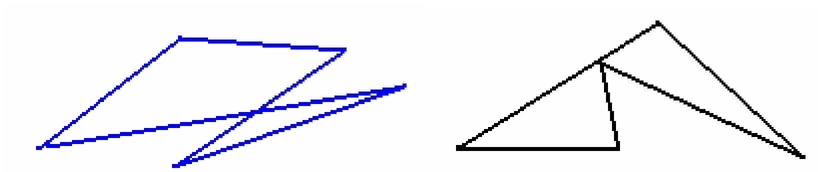
TÍCH NGOÀI CỦA HAI VECTOR - ỨNG DỤNG TRONG TIN HỌC

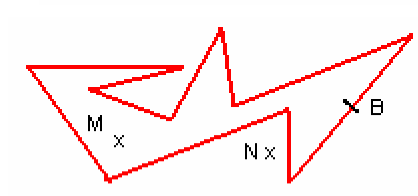
Phần I. Các khái niệm cơ bản

1. ***Đa giác:*** *Là một đường gấp khúc khép kín gồm nhiều đoạn (mỗi đoạn được gọi là một cạnh) sao cho không có hai cạnh liên tiếp nào cùng thuộc một đường thẳng đồng thời không có hai cạnh nào có nhiều hơn một điểm chung.*

2. ***Đa giác tự cắt:*** Là đa giác có ít nhất một cặp 2 cạnh có điểm chung mà là điểm trong của một trong chúng (hình màu xanh và đen).



3. ***Đa giác sao:*** Là đa giác mà không phải là đa giác tự cắt, tức là bất kỳ hai cạnh nào cũng đều không có điểm chung, trừ khi hai cạnh đó là liên tiếp (hình màu đỏ).



4. ***Điểm trong, ngoài, biên:***

   Một đa giác sao ắt sẽ chia mặt phẳng thành 3 phần (liên thông) rời nhau:

   Phần biên: *bao gồm các điểm thuộc các cạnh (kể cả các đỉnh) của đa giác. Điểm thuộc phần biên gọi là điểm biên* (VD: điểm B trong hình).
   Phần ngoài: *gồm các điểm của phần mặt phẳng có thể chứa trọn vẹn một đường thẳng. Điểm thuộc phần ngoài, gọi là điểm ngoài* (VD: điểm N trong hình).
   Phần trong: *là phần còn lại (không thể chứa trọn vẹn đường thẳng nào). Điểm thuộc phần trong, gọi là điểm trong* (VD: điểm M trong hình).

5. ***Góc trong, góc ngoài và số đo của chúng:***
   - *Hai cạnh liên tiếp của đa giác sao cùng xuất phát từ một đỉnh tạo thành 2 góc. Góc có chứa điểm trong, gọi là góc trong. Trái lại, là góc ngoài.*
   - *Với đa giác sao, số đo của các góc trong hay ngoài đều có giá trị trong khoảng $(0, 2\pi)$, tức là $(0^o, 360^o)$.*

6. ***Tích ngoài của hai vector***

   a. *Định nghĩa:*

   Tích ngoài của hai vector $\vec{a} = (x_1, y_1)$ và $\vec{b} = (x_2, y_2)$ là số thực được ký hiệu là $\vec{a} * \vec{b}$ được xác định bởi: $\vec{a} * \vec{b} = \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} = x_1y_2 - x_2y_1$.

b. Các tính chất cơ bản:

TC1 *(tính phản đối xứng):* $\vec{a}*\vec{b} = -(\vec{b}*\vec{a}) = (-\vec{b})*\vec{a} = \vec{b}*(-\vec{a}) = (-\vec{b})*(-\vec{a})$.
TC2 *(tính phân phối với phép cộng vector):* $\vec{a}*(\vec{b}+\vec{c}) = \vec{a}*\vec{b}+\vec{a}*\vec{c}$.
TC3: $(k\vec{a})*(l\vec{b}) = kl(\vec{a}*\vec{b})$
TC4: $\vec{a}*\vec{b} = |\vec{a}||\vec{b}|\sin(\angle(\vec{a},\vec{b}))$, trong đó $\angle(\vec{a},\vec{b})$ là góc định hướng giữa hai vector $\vec{a}$ và $\vec{b}$. *Như vậy, $\vec{a}*\vec{b}$ chính là số đo đại số của diện tích của hình bình hành dựng trên hai vector $\vec{a}$ và $\vec{b}$ đồng thời $|\vec{a}*\vec{b}| = |\vec{a}||\vec{b}|\sin(\vec{a},\vec{b})$ chính là diện tích của hình bình hành đó.*

Hệ quả: *Diện tích của tam giác dựng trên hai vector $\vec{a}$ và $\vec{b}$ là $\frac{1}{2}|\vec{a}||\vec{b}|\sin(\vec{a},\vec{b})$.*

TC5 *(tính định hướng mặt phẳng của tích ngoài):*
- $\vec{a}*\vec{b} = 0 \Leftrightarrow$ *$\vec{a}$ và $\vec{b}$ đồng phương $\Leftrightarrow$ đi thẳng khi đi theo $\vec{a}$ rồi theo $\vec{b}$.*
- $\vec{a}*\vec{b} > 0 \Leftrightarrow$ *góc định hướng giữa $\vec{a}$ và $\vec{b}$ là dương $\Leftrightarrow$ rẽ trái khi đi theo $\vec{a}$ rồi theo $\vec{b}$.*
- $\vec{a}*\vec{b} < 0 \Leftrightarrow$ *góc định hướng giữa $\vec{a}$ và $\vec{b}$ là âm $\Leftrightarrow$ rẽ phải khi đi theo $\vec{a}$ rồi theo $\vec{b}$.*

Phần II. Xác lập các toán tử, hàm và thủ tục cơ bản

Để triển khai thuận lợi các ứng dụng, ta xác lập các yếu tố cơ bản bao gồm cấu trúc dữ liệu và các phép toán với các vector:

1. Kiểu dữ liệu tọa độ của điểm và vector trên mặt phẳng: co2d.
2. Kiểu dữ liệu dãy điểm hay đa giác: polygon.
3. Toán tử cộng + và trừ - hai vector
4. Toán tử tích ngoài * của hai vector
5. Toán tử tích vô hướng / của hai vector

Trên nền tảng đó, có thể xác lập được các hàm quan trọng khác. Cụ thể như sau:

```
Type co2d = record x,y:extended end;
     polygon = array[1..1001] of co2d;
```

Cộng (+) hai vector	**Trừ (-) hai vector**

`Operator +(a,b:co2d)c:co2d;`
`Begin`
` c.x:=a.x+b.x;`
` c.y:=a.y+b.y;`
`End;`
	`Operator -(a,b:co2d)c:co2d;`
	`Begin`
	` c.x:=a.x-b.x;`
	` c.y:=a.y-b.y;`
	`End;`

Tích ngoài (*) hai vector	**Tích vô hướng (/) hai vector**

`Operator *(a,b:co2d)c:extended;`
`Begin`
` c:=a.x*b.y-a.y*b.x;`
`End;`
	`Operator /(a,b:co2d)c:extended;`
	`Begin`
	` c:=a.x*b.x+a.y*b.y;`
	`End;`

Hàm Bang(x,y) trả về True nếu số thực x rất gần y, trái lại thì trả về False

```
Function Bang(x,y: Extended): Boolean;
Begin
 Exit(Abs(x-y)<1E-15);
End;
```

Hàm Len(a) trả về độ dài của vector a

```
Function Len(a:co2d): extended;
Begin
 Exit(sqrt(a/a));
End;
Cũng là:
Function Len(a:co2d): extended;
Begin
 Exit(sqrt(sqr(a.x)+sqr(a.y));
End;
```

Hàm Goc(a,b) trả về số đo radian của góc giữa 2 vector a, b

```
Function Goc(a,b:co2d): extended;
Begin
 Exit(arccos((a/b)/Len(a)/Len(b)));
End;
Cũng là:
Function Goc(a,b:co2d): extended;
Var t: extended;
Begin
 t:=(a/b)/Len(a)/Len(b);
 If Bang(t,0) Then Exit(pi/2) Else
  Begin
   t:=sqrt(1-sqr(t))/t;
   If t<0 Then Exit(pi+arctan(t)) Else Exit(arctan(t));
  End;
End;
```

Hàm GocDH(a,b) trả về số đo radian của góc định hướng tạo bởi vector a và vector b

```
Function GocDH(a,b: co2d):extended;
Begin
  If a*b>0 Then Exit(Goc(a,b)) Else Exit(-Goc(a,b));
end;
```

Phần III. Các ứng dụng điển hình trong tin học

Bài toán 1: Kiểm tra tính thẳng hàng của 3 điểm A,B,C?

Hàm **ThangHang**(A,B,C) sau đây trả về TRUE nếu và chỉ nếu 3 điểm A,B,C thẳng hàng:

```
Function ThangHang(A,B,C: co2d): Boolean;
Begin
 Exit(Bang(A-B)*(A-C),0));
End;
```

Bài toán 2: Kiểm tra tính đồng phương của hai vector a và b?

Hàm **DongPhuong**(a,b) sau đây trả về TRUE nếu và chỉ nếu 2 vector a và b đồng phương:

```
Function DongPhuong(A,B: co2d): Boolean;
Begin
 Exit(Bang(A*B,0));
End;
```

Bài toán 3a: Kiểm tra tính "đi thẳng" nếu đi theo vector a rồi theo tiếp vector b:

Hàm **DiThang**(a,b) sau đây trả về TRUE nếu và chỉ nếu 2 vector a và b cùng hướng:

```
Function DiThang(a,b: co2d): Boolean;
Begin
 Exit(Bang(a*b,0)And(a/b>=0));
End;
```

Bài toán 3a': Kiểm tra tính "đi thẳng" nếu đi từ A đến B rồi tiếp đến C?

Hàm **DiThang**(A,B,C) sau đây trả về TRUE nếu và chỉ nếu là đi thẳng nếu đi từ A đến B rồi đến C:

```
Function DiThang(A,B,C: co2d): Boolean;
Begin
 Exit(DiThang(B-A,C-B));
End;
```

Bài toán 3b: Kiểm tra tính "rẽ trái" nếu đi theo vector a rồi theo tiếp vector b?

Hàm **ReTrai**(a,b) sau đây trả về TRUE nếu và chỉ nếu đi theo vector a rồi rẽ trái theo vector b:

```
Function ReTrai(a,b: co2d): Boolean;
Begin
 Exit(a*b>0);
End;
```

Bài toán 3b': Kiểm tra tính "rẽ trái" nếu đi từ A đến B rồi tiếp đến C?

Hàm **ReTrai**(A,B,C) sau đây trả về TRUE nếu và chỉ nếu là rẽ trái:
Bài toán 3c: Kiểm tra tính "rẽ phải" nếu đi theo vector a rồi theo tiếp vector b?

Hàm **RePhai**(a,b) sau đây trả về TRUE nếu và chỉ nếu đi theo vector a rồi rẽ phải theo vector b:

```
Function RePhai(a,b: co2d): Boolean;
```

```
Begin
 Exit(a*b<0);
End;
```

Bài toán 3c': Kiểm tra tính "rẽ phải" nếu đi từ A đến B rồi đến C?
Hàm **RePhai**(A,B,C) sau đây trả về TRUE nếu và chỉ nếu là rẽ phải:

```
Function RePhai(A,B,C: co2d): Boolean;
Begin
 Exit((B-A)*(C-B)<0);
End;
Cũng là:
Function RePhai(A,B,C: co2d): Boolean;
Begin
 Exit(RePhai(B-A,C-B));
End;
```

Bài toán 4a: Kiểm tra điểm M có thuộc đoạn thẳng AB hay không?

Hàm **DiemThuocDoan**(M,A,B) sau đây trả về TRUE nếu và chỉ nếu M thuộc đoạn AB:

```
Function DiemThuocDoan(M,A,B: co2d): Boolean;
Begin
 Exit(DiThang(M-A,B-M));
End;
```

Bài toán 4a': Kiểm tra điểm M có là điểm trong của đoạn thẳng AB hay không?

Hàm **DiemTrongDoan**(M,A,B) sau đây trả về TRUE nếu và chỉ nếu M là điểm trong của AB:

```
Function DiemTrongDoan(M,A,B: co2d): Boolean;
Begin
 Exit(DiThang(A,M,B)And((M-A)/(M-B)<0));
End;
```

Bài toán 4b: Kiểm tra điểm M có thuộc tia AB hay không?

Hàm **DiemThuocTia**(M,A,B) sau đây trả về TRUE nếu và chỉ nếu M thuộc tia AB:

```
Function DiemThuocTia(M,A,B: co2d): Boolean;
Begin
 Exit(DiThang(B-A,M-A));
End;
```

Bài toán 5: Kiểm tra 2 đoạn thẳng AB và CD có cắt trong nhau hay không?

Hàm **CatNhau**(A,B,C,D) sau đây trả về TRUE nếu và chỉ nếu AB và CD cắt trong nhau:

```
Function CatNhau(A,B,C,D: co2d): Boolean;
Begin
 Exit((((B-A)*(C-A))*((B-A)*(D-A))<0)And
      (((D-C)*(A-C))*((D-C)*(B-C))<0));
```

```
End;
```

Bài toán 6*: Kiểm tra xem dãy điểm $A_1, A_2, ..., A_n$ có lập thành một đa giác không?

Hàm **LaDaGiac**(A, n) sau đây trả về TRUE nếu và chỉ nếu dãy điểm $A_1, A_2, ..., A_n$ lập thành đa giác, tức là không có 3 đỉnh liên tiếp nào thẳng hàng đồng thời không có hai cạnh nào có quá một điểm chung:

```
Function LaDaGiac(A: polygon; n: integer): Boolean;
Var i,j: integer;
Begin
 A[n+1]:=A[1];
 A[n+2]:=A[2];
 For i:=1 To n Do
  If ThangHang(A[i],A[i+1],A[i+2]) Then Exit(False);
 For i:=1 To n-2 Do
  For j:=i+2 To n Do
    If DongPhuong(A[i]-A[i+1],A[j]-A[j+1]) Then
     Begin
      If DiemTrongDoan(A[i],A[j],A[j+1]) Then Exit(False);
      If DiemTrongDoan(A[i+1],A[j],A[j+1]) Then Exit(False);
      If DiemTrongDoan(A[j],A[i],A[i+1]) Then Exit(False);
      If DiemTrongDoan(A[j+1],A[i],A[i+1]) Then Exit(False);
     End;
   End;
 Exit(True);
End;
```

Bài toán 7*: Cho trước đa giác $A_1 A_2 ...A_n$, hãy kiểm tra xem đa giác đó có là đa giác lồi không?

Hàm **DaGiacLoi**(A, n) sau đây trả về TRUE nếu và chỉ nếu đa giác n đỉnh là đa giác lồi:

```
Function DaGiacLoi(A: polygon; n: integer): Boolean;
Var Re: boolean;
Begin
 A[n+1]:=A[1];
 A[n+2]:=A[2];
 Re:=ReTrai(A[1],A[2],A[3]);
 For n:=2 To n Do
  If Re Xor ReTrai(A[n],A[n+1],A[n+2]) Then Exit(False);
 Exit(True);
End;
```

Bài toán 8*: Cho trước đa giác $A_1 A_2 ...A_n$, hãy kiểm tra xem đa giác đó có là đa giac sao không?
Hàm **DaGiacSao**(A, n) sau đây trả về TRUE nếu và chỉ nếu đa giác n đỉnh là đa giác sao:

```
Function DaGiacSao(A: polygon; n: integer): Boolean;
Var i,j: integer;
Begin
 A[n+1]:=A[1];
 A[n+2]:=A[2];
 For i:=1 To n-2 Do
```

```
  For j:=i+2 To n Do
   Begin
    If CatNhau(A[i],A[i+1],A[j],A[j+1]) Then Exit(False);
    If DiemThuocDoan(A[i],A[j],A[j+1]) Then Exit(False);
    If DiemThuocDoan(A[i+1],A[j],A[j+1]) Then Exit(False);
    If DiemThuocDoan(A[j+1],A[i],A[i+1]) Then Exit(False);
   End;
 Exit(True);
End;
```

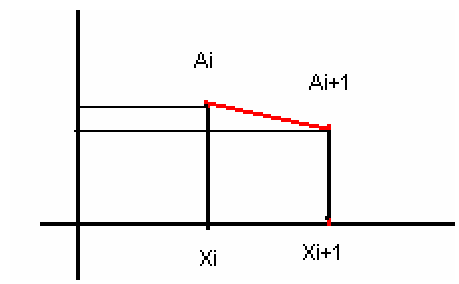
Bài toán 9a*: Cho trước đa giác sao (không tự cắt), hãy tính diện tích của đa giác.

Cách 1: Sử dụng công thức tính diện tích đại số của hình thang có các cạnh đáy song song với trục tọa độ. Với hai điểm $A_i(x_i, y_i)$ và $A_{i+1}(x_{i+1}, y_{i+1})$, hình thang $A_iA_{i+1}X_{i+1}X_i$ có diện tích đại số là: $(x_i - x_{i+1})(y_i + y_{i+1})/2 = (x_i + x_{i+1})(y_i - y_{i+1})/2$. Chú ý: $A_{n+1} \equiv A_1$.

Từ đó, diện tích S của đa giác là:

$$S = \frac{1}{2}\left|\sum_{i=1}^{n}(x_i - x_{i+1})(y_i + y_{i+1})\right| \qquad (1)$$

$$= \frac{1}{2}\left|\sum_{i=1}^{n}(x_i + x_{i+1})(y_i - y_{i+1})\right| \qquad (1')$$

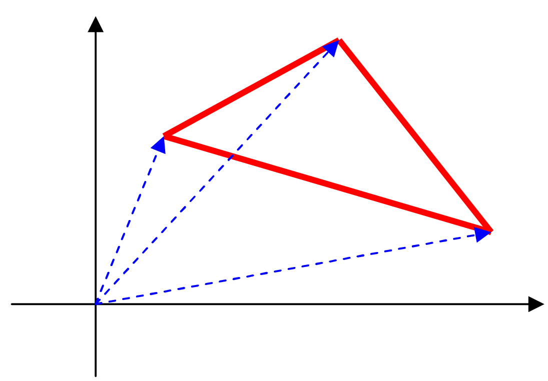


Hàm **DienTich1**(A, n) sau đây trả về diện tích của đa giác sao n đỉnh, tính theo cách 1:

```
Function DienTich(A: polygon; n: integer): Extended;
Begin
 A[n+1]:=A[1];
 DienTich1:=0;
 For n:=1 To n Do
  DienTich1:=DienTich1+(A[n].x-A[n+1].x)*(A[n].y+A[n+1].y);
 DienTich1:=Abs(DienTich1)/2;
End;
```

Cách 2 (rất hay - là phát hiện của tác giả bài viết này): Sử dụng tích ngoài để tính diện tích đại số của tam giác dựng trên 2 vector. Áp dụng cho mỗi tam giác OA_iA_{i+1} (O lấy là gốc tọa độ cho tiện lợi!) dựng tên các vector $\overrightarrow{OA_i}$ và $\overrightarrow{OA_{i+1}}$, ta được:

$$S = \frac{1}{2}\left|\sum_{i=1}^{n}\overrightarrow{OA_i} \otimes \overrightarrow{OA_{i+1}}\right| = \frac{1}{2}\left|\sum_{i=1}^{n}(x_i y_{i+1} - x_{i+1} y_i)\right| \qquad (2)$$



Hàm **DienTich2**(A, n) tính theo cách 2 (rất ngắn gọn):

```
Function DienTich2(A: polygon; n: integer): Extended;
```

```
Begin
 A[n+1]:=A[1];
 DienTich2:=0;
 For n:=1 To n Do DienTich2:=DienTich2+A[n]*A[n+1];
 DienTich2:=Abs(DienTich2)/2;
End;
```

Bài toán 9a': Tính diện tích tam giác ABC?

Xem tam giác ABC như đa giác $A_1A_2A_3$ và tính theo cách 1:

```
Function DTTG1(A,B,C: co2d): Extended;
Begin
 DTTG1:=Abs((A.x-B.x)*(A.y+B.y)+(B.x-C.x)*(B.y+C.y)+(C.x-
A.x)*(C.y+A.y))/2;
End;
```

Xem tam giác ABC như đa giác $A_1A_2A_3$ và tính theo cách 2 (thực chất cũng là cách 1):

```
Function DTTG2(A,B,C: co2d): Extended;
Begin
 DTTG2:=Abs(A.x*B.y-A.y*B.x+B.x*C.y-B.y*C.x+C.x*A.y-C.y*A.x)/2;
End;
```

Tính dựa theo Tính chất 4: $S_{ABC} = \frac{1}{2}\left|\overrightarrow{AB} * \overrightarrow{BC}\right|$:

```
Function DTTG3(A,B,C: co2d): Extended;
Begin
 A.x:=B.x-A.x;A.y:=B.y-A.y;
 B.x:=C.x-B.x;B.y:=C.y-B.y;
 DTTG3:=Abs(A.x*B.y-A.y*B.x)/2;
End;
```

Bài toán 10*: Cho trước đa giác sao, hãy tính số đo radian (thuộc khoảng $(0, 2\pi)$) của các góc trong (tại các đỉnh) của đa giác.

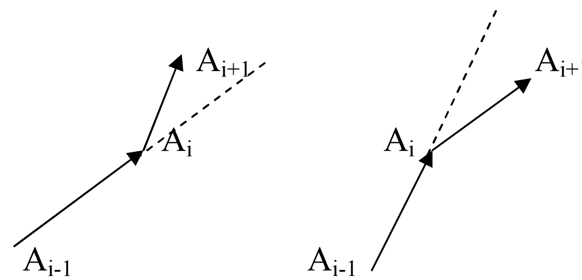
Thuật toán để tính số đo các góc trong (suy ra số đo các góc ngoài):

+ Để tính số đo góc trong (từ đó hiển nhiên suy ra số đo góc ngoài) tại đỉnh A_i, ta có thể dựa vào số đo góc (vô hướng) $\beta = \angle\left(\overrightarrow{A_iA_{i-1}}, \overrightarrow{A_iA_{i+1}}\right)$ tạo bởi các vecto $\overrightarrow{A_iA_{i-1}}$ và $\overrightarrow{A_iA_{i+1}}$ (i = 1,2, ..., n và $A_0 \equiv A_n$). Số đo góc trong tại A_i khi đó sẽ hoặc là β hoặc là $2\pi-\beta$. Tuy nhiên, việc xác định giá trị nào trong hai giá trị này là không đơn giản!

+ Ta chọn một cách tiếp cận khác phù hợp với hướng di chuyển dọc biên đa giác: ta sẽ căn cứ vào số đo số học $\alpha = \angle\left(\overrightarrow{A_{i-1}A_i}, \overrightarrow{A_iA_{i+1}}\right)$ của góc tạo bởi hai vector $\overrightarrow{A_{i-1}A_i}$ và $\overrightarrow{A_iA_{i+1}}$, khi đó số đo (số học) của góc trong tại đỉnh A_i sẽ hoặc là $\pi - \alpha$ hoặc là $\pi + \alpha$.

+ Từ công thức đã biết: $\cos\alpha = \dfrac{\overrightarrow{A_{i-1}A_i}.\overrightarrow{A_iA_{i+1}}}{\left|\overrightarrow{A_{i-1}A_i}\right|\left|\overrightarrow{A_iA_{i+1}}\right|} = t$, suy ra: $\alpha = \begin{cases} \arctan\left(\dfrac{\sqrt{1-t^2}}{t}\right), t > 0 \\ \dfrac{\pi}{2}, t = 0 \\ \pi + \arctan\left(\dfrac{\sqrt{1-t^2}}{t}\right), t < 0 \end{cases}$

+ Để biết chính xác giá trị nào trong hai giá trị π - α , π + α là số đo góc trong, ta còn phải xác định một yếu tố quan trọng (mang tính tổng thể của đa giác): việc di chuyển dọc biên (từ A_{i-1}, đến A_i rồi đến A_{i+1}) là rẽ trái (theo hướng dương-ngược chiều kim đồng hồ) hay rẽ phải (hướng âm-cùng chiều kim đồng hồ). Yếu tố này chỉ được xác định vào thời điểm cuối cùng khi $i = n$ nếu ta bắt đầu với $i = 1$.



+ Để giải quyết vấn đề *hướng* này, ta có thể dựa vào một đặc trưng thú vị (!!!): tổng S các số đo đại số của tất cả các góc định hướng: $S = \sum_{i=1}^{n}\left(\overrightarrow{A_{i-1}A_i}, \overrightarrow{A_iA_{i+1}}\right)$, cụ thể là:

- Số đo đại số $\left(\overrightarrow{A_{i-1}A_i}, \overrightarrow{A_iA_{i+1}}\right)$ của góc định hướng tạo bởi hai vecto $\overrightarrow{A_{i-1}A_i}$ và $\overrightarrow{A_iA_{i+1}}$ là chính $\angle\left(\overrightarrow{A_{i-1}A_i}, \overrightarrow{A_iA_{i+1}}\right)$ nếu (tích ngoài) $\overrightarrow{A_{i-1}A_i} * \overrightarrow{A_iA_{i+1}} \geq 0$ và là $-\angle\left(\overrightarrow{A_{i-1}A_i}, \overrightarrow{A_iA_{i+1}}\right)$ nếu $\overrightarrow{A_{i-1}A_i} * \overrightarrow{A_iA_{i+1}} < 0$.
- Nếu S > 0 thì hướng di chuyển dọc biên là hướng dương (do đó phần trong của đa giác nằm bên trái hướng di chuyển) nên số đo góc trong tương ứng sẽ là π - $\angle\left(\overrightarrow{A_{i-1}A_i}, \overrightarrow{A_iA_{i+1}}\right)$ nếu $\left(\overrightarrow{A_{i-1}A_i}, \overrightarrow{A_iA_{i+1}}\right) > 0$ và sẽ là π + $\angle\left(\overrightarrow{A_{i-1}A_i}, \overrightarrow{A_iA_{i+1}}\right)$ nếu $\left(\overrightarrow{A_{i-1}A_i}, \overrightarrow{A_iA_{i+1}}\right) < 0$, nghĩa là trong mọi trường hợp, số đo góc trong tại A_i sẽ là π - $\left(\overrightarrow{A_{i-1}A_i}, \overrightarrow{A_iA_{i+1}}\right)$.
- Tương tự, nếu S < 0 thì hướng di chuyển dọc biên là hướng âm (phần trong của đa giác nằm bên phải hướng di chuyển) nên số đo góc trong tương ứng sẽ là π - $\angle\left(\overrightarrow{A_{i-1}A_i}, \overrightarrow{A_iA_{i+1}}\right)$ nếu $\left(\overrightarrow{A_{i-1}A_i}, \overrightarrow{A_iA_{i+1}}\right) < 0$ và sẽ là π + $\angle\left(\overrightarrow{A_{i-1}A_i}, \overrightarrow{A_iA_{i+1}}\right)$ nếu $\left(\overrightarrow{A_{i-1}A_i}, \overrightarrow{A_iA_{i+1}}\right) > 0$, nghĩa là trong mọi trường hợp, số đo góc trong tại A_i sẽ là π + $\left(\overrightarrow{A_{i-1}A_i}, \overrightarrow{A_iA_{i+1}}\right)$.
- Trường hợp S = 0 chỉ xảy ra khi đa giác tự cắt hoặc suy biến thành đoạn thẳng (!)

GocTrong[i] sẽ là số đo radian của góc trong tại đỉnh A[i]:

```
Procedure TinhGocTrong(A:polygon;n:integer;Var goctrong:
polygonRad);
Var s: extended;
    i: integer;
```

```
Begin
 A[0]:=A[n];
 For i:=n Downto 1 Do
  Begin
   A[i].x:=A[i].x-A[i-1].x;
   A[i].y:=A[i].y-A[i-1].y;
  End;
 A[n+1]:=A[1];
 s:=0;
 For i:=1 To n Do s:=s+GocDH(A[i],A[i+1]);
 If s>0 Then
  For i:=1 To n Do goctrong[i]:=pi-GocDH(A[i],A[i+1])
 Else
  For i:=1 To n Do goctrong[i]:=pi+GocDH(A[i],A[i+1]);
End;
```

Bài toán 11*: Cho trước đa giác sao và điểm M, hãy cho biết điểm M đó là điểm biên, điểm trong, hay điểm ngoài?

f) Xác định vị trí tương đối của một điểm đối với đa giác sao:

Để kiểm tra điểm M(x, y), có 2 cách tiếp cận tương đương về mặt lý thuyết song độ chính xác khác nhau khi triển khai cài đặt do mức độ sai số tính toán không thể khắc phục tuyệt đối.

+ **Thuật toán 1** (cách 1):

Do dễ dàng biết M có là điểm biên hay không nên ta chỉ cần quan tâm khi M không là điểm biên. Khi đó, ta hãy tính tổng S là tổng số đo đại số của tất cả các góc định hướng: $S = \sum_{i=1}^{n} \measuredangle\left(\overrightarrow{MA_i}, \overrightarrow{MA_{i+1}}\right)$ (Chú ý: tổng này khác tổng S nói trong phần d) ở trên). Có thể thấy rằng S như vậy chỉ có thể nhận một trong 3 giá trị: là 0 (khi và chỉ khi M là điểm ngoài), là 2π hoặc -2π (khi và chỉ khi M là điểm trong).

- Thuật toán này có vẻ như là rất tốt nhưng thực tế thì sai số tính toán khá lớn, thường không chính xác khi điểm M rất gần biên.

+ **Thuật toán 2** (cách 2):

Tính số giao điểm của tia gốc M thích hợp với các cạnh. Tia này phải không được đi qua đỉnh nào của đa giác. Căn cứ vào tính chẵn-lẻ của số giao điểm để đưa ra khẳng định: nếu là lẻ thì M là điểm trong và ngược lại, là chẵn thì M là điểm ngoài.

- Có thể thấy cách này đạt độ chính xác cao hơn nhiều cách trên, kể cả khi M là điểm rất gần biên.

Vấn đề là ở chỗ cách xác định tia gốc M sao cho tia này không đi qua đỉnh nào của đa giác. Sau đây là một đề xuất cách xác định tia, tức là cách xác định điểm thứ hai Z của tia:

- Tìm chỉ số k sao cho cạnh A_kA_{k+1} thỏa mãn điều kiện: đường thẳng A_kA_{k+1} không đi qua M. Có thể là tốt hơn nếu thêm những đòi hỏi như: độ dài cạnh A_kA_{k+1} này là lớn nhất

(hay càng lớn càng tốt), góc giữa các đường thẳng MA_k và A_kA_{k+1} càng lớn (càng gần vuông góc) càng tốt.
- Lần lượt lấy thử Z là trung điểm của đoạn thẳng A_kA_{k+1}, nếu tia MZ chưa đạt yêu cầu thì thay Z bởi trung điểm của MZ và cứ thế. Do đa giác gồm hữu hạn đỉnh nên sau một số không nhiều bước, ắt tìm được Z thỏa mãn.
- Việc đếm số giao điểm từ đây trở nên dễ dàng (sử dụng hàm RayCutInSeg() - Tia cắt trong đoạn thẳng).

III. BÀI TẬP BỔ SUNG

1. Viết chương trình tạo ra những đa giác tự cắt và đa giác sao với số đỉnh cho trước (không quá 100 đỉnh). Xem kết quả tạo được như là dữ liệu cho chương trình giải quyết các bài toán 1 và 2.
2. Viết chương trình tạo ra những đa giác lồi với số đỉnh cho trước (không quá 1000 đỉnh). Xem kết quả tạo được như là dữ liệu cho chương trình giải quyết bài tập 3 dưới đây.
3. Viết chương trình tạo một đa giác nội tiếp, ngoại tiếp một đường tròn với số đỉnh cho trước (không quá 100 đỉnh). Xem kết quả tạo được như là dữ liệu cho chương trình ở bài tập 4 dưới đây.
4. Kiểm tra một đa giác lồi có phải là nội tiếp, ngoại tiếp không? Nếu phải, hãy xác định tâm và bán kính của mỗi đường tròn đó (nếu có).
5. Kiểm tra hai đa giác n ($3 \le n \le 100$) cạnh có đồng dạng với nhau không.

- HẾT-