

### Bài 3. Đường đi nào

Đất nước X đang gặp nguy hiểm, có một con quỷ từ đâu tới phá rối người dân nơi đây. Vốn là một đất nước có tinh thần yêu nước, không phải chờ đợi lâu, một dũng sĩ đã ngay lập tức xách gươm lên và đi đánh quỷ. Dũng sĩ cần đi từ điểm A thẳng tới điểm B rồi tới điểm C (Nơi con quỷ đang phá rối). Nhưng không may, khi đến điểm B, dũng sĩ quên mất đi tới C bằng con đường nào bởi ở B có 3 sự lựa chọn một là rẽ vuông góc sang trái hoặc rẽ vuông góc sang phải hoặc đi thẳng để tới C. Đúng lúc đó có một con đại bàng bay ngang qua và thấy rõ được con quỷ ở hướng nào. Giả sử nếu bạn là con đại bàng và được cho biết trước tọa độ 3 điểm A, B, C, hãy nói cho dũng sĩ nên rẽ trái, phải, hay đi thẳng nhé.

#### Input

Dòng 1, 2, 3 lần lượt chứa tọa độ 3 điểm A, B, C.

Mỗi dòng gồm 2 số nguyên là tọa độ tương ứng. ( $|x|, |y| \leq 10^9$ )

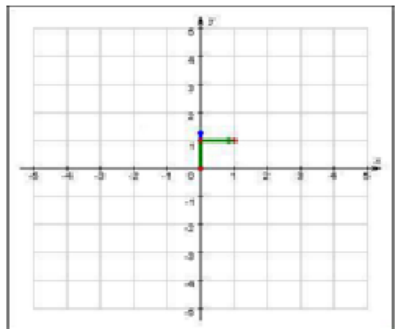
Dòng 1, 2, 3 lần lượt chứa tọa độ 3 điểm A, B, C.

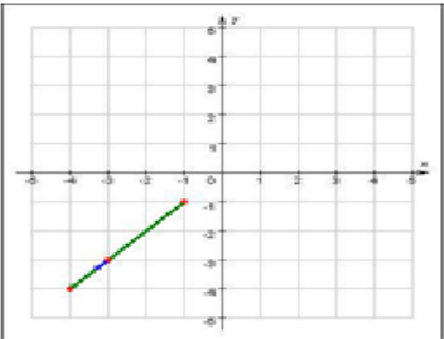
Mỗi dòng gồm 2 số nguyên là tọa độ tương ứng. ( $|x|, |y| \leq 10^9$ )

#### Output

Gồm một dòng duy nhất. In ra “RIGHT” nếu dũng sĩ nên rẽ phải, “LEFT” nếu rẽ trái và “TOWARDS” nếu đi thẳng.

Ví dụ:

Input	Output	
0 0 0 1 1 1	RIGHT	

-1 -1 -3 -3 -4 -4	TOWARDS	
-4 -6 -3 -7 -2 -6	LEFT	

### ***Hướng dẫn thuật toán***

Sử dụng hàm CCW để kiểm tra vị trí tương đối giữa 3 đỉnh. Độ phức tạp  $O(1)$

### **Bài 4. Rào cây**

Người ta muốn rào các cây ở một khu vườn để bảo vệ. Hàng rào được tạo bởi một đường gấp khúc khép kín có đỉnh là một số cây làm cột mốc, sao cho các cây khác phải nằm trong hàng rào (một số cây có thể nằm trên biên)

Hãy xác định một phương án rào cây sao cho số cây phải làm cột mốc là ít nhất.

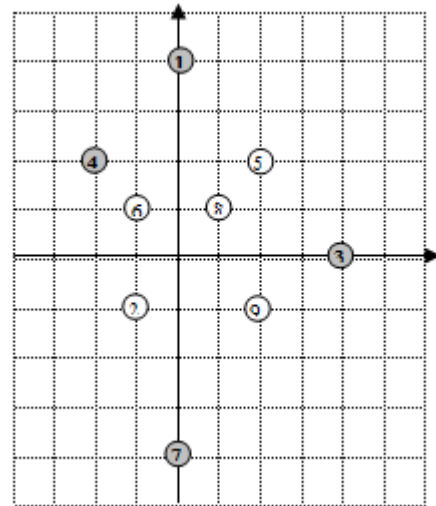
**Input:** Cho bởi file văn bản **RAOCAY.INP**

Dòng đầu số  $n \leq 10000$  là số cây, các dòng tiếp, mỗi dòng là một cặp số nguyên (ghi cách nhau ít nhất một khoảng trắng) mô tả hoành độ và tung độ của một cây. Các cây được đánh số từ 1 theo trình tự xuất hiện trong file.

**Output:** ghi ra file văn bản **RAOCAY.OUT**

Dòng đầu là số cây làm cột mốc, dòng sau là số hiệu các cây này theo đúng thứ tự tạo hàng rào (ghi cách nhau ít nhất một dấu trắng)

Ví dụ: Hình vẽ trên mô tả một khu vườn gồm 9 cây, hàng rào đi qua 4 cây làm cột mốc (màu xám) có số hiệu (theo thứ tự) 7, 3, 1, 4 tương ứng với các file vào, ra dưới đây:



RAOCAY . INP	RAOCAY . OUT
9	4
0 4	7 3 1 4
-1 -1	
4 0	
-2 2	
2 2	
-1 1	
0 -4	
1 1	
2 -1	

### ***Hướng dẫn thuật toán***

- Sử dụng thuật toán Graham để tìm bao lồi trong một tập đỉnh cho trước. Độ phức tạp thuật toán  $O(n \log n)$

### **Bài 5. Bắn máy bay**

Vương quốc C11 sắp chuẩn bị gặp hiểm họa cực lớn từ vương quốc XYZ. Theo thông tin mật báo của điệp viên "Không Không Thấy", rằng XYZ sẽ gửi một Đại đội máy bay B52 ra thả bom tại HN (thủ đô của C11), và quyết đem HN về thời kỳ đồ đá. Và thật may mắn, điệp viên của ta đã rất nhanh tay lấy được sơ đồ chiến thuật tấn công của địch.

Đại đội máy bay bao gồm tổng cộng  $N$  chiếc, được đánh số từ  $1 \rightarrow N$ . Mỗi chiếc sẽ có một vị trí xác định trong đội hình, và được cụ thể bằng một cặp số  $(x, y)$  cho biết tọa độ tương đối của nó. Theo nghiên cứu, cứ mỗi 3 chiếc trong đội hình

tạo thành một hình tam giác sẽ tăng chỉ số chắc chắn của đội hình lên một đơn vị. Và độ an toàn của một chiếc máy bay trong đội hình là số lượng đội hình tam giác mà nó tham gia. Vì thế ta cũng đã đề ra một chiến thuật, mỗi lần ta sẽ bắn rơi chiếc máy bay có độ an toàn nhỏ nhất trong đội hình của chúng.

**Input: - Dòng đầu tiên là số nguyên dương N ( $N \leq 200$ ).**

- N dòng tiếp theo là các cặp số nguyên (x,y) cho biết tọa độ của các chiếc máy bay. ( $|x|, |y| \leq 1000000000$ )

**Output: - Cho biết chỉ số chắc chắn của đội hình lúc đầu và số thứ tự của chiếc máy bay được bắn rơi đầu tiên (nếu có nhiều máy bay cùng độ an toàn thì chọn chiếc có số thứ tự nhỏ nhất).**

Ví dụ:

Input	Output
4	3 2
1 2	
0 0	
1 0	
2 0	

**Hướng dẫn thuật toán:**

- Kiểm tra 3 điểm đôi một khác nhau  $P[i], P[j], P[k]$  có tạo thành một tam giác hay không bằng cách kiểm tra 3 điểm  $P[i], P[j], P[k]$  không thẳng hàng, với mọi  $i, j, k = 1, 2, \dots, n$  ( $i \neq j, j \neq k, i \neq k$ ).

- Để kiểm tra  $P[i], P[j], P[k]$  không thẳng hàng ta dùng hàm  $CCW(p[i], p[j], p[k]) \neq 0$  là được.

- Gọi  $dem[i]$  là số lượng tam giác mà có máy bay thứ  $i$  tham gia vào đội hình tam giác. Kết quả là vị trí  $i$  có  $d[i]$  nhỏ nhất. (nếu có nhiều giá trị  $i$  hiển thị vị trí  $i$  nhỏ nhất).

- Tìm chỉ số chắc chắn của đội hình lúc đầu: Mỗi khi  $P[i], P[j], P[k]$  tạo thành một tam giác và  $i < j < k$  thì chỉ số chắc chắn của đội hình lúc đầu tăng lên 1.

Độ phức tạp thuật toán:  $O(n^3)$

## Bài 6. Mưa thiên thạch

Phú ông nhận được thông tin về một trận mưa thiên thạch sắp ập xuống trái đất. Không những thế, Phú ông còn biết tọa độ của vị trí điểm rơi của mỗi một thiên thạch. Phú ông nhờ Cuội xác định xem có bao nhiêu thiên thạch có thể rơi xuống cánh đồng của ông ta. Cánh đồng của Phú ông có dạng một hình đa giác lồi được xác định bởi danh sách các đỉnh được liệt kê theo thứ tự ngược chiều kim đồng hồ.

**Yêu cầu:** Xác định xem trong tập cho trước các điểm rơi của thiên thạch, có bao nhiêu điểm nằm trong cánh đồng của Phú ông. Các điểm nằm trên biên của cánh đồng không được tính là điểm nằm trong cánh đồng.

### Input:

- Dòng đầu tiên là số nguyên  $n$  ( $3 \leq n \leq 5000$ ) là số đỉnh của đa giác lồi mô tả cánh đồng của Phú ông.
- Mỗi dòng trong  $n$  dòng tiếp theo chứa cặp tọa độ của một đỉnh của đa giác lồi.
- Dòng tiếp theo là số nguyên  $m$  ( $2 \leq m \leq 5000$ ) - số thiên thạch rơi xuống.
- Mỗi dòng trong số  $m$  dòng cuối cùng chứa 2 số là tọa độ điểm rơi của một thiên thạch.

Các tọa độ là các số nguyên có trị tuyệt đối không quá  $10^6$ .

### Output

Ghi ra  $m$  dòng, mỗi dòng tương ứng với 1 điểm rơi của thiên thạch. Ghi "YES" nếu điểm rơi của thiên thạch nằm trong cánh đồng và ghi "NO" nếu trái lại.

Ví dụ:

Input	Output
4	NO
2 4	NO
8 4	YES
6 8	YES
4 6	
4	
3 5	
4 7	

5	5	
6	7	

### **Hướng dẫn thuật toán**

Bài toán kiểm tra điểm nằm trong đa giác

- Sử dụng hàm CCW kết hợp chặt nhị phân
- Giả sử pm là điểm cần kiểm tra thuộc đa giác  $P[1..n]$  hay không?
- Gọi 2 điểm d, c là hai điểm mà có cạnh  $(P[1], P[d])$ ,  $(P[1], P[c])$  thỏa mãn cạnh  $(P[1], pm)$  nằm giữa chúng. Ta sẽ kiểm tra cạnh nối điểm 1 với điểm  $giua = (d+c) \div 2$  nằm cùng phía với cạnh nào. Nếu cùng phía với d thì  $d = giua$ , và ngược lại. Làm như vậy đến khi  $c - d = 1$ . Khi đó ta xét xem diện tích tam giác  $S_{1dc}$  có bằng  $S_{1d(pm)} + S_{1c(pm)} + S_{(pm)dc}$  hay không? Với  $S_{(pm)dc}$  là diện tích tam giác  $Pm, P[d], P[c]$  Chú ý trường hợp điểm Pm nằm trên cạnh đa giác. (nếu Pm nằm trên cạnh đa giác thì hoặc  $S_{1d(pm)} = 0$  hoặc  $S_{1c(pm)} = 0$  hoặc  $S_{(pm)dc} = 0$ )

Độ phức tạp thuật toán:  $O(m \log n)$

### **Bài 7. Khoảng cách**

Trước cửa nhà Mr Bill có một cái hồ rất rộng. Giữa hồ có một hòn đảo nhỏ. Một lần Mr Bill nảy ra ý định bắc một cái cầu từ cửa nhà mình đến đảo giữa hồ để kinh doanh du lịch. Một vấn đề khá hóc búa đối với Mr Bill là làm thế nào xác định được khoảng cách từ nhà mình đến đảo giữa hồ?

Có thể mô tả đảo giữa hồ như là một đa giác lồi còn nhà của Mr Bill như là một điểm nằm ngoài đa giác đó trên mặt phẳng tọa độ. Bạn hãy lập trình giúp Mr Bill tính khoảng cách nhỏ nhất từ nhà mình đến đảo.

**Input:** Vào từ file văn bản **BILL.INP**

- Dòng đầu tiên ghi N là số đỉnh của đa giác ( $N \leq 1000$ )
- Dòng thứ hai ghi tọa độ của điểm được xem như là nhà của Mr Bill
- Tiếp theo là N dòng, mỗi dòng liệt kê tọa độ của một đỉnh của đa giác. Các đỉnh của đa giác được liệt kê ngược theo chiều kim đồng hồ.

**Output:** Ghi ra file **BILL.OUT** một số thực duy nhất là khoảng cách từ nhà của Mr Bill đến đảo giữa hồ (giữ lại 4 chữ số phần thập phân)

Ví dụ:

BILL . INP	BILL . OUT
3	1.4142
0 0	
2 0	
0 2	
2 2	

### Hướng dẫn thuật toán

- Gọi kết quả bài toán là res
- Đầu tiên tính khoảng cách ngắn nhất từ nhà Bill (giả sử đỉnh Q) đến các đỉnh của đa giác,  $res := \min(res, \text{Dist}(Q, p[i]))$ ;
- Tuy nhiên khoảng cách từ đỉnh đa giác tới Q có thể chưa phải là khoảng cách ngắn nhất từ Q đến đa giác, vì thế ta phải tìm khoảng cách ngắn nhất từ Q đến từng cạnh của đa giác. Cụ thể như sau:

Với mỗi cạnh  $(p[i], p[i+1])$ ,  $i = 1, 2, \dots, n$  của đa giác

- Xây dựng đường thẳng (d1) đi qua 2 đỉnh  $(p[i], p[i+1])$  có dạng:  
(d1):  $a_1x + b_1y + c_1$
- Xây dựng đường thẳng (d2) qua Q và vuông góc với (d1) nhận vectơ

$(a_1, b_1)$  làm vectơ chỉ phương có dạng:  $\frac{x - x_M}{a_1} = \frac{y - y_Q}{b_1}$

(d2):  $a_2x + b_2y + c_2$ , với  $a_2 = b_1$ ;  $b_2 = -a_1$ ;  $c_2 = -b_1 \cdot Q.x + a_1 \cdot Q.y$

- Tìm tọa độ giao điểm M của (d1) và (d2)

Xét vị trí tương đối giữa hai đường thẳng

$$D = a_1b_2 - a_2b_1; Dx = b_1c_2 - b_2c_1; Dy = a_1c_2 - a_2c_1$$

Nếu  $D \neq 0$  thì (d1) cắt (d2) tại điểm:  $M.x = Dx/D$ ;  $M.y = -Dy/D$

Nếu  $(D=0)$  và  $(Dx \neq 0)$  thì (d1) song song (d2)

Nếu  $(D=0, Dx=0, Dy=0)$  thì (d1) trùng (d2)

- Nếu M thuộc cạnh  $(p[i], p[i+1])$ , ta tính khoảng cách từ Q đến M và cập nhật kết quả res.

Độ phức tạp thuật toán:  $O(n)$

## Bài 8. BALLGMVN – Đề thi HSG quốc gia năm 2014

Trong một hội thi Ballgame, ban tổ chức chuẩn bị một bàn lớn. Trên mặt bàn có  $n$  bi xanh đánh số từ 1 đến  $n$  và  $n$  bi đỏ đánh số từ  $n + 1$  đến  $2n$ . Mỗi trận đấu, các vận động viên sẽ chơi luân phiên nhau. Đến lượt chơi của mình, Hùng cần tìm 3 bi mà vị trí của chúng là thẳng hàng hanu và sao cho trong số đó có hai bi đỏ và 1 bi xanh (khi đó ăn được một bi đỏ), hoặc là có hai bi xanh và 1 bi đỏ (khi đó được ăn 1 bi xanh).

**Yêu cầu:** Cho biết tọa độ trên mặt phẳng tọa độ Đề-các của vị trí và màu của các bi hiện tại trên bàn, bạn hãy giúp Hùng chọn 3 bi để chơi.

**Dữ liệu:** vào từ file **BALLGMVN.INP**

- Dòng đầu ghi số nguyên dương  $n$ .
- Dòng thứ  $i$  trong số  $n$  dòng tiếp theo ghi hai số nguyên là hoành độ và tung độ trên mặt phẳng tọa độ Đề-các của vị trí đặt bi xanh với chỉ số  $i$ .
- Dòng thứ  $i$  trong số  $n$  dòng cuối cùng ghi hai số nguyên là hoành độ và tung độ trên mặt phẳng tọa độ Đề - các của vị trí đặt bi đỏ với chỉ số  $n + i$ .

Hoành độ và tung độ không vượt quá  $10^6$ , vị trí các bi là đôi một phân biệt.

**Kết quả:** ghi ra tệp **BALLGMVN.OUT**

Ghi ra 3 chỉ số của các viên bi mà Hùng cần chọn, nếu không thể chọn được 3 bi nào, ghi ra -1. Nếu có nhiều đáp án, ghi ra một đáp án bất kì.

**Ví dụ:**

BALLGMVN . INP	BALLGMVN . OUT
3	1 2 4
1 1	
2 2	
4 9	
3 3	
6 20	
8 100	

**Giới hạn:** 30% số test có  $n \leq 2$ ; 30% số test khác có  $n \leq 100$ , 40% số test còn lại có  $n \leq 10^3$ .

**Hướng dẫn thuật toán**

Luân phiên với mỗi điểm  $[i]$  từ 1 đến  $N$ :



Dời gốc tọa độ về điểm[i] nên  $xNew[j]=x[j]-x[i]$  và  $yNew[j]=y[j]-y[i]$  với mỗi  $1 \leq j \leq N$ .

Sắp xếp các điểm theo giá trị  $x/y$ .

Các điểm có cùng giá trị chính là các điểm trên cùng một đường thẳng. Xử lý và in kết quả. Độ phức tạp thuật toán  $O(N^2 \log N)$

Với cách trên các bạn sẽ gặp chút rắc rối ở viên bi màu. vậy các bạn có thể chọn i làm gốc tọa độ của 1 tập, còn tập kia chỉ cần chọn 2 điểm có tỉ số  $x/y$  như hướng dẫn ở trên bằng nhau là được. Như vậy, các bạn có thể chia thành 2 tập viên bi màu xanh và viên bi màu đỏ. và làm tương tự như cách ở trên 2 lần.

***Code tham khảo:***

```
Const fi=' BALLGMVN.INP'; fo=' BALLGMVN.OUT ';
Var f:text;
    X,Y,Res,Vt:array[0..10000] of longint;
    L:array[0..10000] of Real;
    N:int64;
Procedure Doc;
Var i:longint;
Begin
    Assign(f,fi); Reset(f);
    Readln(f,N);
    For i:=1 to 2*N do Readln(f,X[i],Y[i]);
    Close(f);
End;
Procedure QS(d,c:longint);
Var i,j,mid:longint;
    x,tg:real;
Begin
    i:=d; j:=c; x:=L[(i+j) div 2];
    Repeat
        While L[i]<x do Inc(i);
        While L[j]>x do Dec(j);
        If i<=j then
            Begin
                tg:=L[i]; L[i]:=L[j]; L[j]:=tg;
                mid:=Vt[i]; Vt[i]:=Vt[j]; Vt[j]:=mid;
                Inc(i); Dec(j);
            End;
    Until i>j;
    If i<c then QS(i,c);
    If j>d then QS(d,j);
End;
```

```

Procedure Xuly;
Var i,j,d,c,M:longint;
Begin
  For i:=1 to N do
    Begin
      M:=0; d:=0; Res[0]:=i;
      For j:=N+1 to 2*N do
        If Y[i]=Y[j] then
          Begin
            Inc(d);
            Res[d]:=j;
            If d=2 then Exit;
          End
        Else
          Begin
            Inc(M);
            L[M]:=(X[j]-X[i])/(Y[j]-Y[i]);
            Vt[M]:=j;
          End;
      QS(1,M);
      d:=1; c:=1;
      While (d<=M) and (c<=M) do
        Begin
          While (c<M) and (L[c+1]=L[d]) do Inc(c);
          If d<c then
            Begin
              Res[1]:=Vt[d];
              Res[2]:=Vt[c];
              Exit;
            End;
          d:=c+1;
        End;
      End;
  For i:=N+1 to 2*N do
    Begin
      M:=0; d:=0; Res[0]:=i;
      For j:=1 to N do
        If Y[i]=Y[j] then
          Begin
            Inc(d);
            Res[d]:=j;
            If d=2 then Exit;
          End
        Else
          Begin
            Inc(M);
            L[M]:=(X[j]-X[i])/(Y[j]-Y[i]);
            Vt[M]:=j;
          End;
    End;

```

```

        QS(1,M);
        d:=1; c:=1;
        While (d<=M) and (c<=M) do
            Begin
                While (c<M) and (L[c+1]=L[d]) do Inc(c);
                If d<c then
                    Begin
                        Res[1]:=Vt[d];
                        Res[2]:=Vt[c];
                        Exit;
                    End;
                d:=c+1;
            End;
        End;
    End;
    Procedure Ghi;
    Var i,j,d,c,M:longint;
    Begin
        Assign(f,fo); Rewrite(f);
        Xuly;
        If (Res[0]>0) and (Res[1]>0) and (Res[2]>0) then
            Writeln(f,Res[0],' ',Res[1],' ',Res[2])
        Else
            Writeln(f,-1);
        Close(f);
    End;
    BEGIN
        doc;
        ghi;
    END.

```