

CHUYÊN ĐỀ  
QUY HOẠCH ĐỘNG

## MỤC LỤC

	Trang
<b>CHUYÊN ĐỀ: QUY HOẠCH ĐỘNG</b>	3
<b>I. MỘT SỐ KIẾN THỨC VỀ LẬP TRÌNH ĐỘNG</b>	3
<b>II. MỘT SỐ BÀI TOÁN QUY HOẠCH ĐỘNG ĐIỂN HÌNH</b>	4
<b>BÀI TOÁN 1: Đường đi</b>	4
<b>BÀI TOÁN 2. Cầu đá</b>	9
<b>BÀI TOÁN 3. Xe Buýt</b>	12
<b>BÀI TOÁN 4. Nối mạng máy tính</b>	15
<b>BÀI TOÁN 5. Chia kẹo</b>	17
<b>BÀI TOÁN 6. Đoàn xe qua cầu</b>	20
<b>BÀI TOÁN 7: Dây con đơn điệu tăng có độ dài lớn nhất</b>	23
<b>BÀI TOÁN 8: Đổi tiền</b>	24
<b>BÀI TOÁN 9: Dây con có tổng bằng S</b>	26
<b>BÀI TOÁN 10. Dây con có tổng lớn nhất (2)</b>	27
<b>BÀI TOÁN 11. Chọn phần thưởng (dạng bài toán tìm dãy con có tổng lớn nhất theo điều kiện không được chọn 3 phần tử liên tiếp)</b>	30
<b>BÀI TOÁN 12. Dây con chung dài nhất của hai dãy số</b> {COMSEQ.INP}	33
Bài toán 12.1: Dây con chung dài nhất (2) {DAYCON.PAS}	34
<b>BÀI TOÁN 13: SƯU TẬP TEM</b> {STAMP.pas}	36
<b>BÀI TOÁN 14. Phở</b>	38
<b>BÀI TOÁN 15. Chuyên toán tin</b>	41
<b>BÀI TOÁN 16. Đoạn đầu dài nhất</b>	45
<b>BÀI TOÁN 17. Xóa đơn điệu - k gián đoạn</b>	48
<b>BÀI TOÁN 18. Phát quà Noen</b>	51
<b>BÀI TOÁN 19. Hộp thư điện tử</b>	54
<b>BÀI TOÁN 20. Khôi phục biểu thức đúng</b>	57
<b>BÀI TOÁN 21. Quả bóng rơi</b>	59
<b>BÀI TOÁN 22. Hành tinh đâm lầy</b>	63
<b>BÀI TOÁN 23. Biến đổi xâu ký tự (khoảng cách Levenshtein)</b>	67
<b>BÀI TOÁN 24. Xâu FIBONACI</b>	70
<b>BÀI TOÁN 25. Tháp Hà Nội</b>	72
<b>BÀI TOÁN 26. Xếp lịch giảng</b>	74
<b>BÀI TOÁN 27. Kinh doanh bất động sản.</b>	76
<b>BÀI TOÁN 28. Ca nhạc</b>	78
<b>BÀI TOÁN 29. Số lớn nhất</b>	81
<b>BÀI TOÁN 30. Chuỗi đối xứng</b>	83
<b>BÀI TOÁN 31. Palindrome</b>	85
<b>BÀI TOÁN 32. Phân tích số</b>	87
<b>BÀI TOÁN 33. Điền dấu cộng, trừ thích hợp</b>	89
<b>BÀI TOÁN 34. Làm việc tập thể</b>	89
<b>BÀI TOÁN 35. Hai bên sông</b>	91

## CHUYÊN ĐỀ: QUY HOẠCH ĐỘNG

\*\*\*

### I. MỘT SỐ KIẾN THỨC VỀ LẬP TRÌNH ĐỘNG

#### 1. Phương pháp quy hoạch động

Phương pháp quy hoạch động cùng nguyên lý tối ưu được nhà toán học Mỹ R. Bellman đề xuất vào những năm 50 của thế kỷ 20. Phương pháp này đã được áp dụng để giải hàng loạt bài toán thực tế trong các quá trình kỹ thuật công nghệ, tổ chức sản xuất, kế hoạch hoá kinh tế... Tuy nhiên cần lưu ý rằng có một số bài toán mà cách giải bằng quy hoạch động tỏ ra không thích hợp.

Trong thực tế, ta thường gặp một số bài toán tối ưu loại sau: Có một đại lượng  $f$  hình thành trong một quá trình gồm nhiều giai đoạn và ta chỉ quan tâm đến kết quả cuối cùng là giá trị của  $f$  phải lớn nhất hoặc nhỏ nhất, ta gọi chung là giá trị tối ưu của  $f$ . Giá trị của  $f$  phụ thuộc vào những đại lượng xuất hiện trong bài toán mà mỗi bộ giá trị của chúng được gọi là một *trạng thái* của hệ thống và phụ thuộc vào cách thức đạt được giá trị  $f$  trong từng giai đoạn mà mỗi cách tổ chức được gọi là một *điều khiển*. Đại lượng  $f$  thường được gọi là *hàm mục tiêu* và quá trình đạt được giá trị tối ưu của  $f$  được gọi là *quá trình điều khiển tối ưu*.

Bellman phát biểu nguyên lý tối ưu (cũng gọi là *nguyên lý Bellman*) mà ý tưởng cơ bản là như sau: “Với mỗi quá trình điều khiển tối ưu, đối với trạng thái bắt đầu  $A_0$ , với trạng thái  $A$  trong quá trình đó, phần quá trình kể từ trạng thái  $A$  xem như trạng thái bắt đầu cũng là tối ưu”.

*Chú ý rằng nguyên lý này được thừa nhận mà không chứng minh.*

Phương pháp tìm điều khiển tối ưu theo nguyên lý Bellman thường được gọi là *quy hoạch động*. Thuật ngữ này nói lên thực chất của quá trình điều khiển là động: có thể trong một số bước đầu tiên lựa chọn điều khiển tối ưu dường như không tốt nhưng tựu chung cả quá trình lại là tốt nhất.

Ta có thể giải thích ý này qua bài toán sau: Cho một dãy  $N$  số nguyên  $A_1, A_2, \dots, A_N$ . Hãy tìm cách xoá đi một số ít nhất số hạng để dãy còn lại là đơn điệu hay nói cách khác hãy chọn một số nhiều nhất các số hạng sao cho dãy  $B$  gồm các số hạng đó theo trình tự xuất hiện trong dãy  $A$  là đơn điệu.

Quá trình chọn  $B$  được điều khiển qua  $N$  giai đoạn để đạt được mục tiêu là số lượng số hạng của dãy  $B$  là nhiều nhất, điều khiển ở giai đoạn  $i$  thể hiện việc chọn hay không chọn  $A_i$  vào dãy  $B$ .

Giả sử dãy đã cho là 1 8 10 2 4 6 7. Nếu ta chọn lần lượt 1, 8, 10 thì chỉ chọn được 3 số hạng nhưng nếu bỏ qua 8 và 10 thì ta chọn được 5 số hạng 1, 2, 4, 6, 7.

Khi giải một bài toán bằng cách “chia để trị” chuyển việc giải bài toán kích thước lớn về việc giải nhiều bài toán cùng kiểu có kích thước nhỏ hơn thì thuật toán này thường được thể hiện bằng các chương trình con đệ quy. Khi đó, trên thực tế, nhiều kết quả trung gian phải tính nhiều lần.

Vậy ý tưởng cơ bản của quy hoạch động thật đơn giản: tránh tính toán lại mọi thứ hai lần, mà lưu giữ kết quả đã tìm kiếm được vào một bảng làm giả thiết cho việc tìm kiếm những kết quả của trường hợp sau. Chúng ta sẽ làm đầy dần giá trị của bảng này bởi các kết quả của những trường hợp trước đã được giải. Kết quả cuối cùng chính là kết quả của bài toán cần giải. Nói cách khác phương pháp quy hoạch động đã thể hiện sức mạnh của nguyên lý chia để trị đến cao độ.

Quy hoạch động là kỹ thuật thiết kế bottom-up (từ dưới lên). Nó được bắt đầu với những trường hợp con nhỏ nhất (thường là đơn giản nhất và giải được ngay). Bằng cách tổ hợp các kết quả đã có (*không phải tính lại*) của các trường hợp con, sẽ đạt được tới kết quả của trường hợp có kích thước lớn dần lên và tổng quát hơn, cho đến khi cuối cùng đạt tới lời giải của trường hợp tổng quát nhất.

Trong một số trường hợp, khi giải một bài toán  $A$ , trước hết ta tìm họ bài toán  $A(p)$  phụ thuộc tham số  $p$  (có thể  $p$  là một véc tơ) mà  $A(p_0) = A$  với  $p_0$  là trạng thái ban đầu của bài toán  $A$ . Sau đó tìm cách giải họ bài toán  $A(p)$  với tham số  $p$  bằng cách áp dụng nguyên lý tối ưu của Bellman. Cuối cùng cho  $p = p_0$  sẽ nhận được kết quả của bài toán  $A$  ban đầu.

#### 2. Các bước thực hiện quy hoạch động

##### Bước 1: Lập hệ thức

Dựa vào nguyên lý tối ưu tìm cách chia quá trình giải bài toán thành từng giai đoạn, sau đó tìm hệ thức biểu diễn tương quan quyết định của bước đang xử lý với các bước đã xử lý trước đó.

Hoặc tìm cách phân rã bài toán thành các “bài toán con” tương tự có kích thước nhỏ hơn, tìm hệ thức nêu quan hệ giữa kết quả bài toán kích thước đã cho với kết quả của các “bài toán con” cùng kiểu có kích thước nhỏ hơn của nó nhằm xây dựng phương trình truy toán (dạng hàm hoặc thủ tục đệ quy).

Về một cách xây dựng phương trình truy toán:

Ta chia việc giải bài toán thành  $n$  giai đoạn. Mỗi giai đoạn  $i$  có trạng thái ban đầu là  $t(i)$  và chịu tác động điều khiển  $d(i)$  sẽ biến thành trạng thái tiếp theo  $t(i+1)$  của giai đoạn  $i+1$  ( $i=1,2,\dots,n-1$ ). Theo nguyên lý tối ưu của Bellman thì việc tối ưu giai đoạn cuối cùng không làm ảnh hưởng đến kết quả toàn bài toán. Với trạng thái ban đầu là  $t(n)$  sau khi làm giai đoạn  $n$  tốt nhất ta có trạng thái ban đầu của giai đoạn  $n-1$  là  $t(n-1)$  và tác động điều khiển của giai đoạn  $n-1$  là  $d(n-1)$ , có thể tiếp tục xét đến giai đoạn  $n-1$ . Sau khi tối ưu giai đoạn  $n-1$  ta lại có  $t(n-2)$  và  $d(n-2)$  và lại có thể tối ưu giai đoạn  $n-2$  ... cho đến khi các giai đoạn từ  $n$  giảm đến 1 được tối ưu thì coi như hoàn thành bài toán. Gọi giá trị tối ưu của bài toán tính đến giai đoạn  $k$  là  $F_k$  giá trị tối ưu của bài toán tính riêng ở giai đoạn  $k$  là  $G_k$  thì

$$F_k = F_{k-1} + G_k$$

$$F_1(t(k)) = \max_{\forall d(k)} \{G_k(t(k), d(k)) + F_{k-1}(t(k-1))\} \quad (*)$$

Hay là:

**Bước 2: Tổ chức dữ liệu và chương trình**

Tổ chức dữ liệu sao cho đạt các yêu cầu sau:

- Dữ liệu được tính toán dần theo các bước.
- Dữ liệu được lưu trữ để giảm lượng tính toán lặp lại.
- Kích thước miền nhớ dành cho lưu trữ dữ liệu càng nhỏ càng tốt, kiểu dữ liệu được

chọn phù hợp, nên chọn đơn giản để truy cập.

**Cụ thể**

- Các giá trị của  $F_k$  thường được lưu trữ trong một bảng (mảng một chiều hoặc hai, ba, v.v... chiều).

- Cần lưu ý khởi trị các giá trị ban đầu của bảng cho thích hợp, đó là các kết quả của các bài toán con có kích cỡ nhỏ nhất của bài toán đang giải:

$$F_1(t(1)) = \max_{\forall d(1)} \{G_1(t(1), d(1)) + F_0(t(0))\}$$

- Dựa vào công thức, phương trình truy toán (\*) và các giá trị đã có trong bảng để tìm dần các giá trị còn lại của bảng.

- Ngoài ra còn cần mảng lưu trữ nghiệm tương ứng với các giá trị tối ưu trong từng giai đoạn.

- Dựa vào bảng lưu trữ nghiệm và bảng giá trị tối ưu trong từng giai đoạn đã xây dựng, tìm ra kết quả bài toán.

**Bước 3: Làm tốt**

Làm tốt thuật toán bằng cách thu gọn hệ thức (\*) và giảm kích thước miền nhớ. Thường tìm cách dùng mảng một chiều thay cho mảng hai chiều nếu giá trị một dòng (hoặc cột) của mảng hai chiều chỉ phụ thuộc một dòng (hoặc cột) kề trước.

Trong một số trường hợp có thể thay mảng hai chiều với các giá trị phân tử chỉ nhận giá trị 0, 1 bởi mảng hai chiều mới bằng cách dùng kỹ thuật quản lý bit.

### 3. Hạn chế của quy hoạch động

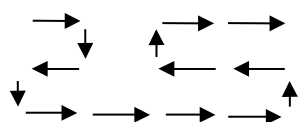
- Việc tìm công thức, phương trình truy toán hoặc tìm cách phân rã bài toán nhiều khi đòi hỏi sự phân tích tổng hợp rất công phu, dễ sai sót, khó nhận ra như thế nào là thích hợp, đòi hỏi nhiều thời gian suy nghĩ. Đồng thời không phải lúc nào kết hợp lời giải của các bài toán con cũng cho kết quả của bài toán lớn hơn.

- Khi bảng lưu trữ đòi hỏi mảng hai, ba chiều ... thì khó có thể xử lý dữ liệu với kích cỡ mỗi chiều lớn hàng trăm.

- Có những bài toán không thể giải được bằng quy hoạch động.

## II. MỘT SỐ BÀI TOÁN QUY HOẠCH ĐỘNG ĐIỂN HÌNH

### BÀI TOÁN 1: Đường đi



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

Cho bảng kích thước 3 x N ( $1 < n \leq 100$ ). Trong mỗi ô có ghi một số 1 hoặc 0. Bằng các phép di chuyển U, D, L, R người ta đi từ ô trên trái của bảng tới ô bất kỳ bên phải nhất của bảng (cột thứ n). Không được đi lại vào ô đã đi qua. Đường đi được đánh giá bằng giá trị S. Ban đầu S nhận giá trị ghi ở ô trên trái. Sau mỗi bước, giá trị của S tăng lên gấp đôi và cộng với nội dung của ô mới đến. Hãy tìm đường đi tương ứng với S lớn nhất. Hình bên trên tương ứng với N=5 và đường đi cần tìm là RDLDRRRRULLURR.

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

- Dòng đầu tiên chứa số nguyên N
- Ba dòng sau chứa 3 x N các ký tự 0,1 xác định giá trị ghi trong các ô, liệt kê từ trái sang phải, từ trên xuống dưới.

*Kết quả:* Đưa ra file BANK.OUT một số nguyên hệ thập phân là giá trị lớn nhất tìm được của S

*Ví dụ:*

BANK.INP

5

00011

01100

00011

BANK.OUT

RDLDRRRRULLURR

### CHƯƠNG TRÌNH

{ \$A+,B-,D+,E+,F-,G-,I+,L+,N+,O-,P-,Q-,R-,S+,T-,V+,X+ }  
{ \$M 16384,0,655360 }

uses crt;

const

tfi = 'BANK.INP';

tfo = 'BANK.OUT';

NN = 100;

maxN = 100;

maxV = 2\*(maxN+1);

maxS = 100;

type

SoNguyen=string[maxS];

var

a: array[1..3,1..maxN] of byte;

n: integer;

fi,fo: text;

CS,TT: array[1..maxV] of integer;

sTT: integer;

Tr: array[1..maxV] of integer;

S : array[1..maxV] of SoNguyen;

x: array[1..maxV] of integer;

slx: integer;

Max: SoNguyen;

AmMot,zero: SoNguyen;

procedure Sinhddl;

var ch: char;

i,j: integer;

begin

clrscr;

writeln('Ban co tao file ',tfi,' (C/K)?');

```
repeat ch:=readkey until upcase(ch) in ['C','K'];
if upcase(ch)='K' then exit;
randomize;
N:=NN;
assign(fi,tfi); rewrite(fi);
writeln(fi,N);
for i:=1 to 3 do
begin
for j:=1 to N do write(fi,random(2));
writeln(fi);
end;
close(fi);
end;
```

```
procedure Gapdoi(var T: SoNguyen);
var nho,i,tich: integer;
begin
nho:=0;
for i:=maxS downto 1 do
begin
tich:=(ord(T[i])-48)*2+nho;
T[i]:=chr(tich mod 10+48);
nho:=tich div 10;
end;
end;
```

```
procedure Cong(var T: SoNguyen; k: byte);
begin
if k=0 then exit;
T[maxS]:=succ(T[maxS]);
end;
```

```
function Doiso(k,i: integer): integer;
begin
Doiso:=(k-1)*(n+1)+i+1;
end;
procedure Doidinh(u: integer; var k,i: integer);
begin
k:=(u-1) div (n+1)+1;
i:=(u-1) mod (n+1);
end;
```

```
procedure Docdl;
var i,j: integer;
c: char;
begin
fillchar(a,sizeof(a),0);
assign(fi,tfi); reset(fi);
readln(fi,n);
for i:=1 to 3 do
begin
for j:=1 to n do
begin
read(fi,c);
```

```
        a[i,j]:=ord(c)-48;
    end;
    readln(fi);
end;
close(fi);
zero:="";
for i:=1 to maxS do zero:=zero+'0';
AmMot:=zero;
AmMot[1]:=Pred(AmMot[1]);
end;

procedure Loang(u: integer);
var v,k,i,j: integer;
begin
    TT[u]:=-1;
    Doidinh(u,k,i);
    for j:=i+1 to n do
        begin
            v:=Doiso(3-k,j);
            if TT[v]=0 then Loang(v);
        end;
    inc(sTT);
    TT[u]:=sTT;
    CS[2*(n+1)-sTT+1]:=u;
end;

procedure SxTopo;
var i: integer;
begin
    Fillchar(TT,sizeof(TT),0);
    Fillchar(CS,sizeof(CS),0);
    sTT:=0;
    for i:=1 to 2*(n+1) do
        if TT[i]=0 then
            Loang(i);
    end;

function Ke(v,u: integer): boolean;
var k,l,i,j: integer;
begin
    Doidinh(CS[v],k,i);
    Doidinh(CS[u],l,j);
    Ke:=(k+l=3) and (i<j);
end;

function Tong(v,u: integer): SoNguyen;
var k,l,i,j,r: integer;
    T: SoNguyen;
begin
    Doidinh(CS[v],k,i);
    Doidinh(CS[u],l,j);
    T:=S[v];
    if k=2 then k:=3;
    if l=2 then l:=3;
```

```
for r:=i+1 to j do
  Begin
    Gapdoi(T);
    Cong(T,a[k,r]);
  end;
for r:=j downto i+1 do
  begin
    GapDoi(T);
    Cong(T,a[2,r]);
  end;
for r:=i+1 to j do
  begin
    GapDoi(T);
    Cong(T,a[l,r]);
  end;
Tong:=T;
end;
```

```
procedure Tinh;
var i,u,v: integer;
    T: SoNguyen;
begin
  fillchar(Tr,sizeof(Tr),0);
  i:=0;
  repeat inc(i) until Cs[i]=1;
  Tr[i]:=-1; S[i]:=zero;
  for u:=i+1 to 2*(n+1) do
    begin
      Tr[u]:=-1;
      S[u]:=AmMot;
      for v:=u-1 downto i do
        if Ke(v,u) then
          begin
            T:=Tong(v,u);
            if T>S[u] then
              begin
                S[u]:=T;
                Tr[u]:=v;
              end;
          end;
      end;
    end;
end;
```

```
procedure TimDuong;
var i,u,v,kt: integer;

begin
  u:=Doiso(1,n);
  v:=Doiso(2,n);

  for i:=1 to 2*(n+1) do
    if (Cs[i]=u) then
      begin
        Max:=S[i];
```



```
    kt:=i;
    break;
end;

for i:=1 to 2*(n+1) do
  if (cs[i]=v) then
    begin
      if Max<S[i] then
        begin
          Max:=S[i];
          Kt:=i;
        end;
      break;
    end;
end;

slx:=0;
u:=kt;
repeat
  inc(slx);
  x[slx]:=Cs[u];
  u:=tr[u];
until u=-1;
end;

procedure Indoan(u,v: integer);
var r,k,i,l,j: integer;
begin
  Doidinh(u,k,i);
  Doidinh(v,l,j);
  for r:=i+1 to j-1 do write(fo,'R');
  if k=1 then write(fo,'D') else write(fo,'U');
  for r:=j-1 downto i+1 do write(fo,'L');
  if k=1 then write(fo,'D') else write(fo,'U');
  for r:=i+1 to j-1 do write(fo,'R');
end;

procedure Inkq;
var i,u,v: integer;
begin
  assign(fo,tfo); rewrite(fo);
  for i:=slx downto 2 do
    begin
      u:=x[i]; v:=x[i-1];
      Indoan(u,v);
      if i>2 then write(fo,'R');
    end;
  close(fo);
end;

BEGIN
  {Sinhdl; }
  Docdl;
  SxTopo;
  Tinh;
```

TimDuong;  
Inkq  
END.

## BÀI TOÁN 2. Cầu đá

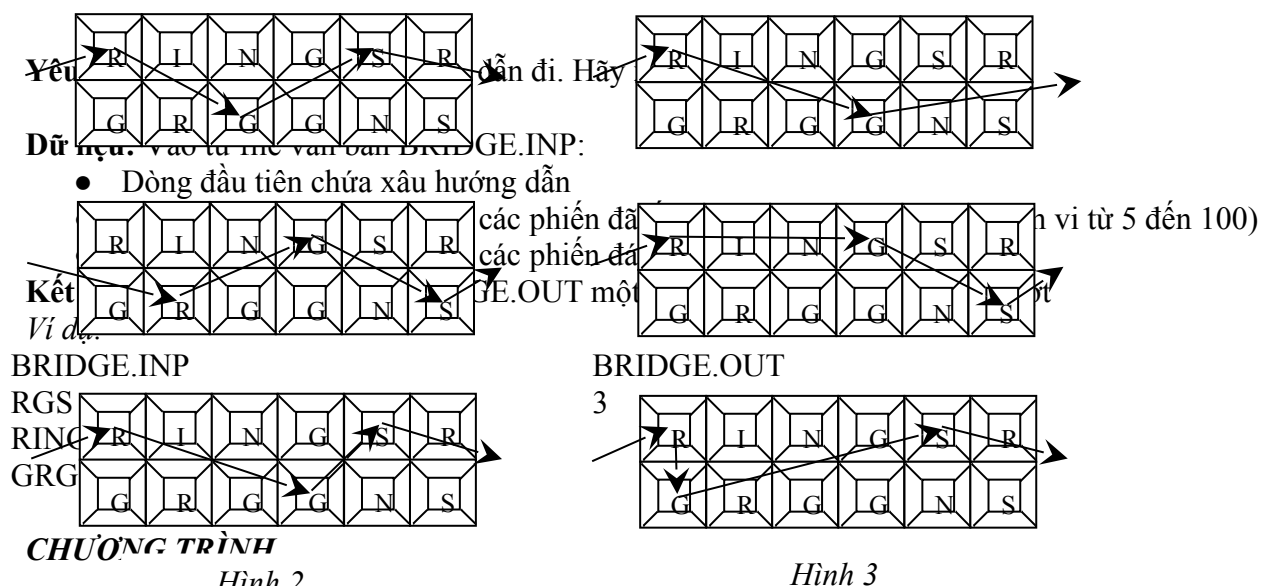
Trên đường đi khảo sát đoàn thám hiểm phải vượt qua một cây cầu đá bắc qua miệng núi lửa. Cây cầu gồm 2 dãy phiến đá - dãy Ác quỷ và dãy Thiên thần.

Hình 1 nêu ví dụ về một cầu độ dài 6. Dãy trên là dãy phiến đá Ác quỷ và dãy dưới là dãy phiến đá thiên thần. Hai dãy phiến đá có độ dài như nhau. Từng cặp phiến đá Ác quỷ và Thiên thần được gắn với nhau tạo thành một tấm lát, tương ứng với 1 cột trên hình vẽ. Trên mỗi phiến đá có khắc một ký tự tổng tập {R, I, N, G, S}

Đoàn thám hiểm phải đi từ bờ trái sang bờ phải. Để vượt cầu, cần phải lần lượt dẫm lên các tầng đá theo một hướng dẫn cho trước dưới dạng một xâu ký tự. Nếu đi sai, cầu sẽ sập xuống núi lửa phía dưới. Ngoài ra phải bảo đảm các qui tắc sau:

- Các ký tự trên những phiến đá được dẫm ghi theo trình tự bước tạo thành xâu hướng dẫn.
- Phải bước đan xen giữa các phiến đá Ác quỷ và Thiên thần bắt đầu từ loại nào - không quan trọng.
- Luôn tiến sang các tấm lát bên phải, độ lớn của mỗi bước đi (số tấm lát giữa hai lần bước liên tiếp) là không quan trọng.

Ví dụ, với cầu ở hình 1 và hướng dẫn 'RGS' hình 2, nêu các cách vượt cầu và hình 3 nêu các cách vượt cầu sai:



```
const
  tfi='BRIDGE.INP';
  tfo='BRIDGE.OUT';

type
  xaukt=string[101];

var
  fi, fo: text;
  mau: xaukt;
  s: array[1..2] of xaukt;
  M,N: integer;
  a: array[0..100,0..100,1..2] of integer;
  kq: integer;
```

```
procedure Docdl;
begin
  assign(fi,tfi); reset(fi);
  readln(fi,mau);
  readln(fi,s[1]);
  readln(fi,s[2]);
  close(fi);
end;

procedure XDB;
var l,i,j: integer;
begin
  m:=length(mau);
  n:=length(s[1]);
  fillchar(a,sizeof(a),0);
  for i:=0 to m do
    begin
      a[i,0,1]:=1;
      a[i,0,2]:=1;
    end;
  for j:=1 to N do
    for i:=j downto 1 do
      begin
        if s[1][j]=mau[i] then
          for l:=j-1 downto i-1 do a[i,j,1]:=a[i,j,1]+a[i-1,l,2];
        if s[2][j]=mau[i] then
          for l:=j-1 downto i-1 do a[i,j,2]:=a[i,j,2]+a[i-1,l,2];
      end;
    end;
end;

procedure XLB;
var i: integer;
begin
  kq:=0;
  for i:=m to n do
    kq:=kq+a[m,i,1]+a[m,i,2];
end;

procedure Inkq;
begin
  assign(fo,tfo); rewrite(fo);
  writeln(fo,kq);
  close(fo);
end;

BEGIN
  Docdl;
  XDB;
  XLB;
  Inkq;
END.
```

### **BÀI TOÁN 3. Xe Buýt**

Một xe buýt của công ty có nhiệm vụ đón nhân viên đến trụ sở làm việc. Trên hành trình xe buýt sẽ tiếp nhận nhân viên đứng chờ ở các điểm hẹn nếu như xe còn chỗ trống. Xe buýt có thể sẽ đỗ lại để chờ những công nhân còn chưa kịp đến điểm hẹn. Cho biết thời điểm mà mỗi nhân viên đến điểm hẹn của mình và thời điểm qua mỗi điểm hẹn của xe buýt. Giải thiết rằng xe buýt đến điểm hẹn đầu tiên tại thời điểm 0, thời gian xếp khách lên xe coi như bằng 0. Hãy xác định khoảng thời gian ngắn nhất để xe buýt có thể chở một số lượng các nhân viên đến trụ sở làm việc lớn nhất có thể được.

Dữ liệu vào từ file BUS.DAT:

- Dòng đầu tiên chứa hai số nguyên dương N và M theo thứ tự là số điểm hẹn và số chỗ ngồi của xe buýt.
- Dòng thứ i trong số N dòng tiếp theo chứa số nguyên  $t_i$  là thời gian cần thiết để xe buýt di chuyển từ điểm hẹn i đến điểm hẹn i+1 (điểm hẹn thứ N+1 sẽ là trụ sở làm việc của công ty), số nguyên K là số lượng nhân viên đến điểm hẹn i và tiếp đến là K số nguyên là các thời điểm đến điểm hẹn của K nhân viên.

Giới hạn:  $1 \leq M \leq 2000$ ,  $1 \leq N \leq 200000$

Kết quả ghi ra file văn bản BUS.OUT thời gian ngắn nhất tìm được.

Ví dụ:

BUS.DAT

3 5

1 2 0 1

1 1 2

1 4 0 2 3 4

BUS.OUT

4

{ \$r- }

const

fi = 'bus.da0';

fo = 'bus.ou0';

nmax = 200000;

var

f : text;

sl : array [0..10000] of real;

n, max, t : longint;

m : real;

procedure open\_file;

begin

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

readln(f, n, m); {n: so diem hen, m: so cho ngoi}

fillchar(sl, sizeof(sl), 0);

end;

function getmax(i, j : longint) : longint;

begin

if i > j then getmax := i

else getmax := j;

end;

procedure solve;

var ii, j, i, time, k, cs : longint;

begin

t := 0; max := 0;

```
for ii := 1 to n do
begin
  read(f, time); {thời gian xe đi từ điểm ii đến điểm ii+1}
  read(f, k);    {số lượng công nhân đến điểm hen ii}

  for i := 1 to k do
  begin
    read(f, j); {thời điểm đến điểm hen của công nhân i}

    {gia sư lui điểm hen về ben góc:}
    cs := getmax(0, j - t); {nếu j-t<0 thì coi như có 1 người lên xe
    tại ben góc, lúc thời điểm 0}
    { Nhưng người đến chậm, ở to phải cho
    nên coi như đôi ở ben góc vào lúc thời điểm j-t trước giờ xe chạy}

    sl[cs] := sl[cs] + 1;    {tăng thêm 1 người đôi ở ben góc,
    vào lúc cs trước giờ xe chạy,
    người này tất nhiên lên được xe nếu số người đôi <M
    sl[cs] là số người }

    if cs > max then max := cs; {max: thời gian cho đôi tối đa}
  end;

  t := t + time; {tổng thời gian ở to chuyển động}
end;
end;

procedure print;
var
  i : longint;
  kq : real;

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

  kq := sl[0]; {số người lên xe ngay}
  i := 0;
  while (kq < m) and (i < max) do
  begin
    inc(i); {i: khoảng thời gian đôi, tại ben góc trước giờ xe chạy}
    kq := kq + sl[i]; {cộng thêm người chờ ở ben góc theo thứ tự thời
    gian từ 0 -> max (thời gian cho đôi tối đa)}

    t := t + 1; {tăng thêm 1 đơn vị thời gian}
  end;

  writeln(f, t);

  close(f);
end;

begin
  open_file;
  solve;
```

```
    print;
end.
cách 2
{ SOLVING PROBLEM BUS }
```

```
uses crt;
const
    fi  = 'BUS.da0';
    fo  = 'BUS.ou0';
    maxM = 2000;
    VoCung = 10000000000;
type
    mang1 = array[0..maxM+1] of longint;
var
    A      : mang1;
    M, N   : integer;
    ThoiGian: longint;
    inp, out: text;

function Max(a, b: longint): longint;
begin
    if a > b then max:= a else max:= b;
end;

procedure Xuly(TG: longint);
var
    i, j, t: longint;
begin
    i:= M-1;
    while (i >= 0) and (A[i+1] > tg) do
    begin
        t:= Max(A[i], TG);
        if A[i+1] > t then A[i+1]:= t;
        dec(i);
    end;
end;

procedure Solve;
var
    i, j, SL, Time : longint;
begin
    assign(inp, fi); reset(inp);
    readln(inp, N, M);
    fillchar(A, sizeof(A), 0);
    for i:=1 to M do a[i]:= VoCung;
    for i:=1 to N do
    begin
        read(inp, ThoiGian);
        read(inp, SL);
        for j:=1 to SL do
        begin
            read(inp, Time);
            Xuly(Time);
        end;
    end;
end;
```

```

    for j:=0 to M do inc(A[j], ThoiGian);
end;
close(inp);

{ In kết quả }
assign(out, fo); rewrite(out);
writeln(out, A[M]);
close(out);
end;

BEGIN
    Solve;
end.

```

#### BÀI TOÁN 4. Nối mạng máy tính

Các học sinh khi đến thực tập trong phòng máy tính thông hay chơi trò chơi điện tử trên mạng. Để ngăn ngừa, người trực phòng máy đã ngắt tất cả các máy tính ra khỏi mạng và xếp chúng thành một dãy trên một cái bàn dài và gắn chặt máy xuống mặt bàn rồi đánh số thứ tự các máy từ 1 đến N theo chiều từ trái sang phải. Các học sinh tinh nghịch không chịu thua, họ đã quyết định tìm cách nối các máy trên bàn bởi các đoạn dây nối sao cho *mỗi máy được nối với ít nhất một máy khác*. Để tiến hành công việc này, họ đã đo khoảng cách giữa hai máy liên tiếp. Bạn hãy giúp các học sinh này tìm cách nối mạng thỏa mãn yêu cầu đặt ra sao cho tổng độ dài cáp nối phải sử dụng là ít nhất.

**Dữ liệu:** vào từ file văn bản CABLE.INP:

- Dòng đầu tiên chứa số lượng máy N ( $1 \leq N \leq 25000$ )
- Dòng thứ i trong số N-1 dòng tiếp theo chứa các khoảng cách từ máy i đến máy i+1 ( $i=1,2,...,N-1$ ). Giả thiết rằng khoảng cách từ máy 1 đến máy N không vượt quá  $10^6$ .

**Kết quả:** Ghi ra file văn bản CABLE.OUT độ dài của cáp nối cần sử dụng.

Ví dụ:

CABLE.INP	CABLE.OUT
6	7
2	
2	
3	
2	
2	

```

{$A+,B-,D+,E+,F-,G-,I+,L+,N-,O-,P-,Q+,R+,S+,T-,V+,X+}
{$M 16384,0,655360}

```

```
uses crt;
```

```

const
    tfi='CABLE.INP';
    tfo='CABLE.OUT';
    NN=25000;

```

```

type
    mang=array[1..13000] of LongInt;

```

```
var
```

```
fi,fo: text;
N: integer;
a: array[1..2] of ^mang;
S: array[1..2] of ^mang;

procedure SinhDL;
var ch: char;
    i: integer;
    u: word;
begin
    clrscr;
    writeln('Ban co tao file ',tfi,' (C/K)?');
    repeat ch:=readkey until upcase(ch) in ['C','K'];
    if upcase(ch)='K' then exit;
    randomize;
    N:=NN;
    assign(fi,tfi); rewrite(fi);
    writeln(fi,N);
    for i:=1 to N-1 do
        begin
            u:=65000;
            u:=random(u)+100;
            writeln(fi,u);
        end;
    close(fi);
end;

procedure CapPhat;
var i: integer;
begin
    for i:=1 to 2 do new(a[i]);
    for i:=1 to 2 do new(s[i]);
end;

procedure DoiDinh(i: integer; var u,v: integer);
begin
    u:=(i-1) div 13000+1;
    v:=(i-1) mod 13000+1;
end;

procedure Docdl;
var i,u,v: integer;
begin
    assign(fi,tfi); reset(fi);
    readln(fi,N);
    for i:=1 to N-1 do
        begin
            DoiDinh(i,u,v);
            readln(fi,a[u]^v);
        end;
    close(fi);
end;

procedure XDB;
```



```

var i,u,v,u1,v1,u2,v2: integer;
begin
  s[1]^1:=0;
  s[1]^2:=a[1]^1;
  s[1]^3:=a[1]^1+a[1]^2;
  for i:=4 to N do
    begin
      DoiDinh(i,u,v);
      DoiDinh(i-1,u1,v1);
      DoiDinh(i-2,u2,v2);
      s[u]^v:=s[u1]^v1+a[u1]^v1;
      if s[u]^v>s[u2]^v2+a[u1]^v1 then
        s[u]^v:=s[u2]^v2+a[u1]^v1;
      end;
    end;
  end;

procedure Inkq;
var u,v: integer;
begin
  assign(fo,tfo); rewrite(fo);
  DoiDinh(n,u,v);
  writeln(fo,s[u]^v);
  close(fo);
end;

BEGIN
  CapPhat;
  {SinhDL; }
  Docdl;
  XDB;
  Inkq;
END.

```

## BÀI TOÁN 5. Chia kẹo

Có N gói kẹo, gói thứ i có A[i] cái kẹo.

**Yêu cầu:** Hãy tìm cách chia các gói kẹo này thành 2 phần sao cho độ chênh lệch giữa số kẹo ở hai phần là ít nhất có thể được.  $0 < A[i] < 10000$ ,  $2 \leq N \leq 1000$

**Dữ liệu vào:** Cho trong file CANDY.INP: Gồm N dòng, dòng thứ i chứa số nguyên A[i] là số kẹo trong gói thứ i

**Kết quả:** Ghi ra file CANDY.OUT:

- Dòng đầu tiên ghi 3 số: Tổng số kẹo ở phần 1, phần 2 và độ chênh lệch giữa hai phần
- Dòng 2, 3 là số hiệu các gói kẹo ở mỗi phần được chia.

*Ví dụ:*

CANDY.INP	CANDY.OUT
3	14 12 2
4	3 4 7
7	12
12	

```

{$A+,B-,D+,E+,F-,G-,I+,L+,N+,O-,P-,Q+,R+,S+,T-,V+,X+}
{$M 16384,0,655360}

```

const

```
tfi='CANDY.INP';
tfo='CANDY.OUT';
maxN=1000;

var
  fi,fo: text;
  N: longint;
  a: array[1..maxN] of longint;
  S: array[1..2] of longint;
  x: array[1..maxN] of longint;
  Delta: longint;

procedure Docdl;
begin
  assign(fi,tfi); reset(fi);
  N:=0;
  while not seekeof(fi) do
    begin
      inc(N);
      read(fi,a[N]);
    end;
  close(fi);
end;

procedure KhoiDau;
var i: longint;
begin
  S[1]:=0;
  for i:=1 to N do
    begin
      x[i]:=1;
      S[1]:=S[1]+a[i];
    end;
  S[2]:=0;
  Delta:=S[1];
end;

function Tim(i: longint): longint;
var k: longint;
begin
  for k:=1 to N do
    if (x[k]=i) then
      if Delta>abs(S[i]-S[3-i]-2*a[k]) then
        begin
          Tim:=k;
          exit;
        end;
  Tim:=0;
end;

procedure Chuyen(i,k: longint);
begin
  Delta:=abs(S[i]-S[3-i]-2*a[k]);
  x[k]:=3-i;
```

```
S[i]:=S[i]-a[k];  
S[3-i]:=S[3-i]+a[k];  
end;
```

```
procedure TinhDoi(var u,v: longint; var ok: boolean);  
var i,j: longint;  
begin  
  ok:=true;  
  for i:=1 to N do  
    if x[i]=1 then  
      for j:=1 to N do  
        if x[j]=2 then  
          if Delta>abs(S[1]-S[2]-2*(a[i]-a[j])) then  
            begin  
              u:=i;  
              v:=j;  
              exit;  
            end;  
          ok:=false;  
        end;  
      end;  
end;
```

```
procedure Chia;  
var ok,stop: boolean;  
    i,j,k: longint;  
begin  
  KhoiDau;  
  stop:=false;  
  while not stop do  
    begin  
      stop:=true;  
      for i:=1 to 2 do  
        begin  
          k:=Tim(i);  
          if k>0 then  
            begin  
              Chuyen(i,k);  
              Stop:=false;  
              break;  
            end;  
          end;  
        if stop then  
          begin  
            TinhDoi(i,j,ok);  
            if ok then  
              begin  
                Chuyen(1,i);  
                Chuyen(2,j);  
                stop:=false;  
              end;  
            end;  
          end;  
        end;  
      end;  
end;
```

```
procedure Inkq;
```

```

var i: longint;
begin
  assign(fo,tfo); rewrite(fo);
  writeln(fo,S[1],',',S[2],',',Delta);
  for i:=1 to N do
    if x[i]=1 then write(fo,i,' ');
  writeln(fo);
  for i:=1 to N do
    if x[i]=2 then write(fo,i,' ');
  close(fo);
end;

```

BEGIN

Docdl;

Chia;

Inkq;

END.

### BÀI TOÁN 6. Đoàn xe qua cầu

Cho một đoàn xe gồm  $n$  chiếc đi trên một đường một chiều và đoàn xe đã được bố trí theo thứ tự từ 1 đến  $n$ . Mỗi một xe trong đoàn có vận tốc là  $v_i$  và trọng lượng  $w_i$ .

Khi đi qua một chiếc cầu có trọng tải giới hạn là  $P$  thì đoàn xe phải chia thành các nhóm sao cho tổng trọng lượng của mỗi nhóm không quá  $P$  (Lưu ý rằng không được đảo thứ tự đoàn xe). Các nhóm phải đi tuần tự có nghĩa là nhóm thứ  $i$  chỉ được khởi hành khi mà toàn bộ xe của nhóm thứ  $i - 1$  đã qua cầu. Giả thiết rằng  $P > w_i$  với  $1 \leq i \leq n$ .

Rõ ràng khi đó thời gian để một nhóm xe qua cầu phụ thuộc vào xe chậm nhất trong nhóm đó nếu coi như chiều dài cũng như khoảng cách của các xe là không đáng kể.

**Hãy tìm cách chia đoàn xe thành các nhóm sao cho thời gian mà đoàn xe sang được cầu là nhỏ nhất có thể được.**

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

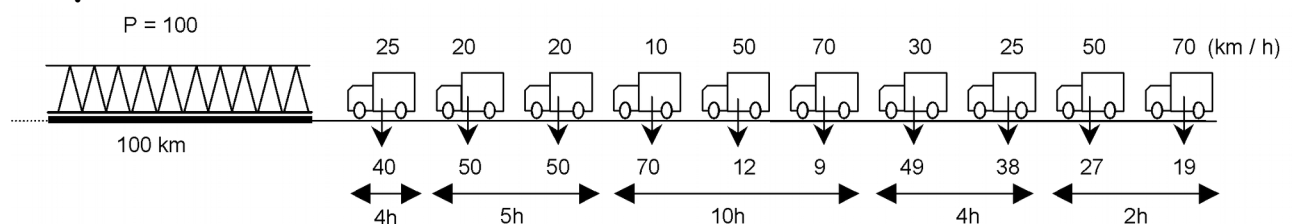
- Dòng đầu là 3 số nguyên dương  $n$ ,  $P$  và  $L$  ( $n, P, L \leq 1000$ ) thể hiện cho số xe, trọng lượng giới hạn của cầu và độ dài của cầu.
- Dòng thứ  $i$  trong  $n$  dòng kế tiếp gồm 2 số nguyên dương  $w_i$  và  $v_i$  ( $w_i, v_i \leq 100$ )

**Kết quả: Ghi ra file văn bản CARGROUP.OUT**

- Dòng đầu ghi một số thực là tổng thời gian nhỏ nhất để xe qua cầu, cho phép làm tròn lấy 2 chữ số sau dấu chấm thập phân.
- Dòng kế tiếp gồm các số  $x_1, x_2, \dots, x_k$  thể hiện: nhóm 1 gồm các xe từ 1 đến xe thứ  $x_1$ , nhóm 2 gồm các xe thứ  $x_1+1$  đến xe thứ  $x_2$ , ..., nhóm  $k$  từ xe thứ  $x_{k-1}+1$  tới  $x_k$

**Các số trên một dòng của Input / Output file ghi cách nhau ít nhất một dấu cách.**

**Ví dụ:**



CARGROUP.INP

CARGROUP.OUT

**10 100 100**  
**40 25**  
**50 20**  
**50 20**  
**70 10**  
**12 50**  
**09 70**  
**49 30**  
**38 25**  
**27 50**  
**19 70**

**25.00**  
**1 3 6 8 10**

```
uses crt;
const
  InputFile = 'CARGROUP.IN';
  OutputFile = 'CARGROUP.OU';
  max = 1000;
  maxReal = 1E10;
var
  T: array[1..max] of Real;
  W: array[1..max] of Byte;
  n, P, diem: Integer;
  F: array[0..max] of Real;
  EndofPrevGroup: array[1..max] of Integer;
  c : Char;

procedure Enter;
var
  fi: Text;
  L, i, v: Integer;
begin
  Assign(fi, InputFile + c); Reset(fi);
  Readln(fi, n, P, L);
  for i := 1 to n do
    begin
      Readln(fi, w[i], v);
      t[i] := L / v;
    end;
  Close(fi);
end;

procedure Optimize;
var
  i, j, k, TotalW: Integer;
  MaxTime: Real;
begin
  F[0] := 0;
  for i := 1 to n do
    begin
      F[i] := maxReal; TotalW := 0; MaxTime := 0;
      for j := i downto 1 do
        begin
          TotalW := TotalW + W[j];
          if TotalW > P then Break;
```

```
    if MaxTime < T[j] then MaxTime := T[j];
    if F[i] > F[j - 1] + MaxTime then
        begin
            F[i] := F[j - 1] + MaxTime;
            EndofPrevGroup[i] := j - 1;
        end;
    end;
end;

procedure Display;
begin
    clrscr;
    writeln('          CHAM BAI CARGROUP');
    writeln('=====');
    writeln; writeln; writeln;
    writeln(' TEST : CARGROUP.IN'+c);
    writeln('*****');
end;

procedure error(s : string);
begin
    writeln(#7, s);
    readln;
end;

procedure Check;
var
    g : text;
    res : real;
begin
    assign(g, outputfile+c); reset(g);
    readln(g, res);
    writeln(' Ket qua cua DA : ', f[n] : 0 : 2);
    writeln(' Ket qua cua HS : ', res : 0 : 2);
    writeln('-----');
    if trunc(res - f[n]) > 1 then
        begin
            error(' Diem cua test : 0!');
            close(g);
            exit;
        end;
    inc(diem);
    writeln(' Diem cua test : 1 !');
    readln;
    close(g);
end;

begin
    diem := 0;
    for c := '0' to '9' do
        begin
            Display;
            Enter;
```

```

    Optimize;
    Check;
end;
writeln('=====');
writeln(' Tong so diem : ', diem);
readln;
end.

```

### BÀI TOÁN 7: Dãy con đơn điệu tăng có độ dài lớn nhất

Cho dãy số nguyên  $A(N)$  ( $n \leq 1000$ ,  $-100000 \leq a_i \leq 10000$ ). Một dãy con của  $A$  là một cách chọn ra trong  $A$  một số phần tử giữ nguyên thứ tự. Như vậy  $A$  có  $2^n$  dãy con.

**Yêu cầu:** Tìm dãy con đơn điệu tăng của  $A$  có độ dài lớn nhất.

(Chú ý: đ/n dãy tăng trong bài này khác bài toán 12)

Bổ sung vào  $A$  hai phần tử:  $a_0 = -\infty$  và  $a_{n+1} = +\infty$ . Khi đó dãy con đơn điệu tăng dài nhất chắc chắn sẽ bắt đầu từ  $a_0$  và kết thúc ở  $a_{n+1}$ .

Với  $\forall i: 0 \leq i \leq n+1$ . Ta sẽ tính  $L[i]$  = độ dài dãy con đơn điệu tăng dài nhất bắt đầu tại  $a_i$ .

#### 1. Cơ sở quy hoạch động (bài toán nhỏ nhất):

$L[n+1]$  = Độ dài dãy con đơn điệu tăng dài nhất bắt đầu tại  $a_{n+1} = +\infty$ . Dãy con này chỉ gồm mỗi một phần tử ( $+\infty$ ) nên  $L[n+1]=1$ .

#### 2. Công thức truy hồi:

Giả sử với  $i$  từ  $n$  đến  $0$ , ta cần tính  $L[i]$ : độ dài dãy con tăng dài nhất bắt đầu tại  $a_i$ .

$L[i]$  được tính trong điều kiện  $L[i+1]$ ,  $L[i+2]$ , ...,  $L[n+1]$  đã biết:

Dãy con đơn điệu tăng dài nhất bắt đầu từ  $a_i$  sẽ được thành lập bằng cách lấy  $a_i$  ghép vào đầu một trong số những dãy con đơn điệu tăng dài nhất bắt đầu tại vị trí  $a_j$  đứng sau  $a_i$ . Ta sẽ chọn dãy nào để ghép  $a_i$  vào đầu? Tất nhiên là chỉ được ghép  $a_i$  vào đầu những dãy con bắt đầu tại  $a_j$  nào đó lớn hơn  $a_i$  (để đảm bảo tính tăng) và dĩ nhiên ta sẽ chọn dãy dài nhất để ghép  $a_i$  vào đầu (để đảm bảo tính dài nhất). Vậy  $L[i]$  được tính như sau:

Xét tất cả các chỉ số  $j$  trong khoảng từ  $i+1$  đến  $n+1$  mà  $a_j > a_i$ , chọn ra chỉ số  $j_{\max}$  có  $L[j_{\max}]$  lớn nhất. Đặt  $L[i] := L[j_{\max}] + 1$ .

#### 3. Truy vết

Tại bước xây dựng dãy  $L$ , mỗi khi tính  $L[i] = L[j_{\max}] + 1$ , ta đặt  $T[i] = j_{\max}$ . Để lưu lại rằng: Dãy con dài nhất bắt đầu tại  $a_i$  sẽ có phần tử thứ hai kế tiếp là  $a_{j_{\max}}$ .

Sau khi tính xong hay dãy  $L$  và  $T$ , ta bắt đầu từ  $0$ .

$T[0]$  là phần tử đầu tiên được chọn,

$T[T[0]]$  là phần tử thứ hai được chọn,

$T[T[T[0]]]$  là phần tử thứ ba được chọn ... Quá trình truy vết có thể diễn tả như sau:

$i := T[0];$

while  $i < n+1$  do {Chùng nào chưa duyệt đến số  $a_{n+1}=+\infty$  ở cuối}

begin

<Thông báo chọn  $a_i$ >

$i := T[i];$

end;

program LongestSubSequence;

const

max = 10000;

var

a, L, T: array[0..max + 1] of Integer;

n: Word;

procedure Enter; {Nhập dữ liệu từ thiết bị nhập chuẩn theo đúng khuôn dạng Input}

var

```

i: Word;
begin
  ReadLn(n);
  for i := 1 to n do
    begin
      write('A[', i, ']=');
      Read(a[i]);
    end;
end;

procedure Optimize; {Quy hoạch động}
var
i, j, jmax: Word;
begin
  a[0] := -32768; a[n + 1] := 32767; {Thêm hai phần tử canh hai đầu dãy a}
  L[n + 1] := 1; {Điền cơ sở quy hoạch động vào bảng phương án}
  for i := n downto 0 do {Tính bảng phương án}
    begin
      {Chọn trong các chỉ số j đứng sau i thoả mãn  $a_j > a_i$  ra chỉ số jmax có L[jmax]
      lớn nhất}
      jmax := n + 1;
      for j := i + 1 to n + 1 do
        if (a[j] > a[i]) and (L[j] > L[jmax]) then jmax := j;
      L[i] := L[jmax] + 1; {Lưu độ dài dãy con tăng dài nhất bắt đầu tại ai}
      T[i] := jmax; {Lưu vết: phần tử đứng liền sau ai trong dãy con tăng dài
      nhất đó là ajmax}
    end;
  Writeln('Length of result: ', L[0]-2); {chiều dài dãy con tăng dài nhất}
  i := T[0];
  while i <> n + 1 do
    begin
      writeln('a[ ', i, ']= ', a[i]);
      i := T[i];
    end;
end;
BEGIN
  Enter;
  Optimize;
END.

```

## BÀI TOÁN 8: Đổi tiền

Có N loại tiền mệnh giá  $m_i$  và số lượng  $s_i$ ,  $i = 1..N$ . Xác định số lượng mỗi loại để có thể trả lại V đồng.

Dữ liệu vào: tệp văn bản **TRATIEN.INP**

Dòng đầu tiên: hai số tự nhiên N và V,  $2 \leq N \leq 15$ .

Dòng thứ hai: N số tự nhiên  $m_1, m_2, \dots, m_N$ .

Dòng thứ ba: N số tự nhiên  $s_1, s_2, \dots, s_N$ .

Dữ liệu ra: tệp văn bản **TRATIEN.OUT**

N số tự nhiên  $c_1, c_2, \dots, c_N$  thể hiện số lượng tờ tiền mỗi loại cần trả,  $c_1 m_1 + c_2 m_2 + \dots + c_N m_N = V$ .  
Nếu vô nghiệm: ghi số 0.

Trong các tệp \*.INP và \*.OUT các số trên cùng dòng cách nhau qua dấu cách.

TRATIEN.INP	TRATIEN.OUT
-------------	-------------



6 156 1 2 5 10 20 50 4 7 2 3 6 2	0 3 0 0 5 1
--	-------------

### Thuật toán

Đây là loại toán Ba lô với dữ liệu nhỏ vì trong thực tế số mệnh giá không nhiều, thí dụ, tiền Việt chỉ có các loại sau đây là thông dụng 100, 200, 500, 1.000, 2.000, 5.000, 10.000, 20.000, 50.000, 100.000, 200.000, 500.000. Nếu tính theo đơn vị 100 đồng thì ta có thể viết lại dãy trên cho gọn hơn như sau:

1, 2, 5, 10, 20, 50, 100, 200, 500, 1.000, 2.000, 5.000.

Ta duyệt các tổ hợp số tờ tiền phải trả cho mỗi loại mệnh giá, cận dưới là 0 cận trên là  $\min(s_i, v \div m_i)$  vì để trả lại  $v$  đồng bằng loại mệnh giá  $m_i$  ta dùng tối đa  $(v \div m_i)$  tờ.

Độ phức tạp:  $(b_1 - a_1 + 1)(b_2 - a_2 + 1) \dots (b_v - a_v + 1)$ ,  $v = M + N$ .

Chú ý: Sau này ta sẽ xây dựng thuật toán tốt hơn cho bài toán trả tiền. Thuật toán này dựa trên một số kiến thức số học.

*program TraTien;*

*uses crt;*

*const mn = 20; bl = #32; nl = #13#10;*

*fn = 'TRATIEN.INP'; gn = 'TRATIEN.OUT';*

*type ml = array[0..mn] of integer;*

*(\*-----*

*n – số lượng các loại tiền*

*v – số tiền cần trả lại*

*vt – giá trị tạm thời*

*m[1..n] – các mệnh giá*

*s[1..n] – số lượng tờ tiền*

*c[1..n] – số lượng cần chọn*

*-----\*)*

*var n,v,vt: integer;*

*m,s,c: ml;*

*f,g: text;*

*procedure Doc;*

*var i: integer;*

*begin*

*assign(f,fn); reset(f); readln(f,n,v);*

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

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

*close(f);*

*end;*

*function Min(a,b: integer): integer;*

*begin*

*if a <= b then min := a else min := b;*

*end;*

*function Next: Boolean;*

*var i: integer;*

```
begin
  Next := false;
  i := n;
  while (c[i] = s[i]) do
    begin
      vt := vt - c[i] * m[i];
      c[i] := 0;
      i := i - 1;
    end;
  if (i = 0) then exit;
  c[i] := c[i] + 1;
  vt := vt + m[i];
  Next := true;
end;
function Duyet: Boolean;
  var i: integer;
begin
  { Khoi tri }
  for i := 1 to n do
    begin
      s[i] := min(s[i], v div m[i]);
      c[i] := 0;
    end;
  c[0] := -1; vt := 0; { tong gia tri cua 1 phuong an }
  Duyet := true;
  repeat
    if (vt = v) then exit;
  until not Next;
  Duyet := false;
end;
procedure Run;
  var i: integer;
begin
  Doc; assign(g, gn); rewrite(g);
  if (Duyet) then
    for i := 1 to n do write(g, c[i], bl);
  else writeln(g, 0);
  close(g);
end;
BEGIN
  Run;
END.
```

### **BÀI TOÁN 9: Dãy con có tổng bằng S**

Cho N số nguyên dương tạo thành dãy  $A = \{A_1, \dots, A_N\}$ . Tìm một dãy con của A có tổng các phần tử bằng S.

Dữ liệu vào từ file DAYCON.INP

Dòng đầu tiên ghi hai số nguyên dương N ( $0 \leq N \leq 200$ ) và S ( $0 \leq S \leq 40000$ )

Các dòng tiếp theo lần lượt ghi N số hạng của dãy A ( $0 \leq A_i \leq 200$ )

Kết quả ra ghi ra file DAYCON.OUT

Nếu bài toán vô nghiệm ghi số 0.

Nếu bài toán có nghiệm thì trên dòng thứ nhất ghi số 1. Các dòng tiếp theo, mỗi dòng ghi hai số là chỉ số trong dãy A và giá trị của một phần tử được chọn.

(xem chuyên đề bồi dưỡng HSG tin học THPT: Bài tập quy hoạch động, page 30)

\* Nếu bài toán yêu cầu in ra tất cả phương án thì ta có thể sử dụng đoạn chương trình đệ quy dưới đây:

Const

MAX = 100;

Var

a: array [1..MAX] of word;

chon: array [1..MAX] of boolean; {giữ trạng thái được chọn hay không của các phần tử}

N, K, S: word;

procedure KhoiTao;

Var i: word;

Begin

{ Nhập mảng a gồm N phần tử }// hs tự viết

{ Nhập S }// hs tự viết

K:= 0; { Tổng giá trị các phần tử đã chọn khởi đầu là 0 }

{ Khởi tạo cho mảng chon }

For i:=1 to N do

chon[i]:= FALSE; { Giả sử chưa chọn phần tử này }

End;

Procedure XuatDayCon;

Var i: word;

Begin

For i:=1 to N do

If (chon[i]) then

Write(a[i], ' ');

Writeln;

End;

Procedure XuLy(i: word);

Begin

If (i <= N) then begin

XuLy(i + 1);

If (K + a[i] <= S) then begin

chon[i]:= TRUE;

K:= K + a[i];

If (K = S) then XuatDayCon

else XuLy(i + 1);

chon[i]:= FALSE;

K:= K - a[i];

end;

end;

End;

BEGIN

KhoiTao;

XuLy(1);

END.

### BÀI TOÁN 10. Dãy con có tổng lớn nhất (2)

Cho dãy n số nguyên dương  $a_1, a_2, \dots, a_n$ . Một dãy con của dãy nói trên là dãy được lập từ dãy đã cho bằng cách bỏ đi một số số hạng của dãy và giữ nguyên trật tự các số còn lại. Hãy tìm một dãy con thỏa mãn tính chất:

- Không có ba số liên tiếp nào của dãy ban đầu có mặt trong dãy con
- Trong ba số liên tiếp của dãy ban đầu có ít nhất một số có mặt trong dãy con
- Tổng các số hạng của dãy con được chọn là lớn nhất có thể được.

**Dữ liệu:** Vào từ file CHONSO.INP:

- Dòng đầu tiên chứa số nguyên dương N ( $N \leq 1000$ )
- N dòng tiếp theo, dòng thứ i chứa số nguyên dương  $a_i$  ( $a_i \leq 30000$ )

**Kết quả:** Ghi ra file văn bản CHONSO.OUT:

- Dòng đầu tiên chứa hai số nguyên dương M và T trong đó M là số lượng các số hạng của dãy con được chọn, T là tổng các số của dãy con được chọn.
- M dòng tiếp theo lần lượt mô tả các số hạng của dãy con được chọn, dòng thứ k ghi số  $j_k$  là chỉ số của số hạng được chọn thứ k.

*Ví dụ:*

CONSO.INP	CHONSO.OUT
6	21 4
2	2
6	3
5	5
1	6
7	
3	

```
const
  tfi='CHONSO.INP';
  tfo='CHONSO.OUT';
  maxN=1000;

var
  fi, fo: text;
  N: integer;
  x,a,T,Tr: array[0..maxN] of longint;
  kt,count: integer;
```

```
procedure Docdl;
var i: integer;
begin
  assign(fi,tfi); reset(fi);
  readln(fi,N);
  for i:=1 to N do readln(fi,a[i]);
  close(fi);
end;
```

```
procedure XDB;
var i: integer;
begin
  fillchar(T,sizeof(T),0);
  fillchar(Tr,sizeof(Tr),0);
  T[1]:=a[1];
  T[2]:=a[1]+a[2];
  if a[2]>a[1] then
    begin
      T[3]:=a[3]+a[2];
      Tr[3]:=2;
    end
  else
    begin
      T[3]:=a[3]+a[1];
      Tr[3]:=1;
    end;
  for i:=4 to N do
    begin
```

```
T[i]:=a[i]+a[i-1]+T[i-3]; Tr[i]:=i-3;
if T[i]<a[i]+a[i-1]+T[i-4] then
  begin
    T[i]:=a[i]+a[i-1]+T[i-4];
    Tr[i]:=i-4;
  end;
if T[i]<a[i]+T[i-2] then
  begin
    T[i]:=a[i]+T[i-2];
    Tr[i]:=i-2;
  end;
end;
end;
```

```
procedure XLB;
var i,u,v,w: integer;
begin
  kt:=n;
  for i:=1 to 2 do
    if T[kt]<T[n-i] then kt:=n-i;
  u:=kt;
  count:=0;
  while u>3 do
  begin
    v:=tr[u];
    if v=u-2 then
      begin
        inc(count);
        x[count]:=u;
      end
    else
      begin
        inc(count);
        x[count]:=u;
        inc(count);
        x[count]:=u-1;
      end;
    u:=v;
  end;
  case u of
    1: begin
        inc(count);
        x[count]:=1;
      end;
    2: begin
        inc(count); x[count]:=2;
        inc(count); x[count]:=1;
      end;
    3: begin
        v:=tr[u];
        inc(count); x[count]:=3;
        inc(count); x[count]:=v;
      end;
  end;
end;
```

end;

procedure Inkq;

var u: integer;

begin

assign(fo,tfo); rewrite(fo);

writeln(fo,T[kt],', ',count);

for u:=count downto 1 do writeln(fo,x[u]);

close(fo);

end;

BEGIN

Docdl;

XDB;

XLB;

Inkq;

END.

### **BÀI TOÁN 11. Chọn phần thưởng (dạng bài toán tìm dãy con có tổng lớn nhất theo điều kiện không được chọn 3 phần tử liên tiếp)**

Bạn là người đạt giải đặc biệt trong một kỳ thi tin học. Ban tổ chức cho phép bạn chọn phần thưởng cho mình. Các phần thưởng của ban tổ chức được xếp thành một dãy, đánh số từ 1 đến N, phần thưởng thứ i có giá trị là  $a_i$ . Bạn sẽ được phép chọn phần thưởng cho mình theo nguyên tắc không được chọn 3 phần thưởng liên tiếp nhau trong dãy phần thưởng của ban tổ chức.

**Yêu cầu:** Viết chương trình nêu ra cách chọn các phần thưởng của ban tổ chức sao cho tổng giá trị của các phần thưởng mà bạn nhận được là lớn nhất.

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

- Dòng đầu ghi số nguyên N là số phần thưởng của ban tổ chức ( $N \leq 25000$ )
- Trong N dòng tiếp theo, dòng thứ i ghi số nguyên  $a_i$  là giá trị của phần thưởng thứ i ( $a_i \leq 10^6$ )

**Kết quả:** Ghi ra file văn bản PTHUONG.OUT:

- Dòng đầu tiên ghi tổng giá trị của các phần thưởng được lựa chọn.
- Trong các dòng tiếp theo, lần lượt ghi số hiệu của các phần thưởng mà bạn sẽ lấy, mỗi dòng ghi đúng 10 số hiệu, trừ dòng cuối cùng có thể ít hơn 10 số hiệu. Các số hiệu được ghi lần lượt theo trình tự tăng dần, các số hiệu ghi trên một dòng cách nhau ít nhất một ký tự trắng.

**Ví dụ:**

PTHUONG.INP

5

6

9

1

3

5

PTHUONG.OUT

23

1 2 4 5

```
{ $A+,B-,D+,E+,F-,G-,I+,L+,N+,O-,P-,Q+,R+,S+,T-,V+,X+ }
{ $M 16384,0,655360 }
```

uses crt;

const

tfi = 'PTHUONG.INP';

tfo = 'PTHUONG.OUT';

maxN = 25000;

NN = 5000;

```
type  
    mang          =    array[1..maxN] of integer;
```

```
var  
    fi,fo: text;  
    N: integer;  
    Tr: mang;  
    a: array[1..3] of LongInt;  
    S: array[0..3] of LongInt;  
    x: ^mang;  
    slx: integer;
```

```
procedure Sinhdl;  
var ch: char;  
    i,u: LongInt;  
begin  
    clrscr;  
    writeln('Ban co tao file ',tfi,' (C/K)?');  
    repeat ch:=readkey until upcase(ch) in ['C','K'];  
    if upcase(ch)='K' then exit;  
    randomize;  
    N:=NN;  
    assign(fi,tfi); rewrite(fi);  
    writeln(fi,N);  
    for i:=1 to N do  
        begin  
            u:=60;  
            u:=random(u)+1;  
            writeln(fi,u);  
        end;  
    close(fi);  
end;
```

```
procedure Mot;  
var u: integer;  
begin  
    assign(fo,tfo); rewrite(fo);  
    readln(fi,u);  
    writeln(fo,u,' ',1);  
    write(fo,1);  
    close(fo);  
end;
```

```
procedure Hai;  
var u,v: integer;  
begin  
    assign(fo,tfo); rewrite(fo);  
    readln(fi,u);  
    readln(fi,v);  
    writeln(fo,u+v,' ',2);  
    write(fo,1,' ',2);  
    close(fo);  
end;
```

```
procedure XDB;
var i: integer;
begin
  for i:=1 to N do Tr[i]:=0;
  read(fi,a[1]);
  read(fi,a[2]);
  tr[1]:=-1;
  tr[2]:=-1;
  s[0]:=0;
  s[1]:=a[1];
  s[2]:=a[1]+a[2];
  for i:=3 to N do
    begin
      read(fi,a[3]);
      s[3]:=s[2]; Tr[i]:=i-1;
      if s[3]<a[3]+s[1] then
        begin
          s[3]:=a[3]+s[1];
          tr[i]:=i-2;
        end;
      if s[3]<a[3]+a[2]+s[0] then
        begin
          s[3]:=a[3]+a[2]+s[0];
          tr[i]:=i-3;
        end;
      a[1]:=a[2];
      a[2]:=a[3];
      s[0]:=s[1];
      s[1]:=s[2];
      s[2]:=s[3];
    end;
end;
```

```
procedure XLB;
var u,v,i: integer;
begin
  slx:=0;
  v:=N;
  repeat
    u:=Tr[v];
    for i:=v downto u+2 do
      begin
        inc(slx);
        x^[slx]:=i;
      end;
    v:=u;
  until v<=0;
end;
```

```
procedure Inkq;
var i,dem: integer;
begin
  assign(fo,tfo); rewrite(fo);
```



```
writeln(fo,s[3],',',N);
dem:=0;
for i:=slx downto 1 do
begin
inc(dem);
write(fo,x^[i],');
if dem mod 10=0 then writeln(fo);
end;
close(fo);
end;

procedure Solve;
begin
XDB;
XLB;
Inkq;
end;

procedure CapPhat;
begin
New(x);
end;

BEGIN
CapPhat;
{Sinhdl;}
assign(fi,tfi); reset(fi);
readln(fi,N);
case N of
1: Mot;
2: Hai;
else Solve;
end;
close(fi);
END.
```

## BÀI TOÁN 12. Dãy con chung dài nhất của hai dãy số {COMSEQ.INP}

Cho hai số nguyên dương  $M, N$  ( $0 < M, N \leq 100$ ) và hai dãy số nguyên:  $a_1, a_2, \dots, a_m$  (A) và  $b_1, b_2, \dots, b_n$  (B). Tìm một dãy dài nhất C nhận được từ A bằng cách xoá đi một số số hạng và cũng nhận được từ B bằng cách xoá đi một số số hạng (nói chung không cùng chỉ số như đối với dãy A). Ta gọi C là dãy con chung dài nhất của hai dãy A, B.

Input: Cho trong file COMSEQ.INP:

- Dòng đầu tiên chứa hai số  $M, N$
- Dòng thứ hai chứa  $M$  số  $a_1, a_2, \dots, a_m$
- Dòng thứ ba chứa  $N$  số  $b_1, b_2, \dots, b_n$ .

Output: Ghi ra file COMSEQ.OUT

- Dòng thứ nhất ghi K là số lượng số hạng của dãy C
- Dòng thứ hai ghi K số là các số hạng của dãy C
- Dòng thứ ba ghi K số là chỉ số trong dãy A của K số hạng của C
- Dòng thứ tư ghi K số là chỉ số trong dãy B của K số hạng của C

*Ví dụ:*

COMSEQ.INP

5 6

COMSEQ.OUT

3

2 3 5 1 4  
4 2 3 6 5 10

2 3 5  
1 2 3  
2 3 5

(giải tương tự bài toán 12.1)

**Bài toán 12.1: Dãy con chung dài nhất (2)**

**{DAYCON.PAS}**

Cho dãy số nguyên A gồm N phần tử  $a_1, a_2, \dots, a_N$  và dãy số nguyên B gồm M phần tử  $b_1, b_2, \dots, b_M$ . Các phần tử trong một dãy số có giá trị khác nhau từng đôi một.

( $1 \leq a_i, b_j \leq 2 \cdot 10^9$ ;  $1 \leq N \leq 100$ ;  $1 \leq i \leq N$ ;  $1 \leq M \leq 100$ ;  $1 \leq j \leq M$ ).

Dãy C được gọi là dãy con của dãy A nếu dãy C nhận được từ dãy A bằng cách xóa đi một số phần tử và giữ nguyên thứ tự của các phần tử còn lại.

Nếu dãy C là dãy con của dãy A và cũng là dãy con của dãy B thì dãy C được gọi là dãy con chung của hai dãy A và B.

**Yêu cầu:** Hãy tìm dãy C là dãy con chung của hai dãy A và B sao cho số lượng phần tử của dãy C là lớn nhất.

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

- Dòng 1: Ghi số nguyên dương N là số lượng phần tử của dãy A.
- Dòng 2: Ghi N số nguyên là giá trị của các phần tử trong dãy A, các số được ghi cách nhau ít nhất một dấu cách.
- Dòng 3: Ghi số nguyên dương M là số lượng phần tử của dãy B.
- Dòng 4: Ghi M số nguyên là giá trị của các phần tử trong dãy B, các số được ghi cách nhau ít nhất một dấu cách.

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

- Dòng 1: Ghi số nguyên dương K là số lượng phần tử của dãy C.
- Dòng 2: Ghi K số nguyên là giá trị của các phần tử trong dãy C, các số được ghi cách nhau một dấu cách.
- Dòng 3: Ghi K số nguyên dương lần lượt là chỉ số của các phần tử trong dãy A tương ứng với các giá trị của phần tử đó trong dãy C, các số được ghi cách nhau một dấu cách.
- Dòng 4: Ghi K số nguyên dương lần lượt là chỉ số của các phần tử trong dãy B tương ứng với các giá trị của phần tử đó trong dãy C, các số được ghi cách nhau một dấu cách.

**Ví dụ:**

DAYCON.INP	DAYCON.OUT
6	4
9 3 1 12 6 15	3 12 6 15
5	2 4 5 6
3 12 7 6 15	1 2 4 5

Hướng dẫn:

Sử dụng mảng hai chiều  $L[0..M, 0..N]$  để lưu bảng phương án. Với ý nghĩa  $L[i, j]$  là độ dài dãy con chung của hai dãy  $A[0..i]$  và  $B[0..j]$ . Đương nhiên, nếu một dãy là dãy rỗng (số phần tử bằng

0) thì dãy con chung dài nhất là rỗng  $\rightarrow L[0..j]=0, \forall j, j = \overline{1, N}, L[i..0]=0, \forall i, i = \overline{1, M}$

Với  $0 \leq i \leq M, 0 \leq j \leq N$  thì  $L[i, j]$  được tính theo công thức truy hồi sau:

$L[i, j] := \max(L[i-1, j], L[i, j-1], L[i-1, j-1] + x)$  với  $x=0$  nếu  $A[i] \neq B[j]$ ,  $x=1$  nếu  $A[i] = B[j]$

Thật vậy, nếu  $A[i] = B[j]$  thì  $L[i-1, j-1] + 1$

Nếu  $A[i] \neq B[j]$  thì

+ Nếu  $A[i]$  trong  $B[1..j]$  thì nó chỉ thuộc đoạn  $B[1..j-1]$  nên  $L[i, j] = L[i, j-1]$

+ Nếu  $B[j]$  trong  $A[1..i]$  thì nó chỉ thuộc đoạn  $A[1..i-1]$  nên  $L[i, j] = L[i-1, j]$

Program Day\_con;

Const fi='DAYCON.INP';

fo='DAYCON.OUT';

MaxN=101;

Type mang2c=Array[0..maxN, 0..MaxN] of Longint;

```
mang1c:=Array[0..MaxN] of Longint;
var A,B,C:mang1c;
    L:mang2c;
    N,M,K:Longint;
procedure Read_Data;
var f:Text; i:Longint;
begin
    Assign(f,fi);
    Reset(f);
    Readln(f,N);
    for i:=1 to N do
        Read(f,A[i]);
    Readln(f,M);
    for i:=1 to M do
        Read(f,B[i]);
    Fillchar(L,Sizeof(L),0); {for i:=1 to n do for j:=1 to m do L[i,j]:=0;}
    Close(f);
end;
procedure Solution;
var i,j:Longint;
begin
    for i:=1 to N do
        for j:=1 to M do
            begin
                if A[i] = B[j] then
                    L[i,j]:=L[i-1,j-1]+1
                else
                    If L[i-1,j] > L[i,j-1] then L[i,j]:=L[i-1,j] else L[i,j]:=L[i,j-1]
            {hoac neu co khai bao ham max thi chi can ghi L[i,j]:=max(L[i,j-1],L[i-1,j])}
            end;
end;
procedure Write_Data;
var f:Text;i,j,Top:Longint;LA,LB:mang1c;
{LA: mang luu chi so cac phan tu chung o mang A, LB: mang luu cac phan tu chung o mang B}
begin
    Assign(f,fo);
    Rewrite(f);
    Writeln(f,L[N,M]);
    Top:=0;
    i:=N;j:=M;
    While (i>0) and (j>0) do
        begin
            if A[i] = B[j] then
                begin
                    inc(Top);
                    LA[Top]:=i;
                    LB[Top]:=j;
                    dec(i);
                    dec(j);
                end
            else
                if L[i,j] = L[i-1,j] then
                    dec(i)
                else
                    dec(j);
```

```

end;
for i:=Top downto 1 do
  Write(f,A[LA[i]],' '); {ghi day con chung
Writeln(f);
for i:=Top downto 1 do
  Write(f,LA[i],' ');
Writeln(f);
for i:=Top downto 1 do
  Write(f,LB[i],' ');
Close(f);
end;
begin
  Read_Data;
  Solution;
  Write_Data;
end.
* Nếu bài toán chỉ yêu cầu đưa ra dãy con chung của hai dãy thì có thể chỉ cần dùng một
mảng nữa để lưu dãy con chung là được (mảng p trong chương trình bên dưới)
procedure Trace;
var f:text;
    i,j:longint;
begin
  assign(f,fo);
  rewrite(f);
  writeln(f,L[m,n]);
  i:=m; j:=n; count:=0;
  while (i>0) and (j>0) do
    begin
      if s1[i]=s2[j] then
        begin
          inc(count);
          P[count]:= a[i];
          dec(i);
          dec(j);
        end
      else
        if L[i,j]=L[i,j-1] then dec(j) else dec(i);
    end;
  write(f,p);
  close(f);
  readln
end;

```

### **BÀI TOÁN 13: SƯU TẬP TEM {STAMP.pas}**

Peter là một cậu bé rất thích sưu tập tem. Nhân dịp nghỉ 30-4 cậu được mẹ cho đi siêu thị mua sắm. Trên đường ra siêu thị, khi đang đi qua bưu điện, cậu bắt đầu vòi tiền của mẹ mình để mua tem. Ở bưu điện, họ đang bán các loại tem khác nhau bao gồm: N tem loại 1 đô-la, và M tem loại 2 đô la.

Peter được mẹ cho đóng K đô la, và cậu muốn dùng tất cả số tiền này để mua tem. Biết rằng cậu có thể mua nhiều tem cùng loại.

Bạn hãy giúp Peter tính xem cậu bé có bao nhiêu cách để có thể mua tem.

#### **Yêu cầu:**

Cho là các số nguyên N, M, K, và một số nguyên tố P.

Nhiệm vụ của bạn là tính  $Z \bmod P$ , trong đó Z (có thể rất lớn) là số cách mà Peter có thể dùng tất

cả K đô la để mua tem.

\* **Dữ liệu vào:** Từ file **STAMP.INP**

- Dòng đầu tiên chứa 4 số một số nguyên N, M, K và P. ( $3 \leq P \leq 10^6$ , có 70% số test có:  $0 \leq N, M \leq 1000$  và  $1 \leq K \leq 1000$ ; 30% số test có  $0 \leq N, M \leq 300$  và  $1 \leq K \leq 10^{12}$ )

\* **Kết quả:** Ghi ra file **STAMP.OUT**

Gồm duy nhất một dòng ghi ra một số nguyên là số lượng cách khác nhau để mua tem, modulo P.

\* **Ví dụ**

STAMP.INP	STAMP.OUT
2 2 4 47	14

**Giải thích:**

- Mua hai tem 2-đô-la: có 3 cách để làm như vậy

- Mua một con tem 2-đô la và hai tem 1-đô la: có  $2 \times 3 = 6$  cách để làm như vậy

- Mua bốn tem 1-đô-la: có 5 cách để làm như vậy

Vì vậy câu trả lời là  $(3 + 6 + 5) \bmod 47 = 14 \bmod 47 = 14$ .

Hướng dẫn: Quy hoạch động trên mảng hai chiều.

{ \$M 50000000 }

{ \$inline on }

```
const fi='stamp.inp';
```

```
fo='STAMP.OUT';
```

```
maxn=1000;
```

```
var f,g : text;
```

```
n,m,p,mv, test : longint;
```

```
s : array[0..maxn,0..maxn] of longint;
```

```
k,ans : int64;
```

```
function max(const u,v : longint) : longint; inline;
```

```
begin if u>v then exit(u); exit(v); end;
```

```
procedure openf; inline;
```

```
begin
```

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

```
assign(g,fo); rewrite(g);
```

```
end;
```

```
procedure prepare; inline;
```

```
var i,j : longint;
```

```
begin
```

```
s[0,0]:=1;
```

```
for i:=0 to mv do s[1,i]:=1;
```

```
for i:=1 to mv do s[i,0]:=1;
```

```
for i:=2 to mv do
```

```
for j:=1 to mv do s[i,j]:=(s[i-1,j]+s[i,j-1]) mod p;
```

```
end;
```

```
procedure process; inline;
```

```
var i,j : longint;
```

```
begin
```

```
if (m=0) then begin if n>0 then writeln(g,s[n,k]) else writeln(g,0); exit; end;
```

```

ans:=0;
for i:=0 to k div 2 do begin ans:=ans+int64(s[m,i])*s[n,k-i*2]; ans:=ans mod p; end;
writeln(g,ans);
end;

procedure big; inline;
begin
  writeln(g,0);
end;

procedure closef; inline;
begin close(f); close(g); end;

BEGIN
openf;
//readln(f,test);

//repeat
  readln(f,n,m,k,p); mv:=max(m,n);
  if k>1000 then big else begin prepare; process; end;
//  dec(test);
//until test=0;

closef;
END.

```

#### BÀI TOÁN 14. Phủ

Cho  $n$  đoạn trên trục số, đoạn thứ  $i$  là  $[L_i, R_i]$ .

**Hãy chọn ra trong các đoạn kể trên một số ít nhất các đoạn để phủ hết đoạn  $[a, b]$**

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

- Dòng 1: Chứa 3 số  $n, a, b$
- $n$  dòng tiếp theo, dòng thứ  $i$  chứa hai số  $L_i$  và  $R_i$

**Kết quả:** Ghi ra file văn bản COVER.OUT

- Dòng 1: Ghi số  $k$  là số đoạn được chọn (Nếu không có cách chọn thì  $k = -1$ )
- Trong trường hợp có phương án thực hiện yêu cầu thì  $k$  dòng tiếp theo, mỗi dòng ghi chỉ số một đoạn được chọn

**Các số trên một dòng của Input/Output file cách nhau ít nhất một dấu cách**

**Ràng buộc:**  $1 \leq n \leq 100000$ ; các số còn lại là số nguyên dương  $\leq 30000$ ;  $a \leq b$ ;  $\forall i: L_i \leq R_i$

**Ví dụ:**

COVER.INP	COVER.OUT	COVER.INP	COVER.OUT
8 2 10	3	8 1 200	-1
4 8	1	1 4	
1 3	4	2 5	
2 3	6	4 5	
1 4		6 45	
3 4		6 7	
7 10		5 7	
9 11		100 200	

8 11		50 99	
------	--	-------	--

```

{$R+,q+}
const
  nfi='cover.inp';
  nfo='cover.out';
  limit=30000;
  limitn=100;
type
  mang=array[1..100] of longint;
var
  fi,fo: text;
  d: array[1..limit] of integer;
  st: array[1..300] of ^mang;
  n,a,b,dd,sl: integer;
procedure openfile;
begin
  assign(fi,nfi); reset(fi);
  assign(fo,nfo); rewrite(fo);
end;
procedure closefile;
begin
  close(fi); close(fo);
end;
procedure input;
var
  i,u,v,j: integer;
begin
  for i:=1 to 30000 do
    d[i]:=0;
  for i:=1 to 300 do
    new(st[i]);
  readln(fi,n,a,b);
  for i:=1 to 300 do
    for j:=1 to 100 do
      st[i]^j:=0;
    for i:=1 to n do
      begin
        readln(fi,u,v);
        if d[u]<v then
          begin
            d[u]:=v;
            st[(u-1) div limitn+1]^(u-1 mod limitn+1):=i;
          end;
        end;
      dd:=a;
    end;
  procedure process1;
  var
    i,max,cs,k: integer;
  begin
    cs:=1;
    sl:=0;

```

```
repeat
k:=0;
max:=0;
for i:=cs to a do
  if d[i]>0 then
    if max<d[i] then
      begin
        max:=d[i];
        k:=i;
      end;
  if k=0 then exit;
inc(sl);
a:=d[k];
cs:=k+1;
until a>=b;
end;
procedure process2;
var
  i,max,cs,sl,k: integer;
begin
  a:=dd;
  cs:=1;
  repeat
    k:=0;
    max:=0;
    for i:=cs to a do
      if d[i]>0 then
        if max<d[i] then
          begin
            max:=d[i];
            k:=i;
          end;
    if k=0 then exit;
    writeln(fo,st[(k-1) div limitn+1]^[ (k-1) mod limitn+1]);
    a:=d[k];
    cs:=k+1;
  until a>=b;
end;
procedure output;
begin
  if a<b then write(fo,-1)
  else
    begin
      writeln(fo,sl);
      process2;
    end;
end;
begin
  openfile;
  input;
  process1;
  output;
  closefile;
end.
```



## BÀI TOÁN 15. Chuyên toán tin

Tờ báo CTT (Chuyên Toán-Tin) phát hành với số lượng lớn trên  $N$  tỉnh  $1..N$ ,  $N$  chẵn và  $N \leq 1000$  do đó phải in tại hai nhà in đặt ở tỉnh 1 và tỉnh 2. Hàng ngày từ mỗi nhà in có  $N/2$  ô tô chở báo đến  $N$  tỉnh, mỗi ô tô chở báo đi một tỉnh, hai ô tô khác nhau chở báo đi hai tỉnh khác nhau, giữa hai tỉnh khác nhau có không quá một đoạn đường nối trực tiếp và không có hai đoạn đường nào cắt nhau. Các xe khi đi từ nhà in đến bất kỳ tỉnh nào cũng đều biết cách đi theo đường ngắn nhất.

Hãy tìm cách phân phối báo từ các nhà in đến các tỉnh sao cho tổng độ dài đường đi của các ô tô là nhỏ nhất.

Dữ liệu vào được cho bởi file CTT.INP trong đó dòng thứ nhất ghi hai số  $N, M$ , tiếp theo là  $M$  dòng, mỗi dòng ghi ba số nguyên dương  $X, Y, L$  có nghĩa là có đoạn đường nối trực tiếp hai tỉnh  $X, Y$  với độ dài  $L$ ,  $L$  không lớn hơn 1000. Luôn có ít nhất một cách chuyển báo đến tất cả  $N$  tỉnh. Độ dài đường đi từ nhà in tại tỉnh nào đến chính tỉnh đó xem nh bằng 0.

Kết quả ghi ra file CTT.OUT hai dòng, dòng thứ nhất ghi tên các tỉnh nhận báo từ nhà in tại tỉnh 1, dòng thứ hai ghi tên các tỉnh nhận báo từ nhà in tại tỉnh 2.

Ví dụ:

CTT.INP	CTT.OUT
6 8	1 4 5
1 2 5	2 3 6
1 4 8	
1 6 7	
2 3 6	
3 4 5	
3 6 2	
4 5 3	
5 6 2	

```

const
  INP = 'CTT.INP';
  OUT = 'CTT.OUT';
  MaxN = 1000;
  Infinity = 10000000000;

type
  PNode = ^TNode;
  TNode = record
    v, w : Integer;
    Next : PNode;
  end;
  TDist = array[1..MaxN] of LongInt;

var
  List : array[1..MaxN] of PNode;
  Dist1, Dist2 : TDist;
  Chon : array[1..MaxN] of Byte;
  N : Integer;
  f : Text;
  Result : LongInt;

procedure Add(u, v, w : Integer);
var
  tmp : PNode;

```

```
begin
  New(tmp);
  tmp^.v:=v; tmp^.w:=w;
  tmp^.Next:=List[u];
  List[u]:=tmp;
end;

procedure Init;
var
  i, u, v, w : Integer;
  M : LongInt;
begin
  Assign(f, INP); Reset(f);
  Readln(f, N, M);
  for i:=1 to N do
    List[i]:=nil;
  for i:=1 to M do
    begin
      Readln(f, u, v, w);
      Add(u, v, w);
      Add(v, u, w);
    end;
  Close(f);
end;

procedure Dijkstra(Start : Integer; var Dist : TDist);
var
  Heap, Pos : array[1..MaxN] of Integer;
  Chua : array[1..MaxN] of Boolean;
  Last, i : Integer;
  HeapSize : Integer;
  tmp : PNode;

procedure UpHeap(i : Integer);
var
  x, c, r : Integer;
begin
  x:=Heap[i];
  c:=i; r:=i shr 1;
  while r > 0 do
    begin
      if Dist[Heap[r]] > Dist[x] then
        begin
          Heap[c]:=Heap[r];
          Pos[Heap[r]]:=c;
        end
      else
        Break;
      c:=r; r:=r shr 1;
    end;
  Heap[c]:=x;
  Pos[x]:=c;
end;
```

```
procedure DownHeap(i : Integer);
var
  x, c, r : Integer;
begin
  x:=Heap[i];
  c:=i; r:=i shl 1;
  while r <= HeapSize do
    begin
      if (r < HeapSize) and (Dist[Heap[r]] > Dist[Heap[r+1]]) then
        Inc(r);
      if Dist[Heap[r]] < Dist[x] then
        begin
          Heap[c]:=Heap[r];
          Pos[Heap[r]]:=c;
        end
      else
        Break;
      c:=r; r:=r shl 1;
    end;
    Heap[c]:=x;
    Pos[x]:=c;
  end;
```

```
function Extract : Integer;
begin
  Extract:=Heap[1];
  Heap[1]:=Heap[HeapSize];
  Pos[Heap[HeapSize]]:=1;
  Dec(HeapSize);
  DownHeap(1);
end;
```

```
begin
  FillChar(Chua, Sizeof(Chua), True);
  for i:=1 to N do
    Dist[i]:=Infinity;
  Chua[Start]:=False;
  Dist[Start]:=0;
  HeapSize:=0;
  for i:=1 to N do
    if i <> Start then
      begin
        Inc(HeapSize);
        Heap[HeapSize]:=i;
        Pos[i]:=HeapSize;
      end;
  Last:=Start;
  for i:=2 to N do
    begin
      tmp:=List[Last];
      while tmp <> nil do
        begin
          if Chua[tmp^.v] and (Dist[Last] + tmp^.w < Dist[tmp^.v]) then
            begin
```

```
        Dist[tmp^.v]:=Dist[Last] + tmp^.w;
        UpHeap(Pos[tmp^.v]);
    end;
    tmp:=tmp^.Next;
end;
Last:=Extract;
end;
end;

procedure Process;
var
    Sub : array[1..MaxN] of LongInt;
    cs : array[1..MaxN] of Integer;
    i : Integer;

    procedure QuickSort(l, r : Integer);
    var
        x : LongInt;
        i, j, tmp : Integer;
    begin
        if l >= r then Exit;
        x:=Sub[cs[Random(r-l+1)+l]];
        i:=l; j:=r;
        repeat
            while (i < r) and (Sub[cs[i]] < x) do Inc(i);
            while (j > l) and (Sub[cs[j]] > x) do Dec(j);
            if i <= j then
                begin
                    if i < j then
                        begin
                            tmp:=cs[i];
                            cs[i]:=cs[j];
                            cs[j]:=tmp;
                        end;
                    Inc(i); Dec(j);
                end;
            until i > j;
            QuickSort(l, j); QuickSort(i, r);
        end;
    end;

begin
    Dijkstra(1, Dist1);
    Dijkstra(2, Dist2);
    for i:=1 to N do
        begin
            cs[i]:=i;
            Sub[i]:=Dist1[i] - Dist2[i];
        end;
    QuickSort(1, N);
    Result:=0;
    for i:=1 to N div 2 do
        begin
            Chon[cs[i]]:=1;
            Inc(Result, Dist1[cs[i]]);
        end;
    end;
```

```

    end;
    for i:=N div 2 + 1 to N do
    begin
        Chon[cs[i]]:=2;
        Inc(Result, Dist2[cs[i]]);
    end;
    Writeln('Result : ', Result);
end;

procedure Done;
var
    i : Integer;
begin
    Assign(f, OUT); Rewrite(f);
    for i:=1 to N do
        if Chon[i] = 1 then
            Write(f, i, ' ');
    Writeln(f);
    for i:=1 to N do
        if Chon[i] = 2 then
            Write(f, i, ' ');
    Close(f);
end;

BEGIN
    Init;
    Process;
    Done;
END.

```

## BÀI TOÁN 16. Đoạn đầu dài nhất

Cấu trúc của một số vật thể sinh học được biểu diễn bởi một dãy các thành phần của chúng. Các thành phần này được ký hiệu bởi các chữ cái hoa. Các nhà sinh học quan tâm đến việc phân rã một dãy dài thành các dãy ngắn hơn. Các dãy ngắn hơn đó được gọi là các dãy nguyên thủy. Ta nói rằng một dãy S có thể ghép được từ một tập cho trước P các dãy nguyên thủy nếu có thể tìm được N dãy nguyên thủy  $p_1, p_2, \dots, p_n$  thuộc P sao cho ghép của chúng  $p_1 p_2 \dots p_n$  bằng S. Việc ghép các dãy nguyên thủy  $p_1 p_2 \dots p_n$  có nghĩa là đặt chúng liên tiếp theo thứ tự đó. Một dãy nguyên thủy có thể có mặt nhiều lần trong phép ghép và không nhất thiết mọi dãy nguyên thủy đều phải có mặt. Chẳng hạn dãy ABABACABAAB có thể được ghép từ tập các dãy nguyên thủy:

{A, AB, BA, CA, BBC}

K ký tự đầu của một dãy S được gọi là đoạn đầu với độ dài K của S. Viết chương trình với dữ liệu vào là một tập P các dãy nguyên thủy và một dãy T các thành phần. Chương trình cần tính độ dài của đoạn đầu dài nhất của dãy T mà đoạn đầu đó có thể ghép được từ các dãy nguyên thủy trong P.

*Dữ liệu vào:*

Dữ liệu vào được cho trong hai file INPUT.TXT mô tả các dãy nguyên thủy P còn file DATA.TXT chứa dãy T cần được xem xét. Dòng thứ nhất của file INPUT.TXT chứa N là số lượng dãy nguyên thủy có trong P ( $1 \leq N \leq 100$ ). Mỗi dãy nguyên thủy được cho bởi hai dòng liên tiếp. Dòng thứ nhất chứa độ dài L của dãy nguyên thủy ( $1 \leq L \leq 20$ ). Dòng thứ hai chứa một xâu độ dài L gồm các chữ cái hoa (từ 'A' đến 'Z'). N dãy nguyên thủy này từng đôi một khác nhau.

Mỗi dòng của file DATA.TXT chứa một chữ cái hoa ở vị trí đầu tiên. Độ dài của dãy từ 1 đến 500000

*Dữ liệu ra:*

Viết vào dòng thứ nhất của file OUTPUT.TXT độ dài của đoạn đầu dài nhất của T mà đoạn đầu có thể ghép được từ tập P

Ví dụ về dữ liệu vào và ra:

INPUT.TXT	DATA.TXT	OUTPUT.TXT
5	A	11
1	B	
A	A	
2	B	
AB	A	
2	C	
BA	A	
2	B	
CA	A	
3	A	
BBC	B	
	C	
	B	

```
{*****
DOAN DAU DAI NHAT
*****}
```

```
uses crt;
```

```
const
  tfi1      = 'INPUT.TXT';
  tfi2      = 'DATA.TXT';
  tfo       = 'OUTPUT.TXT';
  maxN      = 100;
  maxL      = 20;
```

```
var
  N          : integer;
  p          : array[1..maxN] of string[maxL];
```

```
  MaxLen     : LongInt;
```

```
  Buff       : string[2*maxL];
  cB         : integer;
```

```
  fi,fo      : text;
  dd         : array[0..2*maxL] of 0..1;
```

```
procedure ChuanBi;
```

```
var f: text;
```

```
  i,j,L: integer;
```

```
  c: char;
```

```
begin
```

```
  assign(f,tfi1); reset(f);
```

```
  readln(f,N);
```

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

```
    begin
```

```
      readln(f,L);
```

```
      p[i]:='';
```

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

```
begin
  read(f,c);
  p[i]:=p[i]+c;
end;
readln(f);
end;
close(f);
end;

procedure DocVDem;
var c: char;
    i: integer;
begin
  if length(Buff)>maxL then
  begin
    delete(Buff,1,maxL);
    for i:=maxL to cb do dd[i-maxL]:=dd[i];
    cb:=cb-maxL;
    MaxLen:=MaxLen+maxL;
  end;
  while (not seekeof(fi)) and (length(Buff)<2*maxL) do
  begin
    readln(fi,c);
    Buff:=Buff+c;
  end;
  for i:=cb+1 to length(Buff) do dd[i]:=0;
end;

procedure Xuli;
var i,j,maxCB: integer;
    s: string[2*maxL];
begin
  dd[cb]:=1;
  for i:=0 to Length(Buff) do
    if dd[i]=1 then
    begin
      maxCB:=i;
      s:=copy(Buff,i+1,255);
      for j:=1 to N do
        if pos(p[j],s)=1 then dd[i+length(p[j])]:=1;
      end;
      cb:=maxCB;
    end;
end;

BEGIN
  ChuanBi;
  assign(fi,tfi2); reset(fi);
  assign(fo,tfo); rewrite(fo);
  MaxLen:=0;
  Buff:="";
  cb:=0;
  while not seekeof(fi) do
  begin
    DocVDem;
```

```

Xuli;
if cB<maxL then break;
end;
MaxLen:=MaxLen+cB;
writeln(fo,MaxLen);
close(fi); close(fo);
END.

```

### BÀI TOÁN 17. Xóa đơn điệu - k gián đoạn

Cho một dãy số nguyên gồm  $N$  số  $a_1, a_2, \dots, a_n$ ,  $N \leq 1000$  các số hạng có giá trị tuyệt đối không vượt quá 30000 và một số nguyên dương  $K$ . Hãy tìm cách xóa đi một số ít nhất số hạng sao cho dãy còn lại là một dãy đơn điệu tăng ngoại trừ không quá  $K$  cặp số liên nhau vi phạm điều kiện này.

Dữ liệu vào được cho bởi file DAYK.INP trong đó dòng thứ nhất ghi hai số nguyên dương  $N, K$  ( $K \leq 10$ ). Trong  $N$  dòng tiếp theo, mỗi dòng ghi một số hạng của dãy bắt đầu từ  $a_1$  và cuối cùng là  $a_n$ .

Kết quả ghi ra file DAYK.OUT nh sau: dòng thứ nhất ghi  $B$  là số lượng số hạng cần xóa đi. Nếu  $B > 0$ , trong  $B$  dòng tiếp theo, mỗi dòng ghi chỉ số của một số hạng bị xóa đi theo thứ tự tăng dần. Tiếp theo là một dòng ghi  $M$  là số vi phạm tính đơn điệu, nếu  $M > 0$ , trong  $M$  dòng tiếp theo, mỗi dòng ghi một chỉ số  $u$  với ý nghĩa tại đó điều kiện đơn điệu bị vi phạm tức là  $a_u$  là số hạng còn lại nhng số hạng còn lại tiếp theo không lớn hơn  $a_u$ , các chỉ số  $u$  cũng ghi theo thứ tự tăng dần.

Ví dụ:

DAYK.INP	DAYK.OUT
5 2	0
1	1
2	3
3	
2	
4	

```

{$A+,B-,D+,E+,F-,G-,I-,L+,N+,O-,P-,Q+,R+,S+,T-,V+,X+}
{$M 16384,0,655360}

```

```
uses crt,dos;
```

```
const
```

```

dtfi='DAYK.IN';
dtfo='DAYK.OU';
dtfm="";
kqm:array[0..9] of integer=(811,63,112,266,279,364,450,451,623,717);

```

```
var
```

```

Error: array[0..9] of integer;
Diem: array[0..9] of real;
p: integer;
tfi,tfo,tfm: string[12];
d: SearchRec;
fi,fo,fm: text;

```

```

N,K: integer;
a: array[1..1000] of integer;
count: integer;
dd: array[1..1000] of integer;

```



```
ok: boolean;  
Gdm, Gdo: integer;  
Gd: array[1..11] of integer;
```

```
procedure Docdl;  
var i: integer;  
begin  
  assign(fi,tfi); reset(fi);  
  read(fi,N,K);  
  for i:=1 to N do read(fi,a[i]);  
  close(fi);  
end;
```

```
procedure Check; {thu tuc kiem tra k gian doan}  
var v, iv, i: integer;  
begin  
  ok:=false;  
  v:=-maxint;  
  iv:=0;  
  Gdm:=0; {so luong diem gian doan}  
  for i:=1 to N do  
    if (dd[i]=0) then  
      begin  
        if a[i]<=v then  
          begin  
            inc(Gdm);  
            Gd[Gdm]:=iv;  
            if Gdm>k then exit;  
          end;  
        v:=a[i];  
        iv:=i;  
      end;  
  ok:=true;  
end;
```

```
procedure Cham;  
var u,v,i: integer;  
begin  
  readln(fo,Count); {doc so luong so bi xoa}  
  if count>kqm[p] then  
    begin  
      error[p]:=3; {Ket qua khong toi uu}  
      exit;  
    end;  
  fillchar(dd,sizeof(dd),0);  
  v:=0;  
  for i:=1 to count do  
    begin  
      readln(fo,u);  
      if u<=v then  
        begin  
          error[p]:=4; {Cac vi tri xoa khong in tang dan}  
          exit;  
        end;  
    end;
```

```
    dd[u]:=1;
    v:=u;
end;
Check;
if not ok then
begin
    error[p]:=5; {Vi phạm tính k gian doan}
    exit;
end;
readln(fo,Gdo);
if Gdo<>Gdm then
begin
    error[p]:=6; {Tính số lượng các điểm gian doan sai}
    exit;
end;
for i:=1 to Gdo do
begin
    readln(fo,u);
    if u<>Gd[i] then
begin
        error[p]:=7; {In sai các điểm gian doan}
        exit;
    end;
end;
diem[p]:=1.0;
end;
```

```
procedure InError(p: LongInt);
begin
    textcolor(red);
    case Error[p] of
        0: writeln;
        1: writeln('Khong co file OUTPUT');
        2: writeln('Loi doc file OUTPUT');
        3: writeln('Ket qua khong toi uu');
        4: writeln('Cac vi tri xoa khong in tang dan');
        5: writeln('Vi phạm tính k gian doan');
        6: writeln('Tính số lượng các điểm gian doan sai');
        7: writeln('In sai các điểm gian doan');
    end;
    textcolor(white);
end;
```

```
procedure Inkq;
var T: real;
begin
    clrscr;
    writeln;
    writeln;
    writeln;
    textcolor(white);
    writeln('*****');
    writeln(' *      Ket qua cham bai      *');
    writeln('*****');
```

```
T:=0.0;
for p:=0 to 9 do
begin
  write('          TEST ',p,' ',Diem[p]:4:1,' Diem. ');
  InError(p);
  T:=T+diem[p];
end;
writeln('          -----');
textcolor(green);
writeln('          Tong cong = ',T:4:1,' Diem');
textcolor(white);
end;

procedure XoaFile;
begin
  for p:=0 to 9 do
  begin
    tfo:=dtfo+chr(p+48);
    FindFirst(tfo,archive,d);
    if DosError<>0 then continue;
    assign(fo,tfo);
    erase(fo);
  end;
end;

BEGIN
  fillchar(Error,sizeof(Error),0);
  fillchar(Diem,sizeof(Diem),0);
  for p:=0 to 9 do
  begin
    tfi:=dtfi+chr(p+48);
    tfm:=dtfm+chr(p+48);
    tfo:=dtfo+chr(p+48);
    FindFirst(tfo,Archive,d);
    if DosError<>0 then
    begin
      Error[p]:=1;
      continue;
    end;
    Docdl;
    assign(fo,tfo); reset(fo);
    Cham;
    close(fo);
  end;
  Inkq;
  XoaFile;
END.
```

### **BÀI TOÁN 18. Phát quà Noen**

Nhân dịp lễ Giáng sinh, một công ty dịch vụ cần phải thực hiện việc chuyển phát một số lượng lớn các gói quà đến địa chỉ của chúng không muộn hơn thời gian hạn định. Có hai phương tiện để chuyển phát: xe nhỏ và xe lớn. Các gói quà cũng được phân ra làm hai loại: lớn và nhỏ. Gói quà nhỏ có thể chuyển phát bằng xe nhỏ cũng như xe lớn. nhưng gói quà lớn thì chỉ có thể chuyển phát bằng xe lớn. Đối với mỗi gói quà ta biết thời gian cần thiết để chuyển phát nó mà ta gọi là *thời gian phát* của gói quà. Tại mỗi thời điểm, mỗi xe chỉ có thể chở một gói quà. Công ty

được biết thời gian hạn định, nghĩa là khoảng thời gian mà trong đó cần chuyển phát được số lượng nhiều nhất có thể các gói quà. Không cần lo lắng gì về việc gì sẽ xảy ra sau thời gian hạn định này.

Nh vậy nhiệm vụ chuyển phát quà được thực hiện nh sau:

- Một số gói quà nhỏ sẽ được vận chuyển bởi xe nhỏ và tổng thời gian phát của chúng không vượt quá thời gian hạn định;
- Một số gói quà lớn và nhỏ sẽ được vận chuyển bằng xe lớn và tổng thời gian phát của chúng không vượt quá thời gian hạn định.

**Yêu cầu:** Xác định số lượng lớn nhất các gói quà có thể chuyển phát trong thời gian hạn định.

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

- Dòng đầu chứa số nguyên T ( $1 \leq T \leq 1000$ ) là thời gian hạn định
- Dòng thứ hai chứa số nguyên N ( $1 \leq N \leq 500$ ) là số lượng các gói quà nhỏ
- Mỗi dòng từ dòng thứ 3 đến dòng thứ N+2 chứa một số nguyên dương không vượt quá 1000. Đây là thời gian phát của các gói quà nhỏ. Các số này được liệt kê theo thứ tự không giảm.
- Dòng thứ N+3 chứa số nguyên M ( $1 \leq M \leq 500$ ) là số lượng gói quà lớn;
- Mỗi dòng từ dòng thứ N+4 đến dòng thứ N+M+3 chứa một số nguyên dương không vượt quá 1000. Đây là thời gian phát của các gói quà lớn. Các số này được liệt kê theo thứ tự không giảm.

**Kết quả:** Ghi ra file văn bản DELIVERY.OUT một số duy nhất là số lượng lớn nhất các gói quà có thể chuyển phát trong thời gian hạn định.

**Ví dụ:**

DELIVERY.INP	DELIVERY.OUT	Giải thích
10 8 2 2 2 2 2 4 4 4 4 3 3 6 6	8	Thời gian hạn định: 10.Có: 8 gói quà nhỏ (5 gói có thời gian phát là 2, 3 gói có thời gian phát là 4)  4 gói quà lớn (2 gói với thời gian phát 3, 2 gói với thời gian phát 6)

Đáp số: Nhiều nhất trong thời gian hạn định có thể chuyển phát 8 gói quà: 4 gói nhỏ với thời gian phát 2 trên xe nhỏ, và một gói nhỏ với thời gian phát 4 cùng với 2 gói to với thời gian phát 3 trên xe lớn

```
{ $M 32000,0,655360 }
```

```
{ $R+,S+,Q+ }
```

```
uses crt;
```

```
const
```

```
nn=1000;
```

```
tt=1000;
```

```
inpfile = 'DELIVERY.INP';
```

```
outfile = 'DELIVERY.OUT';
```

```
var
```

```
inp, out: text;
```

```
n, kl, m, mn, t: integer;
```

```
c, ik : array[0..nn*2] of integer;
```

```
f1, f2: array[0..tt] of integer;
```

```
procedure qsort(l,r:integer);
var i,j,tg,mid:integer;
begin
  mid:=c[(l+r) div 2];
  i:=l;j:=r;
  while i<=j do
  begin
    while c[i]<mid do inc(i);
    while c[j]>mid do dec(j);
    if i<=j then
    begin
      tg:=c[i];c[i]:=c[j];c[j]:=tg;
      tg:=ik[i];ik[i]:=ik[j];ik[j]:=tg;
      inc(i);dec(j);
    end;
  end;
  if l<j then qsort(l,j);
  if i<r then qsort(i,r);
end;
```

```
procedure Chuanbi;
var i, tg:Integer;
begin
  assign(inp,inpfile);reset(inp);
  readln(inp,t);
  readln(inp,n);
  mn:=0;
  for i:=1 to n do
  begin
    read(inp, tg);
    inc(mn);
    c[mn]:= tg;
    ik[mn]:=1;
  end;
  readln(inp,m);
  for i:=1 to m do
  begin
    read(inp, tg);
    inc(mn);
    c[mn]:= tg;
    ik[mn]:=2;
  end;
  close(inp);

  qsort(1,mn);

end;
```

```
procedure Solve;
var i, j, jj, tg: integer;
    p: longint;
begin
  fillchar(f1,sizeof(f1),255);
```

```

f1[0]:=0;
f2:=f1;
tg:=0; p:=0;
for i:=1 to mn do
begin
  p:=p+c[i];
  if ik[i]=1 then inc(tg,c[i]);
  if (tg>t*2)and(p>t*2) then break;
  for j:=0 to t do
    if (f1[j]<>-1) then
      begin
        if (p-j<=t)and(f2[j]<=f1[j]) then f2[j]:=f1[j]+1;
        jj:=j+c[i];
        if (ik[i]=1)and(jj<=t)and(f2[jj]<=f1[j]) then f2[jj]:=f1[j]+1;
      end;
  f1:=f2;
end;
kl:=0;
for j:=0 to t do
  if f1[j]>kl then kl:=f1[j];
assign(out,outfile);rewrite(out);
writeln(out,kl);
close(out);
end;

BEGIN
  Chuanbi;
  Solve;
END.

```

## BÀI TOÁN 19. Hộp thư điện tử

Một người sử dụng INTERNET đặt yêu cầu nhận thông tin về một số chủ đề khác nhau từ một số địa chỉ truy nhập. Chủ của các địa chỉ truy nhập này sẽ gửi thông tin yêu cầu vào hòm th của người đặt hàng. Mỗi thông tin nhận được từ địa chỉ truy nhập sẽ được ghi vào một danh mục trong máy của người sử dụng dưới dạng một file mà để ngắn gọn ta sẽ gọi là một thông báo. Để thuận tiện cho việc tra cứu, người sử dụng quyết định xây dựng các cặp tài liệu, mỗi cặp chứa thông tin về cùng một chủ đề. Trước khi đọc tài liệu người sử dụng sẽ sao chép chúng từ danh mục các thông báo nhận được vào các cặp tương ứng.

Chương trình hộp th điện tử gắn trên máy của người sử dụng cho phép sau "*một thao tác*" chuyển từ danh mục thông báo vào cặp tài liệu:

- Một thông báo từ danh mục hoặc
- Một dãy các thông báo liên tiếp nhau trong danh mục về cùng một chủ đề

Việc chuyển thông báo không nhất thiết phải bắt đầu từ đầu danh mục.

Cần tìm cách chuyển các thông báo trong danh mục vào các cặp tương ứng đòi hỏi số thao tác phải thực hiện là ít nhất.

**Ví dụ:** Giả sử người sử dụng muốn thu thập thông tin về các chủ đề A, B, C, D. Giả sử danh mục các thông báo nhận được theo trình tự thuộc về các chủ đề (A, C, D, C, B, B, C). Việc di chuyển vào cặp tài liệu có thể thực hiện nh sau: Đầu tiên di chuyển hai thông báo B, khi đó danh mục còn lại là (A, C, D, C, C). Tiếp theo thực hiện việc di chuyển thông báo D, rồi thông báo A và cuối cùng di chuyển nốt 3 thông báo C liên nhau. Cách làm này đòi hỏi 4 thao tác.

**Dữ liệu:** Vào từ file văn bản EMAIL.INP gồm một dòng chứa số nguyên dương N ( $0 < N \leq 50$ ) là số thông báo trong danh mục, tiếp đến là N số nguyên là dãy số cầu các chủ đề của dãy các thông báo trong danh mục cần truyền.

**Kết quả:** Ghi ra file EMAIL.OUT số thao tác ít nhất cần thực hiện.

**Ví dụ:**

EMAIL.INP	EMAIL.OUT
7 1 3 4 3 2 2 3	4

```
{ $A+,B-,D+,E+,F-,G-,I-,L+,N+,O-,P-,Q+,R+,S+,T-,V+,X+}  
{ $M 16384,0,655360}
```

```
uses crt,dos;
```

```
const  
  dtfi="";  
  dtfo='EMAIL.OU';  
  dtfm='EMAIL.AS';
```

```
var  
  Error: array[0..9] of integer;  
  Diem: array[0..9] of real;  
  p: integer;  
  tfi,tfo,tfm: string[12];  
  d: SearchRec;  
  fi,fo,fm: text;  
  Smau,Sout: longint;
```

```
procedure Docdl;  
begin  
  assign(fm,tfm); reset(fm);  
  readln(fm,Smau);  
  close(fm);  
end;
```

```
procedure Cham;  
begin  
  readln(fo,Sout);  
  if ioresult<>0 then  
    begin  
      error[p]:=2;  
      exit;  
    end;  
  if Sout=Smau then diem[p]:=1.0;  
end;
```

```
procedure InError(p: LongInt);  
begin  
  textcolor(red);  
  case Error[p] of  
    0: writeln;  
    1: writeln('Khong co file OUTPUT');  
    2: writeln('Loi doc file OUTPUT');  
  end;  
  textcolor(white);  
end;
```

```
procedure Inkq;  
var T: real;
```

```
begin
  clrscr;
  writeln;
  writeln;
  writeln;
  textcolor(white);
  writeln('*****');
  writeln(' *      Ket qua cham bai THAM HOI CHO (HSGQG 2000) *');
  writeln('*****');
  T:=0.0;
  for p:=0 to 9 do
    begin
      write('      TEST ',p,': ',Diem[p]:4:1,' Diem. ');
      InError(p);
      T:=T+diem[p];
    end;
  writeln('-----');
  textcolor(green);
  writeln('      Tong cong = ',T:4:1,' Diem');
  textcolor(white);
end;

procedure XoaFile;
begin
  for p:=0 to 9 do
    begin
      tfo:=dtfo+chr(p+48);
      FindFirst(tfo,archive,d);
      if DosError<>0 then continue;
      assign(fo,tfo);
      erase(fo);
    end;
end;

BEGIN
  fillchar(Error,sizeof(Error),0);
  fillchar(Diem,sizeof(Diem),0);
  for p:=0 to 9 do
    begin
      tfi:=dtfi+chr(p+48);
      tfm:=dtfm+chr(p+48);
      tfo:=dtfo+chr(p+48);
      FindFirst(tfo,Archive,d);
      if DosError<>0 then
        begin
          Error[p]:=1;
          continue;
        end;
      Docdl;
      assign(fo,tfo); reset(fo);
      Cham;
      close(fo);
    end;
  Inkq;
```



XoaFile;

END.

## BÀI TOÁN 20. Khôi phục biểu thức đúng

Cho một biểu thức chỉ gồm các kí tự (, ), [, ], {, }, Hãy cho biết biểu thức đó có đúng đắn hay không. Nếu không Hãy tìm cách thêm một số ít nhất các kí tự thuộc một trong 6 loại trên để nhận được biểu thức đúng.

VD : ( ) ( { ( ) } ) (

Là biểu thức không đúng, biểu thức cần đưa ra là

( [ ] ) ( { ( ) } ) ( )

Input: Cho trong file EXPRESS.INP

Một chuỗi thể hiện biểu thức gồm n kí tự (n ≤ 200)

Output: Ghi ra file EXPRESS.OUT:

- Dòng đầu ghi một số nguyên là số lượng ít nhất các ký tự cần phải thêm vào để được một biểu thức đúng
- Dòng thứ hai ghi chuỗi ký tự là biểu thức đúng (Từ đầu dòng)

*Ví dụ:*

EXPRESS.INP

( ) ( { ( ) } ) (

{ \$r+ }

const

tfi='Express.inp';

tfo='Express.out';

maxN=201;

type

mang1=array[1..maxN] of integer;

var

fi, fo: text;

s : array[1..maxN] of char;

is: array[1..maxN] of char;

N: integer;

c: array[1..maxN] of ^mang1;

t: array[1..maxN] of ^mang1;

procedure cp;

var i: integer;

begin

for i:=1 to maxN do new(c[i]);

for i:=1 to maxN do new(t[i]);

end;

procedure doc;

begin

assign(fi,tfi); reset(fi);

N:=0;

while not seekeoln(fi) do

begin

inc(n);

read(fi,s[n]);

end;

close(fi);

end;

EXPRESS.OUT

3

( [ ] ) ( { ( ) } ) ( )

```
function Cap(i,j: integer): boolean;
begin
  Cap:=((s[i]='(' and (s[j]='))') or
        ((s[i]='[' and (s[j]=']')) or
        ((s[i]='{' and (s[j]='}')));
end;

procedure Tinh;
var i,j,k,l: integer;
begin
  for i:=1 to n do
    for j:=1 to n do c[i]^j:=0;
  for i:=1 to N do
    begin
      c[i]^i:=1;
      t[i]^i:=0;
    end;
  for k:=1 to n-1 do
    for i:=1 to n-k do
      begin
        j:=i+k;
        if Cap(i,j) then
          begin
            c[i]^j:=c[i+1]^j-1;
            t[i]^j:=-1;
          end
        else
          begin
            c[i]^j:=maxint; t[i]^j:=0;
            for l:=i to j-1 do
              if c[i]^j>c[i]^l+c[l+1]^j then
                begin
                  c[i]^j:=c[i]^l+c[l+1]^j;
                  t[i]^j:=l;
                end;
            end;
          end;
        end;
      end;
    end;

  procedure Trace(k,l: integer);
  var r: integer;
  begin
    if k>l then exit;
    if k=l then
      begin
        case s[k] of
          '(': is[k]:=')';
          ')': is[k]:= '(';
          '[': is[k]:=']';
          ']': is[k]:= '[';
          '{': is[k]:= '}'';
          '}': is[k]:= '{';
        end;
      end;
    end;
```

```

    exit;
end;
if Cap(k,l) then Trace(k+1,l-1) else
begin
    r:=t[k]^l;
    Trace(k,r);
    Trace(r+1,l);
end;
end;

procedure tim;
var i: integer;
begin
    for i:=1 to N do is[i]:=#32;
    Trace(1,N);
end;

procedure viet;
var i: integer;
begin
    assign(fo,tfo); rewrite(fo);
    writeln(fo,c[1]^n);
    for i:=1 to n do
        if is[i]=#32 then write(fo,s[i]) else
            case is[i] of
                '(',')': write(fo,'()');
                '[,]': write(fo,'[]');
                '{,}': write(fo,'{}');
            end;
    close(fo);
end;

BEGIN
    cp;
    doc;
    tinh;
    tim;
    viet;
END.
```

## BÀI TOÁN 21. Quả bóng rơi

Một quả bóng rơi từ trên cao xuống với tốc độ 1m/s, khi chạm vào tấm ván nằm ngang ở dưới (kể cả tại đầu mút của tấm ván), nó sẽ lăn sang phải/trái đều với vận tốc 1m/s tùy theo sự lựa chọn của bạn theo đường thẳng nằm ngang trong mặt phẳng vuông góc với tấm ván và mặt phẳng rơi, sau đó lại tiếp tục rơi xuống dưới vẫn với tốc độ 1 m/s, cứ tiếp tục như vậy cho đến khi chạm tới nền; nền được xem là một tấm ván với kích thước vô hạn. Bóng không được rơi tự do quá một đoạn M cho trước. Quá trình rơi diễn ra trên một mặt phẳng vuông góc với nền nên ta có thể định vị hệ thống bằng một hệ trục tọa độ mà trục Ox là nền, mỗi tấm ván nh một đoạn thẳng song song với nền, quả bóng nh một điểm.

Cho biết vị trí ban đầu của quả bóng, vị trí và độ dài các tấm ván, hãy chọn hành trình đi nhanh nhất cho quả bóng.

**Dữ liệu** vào được cho bởi file FALL.INP, trong đó dòng thứ nhất ghi 4 số N, X, Y, M trong đó N là số tấm ván, (x,y) là tọa độ ban đầu của quả bóng, M là đoạn giới hạn việc rơi tự do, tiếp theo là N dòng, dòng thứ i ghi ba số x1, x2, h, hai số đầu là hoành độ của hai đầu mút ( $x_1 < x_2$ ), số thứ ba

là tung độ của tấm ván thứ i. Mọi số đều nguyên ( $1 \leq N \leq 1000$ ), các hoành độ có giá trị tuyệt đối không quá 20000, các tung độ dương và không quá 20000. Không có hai tấm ván nào có điểm chung và bài toán luôn có lời giải.

**Kết quả** ghi ra file FALL.OUT nh sau: dòng thứ nhất ghi thời gian từ lúc bắt đầu rơi đến khi chạm nền, tiếp theo là một số dòng, mỗi dòng ghi hai số T, H mà T là số hiệu tấm ván, H=0/1 là hướng lần được chọn: 0/1 tương ứng với trái/phải

Ví dụ:

FALL.INP	FALL.OUT
3 8 17 20	23
0 10 8	2 1
0 10 13	1 1
4 14 3	3 1

```
const
  tfi      = 'FALL.INP';
  tfo      = 'FALL.OUT';
  maxN     = 1002;
  VC       = maxlongint;
```

```
type
  van      = record
    x1, x2, H: longint;
    id: longint;
  end;
```

```
var
  fi, fo   : text;
  N        : longint;
  x,y,delta : longint;
  a        : array[0..maxN] of van;
  i0       : longint;

  F        : array[1..2,0..maxN] of longint;
  Tr       : array[1..2,0..maxN] of longint;

  Min      : longint;
  sl       : longint;
  kq       : array[1..maxN] of longint;
```

```
procedure Docdl;
var i: longint;
begin
  assign(fi,tfi); reset(fi);
  readln(fi,N,x,y,delta);
  for i:=1 to N do
    with a[i] do
      readln(fi,x1,x2,H);
  close(fi);
end;
```

```
procedure Trao(var u,v: van);
var w: van;
begin
  w:=u;
```

```
u:=v;  
v:=w;  
end;
```

```
procedure SapXep;  
var i,j: longint;  
begin  
  for i:=1 to N-1 do  
    for j:=i+1 to N do  
      if a[i].H<a[j].H then  
        Trao(a[i],a[j]);  
    end;  
end;
```

```
procedure Qsort(k,l: longint);  
var r,i,j: longint;  
begin  
  r:=(k+l) div 2;  
  i:=k;  
  j:=l;  
  repeat  
    while a[i].H>a[r].H do inc(i);  
    while a[j].H<a[r].H do dec(j);  
    if i<=j then  
      begin  
        Trao(a[i],a[j]);  
        inc(i); dec(j);  
      end;  
  until i>j;  
  if k<j then Qsort(k,j);  
  if i<l then Qsort(i,l);  
end;
```

```
procedure Ke(i1,k1,i2,k2: longint; var ok: boolean; var d: longint);  
var xx,yy,i: longint;  
begin  
  ok:=false;  
  if a[i1].H-a[i2].H>delta then exit;  
  if k1=1 then xx:=a[i1].x1 else xx:=a[i1].x2;  
  yy:=a[i1].H;  
  if (xx<a[i2].x1) or (xx>a[i2].x2) then exit;  
  
  for i:=i1+1 to i2-1 do  
    if (a[i].x1<=xx) and (xx<=a[i].x2) then exit;  
  
  ok:=true;  
  if k2=1 then d:=xx-a[i2].x1+yy-a[i2].H  
    else d:=a[i2].x2-xx+yy-a[i2].H;  
end;
```

```
function RoiDuoc(k,i: longint): boolean;  
var j, xx, yy: longint;  
begin  
  RoiDuoc:=false;  
  if k=1 then xx:=a[i].x1 else xx:=a[i].x2;
```

```
yy:=a[i].H;
for j:=i+1 to N do
  if (a[j].x1<=xx) and (xx<=a[j].x2) then exit;
RoiDuoc:=true;
end;

procedure TimNguoc;
var i,k,ktk,kti: longint;
begin
  (* Tim diem ket thuc *)
  Min:=VC;
  for i:=N downto i0 do
    if (a[i].H<=delta) then
      for k:=1 to 2 do
        if (F[k,i]<VC) and RoiDuoc(k,i) and (F[k,i]+a[i].H<Min) then
          begin
            Min:=F[k,i]+a[i].H;
            ktk:=k;
            kti:=i;
          end;
        if min=VC then exit;
        i:=kti;
        k:=ktk;
        sl:=0;
      repeat
        inc(sl);
        if k=1 then kq[sl]:=a[i].id else kq[sl]:=-a[i].id;
        i:=Tr[k,i];
        if i<0 then
          begin
            i:=-i;
            k:=2;
          end
        else if i<maxN then k:=1;
      until i=maxN;
    end;

procedure Solve;
var i,k,j,l: longint;
    ok: boolean;
    d: longint;
begin
  for i:=1 to N do a[i].id:=i;
  SapXep;
  {Qsort(1,N);}
  i0:=1;
  while (i0<N) and (a[i0].H>y) do inc(i0);
  dec(i0);
  with a[i0] do
    begin
      x1:=x;
      x2:=x;
      H:=y;
    end;
```

```

F[1,i0]:=0; Tr[1,i0]:=maxN;
F[2,i0]:=0; Tr[2,i0]:=maxN;
for i:=i0+1 to N do
  for k:=1 to 2 do
    begin
      F[k,i]:=VC;
      for j:=i-1 downto i0 do
        for l:=1 to 2 do
          if F[l,j]<VC then
            begin
              Ke(j,l,i,k,ok,d);
              if ok and (F[k,i]>F[l,j]+d) then
                begin
                  F[k,i]:=F[l,j]+d;
                  case 1 of
                    1: Tr[k,i]:=j;
                    2: Tr[k,i]:=-j;
                  end;
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  TimNguoc;
end;

procedure Inkq;
var i: longint;
begin
  assign(fo,tfo); rewrite(fo);
  if Min=VC then writeln(fo,-1) else
    begin
      writeln(fo,Min);
      for i:=sl-1 downto 1 do
        if kq[i]>0 then writeln(fo,kq[i],' ',0)
        else writeln(fo,-kq[i],' ',1);
      end;
    end;
  close(fo);
end;

BEGIN
  Docdl;
  Solve;
  Inkq;
END.

```

## BÀI TOÁN 22. Hành tinh đầm lầy

Hành tinh FEN nổi tiếng bởi vô số các đầm lầy kỳ lạ của mình. Mỗi đầm lầy có hình gần nh là chữ nhật. Kết quả quan sát nhiều năm cho thấy đầm lầy này có thể chia thành lối ô vuông. Mỗi ô vuông có 4 trạng thái khác nhau (ký hiệu là 0,1,2,3). Sau 1s thì trạng thái 0 thành trạng thái 3, trạng thái 1 thành trạng thái 0, trạng thái 2 thành trạng thái 1, trạng thái 3 thành trạng thái 2. Trạng thái 0 là trạng thái không thể đứng được ở trên ô đó, các trạng thái khác là trạng thái có thể đứng được ở trên đó.

*Yêu cầu:* Cho bản đồ đầm lầy kích thước  $M \times N$  ô ( $M$  hàng,  $N$  cột. Các hàng được đánh số từ 1 đến  $M$  từ trên xuống dưới và các cột được đánh số từ 1 đến  $N$  từ trái sang phải), hãy xác định chương trình điều khiển đa robot từ ô ở góc tây nam sang ô ở góc đông bắc với thời gian ngắn nhất

có thể. Giả thiết rằng luôn tồn tại một hành trình nh vậy. Tại thời điểm ban đầu (thời điểm 0) trạng thái của ô góc tây nam là 3. Sau 1s robot có thể chuyển đến ô chung cạnh (nếu nh ô này sau giây đó có trạng thái khác 0) hoặc đứng yên (nếu có thể đứng được tại ô đang đứng sau giây đó)

*Dữ liệu:* Vào từ file FEN.INP:

- Dòng đầu ghi số nguyên M, N,  $1 \leq M, N \leq 50$
- M dòng tiếp theo: mỗi dòng ghi N số nguyên dương trong phạm vi từ 0 đến 3

*Kết quả:* Đa ra file FEN.OUT :

- Dòng đầu chứa T là thời gian dịch chuyển robot.
- Dòng thứ 2 chứa chương trình điều khiển với qui ước N- bắc, E- đông, S- nam, W - tây, P - đứng yên (mỗi ký tự là chuyển động sau 1s)

*Ví dụ:*

FEN.INP	FEN.OUT
3 4	4
3 2 3 3	ENEEN
0 0 1 2	
3 3 1 2	

```

{*****
      HANH TINH DAM LAY
*****}
{$A+,B-,D+,E+,F-,G-,I+,L+,N+,O-,P-,Q-,R-,S+,T-,V+,X+,Y+}
{$M 16384,0,655360}
uses crt;

const
  tfi='FEN.INP';
  tfo='FEN.OUT';
  maxN=50;
  Qmax=maxN*maxN;
  GT: array[0..3,0..3] of byte=((0,3,2,1),
                                (1,0,3,2),
                                (2,1,0,3),
                                (3,2,1,0));
  dh: array[1..5] of shortint=(0,-1,0,1,0);
  dc: array[1..5] of shortint=(1,0,-1,0,0);

type
  mang=array[1..60000] of byte;

var
  fi,fo: text;
  M,N: integer;
  a: array[1..maxN,1..maxN] of byte;
  q1f,q1l,q2f,q2l: integer;
  q1,q2: array[1..Qmax,1..2] of byte;
  Pre: byte;
  x: ^mang;
  TGMin: Word;

procedure Docdl;
var i,j: integer;
begin
  readln(fi,M,N);
  for i:=1 to M do

```



```
begin
  for j:=1 to N do read(fi,a[i,j]);
  readln(fi);
end;
end;

procedure InitQ1;
begin
  q1f:=1;
  q1l:=1;
end;

function Inside(i,j: byte): boolean;
begin
  Inside:=(i>=1) and (i<=M) and (j>=1) and (j<=N);
end;

procedure DuyetCR1(hkt,ckt: byte; var tr: byte; var Time: word);
var ok: boolean;
    dd: array[1..maxN,1..maxN] of byte;
    i,j,k,i1,j1: byte;
begin
  time:=0;
  q1f:=1; q1l:=1;
  q1[q1l,1]:=M; q1[q1l,2]:=1; inc(q1l); {PutQ1(M,1)}
  ok:=false;
  repeat
    q2f:=1; q2l:=1; {InitQ2}
    fillchar(dd,sizeof(dd),0);
    time:=time+1;
    while (q1f<>q1l) do
      begin
        i:=q1[q1f,1]; j:=q1[q1f,2]; inc(q1f); {GetQ1(i,j)}
        for k:=1 to 5 do
          begin
            i1:=i+dh[k]; j1:=j+dc[k];
            if inside(i1,j1) and (GT[a[i1,j1],Time mod 4]<>0) then
              if dd[i1,j1]=0 then
                begin
                  q2[q2l,1]:=i1; q2[q2l,2]:=j1; inc(q2l); {PutQ2(i1,j1)}
                  dd[i1,j1]:=1;
                  if (i1=hkt) and (j1=ckt) then
                    begin
                      ok:=true;
                      Tr:=k;
                      exit;
                    end;
                end;
              end;
          end;
        end;
      end;
    Q1:=Q2;
    q1f:=q2f;
    q1l:=q2l;
  until ok;
```

end;

```

procedure DuyetCR2(hkt,ckt: byte; Tg: word; var tr: byte);
var ok: boolean;
    dd: array[1..maxN,1..maxN] of byte;
    i,j,k,i1,j1: byte;
    Time: integer;
begin
    time:=0;
    q1f:=1; q1l:=1;
    q1[q1l,1]:=M; q1[q1l,2]:=1; inc(q1l); {PutQ1(M,1)}
    ok:=false;
    repeat
        q2f:=1; q2l:=1; {InitQ2}
        fillchar(dd,sizeof(dd),0);
        time:=time+1;
        while (q1f<>q1l) do
            begin
                i:=q1[q1f,1]; j:=q1[q1f,2]; inc(q1f); {GetQ1(i,j)}
                for k:=1 to 5 do
                    begin
                        i1:=i+dh[k]; j1:=j+dc[k];
                        if inside(i1,j1) and (GT[a[i1,j1],Time mod 4]<>0) then
                            if dd[i1,j1]=0 then
                                begin
                                    q2[q2l,1]:=i1; q2[q2l,2]:=j1;inc(q2l); {PutQ2(i1,j1)}
                                    dd[i1,j1]:=1;
                                    if (i1=hkt) and (j1=ckt) and (Time=Tg) then
                                        begin
                                            ok:=true;
                                            Tr:=k;
                                            exit;
                                        end;
                                end;
                            end;
                        end;
                    end;
                Q1:=Q2;
                q1f:=q2f;
                q1l:=q2l;
            until ok;
        end;
    end;
end;

```

```

procedure TimNguoc;
var u,v: byte;
    i: word;
begin
    new(x);
    u:=1; v:=N;
    for i:=TgMin downto 1 do
        begin
            DuyetCR2(u,v,i,Pre);
            x^[i]:=Pre;
            u:=u-dh[Pre];
        end;
    end;
end;

```

```

    v:=v-dc[Pre];
end;
end;

procedure Inkq;
var i: word;
begin
    writeln(fo,TgMin);
    for i:=1 to TgMin do
        case x^[i] of
            1: write(fo,'E');
            2: write(fo,'N');
            3: write(fo,'W');
            4: write(fo,'S');
            5: write(fo,'P');
        end;
    end;
end;

```

```

BEGIN
    assign(fi,tfi); reset(fi);
    assign(fo,tfo); rewrite(fo);
    Docdl;
    DuyetCR1(1,N,Pre,TgMin);
    TimNguoc;
    Inkq;
    close(fi); close(fo);
END.

```

### **BÀI TOÁN 23. Biến đổi chuỗi ký tự (khoảng cách Levenshtein)**

Với một chuỗi ký tự S cho trước, ta có thể thực hiện các phép biến đổi sau:

- D: Xoá một ký tự của chuỗi S. Ký hiệu D i trong đó i là vị trí cần xoá
- I: Chèn trước vị trí t của chuỗi S một ký tự c nào đó. Ký hiệu I t c. Qui định thêm về vị trí chèn: nếu chuỗi S có độ dài k, vị trí chèn là 1, 2, 3, ..., k+1, chèn ở vị trí k+1 có nghĩa là viết thêm vào cuối chuỗi S
- R: Thay ký tự thứ t của S bởi ký tự c nào đó. Ký hiệu R t c

Giả sử X và Y là hai chuỗi ký tự. Độ dài chuỗi X là n, độ dài chuỗi Y là m ( $0 \leq m, n \leq 100$ )

Hãy tìm một dãy gồm ít nhất các phép biến đổi biến chuỗi X thành chuỗi Y (số phép biến đổi ít nhất này gọi là khoảng cách giữa hai chuỗi)

**Dữ liệu** vào cho trong file CHANGEST.INP gồm hai dòng

- Dòng thứ nhất là chuỗi X
- Dòng thứ hai là chuỗi Y

**Kết quả** ghi ra file CHANGEST.OUT:

- Dòng thứ nhất ghi số K, đó là khoảng cách giữa hai chuỗi
- K dòng tiếp theo mỗi dòng ghi ký hiệu một phép biến đổi theo trình tự thực hiện để biến X thành Y

*Ví dụ:*

CHANGEST.INP  
 erttryui  
 tyuhj

CHANGEST.OUT  
 6  
 D 1  
 D 1  
 D 1  
 D 1  
 I 4 h  
 R 5 j

```
const
  tfi      = 'CHANGEST.INP';
  tfo      = 'CHANGEST.OUT';
  maxN     = 100;

var
  fi, fo   : text;
  x, y     : string;
  N, M     : integer;

  L        : array[0..maxN,0..maxN] of integer;

  sl       : integer;
  a,b,kieu : array[1..maxN] of integer;
  c        : array[1..maxN] of char;

procedure Docdl;
begin
  assign(fi,tfi); reset(fi);
  readln(fi,x);
  readln(fi,y);
  N:=length(x);
  M:=length(y);
  close(fi);
end;

procedure Tinh;
var i,j: integer;
begin
  for j:=1 to N do L[0,j]:=j;
  for i:=0 to M do L[i,0]:=i;
  for i:=1 to M do
    for j:=1 to N do
      if x[j]=y[i] then L[i,j]:=L[i-1,j-1] else
        begin
          L[i,j]:=L[i,j-1]+1;
          if L[i,j]>L[i-1,j]+1 then L[i,j]:=L[i-1,j]+1;
          if L[i,j]>L[i-1,j-1]+1 then L[i,j]:=L[i-1,j-1]+1;
        end;
    end;
  end;

procedure Tim;
var u,v: integer;
begin
  sl:=0;
  u:=M;
  v:=N;
  repeat
    if y[u]=x[v] then
      begin
        dec(u);
        dec(v);
      end
  end
```

```
else
begin
inc(sl);
if (u>=1) and (L[u,v]=L[u-1,v]+1) then (* chen *)
begin
a[sl]:=v;
b[sl]:=u;
c[sl]:=y[u];
kieu[sl]:=1;
dec(u);
end
else if (v>=1) and (L[u,v]=L[u,v-1]+1) then (* xoa *)
begin
a[sl]:=v;
b[sl]:=u;
c[sl]:=y[u];
kieu[sl]:=2;
dec(v);
end
else
begin
a[sl]:=v;
b[sl]:=u;
c[sl]:=y[u];
kieu[sl]:=3;
dec(u);
dec(v);
end;
end;
until (u=0) and (v=0);
end;
```

```
procedure Inkq;
var i: integer;
begin
assign(fo,tfo); rewrite(fo);
writeln(fo,sl);
for i:=sl downto 1 do
case kieu[i] of
1: writeln(fo,'I ',b[i],' ',c[i]);
2: writeln(fo,'D ',b[i]+1);
3: writeln(fo,'R ',b[i],' ',c[i]);
end;
close(fo);
end;
```

```
BEGIN
Docdl;
Tinh;
Tim;
Inkq;
END.
```

## BÀI TOÁN 24. Xâu FIBONACI

Cho 3 xâu khác rỗng SA, SB, SR, trong đó độ dài của các xâu SA và SB không vượt quá 10, độ dài xâu SR không vượt quá 15. Dãy xâu  $F_0, F_1, F_2, \dots, F_N$  được xây dựng theo qui tắc sau:  $F_0=SA, F_1=SB, F_{k+1}=F_k+F_{k-1}, k=1,2,\dots,N-1, 1 \leq N \leq 35$ .

**Yêu cầu:** Xác định số lần xuất hiện của SR trong  $F_N$  (tức là số các xâu con các ký tự liên tiếp nhau bằng SR). Hai xâu con khác nhau nếu chúng khác nhau ít nhất một ký tự.

**Dữ liệu:** Vào từ file FSTR.INP:

- Dòng đầu tiên chứa số nguyên dương N
- BA dòng tiếp theo chứa các xâu SA, SB, SR, mỗi xâu trên một dòng.

**Kết quả:** Đa ra file FSTR.OUT số lần xuất hiện tìm được (nguyên)

*Ví dụ:*

FSTR.INP	FSTR.OUT
6	4
A	
B	
BAB	

```
{ $A+,B-,D+,E+,F-,G-,I+,L+,N+,O-,P-,Q+,R+,S+,T-,V+,X+ }  
{ $M 16384,0,655360 }
```

```
uses crt;
```

```
const
```

```
  tfi    =    'FSTR.INP';  
  tfo    =    'FSTR.OUT';
```

```
var
```

```
  fi,fo   :    text;  
  N       :    integer;  
  m       :    integer;  
  SA,SB,SR :    string;  
  fa,fb,fc,  
  da,ca,db,cb,  
  dc,cc   :    string;  
  a,b,c   :    extended;
```

```
procedure Sinhdl;
```

```
var ch: char;
```

```
    i: integer;
```

```
begin
```

```
  clrscr;
```

```
  writeln('Ban co tao file ',tfi,' (C/K)?');
```

```
  repeat ch:=readkey until upcase(ch) in ['C','K'];
```

```
  if upcase(ch)='K' then exit;
```

```
  randomize;
```

```
  N:=random(30)+5;
```

```
  m:=random(10)+1;
```

```
  sa:="";
```

```
  for i:=1 to M do sa:=sa+chr(64+random(3)+1);
```

```
  m:=random(10)+1;
```

```
  sb:="";
```

```
  for i:=1 to M do sb:=sb+chr(64+random(3)+1);
```

```
  m:=random(15)+1;
```

```
  sr:=copy(sa+sb,1,m);
```

```
    assign(fi,tfi); rewrite(fi);  
    writeln(fi,N);  
    writeln(fi,sa);  
    writeln(fi,sb);  
    writeln(fi,sr);  
    close(fi);  
end;
```

```
procedure Docdl;  
begin  
    assign(fi,tfi); reset(fi);  
    readln(fi,N);  
    readln(fi,SA);  
    readln(fi,SB);  
    readln(fi,SR);  
    close(fi);  
    m:=length(SR);  
end;
```

```
function Dem(s1,s: string): extended;  
var d: extended;  
    k: integer;  
begin  
    d:=0;  
    while pos(s1,s)>0 do  
        begin  
            k:=pos(s1,s);  
            d:=d+1;  
            delete(s,1,k);  
        end;  
    Dem:=d;  
end;
```

```
function Chung: real;  
begin  
    fc:=ca+db;  
    delete(fc,1,1);  
    delete(fc,length(fc),1);  
    Chung:=Dem(sr,fc);  
end;
```

```
procedure Tinh;  
var i: integer;  
begin  
    fa:=sa;  
    fb:=sb;  
    a:=Dem(sr,sa);  
    b:=Dem(sr,sb);  
    i:=2;  
    while (i<=n) and (length(fa)<=m) do  
        begin  
            fc:=concat(fa,fb);  
            c:=Dem(sr,fc);  
            fa:=fb;
```

```

fb:=fc;
a:=b;
b:=c;
inc(i);
end;
if i<=n then
begin
da:=copy(fa,1,m);
ca:=copy(fa,length(fa)-m+1,m);
db:=copy(fb,1,m);
cb:=copy(fb,length(fb)-m+1,m);
while i<=n do
begin
c:=a+b+Chung;
dc:=da; cc:=cb;
a:=b;
b:=c;
da:=db; ca:=cb;
db:=dc; cb:=cc;
inc(i);
end;
end;
end;

procedure Inkq;
begin
assign(fo,tfo); rewrite(fo);
writeln(fo,c:0:0);
close(fo);
end;

BEGIN
{Sinhdl;}
Docdl;
Tinh;
Inkq;
END.

```

## BÀI TOÁN 25. Tháp Hà Nội

Bài toán Tháp Hà Nội trở thành nổi tiếng vào năm 1883, sau bài báo của Luca là một nhà toán học người Pháp. Tháp là một cọc đĩa đường kính giảm dần từ dưới lên trên. Bài toán đặt ra là cần chuyển chồng đĩa sang một cọc khác sử dụng một cọc trung gian sao cho trong quá trình chuyển đĩa không có đĩa nào có đường kính lớn hơn lại bị đặt lên trên đĩa có đường kính nhỏ hơn.

**Yêu cầu:** Giải bài toán tháp Hà Nội tổng quát. Cho M cọc và tháp N đĩa ( $3 < M \leq 30$ ,  $1 \leq N \leq 30$ ), hãy xác định số lần chuyển đĩa tối thiểu cần thực hiện để chuyển chồng đĩa từ cọc xuất phát sang cọc đích sử dụng M-2 cọc còn lại nh cọc trung gian.

**Dữ liệu:** Vào từ file HANTOWER.INP gồm nhiều dòng, mỗi dòng chứa hai số nguyên N, M được ghi cách nhau theo thứ tự là số đĩa và số cọc trong bài toán tháp Hà Nội.

**Kết quả:** Mỗi dòng trong file dữ liệu vào ghi ra trên một dòng của file văn bản HANTOWER.OUT số lần chuyển tối thiểu cần thực hiện.

**Ví dụ:**

HANTOWER.INP

5 3

HANTOWER.OUT

31

const



```

tfi          =   'HANTOWER.INP';
tfo          =   'HANTOWER.OUT';
maxN         =   30;
maxM         =   30;

var
  fi,fo       :   text;
  N,M         :   integer;

  f           :   array[1..maxN,3..maxM] of extended;

```

```

procedure Tinh;
var i,j,k: integer;
begin
  {Tinh khi m=3: f[n,3]=2f[n-1,3]+1}
  f[1,3]:=1;
  for i:=2 to N do f[i,3]:=2*f[i-1,3]+1;

  {f[n,m]=min(2*f[k,m]+f[n-k,m-1])}
  for j:=4 to m do
    begin
      f[1,j]:=1;
      for i:=2 to N do
        begin
          f[i,j]:=2*f[1,j]+f[i-1,j-1];
          for k:=2 to i-1 do
            if f[i,j]>2*f[k,j]+f[i-k,j-1] then
              f[i,j]:=2*f[k,j]+f[i-k,j-1];
          end;
        end;
      end;
    end;
end;

```

```

BEGIN
  assign(fi,tfi); reset(fi);
  assign(fo,tfo); rewrite(fo);
  while not seekeof(fi) do
    begin
      readln(fi,N,M);
      Tinh;
      writeln(fo,f[n,m]:0:0);
    end;
  close(fi); close(fo);
END.

```

## BÀI TOÁN 26. Xếp lịch giảng

Một giáo viên cần giảng  $N$  vấn đề được đánh số từ 1 đến  $N$  ( $N \leq 1000$ ). Mỗi một vấn đề  $i$  có thời gian là  $A_i$  ( $i=1 \dots N$ ). Mỗi vấn đề chỉ giảng không quá 1 buổi. Thời gian tối đa của một buổi là  $L$  ( $L \leq 500$ ). Vấn đề  $i$  phải được giảng trước vấn đề  $i+1$ . Trong một buổi có thể bố trí giảng vài vấn đề, nhưng nếu thừa lượng thời gian  $t$  thì buổi đó được đánh giá là lãng phí thời gian với mức  $d$ :

$$d = \begin{cases} 0 & t = 0 \\ -c & 1 \leq t \leq 10 \\ (t-10)^2 & t > 10 \end{cases}$$

trong đó c là hằng số nguyên dương cho trước.

Hãy xếp lịch dạy sao cho số buổi ít nhất và tổng các lãng phí thời gian là nhỏ nhất có thể được.

**Dữ liệu** vào từ file LICH.INP gồm:

- Dòng đầu là số N
- dòng tiếp theo là L và C
- Dòng cuối cùng là N số thể hiện  $A_1, A_2, \dots, A_n$

**Kết quả** ghi ra file LICH.OUT gồm:

- Dòng đầu tiên là số buổi của lịch
- Dòng tiếp theo là tổng thời gian lãng phí nhỏ nhất đạt được.

**Ví dụ**

LICH.INP	LICH.OUT
10	6
120 10	2700

80 80 10 50 30 20 40 30 120 100

{ \$A+,B-,D+,E+,F-,G-,I+,L+,N+,O-,P-,Q+,R+,S+,T-,V+,X+,Y+ }

{ \$M 65520,0,655360 }

uses crt;

const

```

tfi      = 'LICH.INP';
tfo      = 'LICH.OUT';
NN       = 100;
maxN     = 100;
```

var

```

fi,fo    : text;
N,M      : integer;
L,C      : integer;
a        : array[1..maxN] of integer;
x        : array[0..maxN] of integer;
d        : array[0..maxN,0..maxN] of LongInt;
LP       : LongInt;
```

procedure Docdl;

var i: integer;

begin

  Readln(fi,N);

  readln(fi,L,C);

  for i:=1 to N do read(fi,a[i]);

end;

procedure TinhSoBuoi;

var i,j,t: integer;

begin

  fillchar(x,sizeof(x),0);

  x[0]:=0;

  for i:=1 to N do

    begin

      x[i]:=MaxInt;

```
t:=0;
for j:=i-1 downto 0 do
begin
t:=t+a[j+1];
if t>L then break;
if x[i]>x[j]+1 then x[i]:=x[j]+1;
end;
end;
M:=x[N];
end;

function CP(t: Integer): LongInt;
var u: LongInt;
begin
u:=t;
if u=0 then CP:=0 else
if u<=10 then CP:=-C else
CP:=sqr(u-10);
end;

procedure TinhLangPhi;
var i,j,k,t: integer;
begin
d[0,0]:=0;
for j:=1 to M do d[0,j]:=d[0,j-1]+CP(l);
for i:=1 to N do
for j:=1 to M do
begin
t:=0;
d[i,j]:=MaxLongInt;
for k:=i-1 downto 0 do
begin
t:=t+a[k+1];
if t>L then break;
if (d[k,j-1]<MaxLongInt) and (d[i,j]>d[k,j-1]+CP(L-t)) then
d[i,j]:=d[k,j-1]+CP(L-t);
end;
end;
LP:=d[N,M];
end;

procedure InKq;
begin
writeln(fo,M);
writeln(fo,LP);
end;

procedure SinhDL;
var ch: char;
i: integer;
begin
clrscr;
write('Ban co tao file ',tfi,' (C/K)?');
repeat ch:=readkey until upcase(ch) in ['C','K'];
```

```

if upcase(ch)='K' then exit;
randomize;
N:=NN;
L:=random(500)+1;
C:=random(10)+1;
for i:=1 to N do a[i]:=random(L)+1;
assign(fi,tfi); rewrite(fi);
writeln(fi,N);
writeln(fi,L,' ',C);
for i:=1 to N do write(fi,a[i],' ');
close(fi);
end;

BEGIN
{sinhDL;}
assign(fi,tfi); reset(fi);
assign(fo,tfo); rewrite(fo);
Docdl;
TinhSoBuo;
TinhLangPhi;
Inkq;
close(fi); close(fo);
END.

```

## BÀI TOÁN 27. Kinh doanh bất động sản.

Tại thành phố Silicon, Một người nọ được thừa kế một khoản tiền N ngàn USD, người đó quyết định đầu t vào việc kinh doanh bất động sản bằng cách mua các mảnh đất hình vuông có kích thước là các số nguyên, biết rằng mỗi mét vuông đất có giá trị 1 ngàn USD. Hãy chỉ cách cho người nọ mua đất sao cho tổng số tiền mua đất đúng bằng N ngàn USD và số mảnh đất mua được càng ít càng tốt.

**Dữ liệu:** Vào từ file văn bản MUADAT.INP gồm một số nguyên dương duy nhất N có giá trị không vượt quá 60000

**Kết quả:** Ghi ra file văn bản MUADAT.OUT một dãy số nguyên dương xếp theo thứ tự giảm dần là kích thước các mảnh đất mua được

*Ví dụ:*

MUADAT.INP	MUADAT.OUT
30	4 3 2 1
{ \$r- }	
const	
tfi='Muadat.inp';	
tfo='Muadat.out';	
maxN=60000;	
maxD=30002;	
type	
mang1=array[1..maxD] of word;	
mang2=array[0..maxN] of byte;	
var	
fi, fo: text;	
N: word;	
M: integer;	
t: mang2;	

```
d: array[1..2] of ^mang1;

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

procedure capphat;
var i: integer;
begin
  for i:=1 to 2 do new(d[i]);
end;

function row(p: word): word;
begin
  row:=p div maxD+1;
end;

function col(p: word): word;
begin
  col:=p mod maxD+1;
end;

procedure Doc;
begin
  assign(fi,tfi); reset(fi);
  readln(fi,N);
  close(fi);
  M:=244;
end;

procedure Tinh;
var j,i: word;
begin
  for j:=0 to N do
    begin
      d[row(j)]^[col(j)]:=j;
      t[j]:=1;
    end;
  for i:=2 to M do
    for j:=N downto i*i do
      if d[row(j)]^[col(j)]>d[row(j-i*i)]^[col(j-i*i)]+1 then
        begin
          d[row(j)]^[col(j)]:=d[row(j-i*i)]^[col(j-i*i)]+1;
          t[j]:=i;
        end;
    end;
end;

procedure Tim;
var u,v: word;
begin
  fillchar(x,sizeof(x),0);
  u:=N;
  repeat
    inc(x[t[u]]);
    v:=t[u];
  until v=0;
```

```

u:=u-v*v;
until u=0;
end;

```

```

procedure Viet;
var i,j: word;
begin
  assign(fo,tfo); rewrite(fo);
  for i:=M downto 1 do
    for j:=x[i] downto 1 do write(fo,i,' ');
  close(fo);
end;

```

```

BEGIN
  Caphat;
  Doc;
  Tinh;
  Tim;
  viet;
END.

```

### BÀI TOÁN 28. Ca nhạc

Trong cuộc thi các tiết mục ca nhạc, N nhóm nghệ sĩ (được đánh số từ 1 đến N) đồng thời trình diễn những tiết mục của mình tại N địa điểm gần nhau, thời gian kết thúc của nhóm thứ i là  $t_i$ . Để thu hút khách đến xem, mỗi nhóm được quyền khuyến mại tặng quà khi khách vào xem, trị giá tặng là  $z_i$ , với điều kiện thời gian khách xem tối thiểu là d đơn vị thời gian.

Hãy lập trình chọn ra các nhóm cần xem sao cho tổng trị giá quà tặng là lớn nhất. Giả sử thời gian bắt đầu của tất cả các nhóm là 0 và thời gian di chuyển giữa các địa điểm là không đáng kể.

**Dữ liệu** vào từ file văn bản MUSIC.INP với cấu trúc nh sau:

- Dòng đầu tiên là 2 số nguyên dương N và d ( $N \leq 1000$ )
- Dòng thứ hai là dãy số nguyên dương  $t_i$ ,  $i=1,2,...,N$
- Dòng thứ 3 là dãy số nguyên dương  $z_i$ ,  $i=1,2,...,N$

**Kết quả** Ghi ra file MUSIC.OUT

- Dòng đầu là tổng giá trị quà tặng
- Dòng sau là dãy thứ tự các nhóm phải xem

*Ví dụ:*

MUSIC.INP	MUSIC.OUT
7 1	230
2 1 4 1 5 2 4	4 1 3 7 5
100 10 15 27 52 19 36	

```

{$R-}
const
  tfi      = 'MUSIC.INP';
  tfo      = 'MUSIC.OUT';
  maxN     = 1000;
  maxB     = 126;

type
  BanNhac = record T,Z,Name: integer end;
  mang    = array[0..maxB] of byte;

var
  fi, fo   : text;
  N,P      : integer;

```

```
a          :   array[1..maxN] of BanNhac;

F          :   array[0..maxN] of longint;
kt         :   integer;

Tr         :   array[0..maxN] of ^mang;

slx        :   integer;
x          :   array[1..maxN] of integer;

procedure CapPhat;
var i: integer;
begin
  for i:=0 to maxN do
    new(Tr[i]);
end;

procedure SetBit(i,j: integer);
var u,v: integer;
begin
  u:=j div 8;
  v:=j mod 8;
  (* Bat bit thu v cua Tr[i,u] *)
  Tr[i]^u:=Tr[i]^u or (1 shl v);
end;

function GetBit(i,j: integer): byte;
var u,v: integer;
begin
  u:=j div 8;
  v:=j mod 8;
  (* kiem tra bit thu v cua Tr[i,u] *)
  GetBit:=(Tr[i]^u and (1 shl v)) shr v;
end;

procedure Docdl;
var i: integer;
begin
  assign(fi,tfi); reset(fi);
  readln(fi,N,P);
  for i:=1 to N do read(fi,a[i].T);
  readln(fi);
  for i:=1 to N do read(fi,a[i].Z);
  close(fi);
  for i:=1 to N do a[i].Name:=i;
end;

procedure Trao(var u,v: BanNhac);
var w: BanNhac;
begin
  w:=u;
  u:=v;
  v:=w;
end;
```

```
procedure SapXep;
var i,j: integer;
begin
  for i:=1 to N-1 do
    for j:=i+1 to N do
      if a[i].T>a[j].T then
        Trao(a[i],a[j]);
end;

procedure Solve;
var k,i,j,u,v: integer;
begin
  for i:=0 to maxN do
    for j:=maxB downto 0 do Tr[i]^j:=0;
  SapXep;
  for k:=1 to N do F[k]:=0; {Phan thuong khi khong xem}
  for k:=1 to N do
    begin
      F[k]:=-1; {Khong xac dinh}
      if (a[k].T>=k*P) and (F[k-1]<>-1) then
        begin
          SetBit(k,k); {Tr[k,k]:=1;}
          F[k]:=F[k-1]+a[k].z;
        end;
      for i:=k-1 downto 1 do
        if (a[k].T>=i*P) and (F[i-1]+a[k].z>F[i]) then
          begin
            F[i]:=F[i-1]+a[k].z;
            SetBit(k,i); {Tr[k,i]:=1;}
          end;
        end;
      end;
    (* Tim diem ket thuc *)
    kt:=0;
    for i:=1 to N do
      if F[i]>F[kt] then kt:=i;
    slx:=0;
    if F[kt]=0 then exit;
    u:=N;
    v:=kt;
    repeat
      if GetBit(u,v)=1 then
        begin
          inc(slx);
          x[slx]:=a[u].name;
          dec(u);
          dec(v);
        end
      else dec(u);
    until (u=0) or (v=0);
  end;

  procedure Inkq;
  var i: integer;
```



```
begin
  assign(fo,tfo); rewrite(fo);
  writeln(fo,F[kt]);
  for i:=slx downto 1 do write(fo,x[i],' ');
  close(fo);
end;
```

```
BEGIN
  CapPhat;
  Docdl;
  Solve;
  Inkq;
END.
```

### BÀI TOÁN 29. Số lớn nhất

Cho 2 số nguyên  $X=x_1x_2...x_N$  và  $Y=y_1y_2...y_M$ . Hãy tìm số  $Z = z_1z_2...z_k$  ( $Z$  nhận được từ  $X$  và  $Y$  bằng cách xóa đi một số chữ số) lớn nhất.

Ví dụ :  $X=12345$

$Y=435012$

Thì  $Z=45$  (nhận được từ  $X$  bằng cách xoá đi  $x_1, x_2, x_3$ ; nhận được từ  $Y$  bằng cách xoá đi  $y_2, y_4, y_5, y_6$ )

Input : NUMBER.IN

- Dữ liệu thứ nhất là  $X$ .
- Dữ liệu thứ hai là  $Y$ .

Output : NUMBER.OUT

- Dữ liệu đầu ghi số  $Z$ .

Giới hạn :  $M, N < 201$ .

NUMBER.I N	NUMBER.OU T
12345 4351023	123

### Thuật toán

Tìm dãy con chung dài nhất đồng thời phải thoả mãn là số lớn nhất. Vậy có 2 yêu cầu:

1. Tìm dãy con chung dài nhất bằng QHĐ: Gọi  $L[i,j]$  là độ dài của dãy con chung dài nhất khi số thứ nhất chỉ  $a[i..length(a)]$  và số thứ hai chỉ dài là  $b[j..length(b)]$  thì:

$$L[i,j] := \text{Max}(L[i+1,j], L[i,j+1], L[i+1,j+1] + \text{ord}(a[i]=b[j]))$$

2. Tìm chữ số ở hàng trái nhất (cao nhất) đó là chữ  $ch$  có tại vị trí  $i$  trong  $a$  và có tại vị trí  $j$  trong  $b$  mà  $m=L[i,j]$  lớn nhất. Đó là độ dài xâu con chung dài nhất (là  $m$ ).
3. Tìm tiếp các hàng tiếp theo bằng vòng lặp:

Tìm chữ số  $k=m-1$  tới 1 (tính từ bên phải dãy con chung sang trái)

```
{
  Duyệt chữ số  $ch$  từ lớn đến nhỏ (từ 9 đến 1)
  {
    Cho  $x$  tăng từ  $i+1$  cho đến khi  $a[x]=ch$ 
    Cho  $y$  tăng từ  $j+1$  đến khi  $b[y]=ch$ 
    Nếu  $L[x,y]=k$  thì  $ch$  là chữ số ở hàng  $k$  (tính từ phải sang)
  }
}
```

Chương trình

const max = 250;

```
        fi      ='number.in1';
        fo      ='number.ou1';
var      l      :array[1..max+1,1..max+1]of byte;
        a,b,c    :string;
procedure docf;
var f     :text;
begin
    assign(f,fi);
    reset(f);
    readln(f,a);readln(f,b);
    close(f);
end;
function maxso(x,y,z:byte):byte;
begin
    if x<y then x:=y;
    if x<z then maxso:=z else maxso:=x;
end;
procedure lam;
var i,j,k,x,y,m :integer;
    ch          :char;
begin
    fillchar(l,sizeof(l),0);
    for i:=length(a) downto 1 do
        for j:=length(b) downto 1 do
            l[i,j]:=maxso(l[i+1,j],l[i,j+1],l[i+1,j+1]+ord(a[i]=b[j]));
        m:=0;c:='0';
        for ch:='9'downto '1' do
            begin
                i:=pos(ch,a);j:=pos(ch,b);
                if (i>0)and(j>0)and(l[i,j]>m) then begin c:=ch;m:=l[i,j];end;
            end;
        i:=pos(c,a)+1;j:=pos(c,b)+1;
        for k:=m-1 downto 1 do
            for ch:='9'downto '0' do
                begin
                    x:=i;y:=j;
                    while (x<=length(a))and(a[x]<>ch) do inc(x);
                    while (y<=length(b))and(b[y]<>ch) do inc(y);
                    if l[x,y]=k then
                        begin
                            c:=c+ch;i:=x+1;j:=y+1;
                            break;
                        end;
                    end;
            end;
        end;
    end;
end;
procedure ghif;
var f     :text;
begin
    assign(f,fo);
    rewrite(f);
    write(f,c);
    close(f);
end;
BEGIN
```

docf;  
lam;  
ghif;  
END.

### BÀI TOÁN 30. Chuỗi đối xứng

Trong một buổi học viết chữ, Bờm phát hiện trong một số từ khi bỏ đi một số ký tự thì đọc ngược hay đọc xuôi đều giống nhau.

Ví dụ từ **IOICAMP**, khi xóa đi các chữ cái C,A,M,P, thì còn lại **IOI** là một từ đối xứng.

Bờm cảm thấy thú vị, và cậu tiếp tục thử xóa các ký tự khác, kết quả là có thêm nhiều từ đối xứng nữa: II, I, O, C... Nhưng nếu với một từ dài, cứ thử từng cách xóa như vậy thì thật mất thời gian.

Bạn Hãy viết chương trình giúp Bờm tính số cách xóa sao cho từ thu được đối xứng. Hai cách xóa chỉ khác nhau bởi thứ tự xóa các ký tự thì coi như trùng nhau.

Input: **PAL.INP**

- Một dạng duy nhất là từ cần tính số cách xóa, từ này chỉ chứa các chữ cái in hoa **A, B, ..., Z**.

Output: **PAL.OUT**

- Một số duy nhất là số cách xóa.

Giới hạn:

- Độ dài từ không quá 120.
- Thời gian: 0.5 s/test
- Bộ nhớ: 1MB

Ví dụ:

<b>PAL.INP</b>	<b>PAL.OUT</b>
IOICAMP	9

const

tfi='Pal.inp';  
tfo='Pal.out';  
maxN=121;  
ONE='1';  
TWO='2';  
THREE='3';

type

xau=string[maxN];  
BigNum=string[40];  
arr1=array[1..maxN] of BigNum;

var

fi, fo: text;  
S: xau;  
N: integer;  
a,b,c: arr1;

procedure Cong(u,v: BigNum; var w: BigNum);

var nho, tong, nu, nv, nw, i: integer;

begin

nu:=length(u);  
nv:=length(v);  
nw:=nu;  
if nw<nv then nw:=nv;  
for i:=nu+1 to nw do u:='0'+u;

```
for i:=nv+1 to nw do v:='0'+v;
w:="";
for i:=1 to nw do w:=w+'0';
nho:=0;
for i:=nw downto 1 do
begin
tong:=ord(u[i])+ord(v[i])-96+nho;
w[i]:=chr(tong mod 10+48);
nho:=tong div 10;
end;
if nho>0 then w:='1'+w;
end;

procedure Tru(u,v: BigNum; var w: BigNum);
var i, nho, hieu, nu, nv, nw: integer;
begin
nu:=length(u);
nv:=length(v);
nw:=nu;
for i:=nv+1 to nw do v:='0'+v;
w:="";
for i:=1 to nw do w:=w+'0';
nho:=0;
for i:=nw downto 1 do
begin
hieu:=ord(u[i])-ord(v[i])-nho;
if hieu<0 then
begin
hieu:=hieu+10;
nho:=1;
end
else nho:=0;
w[i]:=chr(hieu+48);
end;
while (length(w)>1) and (w[1]='0') do delete(w,1,1);
end;

procedure Doc;
begin
readln(fi,s);
N:=length(s);
end;

procedure Tinh;
var i,j,k: integer;
begin
for i:=1 to N do b[i]:=ONE;
for i:=1 to N-1 do
if s[i]=s[i+1] then c[i]:=THREE else c[i]:=TWO;
for k:=2 to n-1 do
begin
move(b,a,sizeof(b));
move(c,b,sizeof(c));
for i:=1 to N-k do
```

```

begin
  j:=k+i;
  if s[i]=s[j] then
    begin
      {c[i]:=b[i]+b[i+1]+1}
      Cong(b[i],b[i+1],c[i]);
      Cong(c[i],ONE,c[i]);
    end
  else
    begin
      {c[i]:=b[i]+b[i+1]-a[i+1];}
      Cong(b[i],b[i+1],c[i]);
      Tru(c[i],a[i+1],c[i]);
    end;
  end;
end;
end;

procedure Viet;
begin
  writeln(fo,c[1]);
end;

procedure Main;
begin
  assign(fi,tfi); reset(fi);
  assign(fo,tfo); rewrite(fo);
  doc;
  tinh;
  viet;
  close(fi); close(fo);
end;

BEGIN
  Main;
END.

```

### **BÀI TOÁN 31. Palindrome** (Bài thi Olympic Quốc tế 2000)

Palindrome là một chuỗi đối xứng, tức là một chuỗi mà đọc từ trái sang phải cũng giống như đọc từ phải sang trái. Bạn cần viết một chương trình với một chuỗi cho trước, xác định số ít nhất các ký tự cần chèn vào chuỗi để nhận được một Palindrome. Ví dụ, bằng cách chèn hai ký tự vào chuỗi “Ab3bd” ta nhận được một Palindrome (chẳng hạn “dAb3bAd” hoặc “Adb3bdA”). Tuy nhiên, nếu chèn ít hơn 2 ký tự thì không thể tạo được Palindrome.

**Dữ liệu vào:** Tên file dữ liệu vào là PALIN.INP. Dòng thứ nhất gồm một số nguyên là độ dài N của chuỗi,  $3 \leq N \leq 5000$ . Dòng thứ hai gồm một chuỗi có độ dài N. Chuỗi gồm các ký tự là các chữ cái hoa A..Z, các chữ cái thường a..z và các chữ số thập phân 0..9, các chữ cái hoa và thông xem như là khác nhau.

**Dữ liệu ra:** Tên tệp dữ liệu ra là PALIN.OUT gồm một số nguyên là số lượng ký tự tối thiểu cần chèn vào.

**Ví dụ:**

PALIN.INP	PALIN.OUT
5	2
Ab3bd	

{ \$A+,B-,D+,E+,F-,G-,I+,L+,N+,O-,P-,Q+,R-,S+,T-,V+,X+}  
{ \$M 16384,0,655360}

uses crt;

const

tfi = 'PALIN.INP';  
tfo = 'PALIN.OUT';  
maxN = 5000;

var

fi,fo : text;  
N : longint;  
c : array[1..maxN] of char;  
a,Tr : array[1..maxN] of longint;

procedure Docdl;

var i: longint;

begin

assign(fi,tfi); reset(fi);  
readln(fi,N);  
for i:=1 to N do read(fi,c[i]);  
close(fi);

end;

function max(u,v: longint): longint;

begin

if u<v then max:=v else max:=u;

end;

procedure XDB;

var i,k,l: longint;

begin

a[1]:=1; a[2]:=0;  
for i:=2 to N do  
begin  
Tr:=a;  
a[i]:=1; a[i+1]:=0;  
for k:=i-1 downto 1 do  
begin  
a[k]:=max(Tr[k],a[k+1]);  
if (c[k]=c[i]) and (a[k]<tr[k+1]+2) then  
a[k]:=tr[k+1]+2;

end;

end;

end;

procedure Inkq;

begin

assign(fo,tfo); rewrite(fo);  
writeln(fo,N-a[1]);  
close(fo);

end;

BEGIN

Docdl;

XDB;

Inkq;

END.

### BÀI TOÁN 32. Phân tích số

Với  $x$  nguyên dương, gọi  $f(x,k)$  là số cách phân tích  $x$  thành tổng các số nguyên tố mà mỗi số nguyên tố có trong tổng không quá  $k$  lần.

$f(12,6)=7$  với 7 cách phân tích:

$$12 = 2+2+2+2+2+2$$

$$12 = 2+2+2+3+3$$

$$12 = 2+2+3+5$$

$$12 = 2+3+7$$

$$12 = 2+5+5$$

$$12 = 3+3+3+3$$

$$12 = 5+7$$

$f(12,3)=5$  vì có 5 cách phân tích:

$$12 = 2+2+2+3+3$$

$$12 = 2+2+3+5$$

$$12 = 2+3+7$$

$$12 = 2+5+5$$

$$12 = 5+7$$

$f(12,1)=2$  vì có 2 cách phân tích

$$12 = 2+3+7$$

$$12 = 5+7$$

Cho một mảng hai chiều  $A[1..M,1..N]$  có  $M$  dòng,  $N$  cột, mỗi phần tử là số nguyên dương không quá 100 và cho số nguyên dương  $k$ . Hãy lập mảng hai chiều  $B$  gồm  $M$  dòng,  $N$  cột sao cho  $B[i,j]=f(A[i,j],k)$

**Dữ liệu** vào lấy từ file văn bản PTS.INP gồm:

- Dòng thứ nhất là 3 số  $M, N, k$
- $M$  dòng tiếp theo là mảng  $A$ , mỗi dòng  $N$  số, hai số trên một dòng cách nhau ít nhất một dấu cách

**Kết quả** ghi ra file PTS.OUT:  $M$  dòng là mảng  $B[1..M,1..N]$ , mỗi dòng  $N$  số, hai số cách nhau ít nhất một dấu cách.

Hạn chế kỹ thuật  $M, N \leq 100$

*Ví dụ:*

PTS.INP	PTS.OUT
2 3 3	0 1 2
1 3 5	3 4 7 5 2 5
7 100 12	

{ \$A+,B-,D+,E+,F-,G-,I+,L+,N+,O-,P-,Q+,R+,S+,T-,V+,X+,Y+ }

{ \$M 65520,0,655360 }

uses crt;

const

tff='PTS.INP';

tfo='PTS.OUT';

maxMN=100;

NTO: array[1..25] of integer=( 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);

```
var
  fi,fo: text;
  M,N,k: integer;
  a: array[1..100,1..100] of integer;
  x,y: array[0..100] of real;

procedure Docdl;
var i,j: integer;
begin
  readln(fi,M,N,k);
  for i:=1 to M do
    begin
      for j:=1 to N do read(fi,a[i,j]);
      readln(fi);
    end;
  end;

procedure ChuanBi;
var i,j,l: integer;
begin
  for i:=0 to 100 do x[i]:=0; {khong xac dinh}
  for j:=0 to k do
    begin
      if j*NTO[1]>100 then break;
      x[j*NTO[1]]:=1;
    end;
  for l:=2 to 25 do
    begin
      y:=x;
      for i:=1 to 100 do
        begin
          x[i]:=0;
          for j:=0 to k do
            begin
              if i-j*NTO[l]<0 then break;
              x[i]:=x[i]+y[i-j*NTO[l]];
            end;
          end;
        end;
      end;
    end;

procedure Inkq;
var i,j: integer;
begin
  for i:=1 to M do
    begin
      for j:=1 to N do write(fo,x[a[i,j]]:0:0,' ');
      writeln(fo);
    end;
  end;

BEGIN
  assign(fi,tfi); reset(fi);
```



```
assign(fo,tfo); rewrite(fo);
Docdl;
ChuanBi;
Inkq;
close(fi); close(fo);
```

END.

### BÀI TOÁN 33. Điền dấu cộng, trừ thích hợp

Hãy điền vào các vị trí có dấu ? trong biểu thức sau:

$$a_1 ? a_2 ? a_3 ? \dots ? a_{n-1} ? a_n$$

Phép toán cộng (+) hoặc trừ (-) để giá trị của biểu thức bằng K cho trước. Ở đây  $a_1, a_2, \dots, a_n, K$  là các số nguyên dương.

Ví dụ: Nếu cho biểu thức

$$1 ? 2 ? 3 ? 4$$

và giá trị  $K=2$  thì một kết quả là

$$1 + 2 + 3 - 4$$

Nếu có nhiều cách điền khác nhau thì chỉ cần đưa ra một cách.

Input: Cho trong file văn bản SIGN.INP

- Dòng đầu tiên ghi N, K ( $N \leq 100, K \leq 10000$ )
- Dòng thứ hai ghi N số  $a_1, a_2, \dots, a_n$  ở đây  $0 < a_i \leq 100$

Output: Ghi ra file văn bản SIGN.OUT: Ký tự đầu tiên là '1' hoặc '0' tương ứng với có hoặc không có cách điền dấu thích hợp. Nếu ký tự đầu là '1' thì n-1 ký tự tiếp theo hoặc là '+' hoặc là '-' tùy theo phép toán tương ứng được điền vào.

*Ví dụ:*

SIGN.INP	SIGN.OUT
4 2	1++-
1 2 3 4	

### BÀI TOÁN 34. Làm việc tập thể

Trong công ty X có N nhân viên rất xuất sắc. Tuy nhiên do tất cả đều quá giỏi và quá tự tin, cứ khi nào 2 nhân viên cùng làm việc với nhau thì hiệu suất gần nh bằng 0. Họ tốn thời gian vào việc tranh cãi và không quyết định được công việc gì. Mỗi nhân viên có giờ làm việc là một khoảng thời gian liên tiếp từ thời điểm  $a_i$  đến thời điểm  $b_i$ . Giờ làm việc của mỗi nhân viên là không thể thay đổi do đặc điểm công việc mà họ đảm trách và tính kỳ quặc của họ. Do các khoảng thời gian này không giống nhau hoàn toàn, có thể có những lúc chỉ có 1 nhân viên làm việc. Lúc này thì họ làm việc rất hiệu quả. Giám đốc muốn giữ lại một số nhân viên sao cho tổng thời gian làm việc hiệu quả là lớn nhất có thể.

Dữ liệu: Vào từ file văn bản WORK.IN

- Dòng đầu tiên ghi n là số nhân viên ( $1 \leq n \leq 10000$ )
- N dòng tiếp theo mỗi dòng ghi hai số  $a_i$  và  $b_i$  là thời điểm bắt đầu và kết thúc giờ làm việc của nhân viên i ( $0 \leq a_i \leq b_i \leq 10^9$ )

Kết quả: Ghi ra file văn bản WORK.OUT một số nguyên duy nhất là tổng thời gian làm việc hiệu quả lớn nhất có thể.

*Ví dụ:*

WORK.IN	WORK.OUT
7	1900
100 150	
0 1000	
900 1000	
1800 2000	
900 1800	
272 314	
1900 2000	

Ghi chú: Các nhân viên được giữ lại là 2, 4 và 5

{\*\*\*\*\*}

```
* Program   : LAM VIEC TAP THE (WORK- Version 1.0)      *
* Date      : 12-01-2006                                *
* Group     : Qui hoạch dong                             *
*****}
{$R+}
const
    tfi      = 'WORK.IN';
    tfo      = 'WORK.OUT';
    maxN     = 10001;

type
    mang1    = array[1..maxN] of longint;

var
    fi, fo   : text;
    N        : integer;
    a,b      : ^mang1;

    f        : mang1;
    Ds       : longint;

procedure capphat;
begin
    new(a);
    new(b);
end;

procedure Docdl;
var i: integer;
begin
    assign(fi,tfi); reset(fi);
    read(fi,N);
    for i:=1 to N do read(fi,a^[i],b^[i]);
    close(fi);
end;

procedure Trao(var u,v: longint);
var w: longint;
begin
    w:=u;
    u:=v;
    v:=w;
end;

procedure sortA(k,l: integer);
var r: longint;
    i,j: integer;
begin
    r:=a^[(k+l) div 2];
    i:=k;
    j:=l;
    repeat
        while a^[i]<r do inc(i);
        while a^[j]>r do dec(j);
```

```

    if i<=j then
        begin
            Trao(a^[i],a^[j]);
            Trao(b^[i],b^[j]);
            inc(i);
            dec(j);
        end;
    until i>j;
    if k<j then sortA(k,j);
    if i<l then sortA(i,l);
end;

procedure Tinh;
var i,j: integer;
    max1,max2: longint;
begin
    sortA(1,n);
    f[1]:=b^[1]-a^[1];
    Ds:=f[1];
    for i:=2 to N do
        begin
            max1:=b^[i]-a^[i];
            max2:=-maxlongint;
            for j:=i-1 downto 1 do
                begin
                    if (b^[j]<=a^[i]) and (max1<f[j]+b^[i]-a^[i]) then
                        max1:=f[j]+b^[i]-a^[i];
                    if (b^[j]>=a^[i]) and (b^[j]<=b^[i])
                        and (max2<f[j]+b^[i]-b^[j]-(b^[j]-a^[i])) then
                            max2:=f[j]+(b^[i]-b^[j])-(b^[j]-a^[i]);
                end;
            f[i]:=max1;
            if f[i]<max2 then f[i]:=max2;
            if f[i]>Ds then Ds:=f[i];
        end;
    end;

procedure Inkq;
begin
    assign(fo,tfo); rewrite(fo);
    writeln(fo,Ds);
    close(fo);
end;

BEGIN
    capphat;
    Docdl;
    Tinh;
    Inkq;
END.

```

### BÀI TOÁN 35. Hai bên sông

Đọc hai bờ sông, mỗi bờ có N làng. Làng i ở bên bờ trái và làng j ở bờ phải có một đường thủy sang nhau nếu  $A[i,j]=1$   
 Hãy lập các đường thủy trên sông thỏa mãn các điều kiện:

- + Các đường thủy không cắt nhau.
- + Không có hai đường thủy nào cùng xuất phát tại 1 làng hoặc cùng tới một làng.
- + Số đường thủy nhiều nhất.

Dữ liệu vào từ file văn bản 2BS.INP

Dòng đầu tiên là N ( $N \leq 100$ )

Các dòng sau, mỗi dòng 2 số i, j thể hiện có thể thiết lập đường thủy giữa làng i bên bờ trái và làng j bên bờ phải.

Kết quả ghi ra file văn bản 2BS.OUT

Dòng đầu là số K – số lượng đường thủy thiết lập được.

K dòng sau, mỗi dòng 2 số i, j thể hiện một đường thủy giữa làng i và làng j.

Ví dụ:

2BS.INP	2BS.OUT
5	4
3 4	2 1
4 2	3 2
4 4	4 4
5 1	5 5
5 2	
5 3	
3 2	
5 5	
1 3	
2 1	

**HD:**

Xây dựng mảng  $L[0..N, 0..N]$ , trong đó  $L[i, j]$  là số đường thủy nhiều nhất có thể lập được khi một bờ sông xét từ làng 1 đến làng i, bờ sông kia xét từ làng 1 đến làng j. Công thức truy hồi tính  $L[i, j]$  là:

$$L[i, j] = \begin{cases} \text{Max}(L[i-1, j], L[i, j-1]) & : A[i, j] = 0 \\ \text{Max}(L[i-1, j-1] + 1, L[i, j-1], L[i-1, j]) & : A[i, j] = 1 \end{cases}$$

const

fi='2BS.INP';

fo = '2BS.OUT';

var

f:text;

A,L:array[0..102,0..102] of longint;

n:longint;

procedure nhap;

var x,y: longint;

begin

assign(f,fi);

reset(f);

readln(f,N);

while not eof(f) do

begin

readln(f,x,y);

a[x,y]:=1;

end;

close(f);

end;

procedure ghi;

var i,j: longint;

begin

```
assign(f,fo);
rewrite(f);
writeln(f,L[N,N]);
i:=n; j:=n;
repeat
    while (L[i,j]=L[i-1,j])and (i>1) do dec(i);
    while (L[i,j]=L[i,j-1])and (j>1) do dec(j);
    writeln(f,i,' ',j);
    dec(i);
    dec(j);
until L[i,j]=0;
close(f);
end;
function max2(x,y: longint):longint;
begin
    if x>y then max:=x else max:=y;
end;
function max3(x,y,z: longint):longint;
var p:longint;
begin
    p:=x;
    if y>p then p:=y;
    if z>p then p:=z;
    max3:=p;
end;
procedure xuli;
var i,j: longint;
begin
    filchar(L, sizeof(L),0);
    for i:=1 to n do
        for j:=1 to n do
            if a[i,j]=0 then L[i,j]:= max2(L[i-1,j],L[i,j-1])
            else L[i,j]:= max3(L[i-1,j],L[i,j-1], L[i-1,j-1]+1);
end;
BEGIN
    NHAP;
    XULI;
    GHI;
END.
```

\*\*\*

### TÀI LIỆU THAM KHẢO

1. Tài liệu giáo khoa chuyên tin, quyển 1. NXB giáo dục.
2. Chuyên đề bồi dưỡng học sinh giỏi môn tin học trung học phổ thông; Trần Đỗ Hùng, Đỗ Đức Đông, Lê Sĩ Quang, NXB giáo dục, 2007.
3. Đề thi HSG tin học THPT các tỉnh.
4. Website: [DayhocIntel.net](http://DayhocIntel.net) > [Góc tin học - công nghệ](#) > [Phần mềm](#) > [Lập trình](#) > Chuyên đề Quy hoạch động để BDHSG

**Giáo viên: Nguyễn Văn Mộng**