until j=n;

end

else

if (s[n] mod s[i]=0) then

begin

dem:=1;

tam:=i;

j:=tam;

repeat

j:=j+1;

tong:=s[j]-s[tam];

if tong=s[i] then

begin

tam:=j;

inc(dem);

end;

if (tong=s[i]) and (j=n) then

if max<dem then

begin

max:=dem;

dem:=0;

end;

until j=n;

end;

write(f,max);

close(f);

END.

---\\

**Cách 2:**

type mang=array[1..1000]of longint;

const fi='DIV.INP';

fo='DIV.OUT';

var f:text;

a,b:mang;

n,i,t,l,e,j,k,tt,sld,s:longint;

kt:boolean;

function ktbang(b:mang;m:longint):boolean;

begin

for i:=1 to m-1 do

if b[i]<>b[i+1] then exit(false);

exit(true);

end;

function tongday(b:mang;m:longint):longint;

var s:longint;

begin

s:=0;

for i:=1 to m do

s:=s+b[i];

tongday:=s;

end;

procedure uoc;

begin

l:=0;

for i:=2 to t do

begin

if t mod i=0 then

begin

l:=l+1;

b[l]:=i;

end;

end;

end;

begin

assign(f,fi);

reset(f);

readln(f,n);

for i:=1 to n do

read(f,a[i]);

assign(f,fo);

rewrite(f);

if ktbang(a,n)=true then

writeln(f,n)

else

Begin

t:=tongday(a,n);

uoc;

k:=0;

for e:=1 to l do

begin

kt:=true;

sld:=0;

i:=1;

while i<=n do

begin

j:=i;

s:=0;

while (j<=n) and (s<>b[e]) do

begin

s:=s+a[j];

j:=j+1;

end;

i:=j;

if s<>b[e] then

begin

kt:=false;

break;

end;

sld:=sld+1;

end;

if kt=true then

if sld>k then

begin

k:=sld;

tt:=b[e];

end;

end;

writeln(f,k);

End;

close(f);

end.

**BÀI 2: ĐOẠN ZERO**

Cho dãy số nguyên a = (a1, a2, ..., an), 1 ≤ n ≤ 10000; ∀i: -10000 ≤ ai ≤ 10000

**Yêu cầu:** *Hãy tìm một đoạn dài nhất gồm m phần tử liên tiếp trong dãy a: ak, ak+1, ..., ak+m-1 có tổng bằng 0*

Dữ liệu vào: Đọc từ file text ZERO.INP

- Dòng 1: Chứa số n

- Dòng 2: Chứa n số a1, a2, ..., an cách nhau ít nhất một dấu cách

Kết quả: Ghi ra file text ZERO.OUT

Chỉ gồm một dòng ghi ba số: m, k và k+m-1 cách nhau ít nhất một dấu cách.

**Ví dụ:**

|  |  |
| --- | --- |
| ZERO.INP | ZERO.OUT |
| 9  2 7 5 -3 -2 4 -9 -2 -1 | 7 2 8 |

*Dữ liệu vào luôn được cho hợp lý để tồn tại một đoạn các phần tử liên tiếp trong dãy a có tổng bằng 0.*

*Cach: Lvu;*

*const fi='zero.inp';*

*fo='zero.out';*

*var f:text;*

*a:array[1..100000] of int64;*

*n,i,j,k,gan,max,d,t:longint;*

*begin*

*assign(f,fi);reset(f);*

*read(f,n);*

*for i:=1 to n do*

*read(f,a[i]);*

*close(f);*

*assign(f,fo);rewrite(f);*

*t:=0;max:=0; d:=0;*

*for i:=1 to n do*

*for j:=1 to n do*

*if i<j then*

*begin*

*for k:=i to j do*

*t:=t+a[k];*

*if (t=0) and (max<j-i+1) then*

*begin*

*gan:=j;*

*d:=j-i+1;*

*max:=j-i+1;*

*end;*

*t:=0;*

*end;*

*write(f,d,' ',gan-d+1,' ',gan);*

*close(f);*

*end.*

*Cach HOANGVU:*

*uses crt;*

*const fi='nhap.inp';*

*fo='xuat.out';*

*var a:array[0..10000] of longint;*

*f1,f2:text; n,i,j,s,d,k,t,c,max:longint;*

*begin*

*clrscr;*

*assign(f1,fi);*

*reset(f1);*

*read(f1,n);*

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

*close(f1);*

*assign(f2,fo);*

*rewrite(f2);*

*max:=0;*

*k:=n;*

*for i:=1 to n do*

*begin*

*while k>i do*

*begin*

*s:=0;*

*d:=0;*

*for j:=i to k do*

*begin*

*s:=s+a[j];*

*d:=d+1;*

*end;*

*if s=0 then*

*begin*

*max:=d;*

*t:=i;*

*c:=0;*

*end;*

*k:=k-1;*

*if c=0 then*

*begin*

*for j:=t to t+max-1 do write(f2,a[j]:4);*

*writeln(f2);*

*c:=c+1;*

*end;*

*end;*

*k:=n;*

*end;*

*close(f2);*

*end.*

***const fi='Zero.inp';***

***fo='Zero.out';***

***var f,fn:text;***

***a:array[0..10000000] of integer;***

***n,i,j,d,m,t,k:longint;***

***begin***

***assign(f,fi); reset(f);***

***assign(fn,fo);rewrite(fn);***

***readln(f,n);***

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

***read(f,a[i]);***

***m:=2;***

***for i:=1 to n-1 do***

***begin***

***d:=1;***

***t:=a[i];***

***for j:=i+1 to n do***

***begin***

***t:=t+a[j];***

***inc(d);***

***if t=0 then***

***begin***

***if d>m then***

***begin***

***m:=d;***

***k:=i;***

***end;***

***end;***

***end;***

***end;***

***write(fn,m,' ',k,' ',k+m-1);***

***close(f);***

***close(fn);***

***end.***

***Cach 1: (thuật toán quicksort)***

***const fi='zero.inp';***

***fo='zorro.out';***

***var f,f1:text;***

***a:array[0..100000] of longint;***

***s,d,dem,vt,i,j,n:longint;***

***begin***

***assign(f,fi);reset(f);***

***readln(f,n);***

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

***read(f,a[i]);***

***close(f);***

***dem:=0;***

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

***begin***

***d:=1;***

***s:=a[i];***

***for j:=i+1 to n do***

***begin***

***s:=s+a[j];***

***inc(d);***

***if (s=0) and (d>dem) then***

***begin***

***vt:=i;***

***dem:=d;***

***end;***

***end;***

***end;***

***assign(f1,fo);rewrite(f1);***

***write(f1,dem:4);***

***write(f1,vt:4);***

***write(f1,dem+vt-1:4);***

***close(f1);***

***end.***

***Cách 2***

const fi='ZERO.INP';

fo='ZERO.OUT';

var a:array [1..100000000] of longint;

i,max,m,k,l,j,s,n:longint;

f:text;

begin

assign(f,fi);

reset(f);

readln(f,n);

for i:=1 to n do

read(f,a[i]);

close(f);

assign(f,fo);

rewrite(f);

for i:=1 to n-1 do

begin

s:=a[i];

for j:=i+1 to n do

begin

s:=s+a[j];

m:=j-i+1;

if (s=0) and (m>max) then

begin

max:=m;

k:=i;

l:=j;

end;

end;

end;

write(f,max,' ',k,' ',l);

close(f);

end.

{C\_L}

const fi='ZERO7.INP';

fo='ZERO7.OUT';

var f:text;

max:longint;

a,tong:array[0..10001] of longint;

l,r,n:longint;

procedure nhap;

var i:longint;

begin

assign(f,fi);reset(F);

readln(f,n);

for i:=1 to n do read(f,a[i]);

closE(F);

end;

procedure tinhtong;

var i:longint;

begin

tong[0]:=0;

tong[1]:=a[1];

for i:=2 to n do tong[i]:=tong[i-1]+a[i];

end;

procedure timdaycon;

var i,j:longint;

begin

for i:=1 to n-1 do

for j:=i+1 to n do

begin

if (tong[j]-tong[i-1]=0) and (j-i+1>max) then

begin

max:=j-i+1;

l:=i;

r:=j;

end;

end;

end;

procedure xuat;

begin

assign(f,fo);rewrite(F);

writeln(f,max,' ',l,' ',r);

closE(F);

end;

BEGIN

nhap;

tinhtong;

timdaycon;

xuat;

END.

**Bài : ĐỔI TIỀN**

Một ngân hàng có N loại tiền mệnh giá A[1], A[2], ..., A[N] với số lượng tiền mỗi loại không giới hạn. Cần chi trả cho khách hàng một số tiền M đồng. Hãy cho biết cần bao nhiêu tờ cho mỗi loại tiền để tổng số tờ là ít nhất.

Dữ liệu vào từ file: TIEN.INP gồm 2 dòng

* Dòng đầu tiên ghi 2 số N, M (0 < N ≤ 10; 0 < M ≤ 1000)
* Dòng thứ hai ghi N số: A[1], A[2], ..., A[N] các số cách nhau bởi dấu cách.

Kết quả ghi ra file: TIEN.OUT gồm 2 dòng

* Dòng đầu tiên ghi số tờ cần dùng, nếu không thể đổi được thì ghi số 0 và không cần thực hiện tiếp.
* Dòng tiếp theo ghi n số ( mỗi số biểu hiện cho số tờ cần dùng cho mỗi loại).

**Ví dụ**:

|  |  |
| --- | --- |
| TIEN.INP | TIEN.OUT |
| 5 98  1 5 10 20 50 | 7  3 1 0 2 1 |

**const fi='tien.inp';**

**fo='tien.out';**

**var f,fn:text;**

**a,b:array[0..1000000] of longint;**

**n,m,d,d1,i:longint;**

**procedure sort(l,r: longint);**

**var**

**i,j,x,y: longint;**

**begin**

**i:=l;**

**j:=r;**

**x:=a[(l+r) div 2];**

**repeat**

**while a[i]<x do**

**inc(i);**

**while x<a[j] do**

**dec(j);**

**if not(i>j) then**

**begin**

**y:=a[i];**

**a[i]:=a[j];**

**a[j]:=y;**

**inc(i);**

**j:=j-1;**

**end;**

**until i>j;**

**if l<j then**

**sort(l,j);**

**if i<r then**

**sort(i,r);**

**end;**

**begin**

**assign(f,fi); reset(f);**

**assign(fn,fo);rewrite(fn);**

**read(f,n);**

**readln(f,m);**

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

**read(f,a[i]);**

**sort(1,n);**

**if m mod a[1]<>0 then write(fn,'0')**

**else**

**begin**

**d:=0;**

**for i:=n downto 1 do**

**begin**

**d1:=0;**

**while (m div a[i]>=1) and (m>0) do**

**begin**

**inc(d);**

**inc(d1);**

**m:=m-a[i];**

**end;**

**b[i]:=d1;**

**end;**

**writeln(fn,d);**

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

**write(fn,b[i],' ');**

**end;**

**close(f);**

**close(fn);**

**end.**

**-----------**

**uses crt;**

**var a:array[0..1000000] of longint;**

**m, c, i,n,k,j,the:longint;**

**begin**

**clrscr;**

**write(' nhap n menh gia:');read(n);**

**write('nhap so tien:');read(k);**

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

**begin**

**write('a[',i,']=');**

**read(a[i]);**

**end;**

**for i:=1 to n-1 do**

**for j:=i+1 to n do**

**if a[i]<a[j] then**

**begin**

**the:=a[j];**

**a[j]:=a[i];**

**a[i]:=the;**

**end;**

**m:=0;**

**c:=1;**

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

**if k mod a[i]= 0 then**

**begin**

**c:=0;**

**break;**

**end;**

**if c=1 then write('o')**

**else**

**while k>0 do**

**begin**

**inc(m);**

**writeln('so to tien menh gia: ',a[m],' la :',k div a[m]);**

**k:=k-a[m]\*(k div a[m]);**

**end;**

**readln;**

**readln;**

**end.**

**CÁCH 1:**

const max=5000;

fi='TIEN.INP';

var loaitien,soluo

ng,kq:array [1..max] of integer;

n,m,nghiem:longint; i:integer;

f:text;

procedure ghinghiem;

var i,dem:integer;

begin

inc(nghiem);

write('Cach ',nghiem,':',n,'=');

dem:=0;

for i:=1 to m do

if kq[i]>0 then

begin

inc(dem);

if dem=1 then

write (loaitien[i],'\*',kq[i])

else

write ('+',loaitien[i],'\*',kq[i]);

end;

writeln;

end;

procedure doitien(tien,i:longint);

var j:longint;

begin

for j:=0 to n div loaitien[i]-kq[i] do

begin

inc(kq[i],j);

tien:=tien-loaitien[i]\*j;

if tien=0 then

ghinghiem

else

if (tien>0) and (i<m) then

doitien(tien,i+1);

dec(kq[i],j);

tien:=tien+loaitien[i]\*j;

end;

end;

BEGIN

assign(f,fi);

reset(f);

readln(f,m,n);

for i:=1 to m do

read (f,loaitien[i]);

doitien(n,1);

writeln(nghiem,' cach');

close(f);

readln

END.

**CÁCH 2(HÀ):**

**const fi='DOITIEN1.INP';**

**fo='DOITIEN1.OUT';**

**var a,luu:array[1..1000] of longint;**

**f,g:text;**

**soto,tongtien,i:longint;**

**begin**

**assign(f,fi);**

**reset(f);**

**assign(g,fo);**

**rewrite(g);**

**read(f,soto,tongtien);**

**for i:=1 to soto do**

**read(f,a[i]);**

**luu[soto]:=tongtien div a[soto];**

**tongtien:=tongtien-luu[soto]\*a[soto];**

**for i:=soto-1 downto 1 do**

**begin**

**luu[i]:=tongtien div a[i];**

**tongtien:=tongtien-a[i]\*luu[i];**

**end;**

**for i:=1 to soto do**

**write(g,luu[i]:3);**

**close(g);**

**close(f);**

**end.**

****BÀI 4: Hàng cây****

Trong khu vườn, người ta trồng một hàng cây chạy dài gồm có N cây, mỗi cây có độ cao là a1, a2,…aN.

Người ta cần lấy M mét gỗ bằng cách đặt cưa máy sao cho lưỡi cưa ở độ cao H (mét) để cưa tất cả các cây có độ cao lớn hơn H (dĩ nhiên những cây có độ cao không lớn hơn H thì không bị cưa).

Ví dụ: Nếu hàng cây có các cây với độ cao tương ứng là 20; 15; 10 và 18 mét, cần lấy 7 mét gỗ. Lưỡi cưa đặt tại độ cao hợp lí là 15 mét thì độ cao của các cây còn lại sau khi bị cưa tương ứng là 15; 15; 10 và 15 mét. Tổng số mét gỗ lấy được là 8 mét (dư 1 mét).

***Yêu cầu:*** Hãy tìm vị trí đặt lưỡi cưa hợp lí (số nguyên H lớn nhất) sao cho lấy được M mét gỗ và số mét gỗ dư ra là ít nhất.

***Dữ liệu***:

+ Dòng thứ nhất chứa 2 số nguyên dương N và M (1≤N≤106; 1≤M≤2x109) cách nhau một dấu cách.

+ Dòng thứ hai chứa N số nguyên dương ai là độ cao của mỗi cây trong hàng (1≤ai≤109; i=1…N), mỗi số cách nhau ít nhất một dấu cách.

***Kết quả***: Đưa ra màn hình một số nguyên cho biết giá trị cần tìm.

***Ví dụ:***

|  |  |
| --- | --- |
| **WOOD.INP** | **WOOD.OUT** |
| 4 7  20 15 10 18 | 15 |

Bài 5: Bài toán ô vuông

File bài làm: **SQUARE.PAS**

Cho một bảng chữ nhật gồm m \* n điểm (gồm m hàng ngang trên mỗi hàng có n điểm). Các điểm liền kề trên cùng một hàng hoặc một cột có thể được nối với nhau bởi một đoạn thẳng có kích thước bằng 1. Trên mỗi hàng có nhiều nhất n - 1 đoạn thẳng nằm ngang nối các điểm liền nhau, trên mỗi cột có nhiều nhất là m - 1 đoạn thẳng thẳng đứng nối các điểm liền nhau. Xem hình dưới:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Ngang: Dọc:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Với bảng trên ta có 3 hàng ngang mỗi hàng 5 điểm. Các đoạn thẳng nối chúng tạo nên 2 ô vuông. Để mô tả bảng người ta dùng hai mảng nhị phân: mảng **Ngang** diễn tả các đoạn thẳng nằm ngang, mảng **Dọc** biểu diễn các đoạn thẳng thẳng đứng. Trong các mảng, số 1 diễn tả có đoạn thẳng nối hai điểm liên tiếp, số 0 diễn tả không có đoạn thẳng nối hai điểm liên tiếp.

***Yêu cầu:*** Lập trình đếm số lượng các ô vuông mà cạnh có độ dài bằng 1 tạo bởi các đoạn nối có trên bảng đã cho.

***Dữ liệu vào:*** **SQUARE.INP**

+ Dòng đầu tiên gồm 2 số nguyên dương lần lượt là m, n (m, n ≤ 1000). Hai số cách nhau một khoảng trắng.

+ m dòng tiếp theo diễn tả các đoạn thẳng nằm ngang (như bảng ngang mô tả ở phần trên). Hai số liền nhau cách nhau một khoảng trắng.

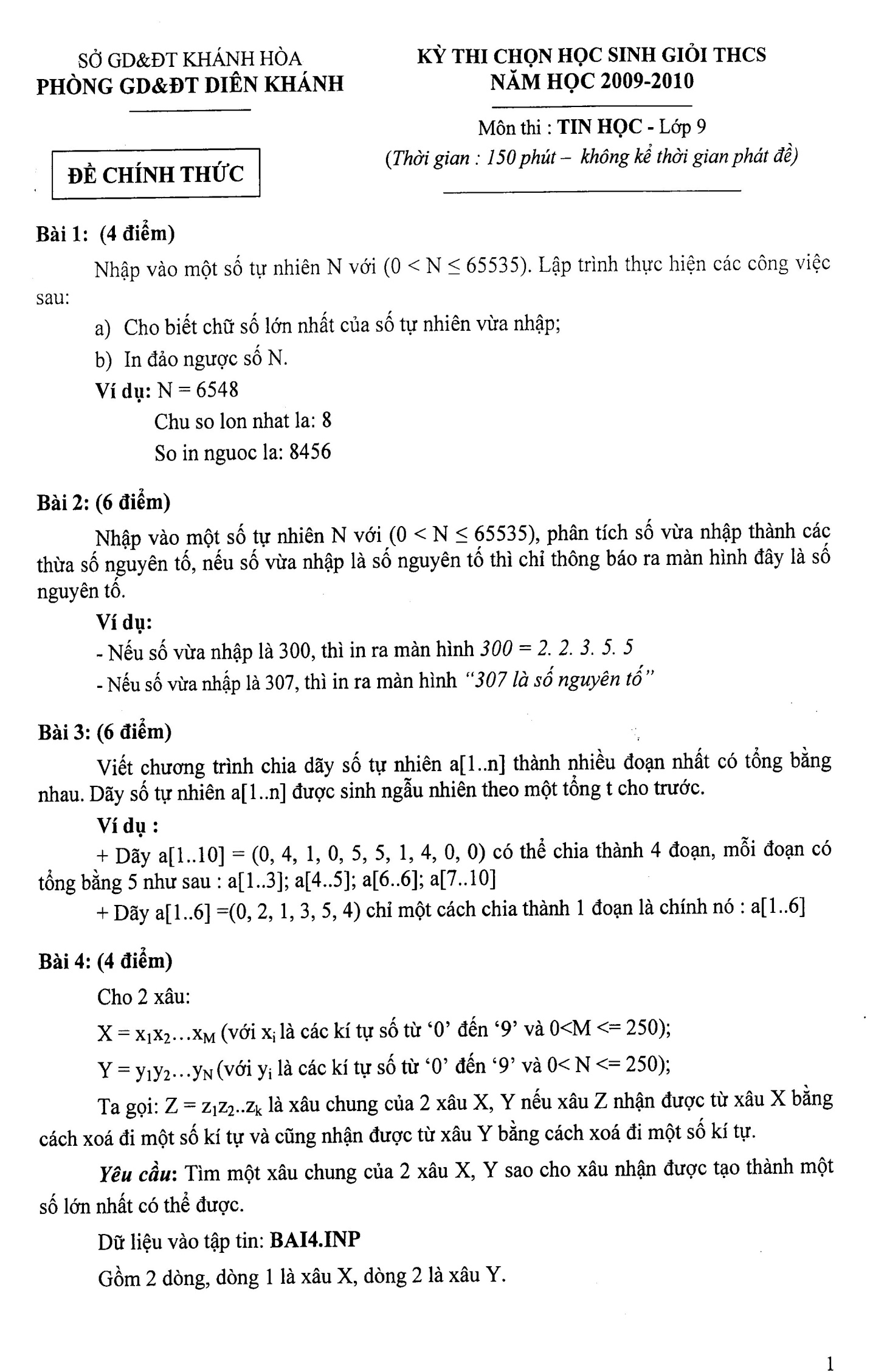
+ m - 1 dòng tiếp theo diễn tả các đoạn thẳng thẳng đứng (như bảng dọc mô tả ở phần trên). Hai số liền nhau cách nhau một khoảng trắng.

***Dữ liệu ra :*** **SQUARE.OUT**

+ Số lượng ô vuông có độ dài cạnh bằng 1.

***Ví dụ:***

|  |  |
| --- | --- |
| **SQUARE.INP** | **SQUARE.OUT** |
| ***3 5***  **1 0 1 0**  **1 0 0 0**  **1 1 1 1**  **1 1 1 0 0**  **1 1 0 0 0** | **2** |



BÀI 6: Tiêu tiền

Đón Tết nguyên đán năm Giáp Ngọ, Hoàng nhận được rất nhiều tiền mừng tuổi, Hoàng đã phân loại tiền theo từng mệnh giá. Mỗi lần đi mua Hoàng cầm mỗi tay 1 xấp tiền có cùng mệnh giá. Vì số tiền quá nhiều nên Hoàng muốn tiêu số tiền mừng tuổi càng nhiều tờ có thể.

Chẳng hạn, nếu Hoàng mang 2 xấp tiền có mệnh giá là 2000 đồng và 5000 đồng, khi phải thanh toán lượng tiền là 17000 đồng, Hoàng có 2 cách trả: cách thứ nhất dùng 6 tờ 2000 đồng và 1 tờ 5000 đồng, cách thứ hai là dùng 1 tờ 2000 đồng và 3 tờ 5000 đồng. Cách thứ nhất sẽ được Hoàng lựa chọn vì giúp Hoàng tiêu được nhiều tờ tiền hơn.

**Yêu cầu:** Cho biết mệnh giá của 2 xấp tiền là A và B, số tiền cần thanh toán là C, hãy xác định số tiền nhiều nhất có thể thanh toán chính xác số tiền C hoặc là không thanh toán được.

**Dữ liệu:** Đọc từ file văn bản COINS.INP chỉ 1 dòng chứa 3 số nguyên A, B, C (1 <= A, B <= 500000; 1 <= C <= 100)

**Kết quả:** Ghi ra file văn bản COINS.OUT chỉ 1 dòng chứa 1 số nguyên là số lượng tờ tiền nhiều nhất có thể sử dụng để thanh toán hoặc ghi -1 nếu không thể thanh toán được.

**Ví dụ:**

|  |  |
| --- | --- |
| **COINS.INP** | **NUM.OUT** |
| 2000 5000 17000 | 7 |

**Bài 3\*:** Cho dãy số tự nhiên a(1), a(2), a(3),...,a(n). Tìm cách chia dãy số trên thành nhiều đoạn nhất sao cho mỗi đoạn có tổng bằng nhau.  
Dữ liệu vào: cho vô File INP.BT3 gồm:  
-Dòng đầu tiên ghi số n (là số phần tử của dãy)  
-Các dòng còn lại ghi các số a(1), a(2), a(3),...,a(n). Các số trên cùng dòng cách nhau một khoảng.  
  
Kết quả: ghi ra File OUT.BT3 gồm k+1 dòng (k là số đoạn được chia nhiều nhất)  
-Dòng đầu tiên chứa hai số k và s (s là tổng số giá trị mỗi đoạn).  
-k dòng còn lại, mỗi dòng ghi theo thứ tự các số của mỗi đoạn chia được. Các số trên cùng dòng cách nhau một khoảng.  
VD:  
INP.BT3 gồm:  
20  
3 5 1 7 8 0 1 2 3 2 10 1 1 1 3 13 0 2 0 1  
OUT.BT3 gồm:  
4 16  
3 5 1 7   
8 0 1 2 3 2   
10 1 1 1 3

var

A, T, Result: array [1..100] of integer;

N, k, d, i, j, x, y: integer;

found\_result: boolean;

f: text;

begin

assign(f, 'INP.BT3');

reset(f);

readln(f, N);

for i := 1 to N do read(f, A[i]);

close(f);

T[1] := A[1];

for i := 2 to N do T[i] := T[i - 1] + A[i];

assign(f, 'OUT.BT3');

rewrite(f);

Result[1] := 1;

found\_result := false;

for k := 1 to N do

begin

Result[2] := k + 1;

d := T[k] + T[k];

j := 3;

for i := k + 1 to N do

begin

if T[i] = d then

begin

Result[j] := i + 1;

d := d + T[k];

if d = T[N] then

begin

found\_result := true;

for x := 1 to j - 1 do

begin

for y := Result[x] to Result[x + 1] - 1 do write(f, A[y], ' ');

writeln(f);

end;

for x := Result[j] to N do write(f, A[x], ' ');

break;

end;

j := j + 1;

end;

end;

if found\_result then break;

end;

if not found\_result then write(f, 'Khong The Chia');

close(f);

end.

Bài 7: Bội có điều kiện

Tìm số S nhỏ nhất đối với 1 số tự nhiên N có các đặc tính sau:

• N <= S

• Giá trị của S viết trong hệ thập phân chỉ chứa các chữ số 7, 2, 0 và không được bắt đầu bằng số 0.

• Dạng thập phân của S chứa nhiều nhất 20 chữ số

• S chia hết cho N

**Dữ liệu:** Đọc từ file văn bản STZ.INP chỉ 1 dòng chứa số nguyên N (0 <= N <= 50000)  
**Kết quả:** Ghi ra file văn bản STZ.OUT chỉ 1 dòng chứa số S, nếu không tồn tại đưa ra “NO”

**Ví dụ:**

|  |  |
| --- | --- |
| **STZ.INP** | **STZ.OUT** |
| 3 | 27 |

|  |  |
| --- | --- |
| **STZ.INP** | **STZ.OUT** |
| 61 | 70272 |

Bài 8: Tách số

Viết chương trình tách số nguyên dương A (A<200) thành tổng của hai số nguyên dương x và y sao cho ước chung lớn nhất d của x và y là lớn nhất trong các cách phân tích có được.

*Yêu cầu:*

● Dữ liệu vào: File text **Tachso.INP** chỉ có 1 dòng chứa số nguyên dương A.

● Dữ liệu ra: File text **Tachso.OUT** gồm 2 dòng:

- Dòng 1: Ghi số nguyên dương A.

- Dòng 2: Ghi ba số nguyên dương x, y, d; Các số cách nhau ít nhất một khoảng trắng.

*Ví dụ:*

|  |  |
| --- | --- |
| **Tachso.INP** | **Tachso.OUT** |
| 95 | 95  19 76 19 |

**var a:array[0..100000] of longint;**

**n,k,i,j,l,m,min,max:longint;**

**begin**

**write(' nhap so n:');read(n);**

**k:=0;**

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

**if n mod i=0 then**

**begin**

**inc(k);**

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

**end;**

**l:=0;**

**for i:=1 to k do**

**begin**

**for j:=1 to (n div a[i]) do**

**for m:=1 to (n div a[i]) do**

**if (j\*a[i]+m\*a[i])=n then begin**

**max:=j;**

**min:=m;**

**l:=a[i];**

**end;**

**end;**

**write(min\*l,max\*l:4,l:4);**

**readln;**

**readln;**

**end.**

**---**

**const fi='Tachso.inp';**

**fo='Tachso.out';**

**var f,fn:text;**

**x,y,i,j,d,a:longint;**

**function ucln(m,n:longint):longint;**

**begin**

**while m<>n do**

**begin**

**if m>n then m:=m-n**

**else n:=n-m;**

**end;**

**ucln:=m;**

**end;**

**begin**

**assign(f,fi);reset(f);**

**assign(fn,fo);rewrite(fn);**

**read(f,a);**

**d:=0;**

**for i:=1 to a do**

**for j:=a downto 1 do**

**if i+j=a then**

**begin**

**if ucln(i,j)>d then**

**begin**

**d:=ucln(i,j);**

**x:=i;**

**y:=j;**

**end;**

**end;**

**write(fn,x,' ',y,' ',d);**

**close(f);**

**close(fn);**

**end.**

**Bài 9 . Khai triển Cantor *Tên file chương trình:* Cantor.PAS**

Khai triển Cantor của số tự nhiên N là biểu diễn số N dưới dạng:

N= ak.k! + ak-1.(k-1)! + … + a2.2! + a1.1!

Trong đó: 0 ≤ ak ≤ k và k!= 1.2.3…(k-1).k

Viết chương trình liệt kê các hệ số ak, ak-1,…, a3, a2, a1 trong khai triển Cantor của số tự nhiên N (N<50000) . Các hệ số cách nhau ít nhất một khoảng trắng.

*Yêu cầu:*

● Dữ liệu vào: File text **Cantor.INP** chỉ có 1 dòng chứa số nguyên dương N.

● Dữ liệu ra: File text **Cantor.OUT** gồm 2 dòng:

- Dòng 1: Ghi số nguyên dương N.

- Dòng 2: Liệt kê các hệ số ak, ak-1,…, a3, a2, a1 trong khai triển Cantor của số tự nhiên N. Các hệ số cách nhau ít nhất một khoảng trắng.

*Ví dụ:*

|  |  |
| --- | --- |
| **Cantor.INP** | **Cantor.OUT** |
| 2014 | 2014  2 4 3 3 2 0 |

***Bài 10 : Bi đổi màu***

Trên bàn có ba loại bi màu đỏ, xanh, vàng với số lượng lần lượt là: x, y, z (x, y, z < 400). Mỗi khi cho hai viên bi khác màu chạm vào nhau thì chúng sẽ đổi sang màu còn lại.

Viết chương trình trình bày quy trình chạm bi sao cho sau khi thực hiện thì trên bàn chỉ còn một màu bi nào đó. Nếu không xử lý được thì xuất ra thông báo “Khong thuc hien duoc”.

*Yêu cầu:*

● Dữ liệu vào: File text **Bidoimau.INP** chỉ có 1 dòng chứa 3 số nguyên dương x, y, z lần lượt là số bi đỏ, xanh, vàng có trên bàn. Các số cách nhau ít nhất một khoảng trắng

● Dữ liệu ra: File text **Bidoimau.OUT** gồm một trông hai trường hợp:

Trường hợp 1 (Nếu bài toán giải được): Gồm nhiều dòng, mỗi dòng gồm 3 số tự nhiên là số lượng của từng loại bi sau một bước va chạm .

Trường hợp 2 (Nếu bài toán không giải được): Đưa ra câu không báo “Khong thực hien dươc”

*Ví dụ 1:*

|  |  |
| --- | --- |
| **Bidoimau.INP** | **Bidoimau.OUT** |
| 2 3 4 | Khong thuc hien duoc |

*Ví du 2:*

|  |  |
| --- | --- |
| **Bidoimau.INP** | **Bidoimau.OUT** |
| 3 4 1 | 3 4 1  5 3 0  4 2 2  6 1 1  8 0 0 |

***Bài 11: SỐ ĐƠN ĐIỆU***

Các số nguyên dương: 3748, 58, 859, 32435465768 được gọi là các số đơn điệu do nếu quan sát các chữ số của các số này, ta thấy chúng luân phiên tăng giảm hoặc giảm tăng. Chẳng hạn:

3 < 7 > 4 < 8 và 3 > 2 < 4 > 3 < 5 > 4 < 6 > 5 < 7 > 6 < 8

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

Nhiệm vụ:Viết chương trình xác định số chữ số đầu tiên lớn nhất tạo thành số đơn điệu của một số cho trước.

Tên tập tin chương trình:

WIGGLE.PAS

Dữ liệu:

Cho trong tập tin văn bản WIGGLE.IN, gồm một dòng duy nhất chứa một số nguyên dương duy nhất có không quá 75 chữ số.

Kết quả:

Cho trong tập tin văn bản WIGGLE.OUT, chứa một số nguyên duy nhất chỉ số chữ số đầu tiên lớn nhất tạo thành số đơn điệu của số tương ứng trong tập tin dữ liệu.

Ví dụ:

|  |  |
| --- | --- |
| **WIGGLE.IN** | **WIGGLE.OUT** |
| 3748 | 4 |

***Bài 12: Tàu cập cảng.***

Một tư nhân có hai tàu thủy chuyên vận chuyển hàng khô rất uy tín đối với khách hàng. Được uy tín như vậy là do lịch trình cập cảng của các tàu đảm bảo số ngày quy định, an toàn và cập cảng ngày nào thì cũng rời cảng trong chính ngày đó. Tại cảng chính, tàu thứ nhất cứ k ngày cập cảng 1 lần; tàu thứ hai cứ s ngày cập cảng 1 lần (k≠s).

***Yêu cầu:*** Lập trình tính xem nếu hai tàu cùng rời cảng thì thời gian ngắn nhất gặp lại nhau tại cảng đó là bao nhiêu ngày?

**- *Dữ liệu vào:*** Đọc từ file văn bản **TAU.INP** chứa một dòng gồm hai số nguyên dương k và s, hai số cách nhau ít nhất một ký tự trống.

**- *Dữ liệu ra****:* Kết quả ghi ra file văn bản **TAU.OUT** thời gian ngắn nhất.

|  |  |
| --- | --- |
| **TAU.INP** | **TAU.OUT** |
| 6 4 | 12 |

***Bài 13: Giải mã thông điệp.***

Trong thời đại ngày nay, thông tin cá nhân của mỗi người dễ dàng bị xâm nhập bởi rất nhiều công nghệ hiện đại dùng nghe lén. Chính vì vậy, một đôi bạn muốn bảo mật thông tin cho nhau đã có sáng kiến nghĩ ra cách trao nhau những thông điệp mà người khác đọc hoặc nghe được thì không hiểu gì. Thông điệp của họ là một câu với nhiều từ được phân biệt bởi ký tự trống (dấu cách) và được sắp xếp lộn xộn để trở thành câu vô nghĩa. Để hiểu nội dung một thông điệp của người bạn gửi cho mình, hai người bạn đã quy ước trước một bộ mã là một dãy số nguyên dương có số lượng số tương ứng với số từ của thông điệp nhằm giải mã thành câu có ý nghĩa.

***Yêu cầu:*** Lập trình để giúp đôi bạn giải mã một thông điệp cho trước.

**- *Dữ liệu vào****:* File văn bản **GIAIMA.INP** gồm 2 dòng:

+ Dòng đầu ghi một câu có không quá 255 kí tự là thông điệp được gởi đi. Trong câu này có không quá 50 từ, mỗi từ có không quá 30 kí tự.

+ Dòng thứ hai ghi bộ mã gồm một dãy các số nguyên dương đôi một khác nhau dùng để giải mã thông điệp trên.

**- *Dữ liệu ra****:* File văn bản **GIAIMA.OUT** gồm một dòng ghi câu đã được giải mã từ thông điệp đã cho.

Ví dụ:

|  |  |
| --- | --- |
| **GIAIMA.INP** | **GIAIMA.OUT** |
| luon chuc hoc ban gioi luon | chuc ban luon luon hoc gioi |

const fi='giaima.inp';

fo='giaima.out';

var f:text;

st:string;

a:array[0..50] of integer;

n,dem:integer;

procedure nhap;

var i:integer;

w:string;

begin

assign(f,fi);reset(F);

readln(f,st);

dem:=0;

while not(eof(f)) do

begin

inc(dem);

read(f,a[dem]);

end;

n:=0;

st:=' '+st;

st:=st+' dong';

for i:=2 to length(st) do

begin

if (st[i-1]=' ') and (st[i]<>' ') then

begin

inc(n);

str(n,w);

insert(w,st,i);

end;

end;

end;

procedure xuli;

var i,q,e:integer;

w:string;

begin

assign(f,fo);rewrite(F);

for i:=1 to dem do

begin

str(a[i],w);

q:=pos(w,st)+1;

e:=q;

while st[e]<>' ' do

begin

write(f,st[e]);

inc(e);

end;

write(f,' ');

end;

closE(F);

end;

BEGIN

nhap;

xuli;

END.

***Bài 14: Mật khẩu***

Để đảm bảo an ninh dữ liệu tại sân bay, nhà chức trách đã xác lập mật khẩu từ bảng số liệu có n hàng (n ≤ 10) h1, h2,...,hn tương ứng với hàng thứ 1, 2, ..., n tính từ trên xuống, mỗi hàng hi (i=1,..,n) có ít nhất từ 3 đến không quá 100 số tự nhiên được phân biệt bởi dấu cách. Mật khẩu được xác định bằng cách viết liên tiếp n số x1, x2, ...xn theo thứ tự, từ trái sang phải, với xi (i= 1, 2,...,n) là số cách chọn ba số tùy ý trên hàng hi, sao cho tích ba số đó chia hết cho tổng của chúng.

***Yêu cầu:*** Lập trình giúp nhà chức trách sân bay xác lập mật khẩu.

***- Dữ liệu vào:***  Đọc từ file văn bản **MATKHAU.INP** chứa n hàng như phần mô tả trong bảng số liệu đã nêu trên.

***- Dữ liệu ra:*** Kết quả ghi ra file văn bản **MATKHAU.OUT** là số mật khẩu tìm được.

***Ví dụ:***

|  |  |
| --- | --- |
| **MATKHAU.INP** | **MATKHAU.OUT** |
| **0 1 3 5 3 4 1 3**  **1 3 2 3 3 1 2**  **1 1 3**  **0 1 3 1** | 271303 |

const fi='matma0.inp';

fo='matma.out';

var f:text;

a:array[0..101] of longint;

dem:array[0..11] of longint;

k,tt:integer;

procedure xuli(H:integer);

var i,j,t:integer;

begin

for i:=1 to tt-2 do

for j:=i+1 to tt-1 do

for t:=j+1 to tt do

if (a[i]\*a[j]\*a[t]) mod (a[i]+a[j]+a[t])=0 then inc(dem[h]);

end;

procedure nhap;

begin

assign(f,fi);reset(f);

fillchar(dem,sizeof(dem),0);

k:=0;

while not eof(F) do

begin

tt:=0;

inc(k);

while not eoln(F) do

begin

inc(tt);

read(f,a[tt]);

end;

xuli(K);

readln(F);

end;

closE(F);

end;

procedure xuat;

var i:integer;

begin

assign(f,fo);rewrite(F);

for i:=1 to k do write(f,dem[i]);

closE(F);

end;

BEGIN

nhap;

xuat;

END.

**Bài 15** XÂU CON

Cho hai xâu X, Y chứa các kí tự số từ 0 đến 9 và được biểu diễn như sau:

X = x1, x2, x3…xn ; Y= y1, y2, y3…ym (n,m ). Hãy viết chương trình tạo ra xâu ST thoả mãn các điều kiện sau:

* Gồm các kí tự số vừa có mặt ở xâu X, vừa có mặt ở xâu Y;
* Các kí tự số trong xâu ST chỉ xuất hiện duy nhất một lần;
* Giá trị xâu ST nhận được là một số đạt giá trị lớn nhất.

Dữ liệu vào cho bởi file INPUT.INP chứa giá trị xâu X và xâu Y, mỗi xâu nằm trên một dòng.

Dữ liệu ra chứa ở file OUTPUT.OUT là số lớn nhất nhận được.

Ví dụ: Xâu X= ‘19012304’; xâu Y= ‘034012’, kết quả là 43210.

**Bài 16. Trò chơi kiểm tra trí nhớ**

Một trò chơi yêu cầu người chơi nhớ nhanh các con số điện tử chạy trên màn hình lớn và tính tổng các số hiện lên theo yêu cầu của cuộc chơi. Các số hiện ra trên màn hình là một dãy số gồm N số nguyên không âm a1, a2, …, aN và cứ lặp đi  lặp lại như thế (nghĩa là sau khi aixuất hiện vài giây đến lượt ai+1 xuất hiện, số xuất hiện sau aNlà a1).

***Yêu cầu:***Bạn hãy giúp người chơi kiểm tra kết quả tính tổng của M số nguyên liên tiếp xuất hiện trên màn hình bắt đầu từ số nguyên thứ T khi giám khảo yêu cầu.

***Dữ liệu vào*:**File text, tên file là BL1.INP, gồm 02 dòng:

+ Dòng đầu ghi ba số nguyên N, T, M (1 £ T £N £104; 1 £ M £ 2. 104 );

+ Dòng thứ hai chứa N số ai (ai< 2.109, i = 1,…,N), cách nhau bởi dấu cách.

***Dữ liệu ra:*** File text, tên File là BL1.OUT, gồm duy nhất một số là tổng tìm được.

*Ví dụ:*

|  |  |
| --- | --- |
| **BL1.INP** | **BL1.OUT** |
| 9 7 5  1 2 5 7 8 5 3 1 5 | 12 |

***Bài 16: Mua quà***

Tý là một học sinh tham dự cuộc thi Olympic Tin học quốc tế năm 2014. Do lần đầu ra nước ngoài nên bạn ấy muốn mua các món quà lưu niệm tặng các bạn ở trường. Sau những buổi thi căng thẳng, ban tổ chức dẫn các thí sinh dự thi đi tham quan danh lam thắng cảnh và những cửa hàng lưu niệm của địa phương.

Sau khi đi lòng vòng Tý bước vào một của hàng lưu niệm bán những món quà mà bạn ấy yêu thích. Chủ cửa hàng giới thiệu cho Tý  () món quà, món quà thứ  ( ) có giá  ().

Tý muốn chọn mua  () món quà sao cho giá trị chênh lệch giữa món quà có giá trị lớn nhất và nhỏ nhất là ít nhất có thể. Vì đi chơi nên Tý không mang theo máy tính nên bạn hãy giúp bạn ấy tính xem có thể mua các món quà với độ chênh lệch nhỏ nhất đó là bao nhiêu nhé.

***Dữ liệu vào***: từ file văn bản **BUYGIFTS.INP** có cấu trúc như sau:

* Dòng 1: ghi hai số nguyên  và ;
* Dòng 2: ghi ****** số nguyên lần lượt là a1, a2, ,an;
* Các số trong file cách nhau một dấu cách.

***Kết quả***: ghi ra file **BUYGIFTS.OUT** gồm duy nhât số  là độ lệnh nhỏ nhất các mòn quà mà Tý có thể mua.

***Ví dụ***:

|  |  |  |
| --- | --- | --- |
| **BUYGIFTS.INP** | **BUYGIFTS.OUT** | **Giải thích** |
| 4 6  10 12 10 5 7 22 | 5 | Tý có thể mua các món quà có giá trị: 10, 12, 10, 7. |

Bài 17: HÌNH CHỮ NHẬT

Trên giấy kẻ ô khổ N x N có vẽ một số hình chữ nhật. Mỗi hình chữ nhật được tạo ra từ các ô nguyên vẹn, các hình chữ nhật khác nhau không chồng lên nhau và không tiếp xúc nhau (Ví dụ : *Hình vẽ dưới đây có 4 hình chữ nhật*).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

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

Hãy viết chương trình tính và cho biết số các hình chữ nhật.

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

- Dòng đầu tiên ghi số nguyên dương N (N <= 250).

- N dòng tiếp theo mỗi dòng ghi N số 0 hoặc 1 là các phần tử của mảng

mỗi số viết cách nhau ít nhất một dấu cách .

**Dữ liệu ra:** Ghi ra File văn bản RECT.OUT gồm duy nhất một số là số hình chữ nhật tìm được .

**Ví dụ:**

|  |  |
| --- | --- |
| RECT.INP | RECT.OUT |
| 5  0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0 0  0 0 0 0 0 | 0 |
| 7  1 0 0 0 0 0 0  0 0 1 1 1 0 0  1 0 1 1 1 0 0  1 0 1 1 1 0 1  0 0 0 0 0 0 0  0 0 0 0 1 1 1  1 0 0 0 1 1 1 | 6 |

const fi='RECT.INP';

fo='RECT.OUT';

var f:text;

a:array[0..251,0..251] of integer;

dem,n:longint;

procedure nhap;

var i,j:longint;

begin

fillchar(a,sizeof(a),0);

assign(f,fi);reset(F);

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 tinh;

var i,j:longint;

begin

dem:=0;

for i:=1 to n do

for j:=1 to n do

if (a[i,j]=1) and (a[i,j-1]=0) and (a[i-1,j]=0) and (a[i-1,j-1]=0) then inc(dem);

end;.

procedure xuat;

begin

assign(f,fo);rewrite(F);

writeln(f,dem);

closE(F);

end;

BEGIN

nhap;

tinh;

xuat;

END

Bài 18: BĂNG NHẠC

Một máy phát nhạc tự động có một băng nhạc đủ lớn để ghi n bài hát, thời gian phát mỗi bài hát tính theo phút và được biết trước. Biết rằng, để phát bài hát thứ i thì máy phải trở về vị trí đầu băng và phải quay để bỏ qua i-1 bài hát trước đó. Máy có thể trở về vị trí đầu băng với thời gian không đáng kể và thời gian quay băng để bỏ qua mỗi bài và thời gian phát bài đó là bằng nhau.

Hãy tìm cách ghi các bài hát trên băng nhạc sao cho tổng số thời gian quay băng trong ngày là nhỏ nhất. Biết rằng, mỗi bài hát được phát một lần.

Chẳng hạn: Băng nhạc có 3 bài hát có thứ tự là 1, 2, 3 với thời gian phát mỗi bài lần lượt là 3, 2 và 6 phút. Nếu ta ghi băng nhạc theo các thứ tự khác nhau thì tổng thời gian phát cả băng nhạc là khác nhau, cụ thể là:

|  |  |  |
| --- | --- | --- |
| Thứ tự bài hát | Thời gian phát mỗi bài | Tổng thời gian phát |
| 1 2 3  2 3 1 | 3 2 6  2 6 3 | 3+(3+2)+(3+2+6) = 19  2+(2+6)+(2+6+3) = 21 |

**Dữ liệu vào:** Đọc từ tệp văn bản BANGNHAC.INP có cấu trúc như sau:

* Dòng thứ nhất chứa số n (n ≤ 100).
* Các dòng tiếp theo chứa các số nguyên t1, t2, ....., tn lần lượt là thời gian phát của các bài hát 1, 2, ......, n. Các số cách nhau ít nhất một dấu cách.

**Dữ liệu ra:** Ghi ra tệp văn bản BANGNHAC.OUT có cấu trúc như sau:

* Dòng thứ nhất ghi số Tmin là tổng thời gian tìm và phát cả băng nhạc.
* Các dòng tiếp theo chứa n là số thứ tự của các bài hát ghi trên băng nhạc, các số cách nhau ít nhất một dấu cách.

Ví dụ: BANGNHAC.INP BANGNHAC.OUT

3 18

3 2 6 2 1 3

BÀI 19: SỐ CHÍNH PHƯƠNG 6 CHỮ SỐ

Viết chương trình tìm tất cả các số chính phương có 6 chữ số, sao cho số tự tạo thành bởi 3 số sau lớn hơn số tạo thành bởi 3 số trước là 4 đơn vị.

Yêu cầu: In ra màn hình tất cả các số tìm được trên một dòng.

Bài 20: DÃY SỐ HẠNH PHÚC

Một dãy số các số nguyên dương được gọi là “dãy số hạnh phúc “ nếu nó thỏa mãn hai điều kiện sau:

1. Các số hạng của dãy là phân biệt và sắp xếp giảm dần.
2. Với mỗi số ai với (i>1), thì ai hoặc là số nguyên tố, hoặc là số nguyên tố cùng nhau với mọi aj với (j=1, .., i-1).

Yêu cầu: Nhập từ bàn phím số nguyên dương N(N<=60000), đưa ra màn hình số lượng phần tử của dãy số hạnh phúc dài nhất bắt đầu N.

|  |  |
| --- | --- |
| Input | Output |
| 10 | 7 |

Bài 21: Tìm hình chữ nhật nhật nhật lớn nhất

Trên một bảng MxN người ta đánh dấu một số hình chữ nhật gồm các ô liên tiếp kề nhau.

Các hình chữ nhật này đôi một không giao nhau và không kề nhau(có thể cùng chung đỉnh như hình H1). Để đánh dấu các hình chữ nhật, ta dùng một bảng A[MxN], gồm M hàng và N cột

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

(Các hang được đánh dấu từ 1 đến M từ trên xuống; các cột được dánh dấu từ 1 đến N từ trái qua) và quy ước giá trị của bảng A như sau:

A[I,j]=1 nếu ô ở hàng i, cột j thuộc một hình chữ nhật nào đó.

A[I,j]=0 nếu ô ở hang i, cột j không thuộc một hình chữ nhật nào cả.

Yêu cầu:Đọc từ file text một bảng A[MxN]. In ra màn hình diện tích lớn nhất lớn nhất có trong một bảng(mỗi ô của bảng được cho là 1 đơn vị diện tích).

H11

Dữ liệu vào: File text, tên file là BL3.INP. Dòng đầu ghi hai số nguyên M, N(M<=100;N<=100); M dòng tiếp theo, mỗi dòng ghi N số 0 hoặc 1, mỗi số cách nhau khoảng trắng.

Dữ liệu ra: Đưa ra màn hình một số nguyên là diện tích của hình chữ nhật lớn nhất tìm được.

BÀI 22: SỐ LƯỢNG CON TĂNG DÀI NHẤT

Cho một dãy số nguyên dương gồm n phần tử ai , i =1,2,3,...,n . Một *dãy con tăng* của dãy số đã cho nếu lấy liên tiếp các phần tử trong dãy đã cho và giữ nguyên trật tự ban đầu, sao cho mỗi phần tử trong dãy con không nhỏ hơn phần tử đứng kề trước nó. Độ dài của dãy con tăng là số phần tử trong dãy.

Hãy viết chương trình bằng ngôn ngữ Pascal, nhập từ bàn phím số nguyên dương **n**, n phần tử : **a1, a2, ..., an** và số nguyên dương **S**. Sau đó in ra màn hình dãy số con tăng dài nhất có tổng các phần tử nhỏ hơn hoặc bằng **S**.

Ví dụ: - Nhập  **n = 6**

Nhập dãy 6 phần tử: **4 5 2 3 5 3**

Nhập **S = 10**

- In ra màn hình: Dãy con dài nhất có chiều dài bằng **3**, gồm các phần tử là : **2 3 5**

program daycondainhat;

uses crt;

const max=200;

var x:array[1..max] of byte;

l,d,c,n,m,luu:byte;

Procedure nhap;

Var i:byte;

Begin

Write('Nhap gia tri tong cua day con: n = ');readln(n);

Write('Nhap so phan tu cua day ban dau : ');readln(m);

Writeln('Nhap ',m,' phan tu cho mang :');

for i:=1 to m do

begin write('X[',i,']= ');readln(x[i]); end;

End;

Procedure inkq;

Var i:byte;

Begin

if m=1 then

begin

if x[1]<=n then write('Day tang dai nhat bang 1 la: ',x[1]:4)

else write(' Khong co loi giai');

readln;

end

else

begin

if l=0 then begin write(' Khong co loi giai');readln; end

else

begin

c:=d+l-1;

writeln('Day tang dai nhat bang ',l,' la: ');

for i:=d to c do write(x[i]:4);readln;

end;

end;

End;

Procedure timdaytang;

Var i,j,s,dem:byte;

ok:boolean;

Begin

l:=0;d:=0;

for i:=1 to m-1 do

if x[i]<=n then

begin

s:=x[i];dem:=1;

ok:=true;j:=i+1;

while (ok and (j<=m)) do

begin

if (x[j]< x[i]) then ok:= false else

begin

s:=s+x[j];

if s<=n then begin inc(dem); inc(j) end

else ok:=false;

end;

if (l<dem) then begin l:=dem; d:=i; end;

end;

end;

End;

BEGIN

clrscr;

nhap;timdaytang;inkq;

END.

***Tes 1: (10 điểm)***

Nhập từ bàn phím: n = 10 và mảng 10 phần tử: 7 5 4 2 1 3 9 10 5; S =20

In ra màn hình: Dãy tăng dài nhất bằng 3 gồm các phần tử 1 3 9.

***Tes 2: (10 điểm)***

Nhập từ bàn phím: n = 2 và mảng 2 phần tử: 5 2; S = 2

In ra màn hình: Dãy tăng dài nhất bằng 1 gồm có phần tử 5.

***Tes 3: (5 điểm)***

Nhập từ bàn phím: n = 3 và mảng 3 phần tử: 4 5 4; S = 2

In ra màn hình: Không có lời giải.

***Tes 4: (5 điểm)***

Nhập từ bàn phím: n = 1 và mảng 1 phần tử: 2; S = 2

In ra màn hình: Dãy tăng dài nhất bằng 1 gồm phần tử 2.

***Tes 5: (5 điểm)***

Nhập từ bàn phím: n = 5 và mảng 5 phần tử: 1 1 1 1 1; S = 5

In ra màn hình: Dãy tăng dài nhất bằng 5 gồm 5 phần tử 1 1 1 1 1.

***Tes 6: (5 điểm)***

Nhập từ bàn phím: n = 10 và mảng 10 phần tử: 5 4 5 9 2 7 8 8 8 4; S = 30

In ra màn hình: Dãy tăng dài nhất bằng 4 gồm các phần tử 2 7 8 8.

BÀI 23: XÁC ĐỊNH THỨ TRONG TUẦN

Đối với những năm có tối đa bốn chữ số, là một *năm nhuận* nếu thỏa mãn các điều kiện sau:

- Nếu năm có tận cùng là hai chữ số 0 thì số năm đó phải chia hết cho 400.

- Nếu năm không tận cùng là hai chữ số 0 thì số năm đó phải chia hết cho 4.

Hãy viết chương trình bằng ngôn ngữ Pascal, nhập từ bàn phím ***ngày,tháng, năm*** và in ra màn hình ***thứ*** trong tuần. *Biết rằng ngày 1 tháng 1 năm 1 là thứ hai.*

Ví dụ: - Nhập ngày, tháng, năm : 15 7 2010

- In ra màn hình : Thứ Năm.

program xacdinhthumay;

uses crt;

const thu:array[0..6] of string[3]=('CN ','Hai','Ba ','Tu ','Nam','Sau','Bay');

s\_thg:array[1..12] of word=(0,31,59,90,120,151,181,212,243,273,304,334);

var ng,thg,nam:word;

id:byte;

Function namnhuan(x:word):boolean;

Begin

namnhuan:= (((x mod 4)=0) and ((x mod 100)<>0) or ((x mod 400)=0))

End;

Function ts\_namnhuan(x:word):word;

var i,d:word;

Begin

d:=0;

for i:=1 to x do if namnhuan(i) then inc(d);

ts\_namnhuan:=d;

End;

Procedure nhap;

Begin

Writeln('Nhap ngay, thang, nam: ');readln(ng,thg,nam);

End;

Procedure inkq;

Var s1:word;

Begin

s1:=nam-1 + ts\_namnhuan(nam-1)+ s\_thg[thg]+ng;

id := (s1 mod 7);

if namnhuan(nam) and (thg>2) then id:=(s1+1) mod 7;

Write('Thu can tra la: ',thu[id]);

readln;

End;

BEGIN

clrscr;

nhap;inkq;

END.

*Test 1(10 điểm):* Nhập từ bàn phím: 15 7 2007 🡪 In ra màn hình: Chủ nhật.

*Test 2(10 điểm):* Nhập từ bàn phím: 20 2 2008 🡪 In ra màn hình: Thứ tư.

*Test 3(10 điểm):* Nhập từ bàn phím: 16 7 2010 🡪 In ra màn hình: Thứ sáu.

Bài 24: BẢN TIN BÓNG ĐÁ

Sau cuối mùa giải bóng đá, căn cứ vào bảng điểm của tất cả các trận đấu, Ban tổ chức biết được số trận thắng, thua, hòa và tổng số điểm của mỗi đội. Từ đó, Ban tổ chức biết được đội bóng đá mạnh nhất trong mùa giải.

*Quy ước:* Trận thắng được 3 điểm, trận hòa được 1 điểm, thua là 0 điểm.

**Yêu cầu**: Người lập trình giúp Ban tổ chức, thống kê số trận thắng, hòa, thua, tổng điểm của mỗi đội, và tìm được đội bóng đá mạnh nhất.

**Dữ liệu**: Vào từ file văn bản BANGDIEM.INP

* Dòng đầu tiên ghi hai số nguyên dương n, m tương ứng là n đội và m trận đấu của mỗi đội, giữa 02 số cách nhau bởi dấu cách.
* Dòng thứ i trong n dòng tiếp theo chứa m số nguyên thuộc một trong các số 0, 1, 3 ứng với điểm mỗi trận đấu, và cách nhau bởi dấu cách.

**Kết quả**: Ghi ra file văn bản TKEGIAI.OUT: Ghi n dòng, mỗi dòng là bốn chữ số, theo thứ tự là số lượng trận thắng, hòa, thua và số cuối là tổng điểm của mỗi đội. Dòng tiếp theo, ghi số thứ tự của đội bóng trong danh sách ban đầu là đội bóng mạnh nhất. Nếu trong trường hợp có nhiều đội bóng có cùng tổng số điểm thì ta liệt kê hết các số thứ tự của đội bóng đó và cách nhau bởi dấu cách.

|  |
| --- |
| **TKEGIAI.OUT** |
| 1 1 2 4  3 1 0 10  0 2 2 2  2 |

|  |
| --- |
| **BANGDIEM.INP** |
| 3 4  1 3 0 0  3 3 3 1  1 1 0 0 |

**Ví dụ**:

Program bangtinbongda;

Const inp='BangDiem.inp';

out='Tkegiai.out';

var a: array[1..100,1..100] of word;

s,t,h: array[1..100] of word;

i,j,n,m,max: byte;

fi,fo:Text;

Procedure docfile\_xuly;

Var i,j:Byte;ss:Word;

Begin

assign(fi,inp);reset(fi);

readln(fi,n,m);

fillchar(s,sizeof(s),0);

fillchar(t,sizeof(t),0);

fillchar(h,sizeof(h),0);

for i:=1 to n do

begin

for j:=1 to m do

begin

read(fi,a[i,j]);

s[i]:=s[i]+a[i,j];

if a[i,j]=3 then t[i]:=t[i]+1;

if a[i,j]=1 then h[i]:=h[i]+1;

end;

readln(fi);

end;

close(fi);

max:=s[1];

for i:=2 to n do if max<s[i] then max:=s[i];

End;

Procedure ghifile;

Var i,j: Byte;

Begin

assign(fo,out);rewrite(fo);

for i:=1 to n do

writeln(fo,t[i]:3,h[i]:3,m-(t[i]+h[i]):3,s[i]:3);

for i:=1 to n do if s[i]=max then write(fo,i:3);

close(fo);

end;

BEGIN

docfile\_xuly;

ghifile;

END.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test 1 *(10 điểm)*** | | **Test 2*(10 điểm)*** | | **Test 3*(10 điểm)*** | |
| **BANGDIEM.INP** | **TKEGIAI.OUT** | **BANGDIEM.INP** | **TKEGIAI.OUT** | **BANGDIEM.INP** | **TKEGIAI.OUT** |
| 3 4  1 3 0 0  3 3 3 1  1 1 0 0 | 1 1 2 4  3 1 0 10  0 2 2 2  2 | 4 5  1 0 0 3 3  0 1 1 1 3  3 1 1 1 1  1 1 1 1 0 | 2 1 2 7  1 3 1 6  1 4 0 7  0 4 1 4  1 3 | 5 5  1 1 1 3 3  1 1 3 3 3  0 0 0 1 1  3 3 3 1 1  1 1 1 0 0 | 2 3 0 9  3 2 0 11  0 2 3 2  3 2 0 11  0 3 2 3  2 4 |

Bài 25: NHÀ CHUNG CƯ

Một khu chung cư được xây dựng nhằm đáp ứng nhu cầu nhà ở đối với những người có thu nhập thấp. Do nhu cầu nhà ở rất lớn và nhu cầu cũng rất khác nhau, nên người kinh doanh nhà ở đã xây dựng với số lượng căn hộ rất lớn và giá trị cho thuê của mỗi căn hộ cũng khác nhau. Mỗi căn hộ được trang bị khá đầy đủ tiện nghi, đảm bảo cho một ga đình sinh hoạt hàng ngày.

**Yêu cầu**: Người lập trình hãy giúp cho người kinh doanh thống kê có bao nhiêu loại căn hộ ứng với các mức giá trị cho thuê. Đồng thời cho biết số lượng căn hộ có giá trị cho thuê bao nhiêu là nhiều nhất?

**Dữ liệu**: Vào từ file văn bản GIACANHO.INP

* Dòng đầu tiên ghi số nguyên dương N là số lượng căn hộ của khu chung cư.
* Dòng thứ hai trở đi, mỗi dòng chứa 10 số (cho đến khi hết N số), mỗi số cách nhau một dấu cách, dòng cuối cùng có thể ít hơn 10 số, số nhỏ nhất là 100 (đơn vị tính là triệu đồng), số lớn nhất là 800.

**Kết quả**: Ghi ra file văn bản LOAIGIA.OUT: dòng đầu ghi M là số lượng loại căn hộ khác nhau ứng với mỗi giá trị cho thuê; dòng thứ hai ghi 2 chứ số cách nhau bởi dấu cách, số đầu ghi số lượng căn hộ có mức giá cho thuê nhiều nhất, số thứ hai là giá trị căn hộ đó.

**Ví dụ**:

|  |
| --- |
| **GIACANHO.INP** |
| 15  100 150 150 200 250 300 250 150 400 150  500 150 600 700 800 |

|  |
| --- |
| **LOAIGIA.OUT** |
| 10  5 150 |

Program giachungcu;

Const inp='giacanho.inp';

out='loaigia.out';

var a,d: array[1..1000] of integer;

n,i,gt,sl,giaLN,dem:integer;

fi,fo:Text;

Procedure docfile\_xuly;

Begin

assign(fi,inp);reset(fi);

readln(fi,n);

fillchar(d,sizeof(d),0);

giaLN:=0;

for i:=1 to n do

begin

if (i mod 10)<> 0 then read(fi,a[i]) else readln(fi,a[i]);

d[a[i]]:=d[a[i]]+1;

if giaLN<a[i] then giaLN:=a[i];

end;

close(fi);

sl:=0;dem:=0;

for i:=1 to giaLN do

if d[i]<> 0 then

begin

dem:=dem +1;

if sl<d[i] then

begin

sl:=d[i];gt:=i;

end;

end;

end;

Procedure ghifile;

Begin

assign(fo,out);rewrite(fo);

write(fo,dem:4);writeln(fo);

write(fo,sl:4,gt:4);

close(fo);

end;

BEGIN

docfile\_xuly;

ghifile;

END.

|  |  |
| --- | --- |
| Test 1 *(20 điểm)* | |
| GIACANHO.INP | LOAIGIA.OUT |
| 15  100 150 150 200 250 300 250 150 400 150  500 150 600 700 800 | 10  5 150 |

|  |  |
| --- | --- |
| Test 2 *(20 điểm)* | |
| GIACANHO.INP | LOAIGIA.OUT |
| 22  100 200 150 200 250 300 300 300 300 650  500 150 300 700 800 750 550 300 200 300  300 450 | 12  8 300 |

BÀI 26: SORT

const fi='sort6.inp';

fo='sort6.out';

var f:text;

dem,n:longint;

kt:array[0..100] of boolean;

a,x:array[0..12] of longint;

procedure nhap;

var i:longint;

begin

assign(f,fi);reset(F);

readln(f,n);

for i:=1 to n do read(f,a[i]);

fillchar(kt,sizeof(kt),true);

end;

procedure ghinghiem;

var w,t,h,k:longint;

dung:boolean;

begin

dung:=true;

for t:=2 to n-1 do

for h:=1 to t-1 do

for k:=t+1 to n do

if 2\*x[t]=x[h]+x[k] then

begin

dung:=false;

break;

end;

if dung=true then

begin

inc(dem);

for w:=1 to n do write(f,x[w],' ');

writeln(F);

end;

end;

procedure duyet(q:longint);

var i:longint;

begin

for i:=1 to n do

if kt[i]=true then

begin

x[q]:=a[i];

kt[i]:=false;

if q=n then ghinghiem

else duyet(q+1);

kt[i]:=true;

end;

end;

BEGIN

nhap;

assign(f,fo);

rewrite(F);

duyet(1);

writeln(f,dem);

closE(F);

END.

BÀI 27: TỌA ĐỘ HÌNH CHỮ NHẬT

Program P100515; {Tim Hinh chu nhat co dien tich lon nhat}

Uses Crt;

Var f: text;

m,n,d1,d2,c1,c2: byte;

max: integer;

a: array[1..100,1..100] of 0..1;

Procedure Nhap;

Var i,j: byte;

Begin

Assign(f,'INP.TXT');

Reset(f);

Readln(f,m,n);

For i:=1 to m do

begin

For j:=1 to n do Read(f,a[i,j]); Readln(f);

end;

Close(f);

End;

Function thoa\_man(i1,j1,i2,j2: byte): boolean;

Var i,j: byte;

Begin

thoa\_man:=false;

For i:=i1 to i2 do

For j:=j1 to j2 do

If a[i,j]=1 then exit;

thoa\_man:=true;

End;

Procedure Xu\_ly;

Var i1,j1,i2,j2: byte;

Begin

max:=0;

For i1:=1 to m do

For j1:=1 to n do

If a[i1,j1]=0 then

For i2:=i1 to m do

For j2:=j1 to n do

If a[i2,j2]=0 then

If thoa\_man(i1,j1,i2,j2) then

If max<(abs(i2-i1+1)\*abs(j2-j1+1)) then

begin

max:=abs(i2-i1+1)\*abs(j2-j1+1);

d1:=i1; d2:=i2;

c1:=j1; c2:=j2;

end;

Writeln('Hinh chu nhat toan so 0 co dien tich lon nhat la: ',max);

Writeln('Toa do trai tren: ',d1,' ',c1);

Writeln('Toa do phai duoi: ',d2,' ',c2);

End;

BEGIN

ClrScr;

Nhap;

Xu\_ly;

Readln;

END.

BÀI 28: ƯỚC NGUYÊN TỐ LỚN NHẤT

Nhập từ bàn phím một số nguyên dương N (2 ≤ N ≤ 109). Hãy in ra màn hình số nguyên tố P lớn nhất là ước của số N.

Ví dụ: Nhập N = 26.

Kết quả trên màn hình là P = 13.

BÀI 29: XÂU CON

Nhập hai xâu kí tự S1và S2 từ bàn phím.

Yêu cầu: Nếu xâu S1 là xâu con của xâu S2 thì in ra màn hình vị trí xuất hiện cuối cùng của xâu S1 trong S2 (có phân biệt chữ hoa và chữ thường). Nếu xâu S1 không phải là xâu con của xâu S2 thì in màn hình ­số 0.

­**Ví dụ: Nhập S1 = ‘abc’ S2 = ‘abcabcAbc’**

Kết quả trên màn hình là 4

BÀI 30: PHÂN SỐ

Cho trước hai số nguyên dương K, P (2 ≤ K ≤ 1500).

**Yêu cầu:**

1. Hãy cho biết có bao nhiêu phân số  đôi một khác nhau (a, b ∈ N\*) thỏa mãn điều kiện: 0 <  < 1 và b ≤ K.
2. Tìm phân số thứ P trong dãy phân số trên sau khi đã sắp xếp theo thứ tự tăng dần.

Ví dụ: Với K = 5 và P = 4 ta có 9 phân số thỏa mãn và được sắp xếp theo thứ tự tăng dần là: . Khi đó phân số thứ P = 4 là 

**Input:** File văn bản **PS.INP** ghi 2 số nguyên dương K và P trên 1 dòng, mỗi số cách nhau ít nhất một dấu cách.

**Output:** File văn bản **PS.OUT** ghi 2 dòng:

* Dòng 1: Ghi số nguyên m là số các phân số  thỏa mãn yêu cầu 1 của bài toán.
* Dòng 2: Nếu tìm được phân số thứ P thì ghi 2 số nguyên a và b (mỗi số cách nhau ít nhất một dấu cách) là **tử và mẫu của phân số thứ P** trong dãy các phân số tìm được sau khi đã sắp xếp theo thứ tự tăng dần. Nếu không tìm được phân số thứ P thì ghi một số 0.

Ví dụ:

|  |  |
| --- | --- |
| **PS.INP** | **PS.OUT** |
| 5 4 | 9  2 5 |

BÀI 31: TỔNG ĐOẠN CON NGẮN NHẤT

Cho dãy số nguyên không âm: a1, a2, a3,…, an. Xếp các số trong dãy theo thứ tự ban đầu trên một vòng tròn theo chiều kim đồng hồ. Ta gọi một đoạn con của vòng tròn là một số các số liên tục theo chiều kim đồng hồ trên vòng tròn đó.

**Yêu cầu:** Hãy tìm một đoạn con có tổng bằng số nguyên S cho trước sao cho độ dài của đoạn con đó là ngắn nhất.

**Input:** File văn bản **SUMSEQ.INP** có cấu trúc như sau:

* Dòng 1: Ghi 2 số nguyên dương n và S (với n ≤ 105).
* Dòng 2: Ghi n số nguyên không âm: a1, a2, a3,…, an (ai ≤ 109).

**Output:** File văn bản **SUMSEQ.OUT**

Ghi một số m là độ dài ngắn nhất của đoạn con có tổng bằng S (nếu không tìm được dãy con nào thi ghi ra một số 0).

Ví dụ:

|  |  |
| --- | --- |
| **SUMSEQ.INP** | **SUMSEQ.OUT** |
| 5 3  1 1 0 0 2 | 2 |

Bài 32: TỔNG LŨY THỪA

Viết chương trình in ra màn hình các số có 4 chữ số, sao cho tổng các lũy thừa 4 của 4 chữ số thì bằng chính nó.

BÀI 33: ƯỚC SỐ CHUNG LỚN NHẤT

Viết chương trình nhập vào một dãy N số nguyên dương lớn hơn 2. In ra màn hình ước số chung lớn nhất của lần lượt từng số từ số thứ 2 đến số N với số đầu tiên.

***Ví dụ:***

|  |  |
| --- | --- |
| Với N = 5  Nhap day so: 3 6 13 22 9 | Cac uoc lan luot la:  3  1  1  3 |

BÀI 34: XE BUS

**Chọn chuyến xe buýt** *( 7 điểm).*

Có một trạm xe buýt ở vùng quê nọ rất uy tín với khách nhờ việc lập lịch khoa học và phục vụ tốt. Chuyến xe buýt khởi hành đầu tiên đi về các tuyến vào buổi sáng, thời gian To và cách một khoảng thời gian D lại có chuyến xe kế tiếp khởi hành để chở khách. Mỗi ngày doanh nghiệp xe buýt này có Ci chuyến, chuyến thứ i khởi hành vào lúc To + (i-1)×D, chuyến 1 khởi hành lúc To.

Có N hành khách thường xuyên đi xe buýt tại bến, hành khách thứ j đến bến ở thời điểm sj và muốn người quản lý trạm xe buýt cho biết rõ chuyến đi để chủ động công việc.

**Yêu cầu:** Cho T0, D và thời điểm các hành khách đến bến, hãy giúp nhà quản lý trạm xe xác định số thứ tự chuyến xe buýt để khách hàng đi.

**Dữ liệu vào:** File văn bản BUS.INP

+ Dòng thứ nhất chứa ba số nguyên N, D, T0 (2≤N≤100;1 ≤D≤100; 0≤T0≤100) .

+ Dòng thứ hai chứa N số nguyên S1,S2,....,SN (0≤Sj ≤ 103).

**Dữ liệu ra:** Ghi ra file văn bản BUS.OUT

Gồm một dòng chứa N số nguyên dương, mỗi số nguyên dương là số thứ tự chuyến xe buýt mà hành khách đi.

Ví dụ:

|  |  |
| --- | --- |
| BUS.INP | BUS.OUT |
| 3 5 8  0 15 23 | 1 3 4 |

//procedure choxebuyt;

const fi='BUS.INP2';

FO='BUS.OUT2';

var a:array[0..1000] of longint;

i,n,t,d,q,w:longint;

f:text;

procedure nhap;

begin

assign(f,fi);reset(F);

readln(f,n,d,t);

for i:=1 to n do read(f,a[i]);

closE(f);

end;

BEGIN

nhap;

assign(f,fo);rewrite(F);

for i:=1 to n do

if a[i]<=t then write(f,1,' ')

else

begin

a[i]:=a[i]-t;

q:=a[i] div d;

w:=a[i] mod d;

if w=0 then write(f,q+1,' ')

else write(f,q+2,' ');

end;

close(F);

END.

type mang=array[0..100]of longint;

const fi='g:\nga\BUS.INP';

fo='g:\nga\BUS.OUT';

var f:text;

n,d,t:longint;

i,j,l,c,e:longint;

s,b:mang;

begin

assign(f,fi);

reset(f);

readln(f,n,d,t);

for j:=1 to n do

read(f,s[j]);

assign(f,fo);

rewrite(f);

i:=t;

l:=0;

b[0]:=0;

c:=t+(j-1)\*d;

while i<=c do

begin

l:=l+1;

b[l]:=i;

i:=i+d;

end;

for i:=1 to j do

begin

if s[i]<=t then

write(f,'1 ')

else if s[i]>c then

write(f,n+1)

else for e:=2 to l do

begin

if (s[i]>b[e-1]) and (s[i]<=b[e]) then

begin

write(f,e,' ');

break;

end

end ;

end;

close(f);

end.

|  |  |  |
| --- | --- | --- |
| Test | BUS.INP | BUS.OUT |
| 1 | BUS.INP2 | 1 3 4 |
| 2 | BUS.INP3 | 1 2 1 |
| 3 | BUS.INP4 | 5 6 1 3 |
| 4 | BUS.INP5 | 5 1 1 9 4 |
| 5 | BUS.INP6 | 3 3 1 3 2 2 |
| 6 | BUS.INP7 | 2 1 1 1 1 1 1 |
| 7 | BUS.INP8 | 11 1 8 1 1 5 7 9 |

BÀI 35: Trò chơi truyền hình

Một trò chơi dành cho nhiều đội chơi được Ban tổ chức quy định như sau: bắt đầu cuộc chơi, Ban tổ chức cho chạy N số trên màn hình từ trái sang phải và yêu cầu đội chơi nào trả lời nhanh nhất số cách chia dãy số trên thành nhiều dãy con nhất sao cho các dãy con có tổng bằng nhau.

Ví dụ: Trên màn hình chạy dòng 19 số: 1 0 2 0 0 3 1 1 1 0 2 1 0 0 2 1 2 0 1. Kết quả trả lời đúng là: tổng các số mỗi dãy con là 3, số dãy con nhiều nhất là 6, bao gồm các dãy con là: 1 0 2, 0 0 3, 1 1 1, 0 2 1, 0 0 2 1, 2 0 1

+ **Dữ liệu vào:** File văn bản GAMESHOW.INP

- Dòng 1 chứa số nguyên dương N.

- Dòng 2 chứa N số chạy trên màn hình, các số cách nhau bởi dấu cách.

**+Dữ liệu ra:** Ghi ra file văn bản GAMESHOW.OUT

- Dòng 1 là tổng các số của một dãy con.

- Dòng 2 là số K (số dãy con nhiều nhất chia được).

- Dòng 3 là các vị trí cuối cùng của từng dãy con, các số vị trí này cách nhau bởi dấu cách.

Ví dụ:

|  |  |
| --- | --- |
| GAMESHOW.INP | GAMESHOW.OUT |
| 19  1 0 2 0 0 3 1 1 1 0 2 1 0 0 2 1 2 0 1 | 3  6  3 6 9 12 16 19 |

//procedure gameshow

const fi='GAMESHOW.inp40';

FO='GAMESHOW.OUT100';

var f:text;

n:longint;

a:array[0..1000] of longint;

procedure nhap;

var i:longint;

begin

assign(F,fi);reset(F);

readln(F,n);

for i:=1 to n do read(f,a[i]);

closE(F);

end;

procedure xuli;

var tong,i,j,s,t,tam:longint;

chiaduoc,stop:boolean;

luu:array[0..1000] of longint;

begin

tong:=0;

for i:=1 to n do tong:=tong+a[i];

chiaduoc:=false;

for i:=n downto 1 do //thu so nhom se chia duoc

if tong mod i=0 then

begin

for j:=1 to i do luu[j]:=0;

s:=tong div i;

stop:=false;

j:=0;

t:=0;

while not(stop) and (j<n)do

begin

tam:=0;

while tam<s do

begin

inc(j);

tam:=tam+a[j];

end;

if tam=s then

begin

inc(t);

luu[t]:=j;

end

else stop:=true;

end;

if luu[i]<>0 then break;

end;

assign(f,fo);rewrite(F);

writeln(f,s);

writeln(F,i);

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

close(F);

end;

BEGIN

nhap;

xuli;

END.

type mang=array[1..1000]of longint;

const fi='g:\nga\GAMESHOW.INP';

fo='g:\nga\GAMESHOW.OUT';

var f:text;

a,b,x:mang;

n,i,t,l,e,j,k,tt,sld,s:longint;

kt:boolean;

function ktbang(b:mang;m:longint):boolean;

begin

for i:=1 to m-1 do

if b[i]<>b[i+1] then exit(false);

exit(true);

end;

function tongday(b:mang;m:longint):longint;

var s:longint;

begin

s:=0;

for i:=1 to m do

s:=s+b[i];

tongday:=s;

end;

procedure uoc;

begin

l:=0;

for i:=2 to t do

begin

if t mod i=0 then

begin

l:=l+1;

b[l]:=i;

end;

end;

end;

begin

assign(f,fi);

reset(f);

readln(f,n);

for i:=1 to n do

read(f,a[i]);

assign(f,fo);

rewrite(f);

if ktbang(a,n)=true then

begin

writeln(f,a[1]);

writeln(f,n);

for i:=1 to n do

begin

write(f,a[i]);

write(f,' ');

end;

end

else

Begin

t:=tongday(a,n);

uoc;

k:=0;

tt:=0;

for e:=1 to l do

begin

kt:=true;

sld:=0;

i:=1;

while i<=n do

begin

j:=i;

s:=0;

while (j<=n) and (s<>b[e]) do

begin

s:=s+a[j];

j:=j+1;

end;

i:=j;

if s<>b[e] then

begin

kt:=false;

break;

end;

sld:=sld+1;

end;

if kt=true then

if sld>k then

begin

k:=sld;

tt:=b[e];

end;

end;

i:=1;

e:=0;

while i<=n do

begin

j:=i;

s:=0;

while (j<=n) and (s<>tt) do

begin

s:=s+a[j];

j:=j+1;

end;

e:=e+1;

x[e]:=j-1;

i:=j;

end;

writeln(f,tt);

writeln(f,k);

for i:=1 to e do

begin

write(f,x[i]);

write(f,' ');

end;

End;

close(f);

end.

|  |  |  |
| --- | --- | --- |
| Test | GAMESHOW.INP | GAMESHOW.OUT |
| 1 | GAMESHOW.INP1 | 3  6  3 6 9 12 16 19 |
| 2 | GAMESHOW.INP2 |  |
| 3 | GAMESHOW.INP3 | 513  1  20 |
| 4 | GAMESHOW.INP4 | 902  1  35 |
| 5 | GAMESHOW.INP5 | 1228  1  47 |
| 6 | GAMESHOW.INP6 | 2098  1  79 |
| 7 | GAMESHOW.INP7 | 3  6  3 6 9 12 16 19 |

Bài 36. Hai hình chữ nhật

Cho một bảng các ô vuông đơn vị có kích thước n×m(n, m ≤100; n số hàng, m số cột của bảng). Trên mỗi ô vuông đơn vị chứa các số nguyên dương.

**Yêu cầu**: Hãy tìm ở bảng trên hai hình chữ nhật (có thể giao nhau nhưng không trùng khít lên nhau) có kích thước p×q (p < n; q < m; p là số hàng, q là số cột của hai hình chữ nhật con) sao cho tổng của tất cả các số trên hai hình chữ nhật là lớn nhất.

**Dữ liệu vào:** File tên HCN.INP

- Dòng 1 ghi bốn số n, m, p, q được ghi cách nhau bởi dấu cách.

- n dòng tiếp theo, mỗi dòng ghi m số nguyên dương và cách nhau bởi dấu cách.

**Dữ liệu ra**: File HCN.OUT, chứa 1 dòng ghi một số nguyên duy nhất S là diện tích lớn nhất cần tìm.

|  |  |
| --- | --- |
| HCN.INP | HCN.OUT |
| 1. 7 3 3 2. 2 3 1 1 3 4 3. 3 1 4 0 1 2 4. 6 3 4 7 1 1   9 3 8 10 0 1 0 | 84 |

Ví dụ:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 1 | 1 | 3 | 4 |
| **5** | **3** | **1** | **4** | **0** | 1 | 2 |
| **4** | **6** | **3** | **4** | **7** | 1 | 1 |
| **9** | **3** | **8** | **10** | **0** | 1 | 0 |

// hai hinh chu nhat

const fi='hcn.inp’;

fo='hcn.out';

var m,n,p,q:longint;

a:array[0..1010,0..1010] of longint;

f:text;

procedure nhap;

var i,j:longint;

begin

assign(F,fi);reset(F);

readln(F,n,m,p,q);

for i:=1 to n do

for j:=1 to m do read(f,a[i,j]);

closE(F);

end;

procedure xuli;

var i,j,tmax1,tmax2,u,v,kq,tam:longint;

s:array[0..1010,0..1010] of longint;

begin

fillchar(s,sizeof(s),0);

for i:=1 to n do

for j:=1 to m do

s[i,j]:=s[i-1,j]+s[i,j-1]-s[i-1,j-1]+a[i,j];

tmax1:=0;

tmax2:=0;

for i:=1 to n-p+1 do

for j:=1 to m-q+1 do

begin

u:=i+p-1;

v:=j+q-1;

tam:=s[u,v]-s[i-1,v]-s[u,j-1]+s[i-1,j-1];

if tam>tmax1 then begin tmax2:=tmax1;tmax1:=tam;end

else if tam>tmax2 then tmax2:=tam;

end;

kq:=tmax1+tmax2;

assign(F,fo);rewrite(F);

writeln(f,kq);

closE(F);

end;

BEGIN

nhap;

xuli;

END.

|  |  |  |
| --- | --- | --- |
| Test | HCN.INP | HCN.OUT |
| 1 | HCN.INP1 | 84 |
| 2 | HCN.INP2 | 11484 |
| 3 | HCN.INP3 | 25899 |
| 4 | HCN.INP4 | 22959 |
| 5 | HCN.INP5 | 69 |
| 6 | HCN.INP6 | 6432 |

Bài 37. Trò chơi kiểm tra trí nhớ

Một trò chơi yêu cầu người chơi nhớ nhanh các con số điện tử chạy trên màn hình lớn và tính tổng các số hiện lên theo yêu cầu của cuộc chơi. Các số hiện ra trên màn hình là một dãy số gồm N số nguyên không âm a1, a2, …, aN và cứ lặp đi  lặp lại như thế (nghĩa là sau khi aixuất hiện vài giây đến lượt ai+1 xuất hiện, số xuất hiện sau aNlà a1).

***Yêu cầu:***Bạn hãy giúp người chơi kiểm tra kết quả tính tổng của M số nguyên liên tiếp xuất hiện trên màn hình bắt đầu từ số nguyên thứ T khi giám khảo yêu cầu.

***Dữ liệu vào*:**File text, tên file là BL1.INP, gồm 02 dòng:

+ Dòng đầu ghi ba số nguyên N, T, M (1 £ T £N £104; 1 £ M £ 2. 104 );

+ Dòng thứ hai chứa N số ai (ai< 2.109, i = 1,…,N), cách nhau bởi dấu cách.

***Dữ liệu ra:*** File text, tên File là BL1.OUT, gồm duy nhất một số là tổng tìm được.

*Ví dụ:*

|  |  |
| --- | --- |
| **BL1.INP** | **BL1.OUT** |
| 9 7 5  1 2 5 7 8 5 3 1 5 | 12 |

BÀI 38: DÃY SỐ HOÀN THIỆN

Một dãy số được gọi là dãy số hoàn thiện nếu được sắp xếp thứ tự giảm dần theo số ước của nó (nếu 2 số có số ước bằng nhau thì số nhỏ đứng trước) và trong dãy không có phần tử nào giống nhau.

Yêu cầu: Cho trước dãy số, em hãy viết chương trình xuất ra dãy số hoàn thiện

**Dữ liệu vào:** File văn bản **DAYSO.INP** gồm 2 dòng

* Dòng 1 ghi một số tự nhiên n (n<107)
* Dòng 2 Ghi n số tự nhiên của dãy, mỗi số cách nhau một khoảng trắng (dấu cách)

**Dữ liệu ra:** File văn bản **DAYSO.OUT** gồm 1 dòng là dãy số hoàn thiện từ dãy số ban đầu

**Ví dụ:**

|  |  |
| --- | --- |
| **DAYSO.INP** | **DAYSO.OUT** |
| **7**  **5 9 12 24 9 5 31** | **24 12 9 5 31** |

Bài 39: Tìm số nguyên dương lớn nhất

Trong cuộc thi giải toán Casio qua mạng internet mỗi học sinh đều có số điểm tích lũy riêng của mình. Số điểm tích lũy của mỗi học sinh là một số nguyên dương k (0 < k ≤ 2×109). Đội tuyển của trường THPT Triển Vọng có N học sinh tham gia dự thi (2 ≤ N ≤ 100). Tại buổi gặp mặt trước kỳ thi cấp tỉnh, thầy hiệu trưởng quyết định thưởng cho các học sinh trong đội tuyển M triệu đồng, biết rằng điểm tích lũy của mỗi học sinh đều chia hết cho M.

***Yêu cầu:*** Hãy tìm số nguyên dương M lớn nhất.

***Dữ liệu vào:*** Cho trong file văn bản PT.INP có cấu trúc như sau:

- Dòng 1: Ghi số nguyên dương N là số lượng học sinh.

- Dòng 2: Ghi N số nguyên dương lần lượt là điểm tích lũy của N học sinh, các số được ghi cách nhau ít nhất một dấu cách.

**Dữ liệu ra:** Ghi ra file văn bản PT.OUT theo cấu trúc như sau:

- Dòng 1: Ghi số nguyên dương M tìm được.

***Ví dụ:***

|  |  |
| --- | --- |
| **PT.INP** | **PT.OUT** |
| **5**  **15 24 45 36 27** | **3** |

Bài 40: Số chính phương nhỏ nhất , nguyên tố nhỏ nhất

Một số tự nhiên được gọi là số chính phương nếu căn bậc hai của số đó là một số tự nhiên.

Cho một dãy gồm n số tự nhiên ai(0 ≤ ai< 109; n < 104;  i =1, 2,...,n).

***Yêu cầu:*** Bạn hãy lập trình tìm ra số chính phương nhỏ nhất không xuất hiện trong dãy số đã cho.

***Dữ liệu vào***: file text, tên file là BL1.INP, gồm 02 dòng:

          -Dòng 1 ghi giá trị của n

          - Dòng thứ 2, ghi n số tự nhiên của dãy, hai số liền kề cách nhau một dấu cách.

***Dữ liệu ra***: file text, tên file là BL1.OUT,gồm 01 dòngghi duy nhất một số chính phương nhỏ nhất không xuất hiện trong dãy số đã cho.

*Ví dụ*:

|  |  |
| --- | --- |
| **BL1.INP** | **BL1.OUT** |
| 6  1 2 5 0 15 16 | 4 |

const fi='e:\ha.pas\scpnn.inp';

      fo='e:\ha.pas\scpnn.out';

var a:array [1..100] of longint;

    i,n,min,d:longint;

    f:text;

procedure qsort(first,last:longint);

var i,j,tg,chot:longint;

begin

    i:=first;

    j:=last;

    chot:=a[(first+last) div 2];

    repeat

       while a[i]<chot do inc(i);

       while a[j]>chot do dec(j);

       if i<=j then

         begin

           tg:=a[i];

           a[i]:=a[j];

           a[j]:=tg;

           inc(i);

           dec(j);

         end;

    until i>j;

    if i<last then qsort(i,last);

    if first<j then qsort(first,j);

end;

function nt(x:longint):boolean;

    var i:longint;

    begin

        if x<2 then exit(false)

        else

           begin

                for i:=2 to x-1 do

                    if x mod i=0 then exit(false);

                    exit(true);

           end;

    end;

  {function scp(x:longint):boolean;

    var i:longint;

    begin

        if x<0 then exit(false)

        else

           begin

                for i:=1 to x do

                    if sqrt(i)\*sqrt(i)<> i then exit(false);

                    exit(true);

           end;

    end; }

BEGIN

    assign(f,fi);

    reset(f);

    readln(f,n);

        for i:=1 to n do

        read(f,a[i]);

    assign(f,fo);

    rewrite(f);

qsort(1,n);

  {// so nguyen to

  d:=0;

    min:=0;

      for i:=1 to n do

       if (a[i]=min )then

        repeat

                begin

                        d:=d+1;

                        min:=d;

                end;

                        inc(min)

        until nt(min)=true; }

    //so chinh phuong

    d:=0;

    min:=0;

      for i:=1 to n do

       if (a[i]=min ) then

        repeat

                begin

                        d:=d+1;

                        min:=d;

                end;

                        inc(min)

        until (sqr(trunc(sqrt(min)))=min);

    if min<a[i] then

           write(f,min)

    else write(f,' khong tim thay');

    close(f);

    END.

-------------------------

const  fi='e:\ha.pas\sntnn.inp';

           fo='e:\ha.pas\sntnn.out';

var a:array [1..100] of longint;

    i,n,min,d:longint;

    f:text;

    kt:boolean;

procedure qsort(first,last:longint);

var i,j,tg,chot:longint;

begin

    i:=first;

    j:=last;

    chot:=a[(first+last) div 2];

    repeat

       while a[i]<chot do inc(i);

       while a[j]>chot do dec(j);

       if i<=j then

         begin

           tg:=a[i];

           a[i]:=a[j];

           a[j]:=tg;

           inc(i);

           dec(j);

         end;

    until i>j;

    if i<last then qsort(i,last);

    if first<j then qsort(first,j);

end;

function  ktnt(x:longint):boolean;

    var i:longint;

    begin

        if x<2 then exit(false)

        else

           begin

                for i:=2 to x-1 do

                    if x mod i=0 then exit(false);

                    exit(true);

           end;

    end;

BEGIN

    assign(f,fi);

    reset(f);

    readln(f,n);

        for i:=1 to n do

        read(f,a[i]);

    assign(f,fo);

    rewrite(f);

    qsort(1,n);

    d:=0;

    min:=0;

      for i:=1 to n do

       if (a[i]=min )then

        repeat

                begin

                        d:=d+1;

                        min:=d;

                end;

                        inc(min)

        until  ktnt(min)=true;

        kt:=false;

        for i:=1 to n do

        begin

                if (min<a[i]) and(min >=2) then

                kt:=true;

                end;

                if kt=true then write(f,min)

                else write(f,' khong tim thay');

    close(f);

    END.

Bài 41: TIN HỌC TRẺ\_2015\_B (CÂU 1)

const fi='KMIN.INP10';

fo='KMIN.OUT10';

var n:longint;

a:array[0..10000001] of longint;

f:text;

procedure nhap;

var i:longint;

begin

assign(f,fi);reset(f);

readln(f,n);

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

closE(F);

end;

procedure sx(q,w:longint);

var dau,cuoi,tam,tam1:longint;

begin

dau:=q;cuoi:=w;

tam:=a[(dau+cuoi)div 2];

repeat

while tam>a[dau] do inc(dau);

while tam<a[cuoi] do dec(cuoi);

if dau<=cuoi then

begin

tam1:=a[dau];

a[dau]:=a[cuoi];

a[cuoi]:=tam1;

inc(dau);

dec(cuoi);

end;

until dau>cuoi;

if q<cuoi then sx(q,cuoi);

if w>dau then sx(dau,w);

end;

procedure duyet;

var i,j:longint;

begin

assign(f,fo);rewrite(F);

a[0]:=0;

for i:=1 to n do

if a[i]-a[i-1]>1 then

begin

writeln(f,a[i-1]+1);

close(f);

exit;

end;

writeln(f,a[n]+1);

closE(F);

end;

BEGIN

nhap;

sx(1,n);

duyet;

END.

Bài 42: THT\_2015 \_B(CÂU 2)

const fi='RECT9.INP10';

FO='RECT9.OUT10';

var f:text;

n,m:longint;

a,b:array[0..1001,0..1001] of longint;

procedure nhap;

var i,j:longint;

begin

assign(f,fi);reset(F);

readln(f,n,m);

for i:=1 to n do

for j:=1 to m do

read(f,a[i,j]);

for i:=1 to n do

for j:=1 to m do

begin

b[i,j]:=b[i-1,j]+b[i,j-1]-b[i-1,j-1]+a[i,j];

end;

end;

procedure xuli;

var i,j,t,h,o,kq,s:longint;

begin

kq:=0;

for i:=1 to n do

for j:=1 to m do

for t:=i to n do

for h:=j to m do

begin

s:=b[t,h]-b[i-1,h]-b[t,j-1]+b[i-1,j-1];

o:=(t-i+1)\*(h-j+1);

if s div 9= o then

begin

if kq<o then kq:=o;

end;

end;

assign(f,fo);rewrite(F);

writeln(f,kq);

closE(F);

end;

BEGIN

nhap;

xuli;

END.

Bài 43: THT\_2015 \_B(CÂU3)

const fi='WORD.INP10';

FO='WORD.OUT10';

var f:text;

st,s,kq:string;

st1,st2,st3,st12,st22,st32:string;

n:longint;

procedure nhap;

begin

assign(F,fi);reset(F);

readln(F,st);

n:=length(st);

closE(F);

end;

function daonguoc(sau:string):string;

var dau,cuoi:longint;

tam:char;

begin

dau:=1;

cuoi:=length(sau);

while dau<cuoi do

begin

tam:=sau[dau];

sau[dau]:=sau[cuoi];

sau[cuoi]:=tam;

inc(dau);

dec(cuoi);

end;

exit(sau);

end;

procedure xuli;

var i,j,t:longint;

begin

kq:=st;

for i:=1 to n-2 do

for j:=i+1 to n-1 do

for t:=j+1 to n do

begin

st1:=copy(st,1,i);

st2:=copy(st,i+1,j-i);

st3:=copy(st,j+1,n-j);

st12:=daonguoc(st1);

st22:=daonguoc(st2);

st32:=daonguoc(st3);

s:=st12+st22+st32;

if s<kq then kq:=s;

end;

assign(f,fo);rewrite(F);

writeln(F,kq);

closE(F);

end;

BEGIN

nhap;

xuli;

END.

**Bài 1 (6điểm):** Test 6 bộ, mỗi bộ 1 điểm- Thời gian chạy mỗi bộ không quá 1s. Quá thời gian trên không cho điểm. Yêu cầu kết quả phải là số chính xác, không chấp nhận kết quả tính gần đúng.

|  |  |  |
| --- | --- | --- |
| Test | KMIN.INP | KMIN.OUT |
| 4 | KMIN.INP4 | 17 |
| 5 | KMIN.INP 5 | 8 |
| 6 | KMIN.INP 6 | 1 |
| 7 | KMIN.INP 7 | 101 |
| 8 | KMIN.INP 8 | 270 |
| 9 | KMIN.INP 9 | 9 |

**Bài 2 (7 điểm):** Test 7 bộ, mỗi bộ 7 điểm. Thời gian chạy mỗi bộ không quá 1s. Quá thời gian trên không cho điểm

|  |  |  |
| --- | --- | --- |
| Test | RECT9.INP | RECT9.OUT |
| 1 | RECT9.INP1 | RECT9.OUT1 |
| 2 | RECT9.INP2 | RECT9.OUT2 |
| 3 | RECT9.INP3 | RECT9.OUT3 |
| 4 | RECT9.INP4 | RECT9.OUT4 |
| 5 | RECT9.INP5 | RECT9.OUT5 |
| 6 | RECT9.INP6 | RECT9.OUT6 |
| 7 | RECT9.INP7 | RECT9.OUT7 |

**Bài 3 (7 điểm)**  Test 7 bộ, mỗi bộ 7 điểm. Thời gian chạy mỗi bộ không quá 1s. Quá thời gian trên không cho điểm.

|  |  |  |
| --- | --- | --- |
| Test | WORD.INP | WORD.OUT |
| 1 | WORD.INP1 | WORD.OUT1 |
| 2 | WORD.INP2 | WORD.OUT2 |
| 3 | WORD.INP3 | WORD.OUT3 |
| 4 | WORD.INP4 | WORD.OUT4 |
| 5 | WORD.INP5 | WORD.OUT5 |
| 6 | WORD.INP6 | WORD.OUT6 |
| 7 | WORD.INP7 | WORD.OUT7 |

===//==

Bài tập 36:dãy con có tông bằng s

Bài tập 37:dãy con tăng dần có tông bằng s

Bài tập 38:tàu thủy

const fi='dctang1.inp';

fo='dctang1.out';

var f,fn:text;

a:array[0..100000] of integer;

n,i,j,max,d,k:longint;

t,s:integer;

begin

assign(f,fi);reset(f);

assign(fn,fo);rewrite(fn);

read(f,n); readln(f,s);

for i:=1 to n do

read(f,a[i]);

max:=1;

for i:=1 to n-1 do

begin

t:=a[i];

d:=1;

for j:=i+1 to n do

begin

t:=t+a[j];

inc(d);

if t=s then

begin

if d>max then

begin

max:=d;

k:=i;

end;

break;

end;

end;

end;

for i:=k to k+max-1 do

write(fn,a[i],' ');

close(f);

close(fn);

end.

Bài 2:

const fi='dctang2.inp';

fo='dctang2.out';

var f,fn:text;

a:array[0..100000] of integer;

n,i,j,max,d,k:longint;

t,s:integer;

begin

assign(f,fi);reset(f);

assign(fn,fo);rewrite(fn);

read(f,n); readln(f,s);

for i:=1 to n do

read(f,a[i]);

max:=1;

for i:=1 to n-1 do

begin

t:=a[i];

d:=1;

j:=i;

while a[j]<a[j+1] do

begin

t:=t+a[j+1];

inc(d);

if t=s then

begin

if d>max then

begin

max:=d;

k:=i;

end;

break;

end;

inc(j);

end;

end;

for i:=k to k+max-1 do

write(fn,a[i],' ');

close(f);

close(fn);

end.

**Bài 3:**

const fi='tauthuy.inp';

fo='tauthuy.out';

var f,fn:text;

s,k:longint;

function ucln(x,y:longint):longint;

begin

while x<>y do

begin

if x>y then x:=x-y

else y:=y-x;

end;

ucln:=x;

end;

begin

assign(f,fi);reset(f);

assign(fn,fo);rewrite(fn);

read(f,s,k);

write(fn,(s\*k)/ucln(s,k):0:0);

close(f);

close(fn);

end.

------------------

Kiểm tra số đơn điệu

