Chương IV: HÌNH HỌC

I Lý Thuyết :

1 . Điểm , đường thẳng, đoạn thẳng :

a . Điểm (Point) :

Trong hình học, chúng ta xét trong hệ DeCac xoy , thì một điểm có toạ độ: (x,y) .

Chính vì thế ta lưu toạ độ một điểm trong một bản Record :

```
Type
    point =  Record
        x , y   :       integer ;
    End ;
    Point_Chung = Record
        x ,y    :       Real ;
    End ;
```

Chính vì vậy khi xét tới toạ độ của P (x,y) thì ta xét P.x , P.y .

Chúng ta biết khoảng cách giữa hai điểm P(x1,y1)và Q(x2,y2) trong mặt phẳng :

```
Function            Khoang_Cach(P,Q : Point_Chung ) : Real ;
Begin
    Khoang_Cach:=Sqrt(Sqr(P.x-Q.x)+Sqr(P.y-Q.y));
End ;
```

b . Đường thẳng (line) :

Trong hình học, chúng ta có phương trình của mộ đường thẳng trong mặt phẳng :

Ax+By + C =0 . Chúng ta coi A , B , C là biểu diễn cho đòng thẳng đó . Nếu một đường thẳng () : Ax + By + C = 0 , đi qua 2 điểm A (x1,y1) và B (x2,y2) thì nó có :

A := y1- y2 ; B := x2 - x1 ; C := -(A.x1+B.y1) .

Chính vì thế chúng ta dùng thủ tục xác định A ,B ,C của một đường thẳng đi qua 2 điểm như sau :

```
Procedure   Xac_DinhABC( P , Q : Point , var A , B , C : Longint ) ;
Begin
    A := P.y-Q.y ;
    B := Q.x-P.x ;
    C := -(A*P.x+B*P.y ) ;
End ;
```

Và chúng ta có thể coi đường thẳng là một kiểu :

```
Type        Lines = Record
                a , b , c  : Longint ;
            End ;
```

Chú ý :

'Lines' chứ không phải 'Line' , vì trong PasCal có thủ tục Line: vẽ đường thẳng. Thông thường chúng ta làm bài hình học để quan sát trực quan thì chúng ta thường biểu diễn lên hình vẽ. Chính vì thế chúng ta cần phải tránh những sai sót không đáng có .

Sở dĩ chúng ta phải khai báo A , B , C trong Longint vì nếu chúng ta lưu A , B , C trong Longint thì lúc chúng ta tính toán phương trình tương quan không tràn bộ nhớ số học .

c. Đoạn thẳng :

Đoạn thẳng là một phần của đòng thẳng , bị giới hạn x , y. Chúng ta xét đoạn thẳng thông thường cho đi qua 2 điểm . Chính vì thế toạ độ x , y bị giới hạn trong khoảng đó .

2. Phương trình tương quan giữa điểm và đường thẳng , đoạn thẳng :

a . Tương quan giữa điểm và đường thẳng :

Cho đường thẳng () có phương trình : Ax +By + C = 0 và P(x,y). Thì phương trình : F(x,y)=A*P.x + B*P.y + C , là phương trình tương quan của P với () . Đúng vậy :

Nếu F (x,y)=0 thì P() . ngược lại nếu F(x,y) 0 thì P() .

Nếu Q (x1,y1) mà nằm cùng phía không thuộc () thì F (x1,y1)*F(x,y)>0 , và nếu khác phía thì F(x1,y1)*F(x,y)<0 . Đây là một trong những điều kiện giúp ích cho ta rất nhiều trong giải toán tin-hình học , và cũng là một phương tiện thiết kế chương trình trong tin học dễ dàng hơn .

Chúng ta xây dựng hàm Phuong_Trinh để xác định mối tương quan của đường thẳng với một điểm :

```
Function        Phuong_trinh ( L : Lines ; P : Point_Chung ) : Real ;
    Begin
        Phuong_Trinh :=L.a*P.x+L.b*P.y+L.c ;
    End ;
```

b. Tương quan của điểm với đoạn thẳng .

Chúng ta biết rằng , đoạn thẳng là một phần đường thẳng. Nên mối tương quan giữa điểm P(x,y) với đoạn thẳng AB , (A(x1,y1) , B(x2,y2)) . là :

- Nếu P[AB] thì :
  + F(x,y)=0 , tức là : a*x + b*y + c = 0 , với a = y1-y2 ; b=x2-x1 và c=-(a*x1+b*y1) .
  + (x-x1)*(x-x2)<=0 và (y-y1)*(y-y2)<=0 .

Nếu P[AB] thì :

+ Nếu F(x,y)=0 thì : (x-x1)*(x-x2)>0 hoặc (y-y1)*(y-y2)>0 .
+ Nếu F(x,y)0 thì P không thuộc đường thẳng qua A ,B .

Ta có thể xây dựng hàm kiểm tra 1 điểm P có thuộc đoạn AB như sau :

```
          Function   thuoc_doan ( P : Point_Chung ; A, B : Point
) : Boolean ;
          Var
               a , b , c : longint ;
               t      : Real ;
               Begin
                    xac_dinhABC ( A, B ,a,b,c) ;
                    thuoc_doan:=false ;
                    t := a * P.x + b*P.y + c ;
                    if t<>0 then exit ;
                    if  (  (P.x-A.x)*(P.x-B.x)>0  )  or  ((P.y-
A.y)*(P.y-B.y)>0)then
                    Exit ;
                    thuoc_doan:=True ;
               End ;
```

Note : Sở dĩ phải khai a , b , c , t : Longint vì có thể tràn bộ nhớ số học (xem phần cần chú ý) .

3. Cắt nhau :

a. Đường thẳng cắt đường thẳng :

(1) : A1*x+B1*y+C1=0 và (2) : A2*x+B2*y+C2=0 . Thì ta gọi mối tương quan giữa (1)và (2) được biểu diễn qua hệ phương trình sau :

A1*x+B1*y=-C1
A2*x+B2*y=-C2

đặt : D=A1*B2-A2*B1; Dx=B1*C2 -B2*C1; Dy=C1*B2-A1*C2

Hai đường thẳng cắt nhau khi và chỉ khi : D0 . Toạ độ điểm giao của hai đường thẳng này là : x=Dx/D , y= Dy/D

Hai đường thẳng song song với nhau khi và chỉ khi : D=0 và Dx0 hoặc Dy0 .

Hai đường thẳng trùng nhau khi : D=Dx=Dy=0 .

Chúng ta có thể xây dựng hàm kiểm tra cắt nhau của hai đường thẳng .

Hàm Lines_Cut có giá trị :

1 : Nếu hai đường thẳng đó cắt nhau
2 : Nếu hai đường thẳng đó song song nhau
3 : Nếu hai đường thẳng đó trùng nhau .

```
          Function     Lines_cut ( L1 , L2  : Lines ) : Byte ;
          var
               D , Dx , Dy          :       Longint ;
               Begin
                    D := L1.a*L2.b -L2.a*L1.b;
                    Dx :=L1.b*L2.c-L1.c*L2.B ;
                    Dy:=L1.c*L2.b-L1.b*L2.c ;
                    If D<>0 then Lines_cut:=1 else
                    If D=0 then
                    Begin
                         If (Dx<>0)Or(Dy<>0)Then Lines_cut:=2 ;
```

```
                If (Dx=0)And(Dy=0) ThenLines_Cut:=3;
            End ;
        End ;
```

b. Đường thẳng cắt đoạn thẳng :

(1) : A1*x+B1*y+C1=0 và đoạn AB , A(x1,y1) , B(x2,y2) .

Đặt A2:=y1-y2 ; B2:=x2-x1 ; C2:=-(A2*x1+B2*y)

D=A1*B2-A2*B1; Dx=B1*C2 -B2*C1; Dy=C1*B2-A1*C2

Mối quan hệ giữa (1) và AB đợc thể hiện :

Nếu D0 và điểm P(Dx/D,Dy/D) nằm trên đoạn AB thì (1) cắt AB.

Nếu D0 và điểm P(Dx/D,Dy/D) nằm ngoài đoạn AB thì (1) cắt đường thẳng chứa AB nhưng không cắt AB .

Nếu D=Dx=Dy=0 thì AB(1) .

Nếu D=0 , Dx/D0 hoặc Dy/D0 thì AB song song với (1) .

Chúng ta xây dựng hàm : Lines_Cut_AB

Lines_Cut_AB bằng :

1 : Nếu Đoạn thẳng cắt đường thẳng .

2 : Nếu Đoạn thẳng song song với đường thẳng

3 : Nếu Đoạn thẳng thuộc đường thẳng .

4 : Nếu đường thẳngchứa Đoạn thẳng cắt đường thẳng một điểm nằm ngoài đường thẳng .

```
Function        Lines_Cut_AB (L1:Lines ; P,Q : Point ) :Byte;
Var
    L2      : Lines ;
    Giao    : Point_Chung ;
    D , Dx , Dy  : Longint ;
    Begin
        With L2 do Xac_DinhABC (P,Q,a,b,c);
        D := L1.a*L2.b -L2.a*L1.b;
        Dx :=L1.b*L2.c-L1.c*L2.B ;
        Dy:=L1.c*L2.b-L1.b*L2.c ;
        If D=0 Then
        Begin
            If ( Dx=0)And (Dy=0)Then Lines_Cut_AB:=3 ;
            If ( Dx<>0)Or(Dy<>0)Then Lines_Cut_AB:=2 ;
        End ;
        If D<>0 Then
        Begin
            Giao.x := Dx/D ; Giao.y := Dy/D ;
            If Thuoc_Doan (Giao,P,Q) then Lines_Cut_AB:=1
            Else Lines_Cut_AB:=4 ;
        End ;
    End ;
```

c. Đoạn thẳng cắt đoạn thẳng :

Xét hai đoạn thẳng : AB và CD thì chúng cắt nhau hay không thì chúng ta có thể xét theo hai cách. Trong bài này tôi xin đề nghị cả hai cách, mỗi cách có những ưu , nhược điểm khác nhau .

Cách 1 : Cách Theo Phương Trình Đường Thẳng :

```
Function                    Cat_Nhau1 ( P,Q,M,N : Point ) : Boolean ;
Var
  L      :      Lines ;
  Begin
        Cat_Nhau1:=False;
        With L Do Xac_DinhABC(P,Q,a,b,c);
        If  Lines_Cut_AB(L,M,N)<>1 then Exit;
        With L Do Xac_DinhABC(M,N,a,b,c);
        If Lines_Cut_AB(L,P,Q)<>1 then Exit ;
        Cat_Nhau1::=True ;
  End ;
```

Tuy hàm này ngắn gọn m nhưng chúng ta lại phải xây dựng cả một thư viện hàm , thủ tục ở trên . Chính vì thế thực ra nó dài , nhưng về tính đơn giản và dễ nhớ hơn vì nó mang tính toán học thuần tuý. Còn sau đây chúng ta xét một cách khác mang tính sáng tạo hơn. Các bạn có thể tham khảo ở quyển sách : "Cẩm Nang Thuật Toán " -Tập 2 của Robert Sedgewick .

Cách 2 :

```
  Function     ccw ( P0 , p1 , p2 : Point ) : Integer ;
  Var
        Dx1,Dx2,Dy1,Dy2 :       Integer ;
        Begin
              Dx1 := P1. x -P0.x ; Dy1 :=P1.y-P0.y ;
              Dx2 := P2. x -P0.x ; Dy2 :=P2.y-P0.y ;
              If Dx1*Dy2>Dy1*Dx2 then ccw :=1 ;
              If Dx1*Dy2<Dy1*Dx2 then ccw :=-1 ;
              If Dx1*Dy2=Dx2*Dy1 then
              Begin
                    If (Dx1*Dx2<0)Or(Dy1*Dy2<0)Then ccw:=-1
                    Else                                          If
(Dx1*Dx1+Dy1*Dy1)>=(Dx2*Dx2+Dy2*Dy2)
                    then ccw:=0 else ccw:=1 ;
              End ;
        End ;
  Function          Cat_Nhau2( P,Q,M,N : Point ) : Boolean ;
  Begin
        Cat_Nhau2:=((ccw(P,Q,M)*ccw(P,Q,N))<=0)And
                  ((ccw(M,N,P)*ccw(M,N,Q))<=0)
```

End ;

4. Đa Giác :

a. Tam Giác :

Một tam giác được định nghĩa là tập ba điểm không thẳng hàng : A(x1,y1);B(x2,y2);C(x3,y3) . Chúng ta có thể tính diện tích tam giác theo công thức tính diện tích đa giác (công thức hình thang- hoặc công thức Pic mà tôi sẽ bàn sau) . Hoặc chúng ta tính theo công thức Herong :

S:=Sqrt((p-a)*(p-b)*(p-c)*p);

Trong đó a , b , c là độ dài ba cạnh của tam giác . P=(a+b+c)/2 ;

b. Hình chữ nhật : (trường hợp cho cách cạnh song song các trục toạ độ) .

Chúng ta xét trong hệ toạ độ một hình chữ nhật : ABCD, A(x1,y1); B(x2,y2) ; C(x3,y3) ; D (x4,y4) . Nhưng có một điều đặc biệt là chúng ta chỉ cần xác định toạ độ đỉnh của hai đỉnh đối nhau thì xác định được một hình chữ nhật duy nhất . Chính vì thế thông thường chúng ta gọi toạ độ của điểm dưới trái và đỉnh trên phải là hai điểm đặc trưng cho hình chữ nhật đó .

c. Một đa giác :

Một Đa giác : A1A2....An có toạ độ : Ai (xi ,yi) .

Ngời ta định nghĩa đa giác đó là lồi khi mọi điểm còn lại của đa giác nằm cùng phía với nhau so với một cạnh nào đó .

Diện tích một đa giác được tính theo công thức hình thang như sau :

S := | [(Xi-Xi+1)*(Yi+Yi+1)/2] |

Nếu toạ độ của các đỉnh đa giác là nguyên thì chúng ta có thể sử dụng công thức sau :

d. Bao lồi :

Bao lồi của một tập điểm là một hình đa giáckhép kín có các đỉnh là một trong các đỉnh của tập điểm đó , và thoả mãn đa giác lồi . Thông thường chúng ta cần phải tìm đa giác bao với chu vi nhỏ nhất . Có nhiều thuật toán để giải quyết bài toán này. Đặc biệt phương pháp quét :

Phương Pháp quét đường thẳng :

Chúng ta đi từ một đỉnh chắc chắn thuộc bao lồi (là những điểm có tung độ /hoành độ lớn nhất hoặc nhỏ nhất) .Chúng ta tìm các đỉnh tiếp theo ,đỉnh nào thoả mãn chứa toàn bộ các đỉnh còn lại một bên mặt phẳng thì ta lấy điểm đó cho đến khi lặp lại điểm ban đầu .

Thủ tục giải quyết nó như sau :

```
Procedure        Scan ;
Begin
        Xác định thuộc đa giác ;
        Repeat
```

tìm đỉnh tiếp theo mà thoả mãn điều kiện lồi của đa giác ;
Until lặp lại đỉnh ban đầu ;
End ;

Lời bàn :

Có thể các bạn cho rằng những đoạn trên là quá dài, hoặc không cần thiết. Nhưng các bạn có thể sẽ mắc phải sai lầm rất nhiều khi ngồi trong phòng thi mà lúc đó thì tâm trạng không ổn định. Mặt khác những bài toán hình học thì không khó về giải thuật nhưng lại rờm rà về chương trình. Chính vì thế chỉ cần một sai sót nhỏ thì chúng ta sẽ không thể nào có thể sửa chương trình trong thời gian cho phép . Chính vì thế học phần này một cách bài bản, nghiêm túc thì chúng sẽ hẳn không khó đối với chúng ta. Không nên khinh thường những thứ sơ đẳng như trên .

Sau đây tôi xin đề nghị những bài toán hình học, có thể rất quen biết với rất nhiều các bạn . Nhưng chúng ta hãy nghiên cứu kĩ từng bài toán, chúng ta sẽ thấy sâu hơn nội dung cần giải quyết của bài toán đó. Đấy chính là một đặc điểm mà chúng ta cần chú ý. Sẽ có những bài rất mới, nhưng tôi sẽ cố gắng đề cập hết tất cả những thuật giải của chúng. (Có thể sẽ có chương trình mẫu và bài test cụ thể cho các bạn kiểm chứng) .

II Bài Toán :

Chúng ta đi từ những bài toán hết sức đơn giản đến những bài toán khá phức tạp . Trong lời giải tôi có đề xuất đến những hàm mà đã xây dựng ở trên. Trong phần hướng dẫn giải tôi chỉ đề cập đến thuật toán và sơ qua về chương trình còn cụ thể các bạn xem trong lời giải tôi cho kèm theo. Chính vì thế có thể tiện theo dõi .

1 . Dạng 1 : Các bài toán về tương điểm , cắt nhau .

Bài toán 1 :

" Cho một đa giác n đỉnh , và một điểm P . Hỏi P có vị trí nh thế nào với đa giác đó "

Bài giải :

Đã có rất nhiều thuật toán để giải quyết bài toán này, ở đây chúng ta chỉ xét với hai thuật toán thường được áp dụng nhất. Trước tiên chúng ta quy định cách lưu các toạ độ của đa giác như sau : lưu bằng mảng một chiều a:aray[1..101]of point

Thuật toán 1:

ý tưởng thuật toán như sau : từ điểm M chúng ta kẻ tia Mx cắt các cạnh của đa giác. Sau đó đếm số giao điểm của tia đã vẽ với đa giác . Nếu số giao điểm là lẻ thì điểm đó nằm trong đa giác, ngược lại thì nằm ngoài .

Về chương trình của thuật toán này hầu hết đều được các sách tin học đưa ra chương trình (Các bạn có thể tự tham khảo) cho nên tôi không đưa ra trong bài viết này. Nhưng hầu hết các cách giải đó đều vẫn chưa giải quyết các nhược điểm của thuật toán này đó là:

Nhược điểm 1: phải kiểm tra tia đã cho có trùng vơi một số cạnh nào không của đa giác. ví dụ :

Nhược điểm 2: khi điểm giao của tia vừa kẻ lại trùng với một đỉnh của đa giác . Thì lúc đó chúng ta không thể biết lúc nào thì điểm giao đó được tính là một giao điểm và khi nào thì không tính .

Hình 1 Hình 2 Hình 3

Ta nhận thấy ở hình 1 thì mặc dù có 1 giao điểm nhưng điểm đó lại nằm ngoài đa giác . Trong khi đỏ hình 2 thì tia đó có 2 giao điểm khác với đỉnh và 2 giao điểm trùng với đỉnh thì vẫn nằm trong Và ở hình 3 thì tia đó có 1 giao điểm không trùng với đỉnh của đa giác và có 4 giao điểm trùng với đỉnh của đa giác .

Việc biết được khi nào thì tính giao với 1 đỉnh thì tính là 1 giao điểm là rất khó .

Công việc khắc phục 2 điểm yếu của thuật toán là rất khó khăn, nhất là nhược điểm thứ 2. Vì vậy mọi kết quả của thuật toán này thì chỉ cho một kết quả gần đúng .

Thuật toán 2:

Chúng ta sử dụng tổng các góc. ý tưởng của thuật toán này như sau :

Đầu tiên chúng ta xây dựng một hệ trục lượng giác lấy chiều kim đồng hồ làm chiều dương : x

Ví dụ : góc ^xoy =45 thì góc ^ yox=45

o y

Ta nối các đỉnh của đa giác với điểm M cần xét . Xi+1

thì tạo thành n góc : Xi

Vậy nếu (^XiOXi+1)=360 thì điểm này nằm trong đa giác và ngược lại thì không . (với X1 Xn)

Thuật toán luôn đúng đắn với mọi trường hợp về hình dạng của đa giác, và một điều kiện là điểm này không thuộc biên của đa giác. Vì vậy nhiệm vụ của chúng ta là cần xây dựng các hàm : hàm kiểm tra thuộc biên, hàm tính góc .

Đầu tiên ta xây dựng hàm kiểm tra thuộc biên của điểm đó như sau :

```
Function kiemtrabien:boolean;
var j:byte;
begin
        kiemtrabien:=true;
```

```
            for j:=1 to n do
            if thuoc_doan ( M , A[i],A[i+1]) then exit ;
            kiemtrabien:=false;
        end;
```

Hàm này cho giá trị đúng nếu điểm đó thuộc biên và ngược lại. Bây giờ nhiệm vụ chính của chúng ta là cần xây dựng hàm tính góc (tính theo radian).

Ta áp dụng các kết quả toán học trong tam giác ABC thì :

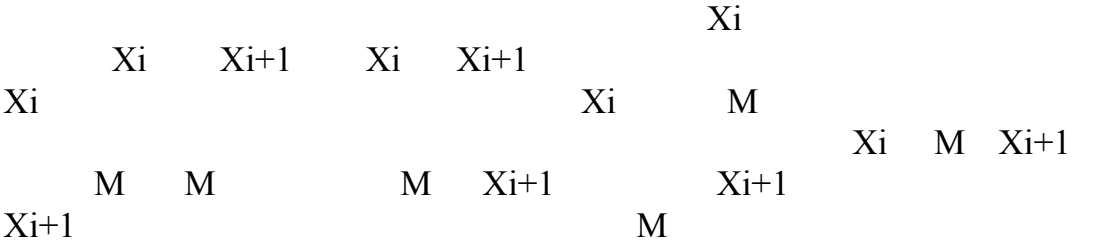
Cos ^ABC=(SQR(AB)+SQR(BC)-SQR(AC))/(2*AB*BC);

1+SQR(tg ^ABC)=1/SQR(cos^ABC);

Vì vậy ta áp dụng với hệ 3 điểm : Xi,Xi+1,M thì ta có : biết được toạ độ 3 đỉnh này thì ta tính được độ dài các cạnh của tam giác tạo bởi 3 đỉnh này. Sau đó áp dụng định lý Cosin ở trên thì ta tính được COS^XiXi+1M .Vì trong Pascal không có hàm arccos hay arcsin mà chỉ có hàm arctan nên ta phải đổi giá trị cos góc đó thành giá trị tag. Rồi sau đó áp dụng ta tính được độ lớn của góc đó (chưa tính chiều) . Nên giá trị góc trong Pascalà theo đơn vị radian . Nên 360o=2*pi .

```
Function      goc ( p , q : point ) : double ;
          var
            k1 , k2 , k3 , t : real ;
            begin
               If ptrinhdthang ( p , q , m )  = 0 then
               begin
                  If thuoc_doan ( M , P , Q ) then goc := 0
                  else
                  goc := pi ;
                  exit ;
               end ;
                  k1 := sqr ( p.x - q.x ) + sqr ( p.y - q.y ) ;
                  k2 := sqr ( p.x - M.x ) + sqr ( p.y - M.y ) ;
                  k3 := sqr ( q.x - M.x ) + sqr ( q.y - M.y ) ;
               if k1 = k2 + k3 then goc := pi/2 else
               begin
                     t:=(K2+k3-k1)/(2*sqrt(k2)*sqrt(k3));
                     t:=sqrt(1/sqr(t)-1);
                  if k1>k2+k3 then
                  goc:=(pi-arctan(t)) else
                     goc:=arctan(t);
               end ;
            end;
```

Bây giờ nhiệm vụ chính của chúng ta là xác định chiều của góc đó . Ví chọn chiều dương là chiều của kim đồng hồ nên ta có với 3 điểm Xi,Xi+1,M thì dấu của góc ^XiMXi+1 là:



Hình 4 Hình 5 Hình 6 Hình 7 Hình 8 Hình 9

Từ các trường hợp của ba điểm này ta có cách xây dựng dấu của góc tạo bởi ba điểm Xi,Xi+1,M tại đỉnh M như sau:

```
Function      giatrigoc ( p , q : point ) : double ;
          var
            t : point ;
            begin
                if  (Q.y=P.y) then
                   begin
                   if P.x<Q.x then
                      begin
                       if M.y>=Q.y then giatrigoc:=-goc(p,q)
                       else
                       if M.y<Q.y then Giatrigoc:=goc(p,q)
                   end
                   else
                   if P.x>Q.x then
                   giatrigoc:=-giatrigoc(q,p);
               end
               else
                   if (Q.y<P.y ) then
                   begin
                         t.x:=p.x;
                         t.y:=q.y;
                         if P.x=Q.x then
                         begin
                       if M.x>=P.x then giatrigoc:=-goc(p,q)
                            else
                            if M.x<Q.x then giatrigoc:=goc(p,q)
                            else  giatrigoc:=0
```

```
                end
                    else
                    if P.x<Q.x then
                    begin
                        t.x:=P.x;
                          t.y:=Q.y;
                          if  ptrinhdthang(p,q,t)*ptrinhdthang(p,q,m)>0 then
                          giatrigoc:=goc(p,q)
                          else
                          if ptrinhdthang(p,q,t)*ptrinhdthang(p,q,m)<0 then
                          giatrigoc:=-goc(p,q)
                          else
                          giatrigoc:=0;
                    end
                    else
                    begin
                        if ptrinhdthang(p,q,t)*ptrinhdthang(p,q,m)>0 then
                          giatrigoc:=-goc(p,q)
                          else
                          if ptrinhdthang(p,q,t)*ptrinhdthang(p,q,m)<0 then
                          giatrigoc:=goc(p,q)
                          else giatrigoc:=0;
                    end;
                end;
                if P.y<Q.y then
                giatrigoc:=-giatrigoc(q,p);
            end;
```

Vì trong các phép tính trên thì giá trị tổng số góc được pascal tính như gần đúng của số thực, cho nên 2*pi chỉ đúng sau khoảng 10 số sau dấu phẩy. Chính vì thế chúng ta sẽ làm tròn gần đúng 10 số sau dấu phẩy. Nếu độ làm tròn càng lớn thì độ chính xác càng cao. Như vậy khi đó chúng ta chỉ cần một phần thân chương trình chính như sau:

```
        procedure       xuli ;
            var
            k : double ;
            t : double ;
            i : integer ;
            f : text ;
              Begin
                assign ( f , fo ) ; rewrite ( f ) ;
                clrscr ;
                a[n+1]:=a[1];
```

```
                if kiemtrabien then writeln(f , ' diem do nam tren bien da
giac')
                else
                begin
                k:=0;
                    for i:=1 to n do
                   k := k + giatrigoc(a[i],a[i+1]);
                (* Tinh pi xap xi ma thoi *)
                    if  trunc(k*10000000)=Hoa_Tra_My then
                writeln(f , ' diem nay nam trong da giac ')
                    else writeln(f ,' diem nay nam ngoai da giac');
                end;
            close ( f ) ;
        end;
```

Bài toán 2 :

" Cho đa giác N đỉnh , hỏi có bao nhiêu điểm nguyên có trong đa giác đó (không kể cả điểm thuộc biên) "

Bài giải :

Có thể có một lời giải, đó là chúng ta tìm từng điểm xem nó có thuộc đa giác không. Cách này sẽ không ổn, bởi vì thứ nhất là nó có chương trình rất dài và lằng nhằng. Thứ hai là chúng ta phải kiểm tra hết tất cả các điểm có trong giới hạn của toạ độ đa giác. thì như vậy sẽ phải xét đến trường hợp xấu nhất đó là sqr(32767*2) . như vậy chương trình sẽ không thể chạy được. Chúng ta sẽ đi từ một công thức, mà đã nêu trong phần lí thuyết, đó là công thức Pic và công thức hình thang, hai công thức tính diện tích của đa giác. Chúng ta hãy xét lại nó :

+ Công thức hình thang : " Các đỉnh của một đa giác (không nhất thiết là lồi) nằm ở các điểm nguyên. Bên trong nó có n điểm nguyên , còn trên biên m điểm nguyên. Chứng minh rằng diện tích của nó bằng S= n + m/2 -1 ."

từ công thức hình thang chúng ta đã thấy được lời giải của nó. Thật vậy chúng ta dùng công thức hình thang để tính diện tích của nó (nhng phải quy đổi làm tròn một dấu phẩy vì chắc chắn diện tích của nó là một phân số). Sau đó ta tìm các toạ độ các điểm nguyên có trên các cạnh của nó . thì ta sẽ có n = (s*2+2-m)/2 ;

Công việc bây giờ là chúng ta phải tìm đợc số điểm nguyên có trên cạnh của nó. Chúng ta xét trên cạnh : i , i + 1 . thì ta sẽ có phương trình đường thẳng cạnh này là :

Ai*x+Bi*y+Ci = 0 .

Để tìm số điểm nguyên trên cạnh này thì chúng ta xét các toạ độ của x [a[i],a[i+1]] . Nếu Ci+Ai*x mod Bi =0 thì điểm có toạ độ (x , - (Ci +

Ai*x)/Bi) là toạ độ nguyên . Chúng ta sẽ dùng hàm để tính số điểm nguyên trên một cạnh là :

```
Function    Count_point ( P, Q: Point) : Integer ;
Var
    L : Lines ;
    Count , i : Integer ;
    Begin
        With L do
        Xac_DinhABC( P,Q,a , b, c ) ;
        Count := 0 ;
        If P.x>Q.x then
        Begin
            Count_Point :=Count_Point ( Q , P ) ;
            Exit ;
        End ;
        For i :=P.x to Q.x do
         If (L.a*i+L.c) Mod L.b = 0 then Inc ( Count ) ;
        Count_Point:=Count ;
    End ;
Procedure   Main_Count ;
Var
    So_Diem , i : Integer ;
    Begin
        So_Diem:=0 ;
        For i := 1 to n do  Inc ( So_Diem, Count_Point ( A[i],A[i+1]) ) ;
        So_Diem := So_Diem - N ; { loại trừ trờng hợp chúng ta tính các đỉnh thành hai lần }
    End ;
```

Công đoạn cuối cùng chúng ta có thể xem ở lời giải. Các bạn nên tự làm lại và thử Test mà tôi cho trong bộ kèm theo .

Bài toán 3 : Quy Hoạch Sân Hình Chữ Nhật

Đề Bài :

" Một khu đất trồng có N cây cổ thụ. Người ta chuẩn bị quy hoạch một khu dân cư trong đó có một sân chơi hình chữ nhật. vì số cây này tản mát trên một diện tích khá rộng nên người ta muốn sân này có diện tích bé nhất mà chứa hết các cây cổ thụ . Hãy lập trình xác định sân chơi này .

Dữ liệu vào : Cho trong file văn bản SAN.INP gồm :

Dòng đầu tiên ghi số N (N<=200)

N dòng sau ghi tạo độ của N cây cổ thụ đó

Kết qủa : Ghi ra file văn bản SAN.OUT như sau :

Dòng đầu tiên ghi diện tích của sân chữ nhật ấy .

Dòng tiếp theo lần lượt ghi 2 cặp toạ độ đối đỉnh của hình chữ nhật ấy .
(các toạ độ là các số thực - làm tròn đến 2 con số 0) . “

Ví Dụ :

SAN.INP SAN.OUT

8 1 2 2 4 3 1 2 3 4 2 5 0 7 2 4 3 10.00 1.00 2.00 7.00 2.00

Bài giải :

Nhưng có rất nhiều Hình chữ nhật bao lấy một đa giác lồi. Ta cũng có thể thấy rằng để tận dụng hết về diện tích thì hình chữ nhật nhỏ nhất phải thoả mãn các điều kiện sau :

Có đường thẳng chứa một cạnh trùng với đường thẳng chứa một cạnh của đa giác bao ấy .

Đường thẳng chứa ba cạnh còn lại, mỗi đường phải đi qua ít nhất một đỉnh của đa giác bao ấy .

Từ các tính chất ấy chúng ta sẽ lần lượt nhận từng cạnh của đa giác bao ấy làm cạnh của hình chữ nhật. Với mỗi trường hợp ta tìm đỉnh có khoảng cách xa nhất tới cạnh ấy làm đỉnh có đường thẳng chứa cạnh đối diện của hình chữ nhật ấy đi qua. Sau đó ta tìm hai đỉnh mà qua đó kẻ đường thẳng vuông góc với hai cạnh đã xác định ấy. Nếu đỉnh nào có đường thẳng vuông góc ấy không cắt cạnh nào của đa giác bao thì đường thẳng đi qua đỉnh ấy là một trong số cạnh còn lại của hình chữ nhật. Sau đó với mỗi lần xác định chúng ta xác định hình chữ nhật có diện tích nhỏ nhất. Thì hình đó là hình cần tìm

Bài toán 4 : Wall

Đề Bài :

“ Có N điểm dân cư một nửa theo thiên chúa giáo, một nửa theo đạo hồi nên hay xẩy ra xung đột. Chính vì thế chính phủ muốn xây một bức tường thẳng để ngăn N điểm dân cư thành hai nửa, mỗi nửa chứa một loại là dân cư theo một đạo .

Dữ liệu vào : Cho từ file văn bản WALL.INP :

Dòng đầu tiên ghi số N (N<=200)

N dòng tiếp, mỗi dòng là bộ ba số (x,y,e) thể hiện cho toạ độ của điểm dân cư đó và loại tôn giáo mà điểm dân cư đó theo (c=0) là thể hiện cho theo đạo thiên chúa, nếu e=1 thì thể hiện cho theo đạo hồi .

Kết qủa : Xuất ra file văn bản : WALL.OUT :

Dòng đầu tiên là số 0 nếu không thể xây dựng được và 1 trong trường hợp còn lại

Dòng sau là 3 số A , B , C là các hệ số biểu diễn đường thẳng chứa bức tường (phương trình Ax+By+C=0).

Ví Dụ :

WALL.INP WALL.OUT

10 10 0 10 15 0 30 12 0 23 13 0 34 14 1 18 28 1 29 20 1 37 22 1
38 26 1 1 -14 -16 612

Bài Giải :

Bài toán này nhìn có vẻ là phức tạp, nhưng khi chúng ta chia N điểm trên thành 2 loại điểm (loại có e=0 và loại có e=1) . Với mỗi loại ta tìm bao lồi của chúng. Sau khi chúng ta xác định được S1 và S2 là hai đa giác chứa hai loại điểm trên. Thì ta có nhận xét rằng :

1) Nếu S1 và S2 cắt nhau, thì dẫn đến có ít nhất một điểm của đa giác này sẽ thuộc miền trong của đa giác kia. Điều này dẫn đến sẽ không thể có một đường thẳng nào mà có thể phân chia được hai loại điểm này .
2) Nếu ở S1 và S2 không cắt nhau. Ta thấy rằng sẽ tồn tại một đỉnh của S1 mà từ đó chúng ta kẻ lần lượt các đòng thẳng song song với các cạnh của S2 thì tồn tại một đường thẳng không cắt S2. Như vậy đường thẳng đó là đường thẳng cần tìm .

Bài toán 5 : Chia Đất

Đề Bài :

Có một mảnh đất, trên biên của nó có cắm một số cột mốc để mảnh đất có thể xem như một đa giác có biên gồm các đoạn thẳng kề nhau nối các cột mốc này. Bắt đầu từ một cột mốc nào đó để được đánh số 1, người ta đi vòng quanh mảnh đất theo một chiều xác định để đánh số các cột mốc kế tiếp (2 , 3 , ..N) theo thứ tự được gặp .

Yêu Cầu : hãy phân chia mảnh đất thành hai phần bằng một đoạn thẳng nối hai cột mốc nào đó để độ chênh lệch diện tích của hai phần được chia là nhỏ nhất. Đoạn thẳng chia không được có phần nàp nằm ngoài mảnh đất và không được chứa cột mốc nào ngoài hai cột mốc đầu mút của nó .

Dữ Liệu : vào từ file : Poly.Inp :

Dòng đầu tiên ghi số nguyên N (4<=N<=100),

N dòng tiếp theo, mỗi dòng ghi hai số nguyên x , y là cặp toạ độ trong một hệ vuông góc nào đấy của một cột mốc (theo thứ tự các cột mốc từ 1 đến N) . Các số trên một dòng đợc ghi cách nhau ít nhất một dấu trắng .

Kết Quả : Ghi ra file : poly.Out một dòng gồm hai số , ghi cách nhau ít nhất một dấu trắng , là số hiệu của hai cột mốc tạo thành đoạn thẳng chia .

Ví Dụ :

Poly.Inp	PoLy.Out
10 2 -3 1 -2 -1 -1 -3 0 -3 0 -2 1 -1	1 1 5 10

2 0 0 -1 2

Bài Giải :

Chúng ta thấy rằng, thuật giải hết sức đơn giản, đó là chúng ta đi thử từng cặp điểm trong n điểm đó, rồi với mỗi trường hợp chúng ta xem chúng có thoả mãn cách chia không . Trong các cách thì cách nào có hiệu diện tích nhỏ nhất là lấy . Nhưng chúng ta cần phải quan tâm rất nhiều đến chương trình của bài toán. Để giải quyết tốt bài toán thì chúng ta phải giải quyết tốt các công đoạn sau :

Công thức tính diện tích của một đa giác có các đỉnh nguyên .

Hàm kiểm tra cắt nhau . Nếu đoạn nối cắt một cạnh nào đó (khác đỉnh) thì loại.

Nếu thoả mãn không cắt nhau , nhưng cũng cần kiểm tra xem đường đó có nằm ngoài đa giác đó không . Ví dụ như trường hợp sau sẽ không thoả mãn :

1

2

33 3

5 4

như ở trường hợp trên thì chúng ta thấy rằng đường chia: 24 không thoả mãn yêu cầu của bài toán. Để giải quyết một cách nhanh, mà lại thuận lợi, đó là chúng ta luôn phải tính S1, S2 là diện tích mỗi phần sau khi chia (các phần này tạo bởi hai đường khép kín có cạnh nối là đoạn chia). Và ta tính diện tích chung của đa giác là S . Thì nếu : S1+S2=S thì đường chia này không có đoạn nào nằm ngoài đa giác .

Bài toán này chính là đề thi học sinh giỏi QG 1999-2000 , bài số 4 , bảng B .

Bài toán 6 : chướng ngại vật

Đề Bài :

Trên mặt phẳng toạ độ có một hình vuông một đỉnh ở gốc toạ độ và một đỉnh có toạ độ (100,100). Trong hình vuông lớn có M hình vuông nhỏ hơn là những vật cản, hai vật cản khác nhau không có điểm chung. Cần tìm một hành trình ngắn nhất từ điểm (0,0) đến điểm (100,100) sao cho hành trình đó không đi xuyên qua các vật cản mà cùng lắm chỉ chạm vào điểm thuộc cạnh.

Dữ liệu: vào được cho bởi file OBSTACLE.INP trong đó dòng thứ nhất ghi số M<=30, trong M dòng tiếp theo, mỗi dòng ghi ba số U, V, W với ý nghĩa có vật cản mà góc trái dưới có toạ độ (U,V) và cạnh bằng W, W nguyên dương và các số U, V nguyên không âm.

Kết quả: ghi ra file OBSTACLE.OUT nh sau: dòng thứ nhất ghi độ dài hành trình, dòng thứ hai ghi số D là số điểm mà tại đó hành trình đổi hướng, nếu D>0, trong D dòng tiếp theo, mỗi dòng ghi một điểm tại đó hành trình đổi hướng theo trình tự trên hành trình. Số thực lấy với 3 số lẻ sau dấu phảy.

Ví dụ

OBSTACLE.INP	OBSTACLE.OUT
5 5 5 5 5 5 15 5 15 10 5 15 20 5 90 90 5	142.980 3 5 10 20 20 95 90

Bài Giải :

Thực chất có rất nhiều đường đi ngắn nhất thoả mãn, thế nhưng để chọn ra một kết quả không phải là dễ. Chúng ta thấy rằng , nếu các điểm quành là một điểm trong hình vuông lớn, thì ta cũng có thể nối một con đường khác mà đi qua các đỉnh của các hình vuông con , nhưng độ dài vẫn thoả mãn bằng nhau. Vậy để đơn giản , chúng ta chọn cách nối từ điểm (0,0)và qua các đỉnh của hình vuông có thể có để đến đỉnh (100,100) là ngắn nhất . Vậy tức là nó trở thành bài toán tìm đường đi ngắn nhất trong đồ thị 4 * N + 2 đỉnh , mà độ dài cung nỗi chúng bằng khoảng cách trong không gian giữa chúng .

Thực chất bài toán này là một bài toán pha trộn giữa hình học và đồ thị một cách đẹp mắt . Đây là một bài toán hay, nếu bạn nào tìm ra thuật toán của nó là rất tốt, còn nếu không thì cần phải học hỏi rất nhiều từ bài toán này. Có thể mở rộng cho bài toán bằng nhiều cách để các bạn tự giải quyết thêm như là : Các hình trong bài toán không nhất thiết là hình vuông mà còn có thể là hình chữ nhật . Hình vuông bao ngoài có thể là một hình có kích thước khác . Chính vì thế mà qua mỗi một thây đổi ta lại cho ra một bài toán mới. Đây chính là điều mà các bạn nên làm cho các bài toán khác. Các bạn có thể tham khảo lời giải mẫu của tôi để biết thêm .

2. Dạng 2 : Các bài toán phủ và chia hình :

Bài toán 7 : Tính diện tích các hình chữ nhật

Đề bài :

Cho một tập hình chữ nhật trong mặt phẳng. Mỗi hình được xác định bởi hai cặp toạ độ đỉnh đối diện (trái dưới và phải trên). Biết rằng các tọa độ của chúng là nguyên .

Yêu Cầu : Hãy tính phần diện tích trong mặt phẳng toạ độ mà tập các hình này phủ lên .

Dữ Liệu : Vào từ file : AreaRec.Inp như sau :

Dòng đầu tiên ghi số hình chữ nhật N .

N dòng tiếp theo , mỗi dòng ghi 4 số thể hiện toạ độ hai toạ độ của đỉnh đối diện của hình chữ nhật đó .

Kết Quả : Ghi ra file : AreaRec.Out nh sau :

Hiện lên một số duy nhất là diện tích bị phủ của các hình chữ nhật đó .

N<=300, và toạ độ của các hình chữ nhật : -32767<=x,y<=32767 .

Ví Dụ :

AreaRec.Inp	AreaRec.Out

3 1 1 4 4 2 1 5 5 3 3 6 6 20

Bài giải :

Phương pháp giải này là một phương pháp chung để cho các bạn giải các bài toán thuộc dạng này : đó là phương pháp Chia Lưới . Tôi có thể trình bày nh sau :

Chúng ta sắp xếp các đỉnh của các đỉnh của các hình chữ nhật theo một trật tự tăng dần

Chúng ta sẽ lần lượt lấy từng cặp đỉnh một theo thứ tự tăng dần, chúng sẽ tạo ra một hình chữ nhật.Vì chúng là phần chia nhỏ, nên nếu chúng thuộc một hình chữ nhật nào đó thì chúng sẽ thuộc miền phủ của hình chữ nhật phủ, cho nên ta tăng diện tích của chúng lên .

Cứ như thế ta thực hiện sẽ tính được phần diện tích bao của các hình chữ nhật mà loại trừ được lặp .

Để các bạn hiểu rõ tôi có thể viết một số thủ tục mẫu cho các bạn tham khảo mà hiểu thêm :

```
(*    Quy ớc :
   Mảng A[i] là chứa toạ độ hoành độ  , B[i] là tung độ
   Mảng Rec[i] là mảng chứa các đỉnh của hình chữ nhật (P1,P2)
 *)
      Procedure    Quy_Doi ;
      Begin
            For i :=1 to n do
            Begin
                  A[i*2-1]:=Rec[i].p1.x ; A[i*2]:=Rec[i].P2.x;
                  B[i*2-1]:=Rec[i].p1.y; B[i*2]:=Rec[i].p2.y;
            End ;
            Sort(A); Sort(B);
      End ;
      Function              KiemTra ( I , J : Integer):Boolean ;
      Var
            t  :  integer ;
            Begin
                  KiemTra:=True ;
                  For t:=1 to n do
                   If (Rec[t].p1.x<=A[i-1])And(A[i]<=Rec[t].p2.x)
                  And(Rec[t].p1.y<=B[j-1])And(B[j]<=Rec[t].P2.y)       then
Exit ;
                  KiemTra:=False;
            End ;

      Procedure    Main ;
      Begin
```

```
        s:=0 ;
        For i:=2 to 2*n do
         For j:=2 to 2*n do
          if KiemTra(i,j) then
          s :=s+(a[i]-a[i-1])*(b[j]-b[j-1]);
End ;
```

Bài toán 8 : TriColor

Đề bài :

Cho N hình chữ nhật trong mặt phẳng toạ độ (N<=200) . Và các toạ độ của chúng là nguyên , có các đỉnh có toạ độ (x,y) và -32767<=x,y<=32767 . Các hình chữ nhật được quy ước là hai đỉnh đối diện, trái dưới và phải trên. Biết rằng mỗi hình có một màu nhất định trong số ba màu : Xanh , Đỏ và Tím . Miền được phủ bởi ba màu, mỗi màu một lần gọi là miền ba màu đơn. Còn miền được phủ cũng được ba màu đó, nhưng số màu Xanh hoặc Đỏ Hoặc Tím có thể nhiều lần thì gọi là miền ba màu. Tức là miền ba màu chứa miên ba màu đơn .

Yêu Cầu : Hãy tính diện tích của miền ba màu và miền ba màu đơn .

Dữ Liệu : Vào từ file : TriColor.Inp nh sau :

Dòng đầu tiên ghi số N

Các dòng tiếp theo ghi 5 số , 4 số đầu là toạ độ của hình chữ nhật đó , còn số cuối là một trong ba số 1 ,2 ,3 được hiểu là hình đó có màu lần lượt là Xanh , Đỏ , Tím .

Kết Quả : Ghi ra file TriColor.Out nh sau :

Dòng đầu tiên là diện tích miền ba màu

Dòng thứ hai là diện tích miền ba màu đơn .

Ví Dụ :

TriColor.Inp	TriCoLor.Out
3 1 1 4 4 1 1 1 5 5 3 3 6 6	2 2

Bài giải :

Cũng dùng phương pháp chia lưới, nhưng trong mỗi hình chứ nhật con, chúng ta xem nó thuộc những hình nào có màu gì , rồi kiểm tra nó có phải miền ba màu hoặc ba màu đơn không .

Bài Toán 9 : Vườn Táo

Đề bài :

Trong một mảnh vườn hình vuông cạnh 1000 m có trồng N cây táo. Vị trí mỗi cây táo được xác định bởi toạ độ (x,y) trong hệ toạ độ vuông góc với gốc toạ độ là góc trái dưới của vườn và các trục toạ độ song song với các cạnh vườn. Hãy tìm hình chữ nhật lớn nhất sao cho các cạnh song song với các cạnh của vườn, và trong nó không chứa cây táo nào (mà chỉ có thể chứa trên biên) .

Dữ Liệu : Vào từ file Tao.Inp :
Dòng thứ nhất ghi N số từ 1 đến 600 .
Trong N dòng sau mỗi dòng ghi hai số nguyên dương x,y ,0<x,y<1000 là toạ độ của các cây táo .
Kết Quả : Ghi ra file Tao.Out :
Dòng thứ nhất ghi diện tích hình chữ nhật
Dòngthứ hai ghi số hai số là toạ độ của đỉnh trái dưới
Dòng thứ ba ghi hai số là toạ độ của đỉnh trên phải của hình chữ nhật .
Ví Dụ :

TAO.INP	TAO.OUT
7 280 100 200 600 400 200 135 800 800 400 600 800 900 210 200 200 800 800	360000

Bài giải :

Ta thêm vào tập các cây những cây giả là hình chiếu của các cây thật trên bốn cạnh của khu vườn. Sắp xếp lại các cây theo thứ tự tăng dần của hoành độ. Thực chất ta đã chia chúng thành các hình nhỏ hơn. Sau đó tìm từng hình một thoả mãn điều kiện của hình chữ nhật (không chứa táo , diện tích lớn nhất) .

Bài Toán 10 : Tam giác

Đề Bài :

Trên mặt phẳng cho N (<=100)tam giác được xác định bởi toạ độ của ba đỉnh (Xi1 , Yi1) , (Xi2,Yi2) , (Xi3,Yi3) . Tất cả các toạ độ đều là những số nguyên có trị tuyệt đối không vợt quá 3000. Cần tìm diện tích của phần mặt phẳng bị phủ bởi các tam giác này. Giả thiết rằng tất cả các tam giác đã cho là không suy biến (thoả mãn điều kiện tồn tại tam giác) .
Dữ Liệu : vào từ file : TamGiac.Inp
Dòng đầu tiên ghi số N
Dòng thứ i trong số N dòng tiếp theo chứa 6 số lần lợt là : Xi1 , Yi1 , Xi2 , Yi2 , Xi3 , Yi3 .
Các số trên một dòng đợc ghi cách nhau một dấu cách .
Kết Quả : Ghi ra file TamGiac.Out duy nhất một số là giá trị của diện tích cần tìm với 4 chữ số sau dấu phẩy .
Ví dụ :

Tamgiac.inp	Tamgiac.Out
2 0 0 2 0 1 2 0 0 2 0 1 1	2.0000

Bài giải :

Bài toán này tương đối khó, nó đòi hỏi chúng ta phải tính toán diện tích với một kết quả gần đúng. Thế nhưng lời giải của nó thì cực kì dài và phức tạp. Nội dung của thuật toán : ta chia nhỏ hệ trục toạ độ ra thành các đơn vị nhỏ hơn. ta sẽ tìm cách xây dựng cách hình chữ nhật con trong đó để tính từng vùng một. Công việc giải quyết nó cũng rất khó khăn. Các bạn nên dùng

tham lam là hơn hết, vì tuy nó ăn chỉ khoản những test có toạ độ mà [-1000,1000] thì có khả năng đúng cả , còn nếu test khó thì chúng ta chỉ mất vài điểm mà thôi .

Bài toán 11 : Part

Đề bài :

Trên mặt phẳng cho một hình vuông kích thước 1000*1000 . Người ta lần lượt vẽ N đường thẳng. Vấn đề đặt ra là cần đếm số phần của hình vuông bị chia bởi N đường thẳng đã vẽ .

Giả thiết Là :

1.) Sau mỗi lần vẽ độ dài của các phần thu được đều không nhỏ hơn 1
2.) Toạ độ của các đỉnh của hình vuông là (0,0),(0,1000),(1000,1000),(1000,0)
3.) Mỗi đường thẳng đều cắt hai cạnh nào đó của hình vuông

Ví Dụ:

Hình vẽ dưới đây cho thấy hình vuông được chia ra làm 10 phần khi vẽ 4 đường thẳng .

Dữ Liệu : vào từ file văn bản Part.Inp :

Dòng đầu tiên ghi số đường thẳng được vẽ N (N<=100)

Dòng thứ i trong số N đường tiếp theo ghi 4 số nguyên a , b , c, d là toạ độ của các giao điểm của đường thẳng thứ i với hai cạnh của hình vuông

Kết Quả : Ghi ra file văn bản Part.Out : số phần đếm được

Ví dụ :

PART.INP	PART.OUT
3 0 0 1000 1000 500 0 500 1000 0 500 1000 500	6

Bài giải :

Ta thấy, với một trạng thái của hình vuông, khi ta vẽ thêm một đường thẳng thì nếu đường thẳng đó cắt các đường thẳng đã vẽ tại X điểm (nếu trùng với điểm đã bị giao bởi hai đường khác thì không tính) thì số phần sẽ tăng lên x+1 phần nữa .

Nhưng để tránh trường hợp các điểm giao nhau có toạ độ trùng nhau, chúng ta dùng mảng A[1..100,1..100]of point để lưu dữ toạ độ điểm giao nhau của đường i với đường j. Ssau đó nếu có điểm giao nào thì ta dùng hàm kiểm tra để rồi tiếp nhận nó. Nên nhớ số phần ban đầu là 1 .

Bài toán 12 : Lưới đánh cá

Đề bài :

Để đánh cá Robinson đã chế tạo một loại lưới đánh cá bằng cách căng các đoạn dây trên các cạnh của một hình chữ nhật . Ví dụ :

Robinson muốn kiểm tra là với việc căng lưới như vậy thì mắt lới lớn nhất có diện tích là bao nhiêu .

Dữ Liệu : Vào File Back.Inp

Dòng đầu tiên ghi số N và toạ độ trái trên và phải dưới của hinh d chữ nhật đó .

N dòng kế tiếp , mỗi dòng là một cặp toạ độ thể hiện cho các đoạn dây

Kết quả : ghi ra file Back.Out diện tích của đa giác lớn nhất .

Bài giải :

Chúng ta dùng một kiểu dữ liệu để lưu dữ, biểu diễn các đa giác. Giải thuật của bài toán như sau :

Số đa giác ban đầu là 0. Đa giác ban đầu là đưa hình chữ nhật .

khi ta thêm một đường thẳng thì ta dùng thủ tục để chia các đa giác bị đường thẳng đó cắt thành hai phần, rồi kết nạp hai đa giác mới đó vào danh sách đa giác .

Cứ tiếp tục thêm cho đến hết đòng thẳng. đa giác có diện tích lớn nhất là đa giác cần tìm .

Tuy thuật toán rất đơn giản, nhưng chúng ta gặp rất nhiều khó khăn, đó là số đa giác có thể lên một số lượng rất lớn, mà mỗi đa giác ta lại lưu dưới dạng dữ liệu mảng, cho nên nếu N lớn thì có đôi lúc tràn dữ liệu . Mặt khác chương trình của bài toán rất khó khăn. Nếu các bạn lúng túng thì nên xem phần bài giải mẫu để hiểu rõ hơn .

3. Dạng 3 : Các bài toán chiếu , và các dạng khác

Bài toán 13 : Dark Street

Đề bài :

Trên một đoạn đường có độ dài là L, được lắp một số ngọn đèn đường tại một số vị trí với chiều cao là h. Trên đường trống một số cây và có độ cao và tán là hoàn toàn xác định và khác nhau. Do tán là của các cay nên có thể một số vị trí không được chiếu sáng. Ví dụ đoạn (A,B) là đoạn không được chiếu sáng . (Hình vẽ) :

A B

Bài toán đặt ra cho chúng ta là hãy tính độ dài không được chiếu sáng khi biết được độ cao của từng cột đèn và cay , tán rộng bao nhiêu .

Dữ Liệu : Vào từ file : Dark.Inp nh sau :

Dòng đầu tiên là số L,N,H và số m (độ dài của đường , số cột đèn ,độ cao của ccs cột đèn và số cây) .

Dòng kế tiếp là N số thể hiện cho vị trí của n coọt đèn với mốc 0 là đầu đường bên trái

Dòng kế tiếp là m bộ 3 số thể hiện cho độ dài của tán lá [Li,Ri] và cây có độ cao là Hi

Kết quả : ghi ra file Dark.Out là tổng độ dài các đoạn không được chiếu sáng trên đoạn đường .

Hạn chế : N,M<=100 .

Bài giải :

Chúng ta sẽ tính các điểm giao có thể có của các đường ánh sáng chiếu từ đèn đến một đầu mút của tán cây (không bị che bởi tán cây khác). Tức là phần ánh sáng có thể cắt trên mặt đường .

A B C D E

Trong ví dụ trên thì các điểm chúng ta sẽ tính đó là : A , B , C , D . Và lấy hai điểm đầu mút (ở ví dụ trên A,E là hai điểm đầu mút , nhưng A cũng là điểm giao). Rồi sau đó chúng ta tính trong từng đoạn một : ví dụ : AB,BC,CD.,DE đoạn nào có bị chiếu sáng không, nếu bị chiếu sáng thì tăng diện tích bị chiếu sáng lên. Sau đó lấy độ dài đoạn đường trừ đi phần chiếu sáng là ra phần không bị chiếu sáng .

ở ví dụ trên : AB : không bị chiếu sáng , BC : được chiếu sáng , CD : không bị chiếu sáng, DE : được chiếu sáng . Như vậy phần không bị chiếu = L-BC-DE .

Bởi vì tính diện tích phần sáng dễ hơn phần tối nên chúng ta làm vậy . Tuy bài toán có dài nhưng đòi hỏi tuy duy sáng tạo trong lập trình chứ không phải trong thuật toán .

Bài toán 14 : Contour

Đề bài :

Cho đường gấp khúc khép kín không tự cắt N cạnh (4< N <= 10000), các cạnh của đường gấp khúc song song với trục toạ độ và có toạ độ đỉnh nguyên, theo giá trị tuyệt đối không vợt quá 30000 . Độ dài của mỗi cạnh lớn hơn 2 * D , trong đó : D là số nguyên dương .

Người ta xây dựng một đường bao ngoài đường cho trước một khoảng là D. Hãy xác định đỉnh của con đường viền đó .

Dữ liệu : vào từ file văn bản CONTOUR.INP gồm n + 1 dòng :
Dòng đầu tiên là số nguyên D
dòng thứ i trong n dòng tiếp theo ghi toạ độ của các đỉnh (được liệt kê một chiều tuỳ ý)
Kết quả : Đưa ra file Contour.Out n dòng, dòng thứ i chứa cặp số nguyên là toạ độ của hình thứ i của đường bao
Ví Dụ :

COUNTOUR.INP	CONTOUR.OUT
1 2 4 2 -3 -4 -3 -4 1 -2 1 -2 4	5 3 -4 -5 4 -5 2 -3 2 -3 5

Bài giải :

Trước hết ta định nghĩa hình viền ngoài của đa giác ban đầu thì ta có mỗi đỉnh viền của nó sẽ tương ứng với mỗi điểm bên trong. Ta cũng đánh số đỉnh viền theo thứ tự đỉnh của đa giác. Gọi đa giác ban đầu sau khi đã đánh lại thứ tự đỉnh là: A1,A2,...An với A1 là đỉnh có toạ độ (x,y) là lớn nhất. Thì ta sẽ có các đỉnh của đa giác viền tương ứng : B1,B2,..Bn. Trong đó ta đã xác định được B1 . với B1(A1.x+d,A1.y+d).
. Chúng ta sẽ giải quyết bài toán nhỏ sau:
" Hãy xác định được Bi+1 khi đã xác định được Bi "
Chúng ta sẽ thấy một số điều thú vị sau :
Bi và Bi+1 sẽ cùng toạ độ x nếu Ai và Ai+1 có cùng toạ độ x, và Bi và Bi+1 cũng có cùng toạ độ y nếu Ai và Ai+1 cũng có cùng toạ độ y. Và chúng sẽ chỉ xẩy ra một và chỉ một trong hai trường hợp này mà thôi .
Nếu Bi và Bi+1 có cùng toạ độ x thì chúng có toạ độ là (Xi,Yi) và (Xi+1,Yi+1). Thì Xi=Xj . và một điều đặc biệt là : Yi+1=Yi+t +[Ai,Ai+1] . với t = 2*d hoặc t=0 hoặc t=-2*d . và [Ai,Ai+1]là độ dài đoạn Ai,Ai+1.
Tương tự cho Bi và Bi+1 có cùng toạ độ y .
Cách xác định t như sau :
ta có các trường hợp sau :
+ Nếu Bi và Ai-1 khác phía nhau bờ là đường [Ai,Ai+1] thì :
Nếu Bi và Ai+2 khác phía nhau bờ là đường [Ai,Ai+1] thì t=2*d
Nếu Bi và Ai+2 cùng phía nhau bờ là đường [Ai,Ai+1] thì t=0
+ Nếu Bi và Ai-1 cùng phía nhau bờ là đường [Ai,Ai+1] thì :
Nếu Bi và Ai+2 cùng phía nhau bờ là đường [Ai,Ai+1] thì t = -2*d
Nếu Bi và Ai+2 khác phía nhau bờ là đường [Ai,Ai+1] thì t =0 .
Các bạn có thể tham khảo lời giải sau để tham khảo :

```
program      countour ;
uses         crt ;
const        fi  =  'CONTOUR.IN1' ;
             fo  =  'CONTOUR.OUT' ;
type         point  =      record
                             x , y   : integer ;
```

```
                    end ;
    arr         =     array[0..10001 ] of point ;
var
  a , b     :     ^ arr ;
  f         :     text ;
  d , n     :     integer ;

  procedure         readfile ;
  var
    i       :     integer ;
    begin
        clrscr ;
        assign ( f , fi ) ;
        reset ( f ) ;
        readln ( f , d ) ;
        n := 0 ;
        while not seekeof ( f ) do
         begin
            inc ( n ) ;
            readln ( f , a ^ [ n ] . x , a ^ [ n ] . y ) ;
         end ;
        a ^ [ n + 1 ] := a ^ [ 1 ] ;
        a ^ [ 0 ] := a ^ [ n ] ;
        close ( f ) ;
    end ;

  procedure         abc(p1,p2 : point; var a , b , c : longint ) ;
  begin
     a := p1 . y - p2 . y ;
     b := p2 . x - p1 . x ;
     c := - ( a * p1 . x + b * p1 . y ) ;
  end ;

  function          khac_phia(p1,p2:point;p3,p4 : point): boolean ;
  var
    a , b , c    :   longint ;
    t1 , t2, t : real ;
    begin
        abc ( p3 , p4 , a , b , c ) ;
        t1 := p1 . x * a + p1 . y * b + c ;
        t2 := p2 . x * a + p2 . y * b + c ;
        t := t1 * t2 ;
        khac_phia := ( t < 0 ) ;
```

```
  end ;

procedure          xuli(i:integer ; p : point ; var p1 : point ) ;
var
  t       :      integer ;
  t1 , t2   :      boolean ;
  begin
      t1 := khac_phia (p,a ^[i - 1 ],a ^ [i],a ^ [ i + 1 ] ) ;
      t2 := khac_phia (p,a ^[ i+ 2 ],a ^ [i], a ^[ i + 1 ] ) ;
      if t1 and t2 then t := 2 * d
      else
      if not t1 and not t2 then t := - 2 * d
      else t := 0 ;
      if a ^ [ i ] . x = a ^ [ i + 1 ] . x then
       begin
           p1 . x := p . x ;
           if a ^ [ i ] . y > a ^ [ i + 1 ] . y then
            p1 . y := p . y - abs(a ^[i] .y - a^[i+1 ] . y + t )
           else
            p1 . y := p . y + abs(a^[i +1].y-a^ [ i ] . y + t ) ;
       end
      else
      if a ^ [ i ] . x < a ^ [ i + 1 ] . x then
       begin
           p1 . y := p . y ;
           p1 . x := p . x + abs(a^[i+1].x- a ^ [ i ] . x + t ) ;
       end
      else
      begin
          p1 . y := p . y ;
          p1 . x := p . x - abs(a^[i].x- a ^[ i + 1 ] . x + t ) ;
      end ;
  end ;

procedure          process ;
var
  min , i , li , dem     :      integer ;
  p , p1                 :      point ;
  g                      :      text ;
  begin
      li := 1 ;
      for i := 2 to n do
       begin
```

```
            if a  ^ [ li ] . x > a ^ [ i ] . x then
             li := i
            else
            if a ^ [ li ] . x = a ^ [ i ] . x then
             if a ^ [ li ] . y < a ^ [ i ] . y then
              li := i ;
          end ;
        dem := 0 ;
        for i := li to n do
         begin
            inc ( dem ) ;
            b ^ [ dem ] . x := a ^ [ i ] . x ;
            b ^ [ dem ] . y := a ^ [ i ] . y ;
         end ;
        for i := 1 to li - 1 do
         begin
            inc ( dem ) ;
            b ^ [ dem ] . x := a ^ [ i ] . x ;
            b ^ [ dem ] . y := a ^ [ i ] . y ;
         end ;
        a ^ := b^ ;
        a ^ [ n + 1 ] := a ^ [ 1 ] ;
        a ^ [ 0 ] := a ^ [ n ] ;
        p . x := a ^ [ 1 ] . x - d ;
        p . y := a ^ [ 1 ] . y + d ;
        writeln ( f , p . x : 7 , p . y : 7 ) ;
        for i := 1 to n - 1 do
         begin
            xuli ( i , p , p1 ) ;
            p := p1 ;
            writeln ( f , p . x : 7 , p . y : 7 ) ;
         end ;
    end ;

  procedure          writefile ;
    begin
        assign ( f , fo ) ;
        rewrite ( f ) ;
        process ;
        close ( f ) ;
    end ;

BEGIN
```

```
    new ( a ) ;
    new ( b ) ;
    readfile ;
    writefile ;
    dispose ( a ) ;
    dispose ( b ) ;
END.
```

Sau đây là một số bài toán khá hay, hoặc là tôi chưa có thuật toán nêu ra cho các bạn hoặc là lời giải là lời giải không tối ưu . Tôi xin nêu ra để các bạn tham khảo, một số bài sẽ có bộ test . Nhưng cũng có một số bài không .

Bài toán 15 : Tìm tập các hình chữ nhật

Đề bài :

Mặt phẳng toạ độ được chia ra thành các hình chữ nhật với các cạnh song song với các trục toạ độ và các đỉnh tại các điểm có toạ độ (M*i ,N*j) , trong đó (i,j) chạy qua tất cả các giá trị nguyên có thể. Ta gọi khoảng cách từ 1 điểm đến một hình chữ nhật là khoảng cách nhỏ nhất trong số các khoảng cách từ P đến các điểm thuộc hình chữ nhật (kể cả các điểm nào nằm trên cạnh của hình chữ nhật). Trong trường hợp riêng, khoảng cách từ P đến hình chữ nhật chứa nó bằng 0 .

Yêu Cầu : Cho trước điểm P với toạ độ (Xp,Yp) và số L , hãy đa ra tất cả các hình chữ nhật mà khoảng cách từ điểm P đến chúng không vợt qúa L. Các hình chữ nhật tìm được đưa theo thứ tự không giảm của khoảng cách từ chúng tới P .

Dữ liệu : Cho từ file văn bản Disrec.Inp chứa các số nguyên M,N,L ,Xp,Yp (0<=M<=10,0<=N<=10,0<=L<=300),-30000<Xp,Yp<30000) được ghi cách nhau bởi một dấu cách hoặc dấu xuống dòng .

Kết quả : Chỉ ghi ra file văn bản : Disrec.Out mỗi dòng chứa toạ độ của đỉnh ở góc trái dưới của một trong số các hình chữ nhật tìm được theo thứ tự đã nêu trong yêu cầu. Các hình chữ nhật cách đều P có thể đa theo thứ tự tuỳ ý.

Ví dụ :

DISREC.INP	DISREC.OUT
3 2 2 4 3	3 2 3 0 0 2 3 4 0 0 0 4 6 2

Bài Toán 16 : Chặt Cây

Đề bài :

Một gia đình nọ có một khu vườn cây và muốn rào khu vườn đó lại. Tuy nhiên, để rào được toàn bộ cây đó thì chúng ta phải chặt một số cây trong

vườn để lấy gỗ rào các cây còn lại. Mỗi cây gỗ sau khi chặt cho một lượng gỗ đủ để rào một đoạn Li .

Hãy tìm cách chặt ít cây nhất để có được một hàng rào bao được tất cả các cây của khu vườn (hàng rào phải là một đa giác bao được tất cả các cây còn lại) .

Dữ liệu Vào từ file văn bản : CAY.INP như sau :

Dòng đầu tiên ghi N (N<=50).

N dòng tiếp , mỗi dòng ghi 3 số (x,y,L) là biểu diễn toạ độ của cây ấy và lợng gỗ thu được khi chặt nó .

Kết quả Ghi ra file CAY.OUT như sau :

Dòng đầu tiên ghi số cây tối thiểu phải chặt

Dòng kế tiếp là các cây cần chặt .

Bài Toán 17: Quản lí các cửa sổ trong môi trường Windows

Đề bài :

Trong môi trường Windows, các ứng dụng được thể hiện trong các vùng hình chữ nhậtg gọi là cửa sổ. Giả sử màn hình được đặt trong một hệ toạ độ mà gốc toạ độ là góc trên bên trái của màn hình. Chiều ngang của màn hình la A, chiều cao của màn hình là B. Các cửa sổ có thể chồng lên nhau, cửa sổ nọ có thể che lấp một phần hay toàn bộ cửa sổ kia trừ trường hợp chúng không giao nhau trên mặt phẳng. Trạng thái mỗi cửa sổ đợc xác định bởi toạ độ của đỉnh trên bên trái, đỉnh dưới bên phải và độ sâu thể hiện thứ tự chồng nhau của cửa sổ. Mỗi cửa sổ có một độ sâu khác nhau chiếm một dải liên tục từ 0 đến M với M là số cửa sổ. Cửa sổ có độ sâu 0 là cửa sổ trên cùng không bị che lấp .

Để quản lí cửa sổ trong hệ điều hành người ta đã xây dựng một số thủ tục như sau :

Thử tục tạo một cửa sổ có dạng “T n x1 y1 x2 y2 “ với n là số hiệu của cửa sổ cho bằng một số tự nhiên do hệ điều hành tự sinh sao cho không có hai cửa sổ trên màn hình có số hiệu trùng nhau , x1,y1 là hoành độ và tung độ của đỉnh trên bên trái cửa sổ còn x2,y2 là hoành độ và tung độ đỉnh dưới bên phải . Thủ tục này sẽ tạo ra một cửa sổ có số hiệu n , có độ sâu 0 và đẩy tất cả các cửa sổ đã có tăng độ sâu thêm một mức .

Thủ tục xoá một cửa sổ trên màn hình có dạng “X n” cho phép xoá (đóng ứng dụng tương ứng) cửa sổ số hiệu n . Độ sâu của các cửa sổ sâu hơn sẽ giảm đi một mức .

Thủ tục di chuyển một cửa sổ đi theo gia số dx.dy có dạng “ D dx dy “ trong đó dx , dy là hai số ,sẽ tịnh tiến cửa sổ theo trục x một khoảng dx và theo trục y một khoảng dy , dx và dy có thể âm . Nếu gia số có giá trị tuyệt đối quá lớn thì cửa sổ có thể nằm ngoài màn hình một phần hoặc toàn bộ và ta sẽ không thấy phần khuất. Việc di chuyển không làm thay đổi độ sâu .

Thủ tục kích hoạt một cửa sổ có dạng "K n" . Khi thực hiện ,cửa sổ số hiệu n sẽ được đặt lên trên cùng, nhận độ sâu là 0 và đẩy các cửa sổ vốn ở trên nó xuống tăng độ sâu hơn một mức .

Ban đầu trên màn hình chưa có cửa sổ nào. Sau khi thực hiện một số thủ tục, trên màn hình có một số cửa sổ. Hãy lập chương trình cho biết mỗi cửa sổ trên màn hình có thể nhìn thấy bao nhiêu phần trăm so với diện tích của nó.

Dữ Liệu : Cho trong file văn bản WInDow.In .

Dòng đầu tiên gồm 3 số A và B tương ứng là chiều ngang và chiều cao của các màn hình và số các lệnh gọi các thủ tục cửa sổ, các số này cách nhau một khoảng trống .

Các dòng tiếp theo, mỗi dòng trong là một lời gọi thủ tục, có dạng như đã mô tả .

Kết quả : Ghi vào file văn bản WinDow.Out gồm nhiều dòng, mỗi dòng tương ứng với một cửa sổ (không kể các cửa sổ đã xoá) sau khi thi thực hiện xong các thủ tục cho trong tệp WinDow.In và gồm hai số , số thứ nhất là sô hiệu cửa sổ, số thứ hai là tỉ lệ phần trăm màn hình thấy lấy đến hai chữ số lẻ. Các dòng xuất hiện theo thứ tự tăng dần của số hiệu cửa sổ .

Ví dụ :

WINDOW.IN	WINDOW.OUT
800 600 9 T 1 120 120 600 300 T 2 300 180 540 480 T 3 420 360 660 540 T 4 200 300 400 500 T 5 300 60 480 480 D 5 -120 180 X 4 T 6 0 360 120 480 K 2	1 58.33 2 100.00 3 66.67 5 58.67 6 100.00

Bài Toán 18 : Prison

(Bài 2 HSGQG 2001-2002 Bảng B)

Đề bài :

Khu vực nhà tù giam dành cho tù binh Al Qaeda ở đảo Guantanamo có dạng một hình chữ nhật được chia thành một lưới M*N ô vuông cạnh 1 đơn vị độ dài (o<M,N<=100). Biên trái và biên dưới của khu vực giam là trục tung và trục hoành của hệ trục toạ độ (xem hình vẽ). Các phòng giam có dạng hình chữ nhật và toạ độ các đỉnh là nguyên. Các ô còn lại là sân. Để canh phòng tù binh trốn trại ngời ta bố trí một số camera ở biên của khu vực nhà giam .

Ta nói một ô được một camera quan sát đầy đủ, nếu từ camera đó có thể nhìn thấy mọi điểm trong ô đó mà không bị một phòng giam nào che lấp .

Yêu cầu : Hãy chỉ ra một phương án lắp một số ít nhất các camera có toạ độ nguyên ở trên biên để mọi ô ở sân đều đợc quan sát đấy đủ bởi ít nhất một camera

Dữ Liệu : Vào từ file Prison.Inp :

Dòng đầu tiên chứa 3 số nguyên dơng M ,N ,C lần lợt là số hàng , số cột của lưới và số phòng giam , 0<C<20 , các số này phân tách nhau bởi các dấu cách

Dòng thứ i trong số C dòng tiếp theo chứa 4 số lần lợt là hoành độ đỉnh dới bên trái , tung độ đỉnh dới bên trái , hoành độ đỉnh trên bên phải , tung độ đỉnh trên bên phải của phòng giam thứ i . Bốn số này cách nhau bởi các dấu cách .

Kết quả : Đưa ra file Prison.Out :

Dòng đầu tiên là số1 hay số 0 tuỳ theo bài toán có nghiệm hay không .

Nếu Bài toán có nghiệm thì dòng tiếp theo sẽ chứa số nguyên S là số camera ít nhất cần đặt . Khi đó S dòng tiếp theo sẽ chứa mỗi dòng 2 số cho toạ độ của các camera trên biên .

Ví Dụ :

Prison.Inp	Prison.Out
10 10 2 2 3 4 8 3 0 7 2	1 2 10 9 0 2

Bài Toán 19 : Chiếu Sáng Trong Công Viên

Đề bài :

Quy ước công viên bao gồm các đoạn thẳng liền nét được nối với nhau. ở phía công viên có N cái đèn được treo ở cùng 1 độ cao T tại các vị trí khác nhau. Mục đích của các đèn là để thắp sáng toàn bộ công viên. 1 điểm trong công viên (nằm trên đờng gấp khúc) được gọi là nhìn thấy từ một chiếc đèn, nếu đoạn thẳng nối điểm đó với chiếc đèn không có điểm chung với các đoạn thẳng khác của công viên .

Yêu cầu : Bạn phải viết một chương trình xác định số lượng đèn ít nhất cần phải bật lên để chiếu sáng toàn bộ công viên (chiếu sáng toàn bộ đường gấp khúc) .

Dữ Liệu : Đọc từ file Light.In có cấu trúc :

Dòng đầu tiên chứa số M là số điểm gãy của đường gấp khúc

Từ dòng thứ hai trở đến dòng thứ M+1 , dòng thứ i+1 chứa hai số nguyên Xi và Hi ,cho ta biết điểm gãy thứ i nằm ở độ cao Hi và hoành độ là Xi . Với mọi 1<=i<=M thì X[i+1]>X[i], 2 điểm liên tiếp tạo thành một đoạn thẳng của đường gấp khúc .

Dòng thứ M+2 chứa 2 số N và T , trong đó N là số đèn còn T là độ cao chung của đèn đó

Dòng cuối cùng gồm N số nguyên B1,B2 ..Bn lần lợt là hoành độ của các đèn (B[i+1]>B[i]).

Kết quả : Ghi ra file văn bản Light.Out nh sau :
Dòng đầu tiên chứa số K là số đèn ít nhất cần phải bật lên
Dòng thứ 2 ghi K số nguyên dương L1,L2,..Lk là số thứ tự các đèn cần bật
Giới Hạn Kĩ Thuật :
1 <=M<=200 , 1<=N<=200 , 1<=Xi , Hi , T<=10000 , T>Hi , X1<B1và Bn<Xm
Các test đều có nghiệm , chỉ đa ra 1 kết quả .
Ví Dụ :

LIGHT.IN	LIGHT.OUT
6 1 1 3 3 4 1 7 1 8 3 11 1 4 5 1 5 6 10	2 1 4

Bài Toán 20 : Chiếu Sáng Khu Nhà
Đề bài :
Cho một khu nhà hình đa giác, có cạnh song song với các trục toạ độ. Các đỉnh toạ độ nguyên. Về ban đêm để canh phòng cẩn mật khu nhà (giữ đồ vật quý), chính vì thế ngời ta phải bố trí các đèn trong khu nhà , sao cho không có một chỗ nào của căn phòng không được chiếu sáng
Yêu Cầu : Hãy bố trí các đèn sao cho thoả mãn yêu cầu mà cần ít đèn nhất .
Dữ Liệu : Cho trong file Chieu.inp :
Dòng đầu tiên ghi số N là số đỉnh của khu nhà
N dòng sau ghi toạ độ các đỉnh đó
Kết Quả : Ghi ra file Chieu.Out :
Dòng đầu tiên ghi số K là số đèn ít nhất cần chiếu
K dòng sau ghi toạ độ các đèn đó .
Ví Dụ :

Chieu.Inp	Chieu.Out
10 1 1 3 1 3 0 5 0 5 2 4 2 4 4 2 4 2 3 1 3	1 3 2

Note :
Khi giải một bài toán chiếu đèn (từ bài 18-> bài 20) chúng ta phải làm tốt các ý chính :
+ Làm tốt bài toán quét (chiếu hình). Tránh sai sót trong khâu cắt nhau ,..dễ dẫn đến các kết quả sai
+ Nên phân thành các thủ tục, để rồi giải quyết từng khâu nhỏ một (để tránh nhầm lẫn) .
Bài toán Chiếu sáng hầu nh không có thuật toán tốt (bài 20 ,bài 21) thì thông thường ta chỉ làm một kết quả tham lam gần đúng là gần như đạt yêu cầu bài toán (nhất trong các kì thi) .

Bài Toán 21 : Counting Triangle
Đề bài :

Một trong những trò chơi giúp trẻ em phát triển trí não là trò chơi đếm tam giác trên một bảng. Trên các ô đợc đặt các đoạn thẳng có độ dài 1 hoặc 2. Các đoạn trong khi đặt có thể cắt nhau

Hình 1 : *những cách cơ bản để đặt các dấu*

Yêu cầu : Bài toán đặt ra là hãy đếm những tam giác được tạo thành từ việc đặt các đoạn . Ví dụ các hình sau sẽ có tam giác là : 2 , 5 , 12 và 0

Dữ Liệu : Vào từ file : CountTri.Inp :

Dòng đầu tiên là số N thể hiện số đoạn

Dòng kế tiếp là N bộ số , mỗi bộ số gồm 2 cặp số (x1,y1)và(x2,y2)thể hiện cho toạ độ của từng đoạn

Kết Quả Ghi ra file CountTri.Out một số duy nhất là số tam giác có

Bài Toán 22 : Trại Hè

(bài 2 - HSGQG 1999-2000 Bảng A)

Đề bài :

Khu đất để tổ chức cắm trại hè có dạng 1 hình chữ nhật được chia thành N*M ô vuông (5<=N,M<=200) , N hàng , M cột. Cần thiết kế vị trí trong lưới để dựng các trại, mỗi trại đều dựng trên nền là một tam giác vuông.Trên cạnh lớn góc vuông dài L đơn vị có một ‘ đoạn cửa ‘ cách đỉnh góc vuông 1 đơn vị và có độ dài là thương số nguyên của phép chia L cho 2 (L div 2) .Tại đỉnh đối diện cạnh có ‘ đoạn cửa ‘ có treo một ngọn đèn. Được biết có K (K<=100) đơn vị đăng kí dựng trại.

Yêu cầu : Tìm cách thiết kế vị trí dựng trại sao cho :

1) Các đỉnh của các nền của các trại phải ở tại các mắt lới không thuộc biên của lưới. Các cạnh góc vuông song song với các cạnh của lưới. Các nền trại từng đôi một không có điểm chung.
2) Có ít nhất một tia sáng của ngọn đèn chiếu qua ‘ đoạn cửa’ đến được một điểm nào đó trên biên của lưới, nghĩa là khi tia sáng đi theo nửa đường thẳng xuất phát từ đỉnh có đặt ngọn đèn đến điểm biên cuả lưới không cắt cạnh của bất cứ nền trại nào ngoại trừ có thể đi qua hai đầu mút của ‘ đoạn cửa’.
3) Số đơn vị đăng ký được phép dựng trại càng nhiều càng tốt.

Dữ liệu : vào từ file văn bản : traihe.inp có cấu trúc :

Dòng đầu ghi 3 số nguyên N,M,K (5<=N,M<=200, 3<=K<=100) tương ứng là độ dài của hai cạnh của khu đất và số lượng đơn vị đăng ký dựng trại.

Trên dòng thứ i trong K dòng tiếp theo ghi cặp số nguyên Xi ,Yi là chiều dài của hai cạnh góc vuông của nền trại thứ i (1<=i<=K, 1<=Yi,Xi<=150).

Kết quả : ghi ra file văn bản : traihe.out theo cấu trúc sau :

Dòng đầu tiên ghi số Kmax là số lượng đơn vị được dựng trại lớn nhất tìm được.

Mỗi dòng trên Kmax dòng tiếp theo ghi số hiệu của một nền trại được dựng tiếp theo là 3 cặp số nguyên theo thứ tự là các cặp toạ độ của đỉnh góc vuông rồi đến hai đỉnh còn lại của nền trại tương ứng. Mắt lưới được đánh số toạ độ từ trên xuống dưới, và từ trái sang phải, bắt đầu từ 0 .

Các dữ liệu số trên một dòng đọc ghi cách nhau ít nhất một dấu trắng .

Bài toán 23 : Picture

Đề bài :

Một số bưu ảnh và các tranh khắc hình chữ nhật có cùng kiểu được dán trên tường. Các cạnh của chúng đều nằm ngang hoặc thẳng đứng. Các hình chữ nhật có thể phủ lên nhau từng phần hoặc toàn bộ. Độ dài của biên của hộp mọi hình chữ nhật đợc gọi là chu vi .

Yêu cầu : Viết chương trình tính chu vi biên các hình

Ví Dụ :

Biên tương ứng là :

Dữ liệu : Vào từ file Picture.In :

Dòng đầu tiên là số lượng các hình chữ nhật có trên tường dán

Trong mỗi dòng tiếp theo ta có thể tìm thấy toạ độ nguyên của đỉnh trái dưới và phải trên của 1 hình chữ nhật .

Kết quả : Ghi ra file Picture.Out :

Dòng duy nhất giá trị của chu vi

Các ràng buộc :

Toạ độ nằm trong khoảng [-10000,10000]

Giá trị kết quả có thể cần đến 32 bit có dấu .

Ví Dụ :

PICTURE.IN PICTURE.OUT

7 -15 0 5 10 -5 8 20 25 15 -4 24 14 0 -6 16 4 2 15 10 22 30 10 36 20 30 0 40 60 228

Hướng dẫn :

Bài toán 24 : Cắt Hình

Đề bài :

Trên hệ toạ độ ĐêCac cho hình đa giác có các cạnh song song với các trục toạ độ. Mỗi cặp cạnh liên kề nhau đều vuông với nhau. Toạ độ các đỉnh đa giác đều là số nguyên. Cắt hình đa giác bằng 1 hình chữ nhật có các cạnh song song với các trục tọa độ.

Yêu cầu : Hãy xác định xem có bao nhiêu phần rời nhau của đa giác sau khi cắt .

Dữ liệu : Vào từ file Okno.Inp :

Dòng đầu tiên chứa x1 , y1 là toạ độ tương ứng cúa góc trái trên , x2 , y2 là toạ độ góc phải dưới của hình chữ nhật

Dòng 2 là N (4<=n<=5000) là số đỉnh của đa giác

Dòng thứ i trong n dòng tiếp theo ghi toạ độ của đỉnh thứ i , các đỉnh liệt kê theo chiều ngược kim đồng hồ, giá trị toạ độ <=10000

Kết quả : Ghi ra file Okno.Out :

Ghi số lượng các phần rời nhau của hình đó .

Hướng Dẫn :

Bài toán 25: Ghép Gỗ

Đề bài :

Chúng ta thấy trong các dụng cụ của gia đình thì các đồ vật thương là các bộ phận nhỏ ghép lại. Chính vì thế các múi ghép giữa các đồ vật, nhất là các vật bằng gỗ thì điều đó là việc thường gặp trong thực tế. Các bạn có thể xem một số mối ghép giữa các miếng gỗ với nhau nh sau :

Hình 1 Hình 2

Hình 3 Hình 4

Múi nối ở hình 1 chúng ta thấy đối với việc khớp giữa hai múi đó là không thể kéo ra được còn đối với ví dụ ở hình 2 thì chúng sẽ dễ dàng bị rơi ra. Chính vì thế việc này đã được các kỹ sư thiết kế máy móc tính đến. Chúng ta biểu diễn các múi nối nh hình 3 và hình 4 tương ứng với các múi ở hình 1 và hình 2. Chúng ta hãy xem các múi nối đó có chắc chắn không.

Dữ Liệu Vào từ File : GHEP.INP là biểu diễn đường gấp khúc ở vị trí ghép nối . Nó được biểu diễn như sau :

Dòng đầu tiên ghi số N Là số điểm biểu diễn đường gấp khúc đó .

N dòng còn lại là ghi toạ độ của các điểm đó trên hệ trục toạ độ đề các (toạ độ nguyên và lớn hơn -1000 , bé hơn 1000).

Dữ Liệu ra File GHEP.OUT Chỉ ghi số 1/0 nếu mối nối đó chắc chắn /hoặc không chắc chắn .

Ví Dụ:

GHEP.INP	GHEP.OUT
7 0 0 1 1 0 2 2 3 3 1 2 0 4 1	1