

# Mục lục

<b>1 Tkz – base</b>	<b>3</b>
1.1 Lệnh tkzInit . . . . .	3
1.2 Lệnh tkzClip . . . . .	3
1.3 Lệnh tkzGrid . . . . .	4
<b>2 Điểm trong tkz– euclide</b>	<b>4</b>
2.1 Định nghĩa điểm bằng lệnh tkzDefPoint và tkzDefPoints . . . . .	4
2.2 Dịch điểm bằng lệnh tkzDefShiftPoint . . . . .	5
2.3 Dịch điểm bằng lệnh tkzDefShiftPointCoord . . . . .	5
2.4 Vẽ điểm bằng lệnh tkzDrawPoint . . . . .	6
2.5 Gán nhãn cho điểm bằng lệnh tkzLabelPoint . . . . .	6
2.6 Chọn kiểu hiển thị điểm bằng lệnh tkzSetUpPoint . . . . .	6
<b>3 Các điểm đặc biệt trong Tkz – euclide</b>	<b>7</b>
3.1 Xác định trung điểm của đoạn thẳng bằng lệnh tkzDefMidPoint . . . . .	7
3.2 Xác định tâm tỉ cự bằng lệnh tkzDefBarycentricPoint . . . . .	7
3.3 Xác định trọng tâm tam giác . . . . .	7
3.4 Xác định tâm đường tròn ngoại tiếp bằng lệnh tkzCircumCenter . . . . .	7
<b>4 Xác định điểm ngẫu nhiên</b>	<b>8</b>
<b>5 Xác định điểm bằng các phép biến hình</b>	<b>9</b>
5.1 Phép tịnh tiến . . . . .	9
5.2 Phép vị tự . . . . .	10
5.3 Phép chiếu vuông góc . . . . .	10
5.4 Phép đối xứng tâm . . . . .	10
5.5 Phép quay . . . . .	11
5.6 Hiển thị cấu trúc của các phép biến hình . . . . .	11
<b>6 Xác định giao điểm của các đường</b>	<b>11</b>
6.1 Xác định giao điểm của 2 đường thẳng . . . . .	11
6.2 Xác định giao điểm của đường thẳng và đường tròn . . . . .	12
6.3 Xác định giao điểm của 2 đường tròn . . . . .	12
<b>7 Xác định đường thẳng</b>	<b>13</b>
7.1 Xác định đường thẳng . . . . .	13
7.2 Vẽ đường thẳng . . . . .	13
7.3 Gán nhãn cho đường thẳng . . . . .	14
7.4 Hiển thị cách vẽ đường thẳng . . . . .	14
7.5 một số ví dụ minh họa . . . . .	14

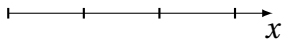
<b>8</b>	<b>Vẽ đoạn thẳng</b>	<b>15</b>
8.1	Vẽ đoạn thẳng bằng lệnh tkzDrawSegment . . . . .	15
8.2	Đánh dấu đoạn thẳng . . . . .	16
8.3	Gán nhãn cho đoạn thẳng . . . . .	16
8.4	Các ví dụ minh họa . . . . .	16
<b>9</b>	<b>Xác định điểm bằng vectơ</b>	<b>16</b>
9.1	Lệnh tkzDefPointWith . . . . .	16
<b>10</b>	<b>Vẽ đa giác</b>	<b>17</b>
10.1	Vẽ tam giác . . . . .	17
10.2	Vẽ đường trung tuyến trong tam giác . . . . .	18
10.3	Vẽ đường cao trong tam giác . . . . .	18
10.4	Vẽ đường phân giác . . . . .	19
10.5	Vẽ hình bình hành . . . . .	19
10.6	Vẽ hình vuông . . . . .	19
10.7	Vẽ hình chữ nhật theo tỷ lệ vàng . . . . .	20
10.8	Vẽ đa giác đi qua các điểm . . . . .	20
<b>11</b>	<b>Vẽ đường tròn</b>	<b>21</b>
11.1	Tô màu hình tròn . . . . .	22
11.2	Cắt phần trong của hình tròn . . . . .	23
11.3	Gán nhãn cho đường tròn . . . . .	23
11.4	Vẽ tiếp tuyến của đường tròn . . . . .	23
<b>12</b>	<b>Vẽ hình quạt</b>	<b>24</b>

# 1 Tkz – base

Các macro TKZ dựa trên chính là:

- `\ tkzInit`
- `\ tkzClip`
- `\ tkzAxeXY`
- `\ tkzAxeX`
- `\ tkzAxeY`
- `\ tkzDrawX`
- `\ tkzDrawY`
- `\ tkzLabelX`
- `\ tkzLabelY`
- `\ tkzGrid`
- `\ tkzRep`

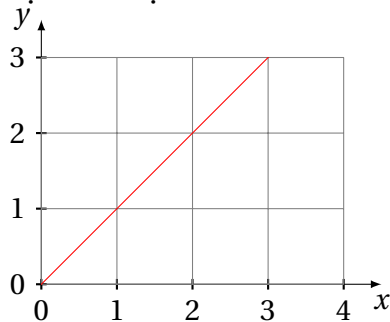
## 1.1 Lệnh `tkzInit`



Lệnh **`tkzInit`** sử dụng để thiết lập khung cho bản vẽ. Các tùy chọn bao gồm `xmax`, `xmin`, `ymax`, `ymin`. Hình vẽ bên là `xmax = 3`, `ymax = 3`.

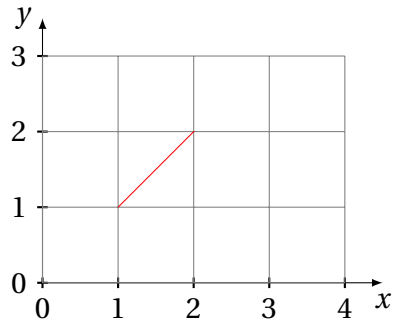
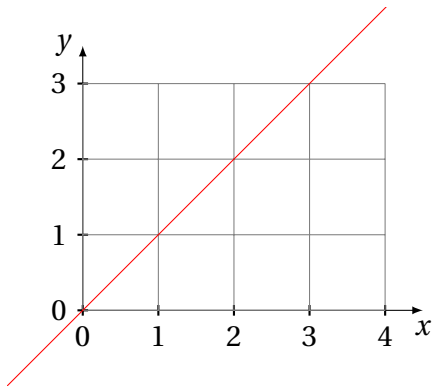
## 1.2 Lệnh `tkzClip`

Lệnh **`tkzClip`** sử dụng để cắt khung hình bản vẽ. Khung được cắt có dạng hình vuông, được quy định bởi lệnh **`tkzInit`**.



### Lệnh `tkzClip` với tùy chọn `space`

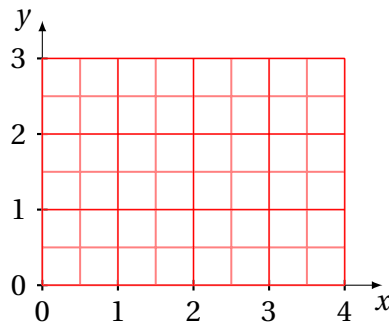
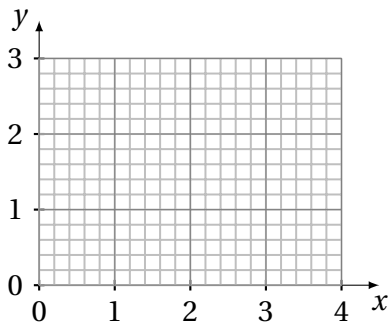
Với tùy chọn **`space`** ta có thể điều chỉnh khung hình vẽ so với khung đã được xác định bởi **`tkzInit`**. Hai hình dưới đây sử dụng các tùy chọn `space = 1` và `space = -1`.



### 1.3 Lệnh tkzGrid

Lệnh **Lệnh tkzGrid** cho phép ta kẻ lưới ô.

- Tùy chọn **sub**: tạo lưới ô nhỏ hơn đơn vị trên trục. Mặc định độ chia là 0.2. Tuy nhiên ta có thể thay đổi bằng tham số **subxstep** và **subystep**.
- Tùy chọn **color**: tô màu lưới ô.



## 2 Điểm trong tkz– euclide

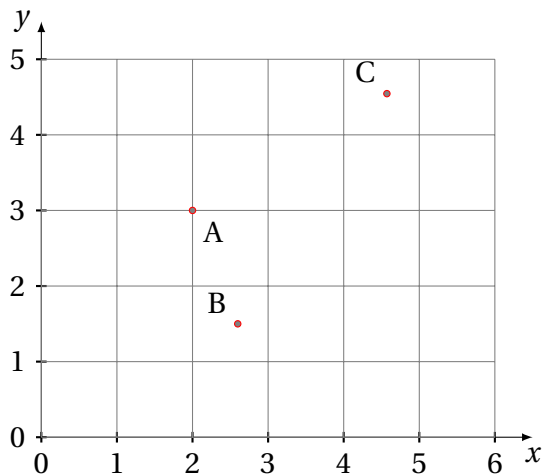
### 2.1 Định nghĩa điểm bằng lệnh tkzDefPoint và tkzDefPoints

Cấu trúc lệnh **tkzDefPoint**

`\tkzDefPoint[tùy chọn](x,y){tên điểm}` hoặc `\tkzDefPoint[option](a:r){tên điểm}`

$(x, y)$  là tọa độ trong hệ Descartes và  $(a : r)$  là tọa độ trong hệ tọa độ cực.

- Tùy chọn **label**: gán nhãn cho điểm. Nhãn của điểm có thể không trùng với tên điểm theo cấu trúc **label=... : tên nhãn**. Trong dấu ... là một số hoặc các tùy chọn *left, right, below, above*. Mặc định: *right*.
- Tùy chọn **shift=(x,y)** cho phép dịch chuyển điểm theo phép tịnh tiến theo vectơ  $(x, y)$ .



Hình vẽ này biểu thị điểm  $A(2,3)$  trên hệ Decartes và  $B(30 : 3)$  trên hệ tọa độ cực và điểm  $C$  là tọa độ điểm  $B$  với tùy chọn  $\text{shift} = \{(2,3)\}$ .

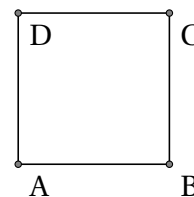
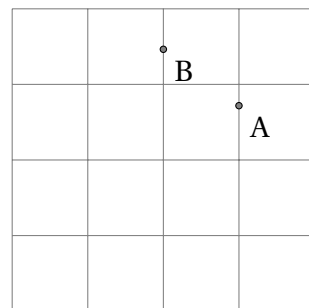
Tọa độ các điểm có thể cho bởi các biểu thức. Chẳng hạn:

### Định nghĩa điểm bằng lệnh `tkzDefpoints`:

lệnh được sử dụng để định nghĩa nhiều điểm một lúc. Cú pháp như sau:

```
\tkzDefPoints[option]{x1/y1/n1,x2/y2/n2, ...}
```

Chẳng hạn, vẽ hình vuông  $ABCD$ .



## 2.2 Dịch điểm bằng lệnh `tkzDefShiftPoint`

### Cú pháp

```
\tkzDefShiftPoint[point](x,y){name} hoặc \tkzDefShiftPoint[option](a:r){name}
```

trong đó  $(x,y)$  là tọa độ trong hệ Decartes,  $(a:r)$  là tọa độ trong hệ tọa độ cực.

## 2.3 Dịch điểm bằng lệnh `tkzDefShiftPointCoord`

Nhằm tạo ra điểm bằng cách dịch theo một vectơ cho trước.

### Cú pháp

```
\tkzDefShiftPointCoord[(a,b)](x,y){name} hoặc (a:r){name}
```

## 2.4 Vẽ điểm bằng lệnh `tkzDrawPoint`

### Cú pháp

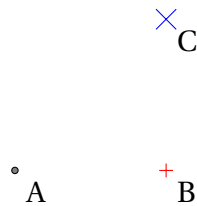
`\tkzDrawPoint[option](name)`

Các thông số của option:

- **shape**: mặc định là **circle**. Có thể là **cross** hoặc **cross out**.
- **size**: mặc định 6.
- **color**: mặc định black.

Vẽ nhiều điểm dùng lệnh **tkzDrawPoints**.

Ví dụ dưới đây thể hiện 3 điểm A, B, C với các tùy chọn **circle**, **cross**, **cross out** với các màu **black**, **red**, **blue**, kích thước 6pt, 12pt, 18pt.



## 2.5 Gán nhãn cho điểm bằng lệnh `tkzLabelPoint`

### Cú pháp

`tkzLabelPoint[option](point){label}`

Để gán nhãn cho nhiều điểm, ta có thể dùng lệnh `tkzLabelPoints` với cú pháp

`\tkzLabelPoints[option](A1,A2,...)`

## 2.6 Chọn kiểu hiển thị điểm bằng lệnh `tkzSetUpPoint`

### Cú pháp

`\tkzSetUpPoint[options]`

Khi dùng lệnh **tkzSetUpPoint** ta có thể chọn kiểu vẽ điểm cho lệnh **tkzDrawpoints**. Các **option** giống với lệnh **tkzDrawpoint**.

### 3 Các điểm đặc biệt trong Tkz – euclide

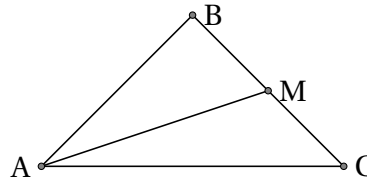
#### 3.1 Xác định trung điểm của đoạn thẳng bằng lệnh tkzDefMidPoint

##### Cú pháp

`\tkzDefMidPoint(point1,point2)`

Để nhận kết quả dùng lệnh **tkzGetPoint**.

Ví dụ sau vẽ tam giác ABC và đường trung tuyến AM.



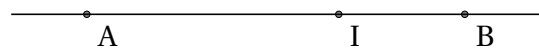
#### 3.2 Xác định tâm tỉ cự bằng lệnh tkzDefBarycentricPoint

##### Cú pháp

`\tkzDefBarycentricPoint(pt1=k1,pt2=k2,...)`

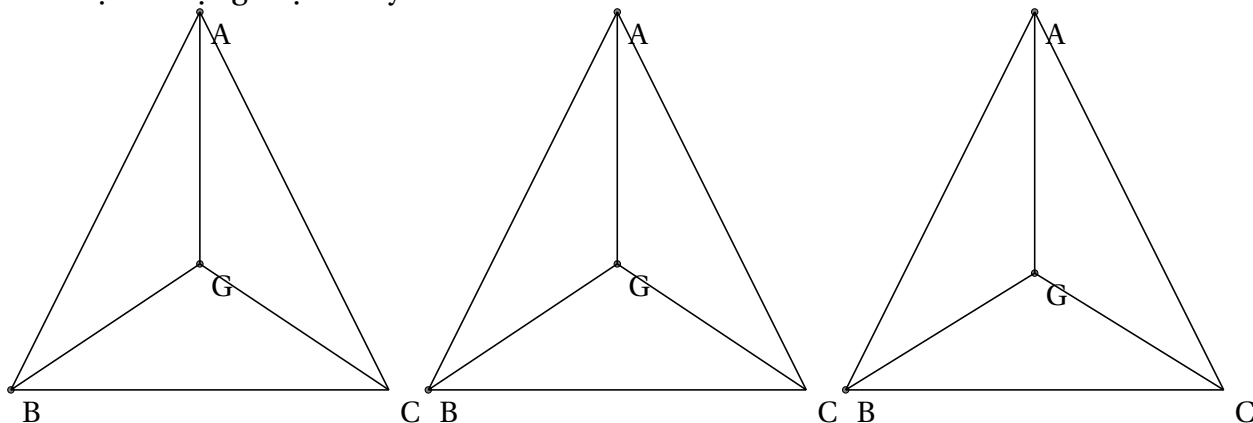
Các hệ số  $k_1, k_2, \dots$  là trọng số của các điểm  $pt_1, pt_2, \dots$

**Ví dụ 1:** Vẽ điểm I với  $\vec{AI} = \frac{2}{3}\vec{AB}$ .



#### 3.3 Xác định trọng tâm tam giác

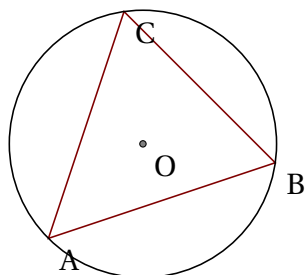
Để xác định trọng tâm tam giác ta có thể dùng lệnh **tkzDefBarycentricPoint** với các hệ số tương ứng 1,1,1. hoặc dùng lệnh **tkzCentroid**, hoặc lệnh **tkzInCenter**. Các hình vẽ dưới đây minh họa cho việc sử dụng 3 lệnh này.



#### 3.4 Xác định tâm đường tròn ngoại tiếp bằng lệnh tkzCircumCenter

##### Cú pháp

`\tkzCircumCenter(pt1,pt2,pt3)`



## 4 Xác định điểm ngẫu nhiên

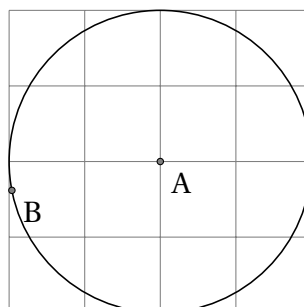
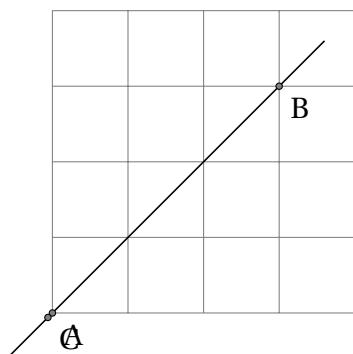
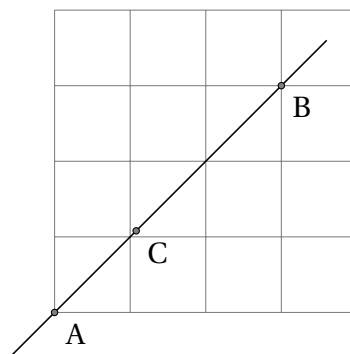
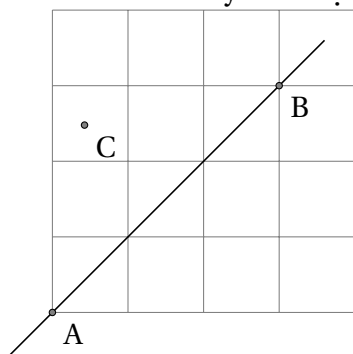
### Cú pháp

`\tkzGetRandPointOn[option]{name}`

Có 4 tùy chọn cho điểm ngẫu nhiên

- **rectangle = #1 and #2**: Điểm ngẫu nhiên trong một hình chữ nhật.
- **segment = #1- -#2**: Điểm ngẫu nhiên trên một đoạn thẳng.
- **line = #1- -#2**: Điểm ngẫu nhiên trên một đường thẳng.
- **circle = center #1 radius #1**: Điểm ngẫu nhiên trên một đường tròn.

Các hình vẽ dưới đây thể hiện lần lượt cho 4 tùy chọn trên.





## 5 Xác định điểm bằng các phép biến hình

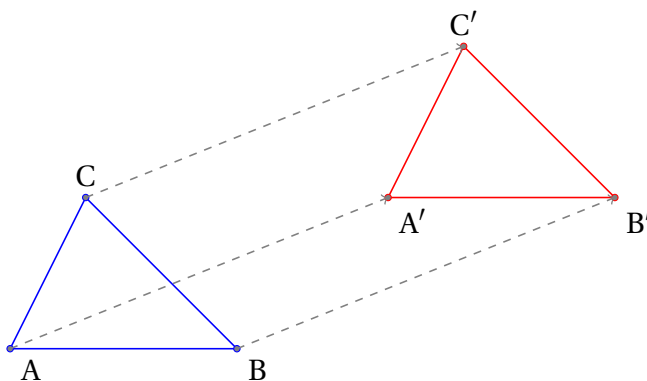
### Cú pháp

`\tkzDefPointBy[option](point)`

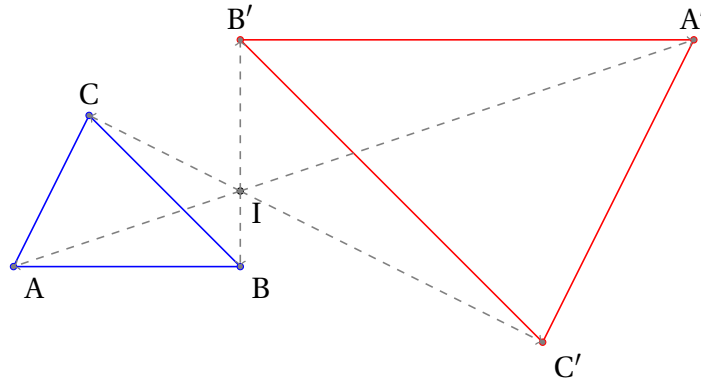
Các tùy chọn như sau:

- **translation = from #1 to #2:** Phép tịnh tiến.
- **homothety = center #1 ratio #2:** Phép vị tự.
- **reflection= over #1 - - #2:** Phép đối xứng qua đường thẳng.
- **symmetry = center #1:** Phép đối xứng tâm.
- **projection= onto #1 - - #2:** Phép chiếu vuông góc trên đường thẳng.
- **rotation = center #1 angle #2:** Phép quay góc đơn vị độ.
- **rotation in rad = center #1 angle #2:** Phép quay góc đơn vị radian.
- **inversion = center #1 through #2:** Phép nghịch đảo.

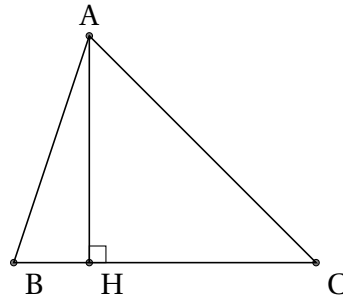
### 5.1 Phép tịnh tiến



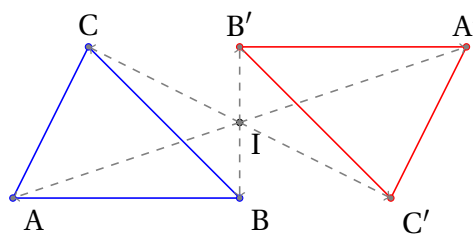
## 5.2 Phép vị tự



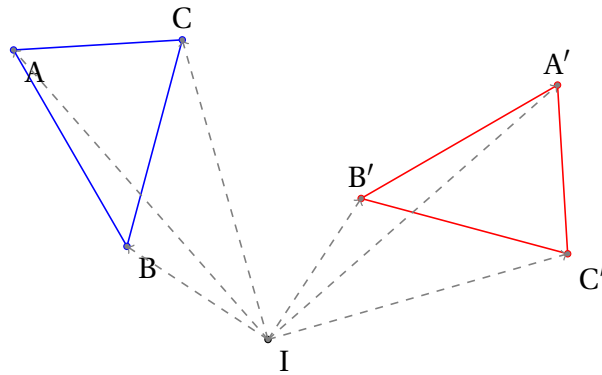
## 5.3 Phép chiếu vuông góc



## 5.4 Phép đối xứng tâm



## 5.5 Phép quay



## 5.6 Hiển thị cấu trúc của các phép biến hình

Lệnh `tkzShowTransformation` chỉ rõ cấu trúc của phép chiếu vuông góc, đối xứng tâm, đối xứng trục giao.

### Cú pháp

`\tkzShowTransformation[option](pt1,pt2)` hoặc `(pt1,pt2,pt3)`

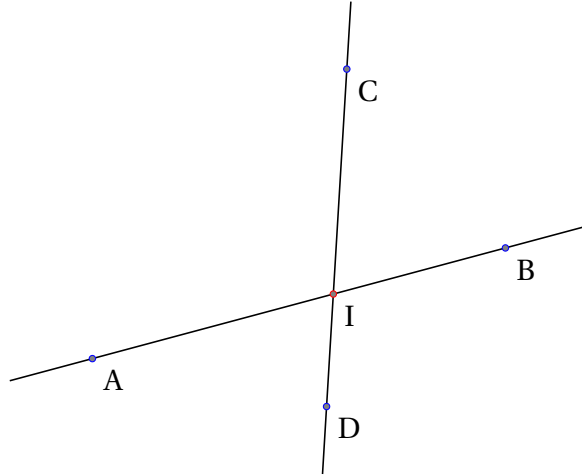
- **reflection= over pt1-pt2**: mặc định reflection.
- **symmetry=center pt** mặc định reflection.
- **projection=onto pt1-pt2** mặc định reflection.
- **translation=from pt1 to pt2** mặc định reflection.
- **K** mặc định 1.
- **length** mặc định 1.
- **ratio** mặc định 0.5
- **gap** mặc định 2.
- **size** mặc định 1.

## 6 Xác định giao điểm của các đường

### 6.1 Xác định giao điểm của 2 đường thẳng

#### Cú pháp

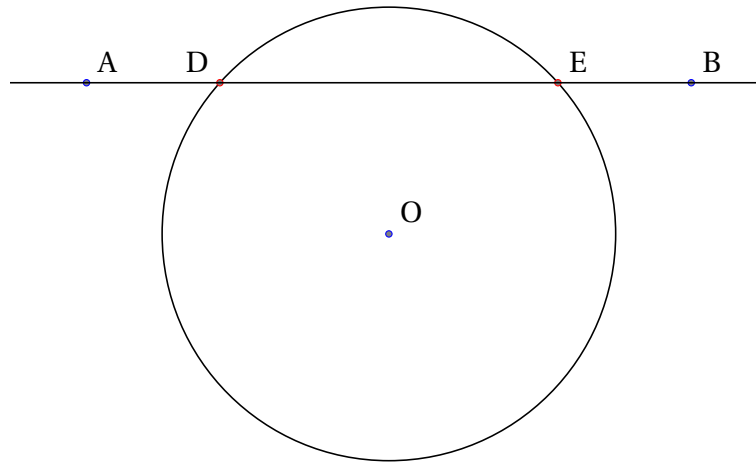
`\tkzInterLL(A,B)(C,D)`



## 6.2 Xác định giao điểm của đường thẳng và đường tròn

### Cú pháp

`\tkzInterLC(A,B)(O,C/r){pt1}{pt2}`



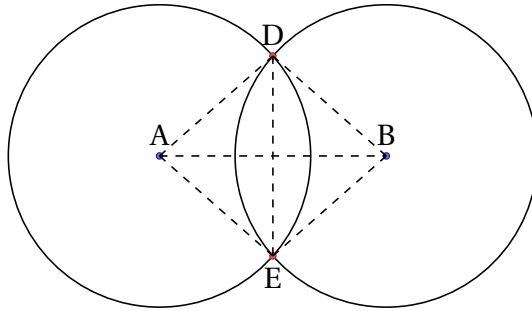
## 6.3 Xác định giao điểm của 2 đường tròn

### Cú pháp

`\tkzInterCC[option](O,A/r)(O',A'/r'){pt1}{pt2}`

Tùy chọn N: đường tròn tâm O, đi qua A

Tùy chọn R: đường tròn tâm O bán kính  $r$ .



## 7 Xác định đường thẳng

### 7.1 Xác định đường thẳng

#### Cú pháp

`\tkzDefLine[option](pt1,pt1)` hoặc `(pt1,pt2,pt)`

Kết quả được lấy từ lệnh **`\tkzGetPoints`** Các tùy chọn:

- **mediator**: vẽ đường trung trực.
- **perpendicular=through...**: vẽ đường vuông góc với đường thẳng cho trước.
- **orthogonal=through...** Vẽ đường trực giao.
- **parallel=through...** Vẽ đường song song.
- **bisector**: Vẽ đường phân giác.
- **bisector out**: Vẽ đường phân giác ngoài.
- **K**: mặc định bằng 1.

### 7.2 Vẽ đường thẳng

#### Cú pháp

`\tkzDrawLine[option](pt1,pt2)` hoặc `\tkzDrawLines[option](pt1,pt2 pt3,pt4 ...)`

Thông số option

- **add #1 and #2**: kéo dài về 2 điểm đầu mút. Mặc định là 0.2.
- **color**: mặc định black.

### 7.3 Gán nhãn cho đường thẳng

#### Cú pháp

`\tkzLabelLine[options](pt1,pt2){name}`

Thông số option: màu sắc, vị trí của nhãn. Chọn vị trí chính xác bằng option **pos=**.

### 7.4 Hiện thị cách vẽ đường thẳng

#### Cú pháp

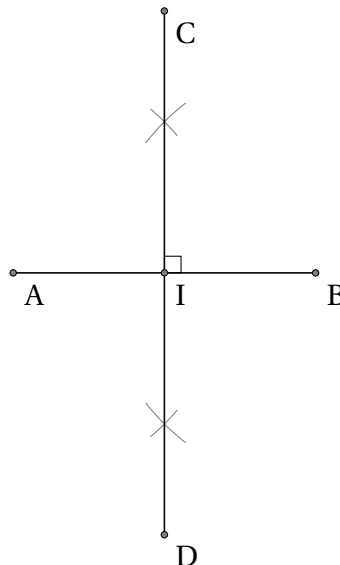
`\tkzShowLine[option](pt1,pt2) ou (pt1,pt2,pt3)`

Các tùy chọn

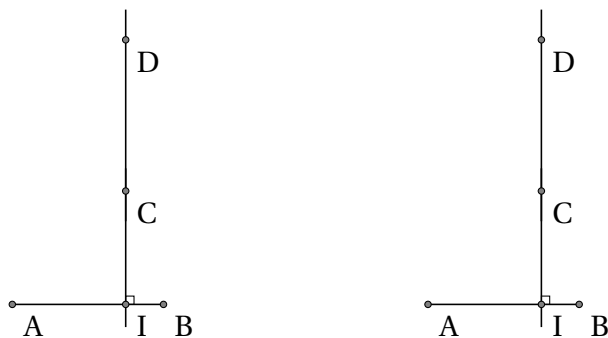
- **mediator**: mediator
- **perpendicular**: mediator
- **orthogonal**:mediator
- **bisector**:mediator
- **K**: 1
- **length**:1
- **ratio**:0.5
- **gap**: 2
- **size**:0.5

### 7.5 một số ví dụ minh họa

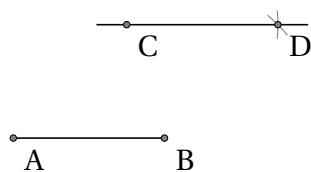
Ví dụ 1. (vẽ trung trực)



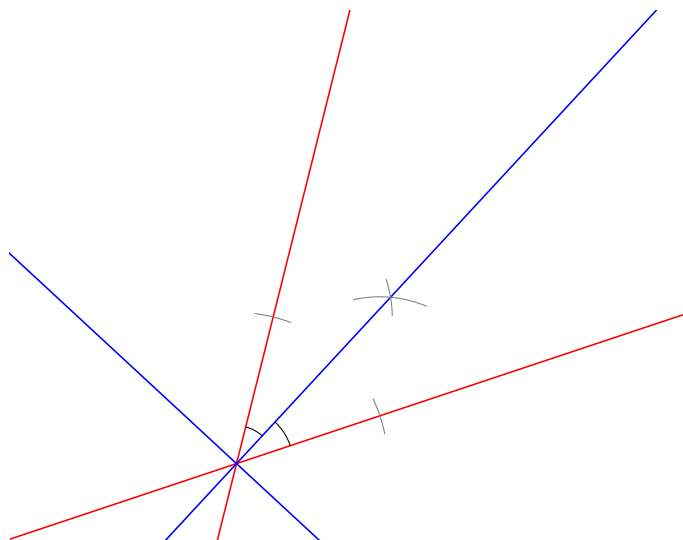
**Ví dụ 2.** (Vẽ đường vuông góc)



**Ví dụ 3.** (Vẽ đường song song)



**Ví dụ 4** (Vẽ đường phân giác trong, ngoài)



## 8 Vẽ đoạn thẳng

### 8.1 Vẽ đoạn thẳng bằng lệnh tkzDrawSegment

Cú pháp

```
\tkzDrawSegment[option] (pt1,pt2)
```

Các tùy chọn gồm màu sắc, kích thước, kiểu dáng. Để vẽ nhiều đoạn thẳng cùng lúc dùng lệnh **tkzDrawSegments**

## 8.2 Đánh dấu đoạn thẳng

### Cú pháp

`\tkzMarkSegment[option](pt1,pt2)`

Các tùy chọn

- **pos**
- **color**
- **mark**
- **size**

Để đánh dấu nhiều đường cùng một kiểu dùng lệnh **tkzMarkSegments**

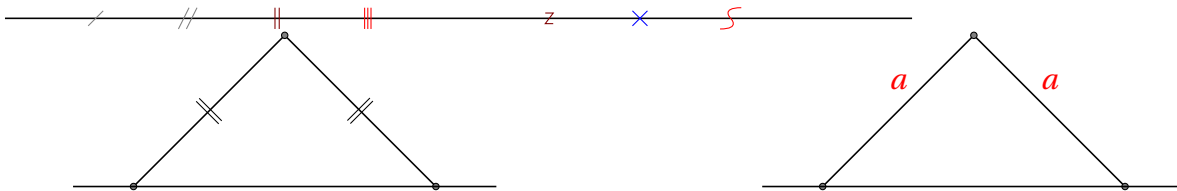
## 8.3 Gán nhãn cho đoạn thẳng

`\tkzLabelSegment[option](pt1,pt2){label name}`

Các tùy chọn gồm: **color**, **pos** (above, below).

Gán nhãn cho nhiều đoạn thẳng dùng lệnh **tkzLabelSegments**

## 8.4 Các ví dụ minh họa



## 9 Xác định điểm bằng vectơ

### 9.1 Lệnh tkzDefPointWith

#### Cú pháp

`\tkzDefPointWith[options](pt1,pt2)`

Kết quả được lấy bằng lệnh **tkzGetPoint** Các tùy chọn

- **orthogonal**  $\vec{AC} \perp \vec{AB}$  và  $AC = AB$ .
- **orthogonal normed**  $\vec{AC} \perp \vec{AB}$  và  $AC = 1$ .
- **linear**  $\vec{AC} = K\vec{AB}$ .



- **linear normed**  $AC = K$  và  $\vec{AC} \parallel \vec{AB}$ .
- **colinear= at #1**  $\vec{CD} = \vec{AB}$ .
- **K**: cho thông số tùy chọn linear.

## 10 Vẽ đa giác

### 10.1 Vẽ tam giác

#### Cú pháp

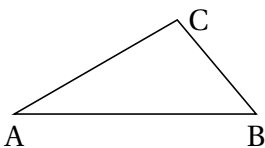
`\tkzDefTriangle[option] (A,B)`

Đỉnh thứ 3 của tam giác được lấy bằng lệnh **tkzGetPoints**.

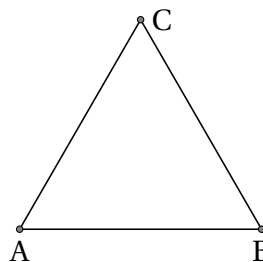
Để vẽ tam giác không cần đỉnh thứ 3, ta dùng lệnh **tkzDrawTriangle** Các tùy chọn

- **two angles= #1 and #2**: Vẽ tam giác có 2 góc cho trước.
- **equilateral**: Vẽ tam giác đều.
- **pythagore**: Vẽ tam giác Pythagore, với bộ ba số Pythagore là 3,4,5.
- **school**: Vẽ tam giác với các góc  $30^\circ, 60^\circ, 90^\circ$ .
- **gold**: vẽ tam giác cân với các góc  $72^\circ, 72^\circ, 36^\circ$ , trong đó, pt1 là đỉnh.
- **euclide**: Tam giác cân với các góc  $72^\circ, 72^\circ, 36^\circ$ , trong đó, AB là đáy.
- **golden**
- **cheops**

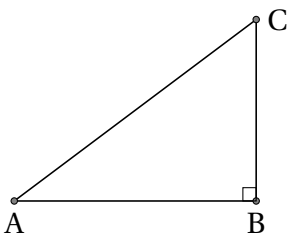
Tùy chọn **two angles= 30 and 50**



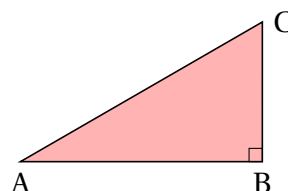
Tùy chọn **equilateral**



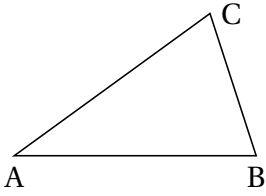
Tùy chọn **pythagore**



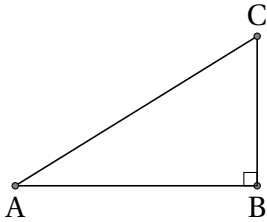
Tùy chọn **school**



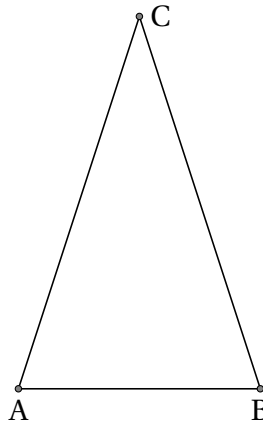
Tùy chọn **gold**



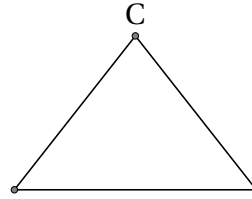
Tùy chọn **golden**



Tùy chọn **euclide**



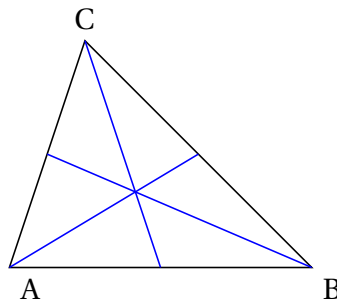
Tùy chọn **cheops**



## 10.2 Vẽ đường trung tuyến trong tam giác

Cú pháp

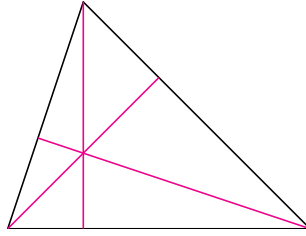
`\tkzDrawMedian[option] (pt1,pt2) (pt3)`



## 10.3 Vẽ đường cao trong tam giác

Cú pháp

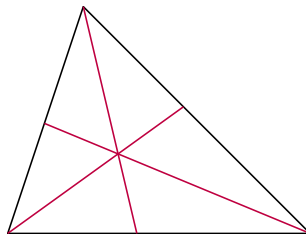
`tkzDrawAltitude[option] (pt1,pt2) (pt3)`



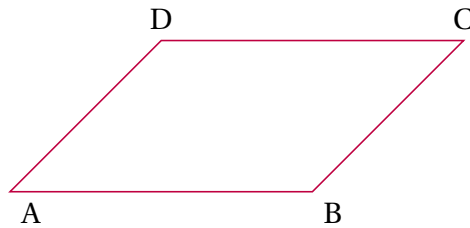
## 10.4 Vẽ đường phân giác

### Cú pháp

```
\tkzDrawBisector[option](pt1,pt2,pt3)
```



## 10.5 Vẽ hình bình hành

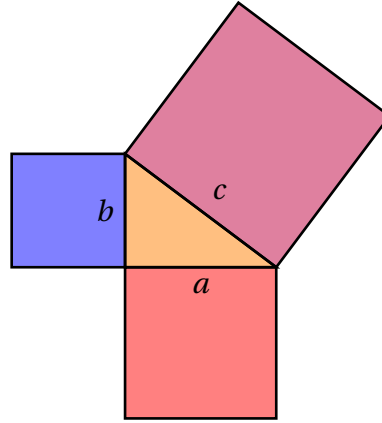
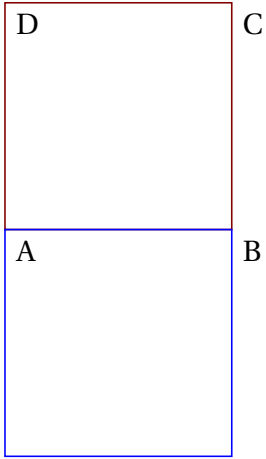


## 10.6 Vẽ hình vuông

### Cú pháp

```
\tkzDefSquare(pt1,pt2)
```

Kết quả các điểm được nhận bởi **tkzGetFirstPoint** và **tkzGetSecondPoint**. Ngoài ra có thể vẽ trực tiếp hình vuông mà không cần xác định các đỉnh còn lại bằng lệnh **tkzDrawSquare**.



## 10.7 Vẽ hình chữ nhật theo tỷ lệ vàng

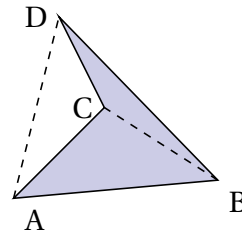
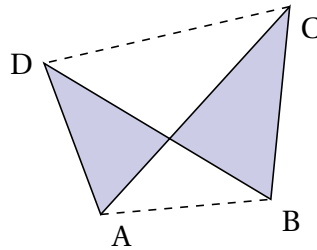
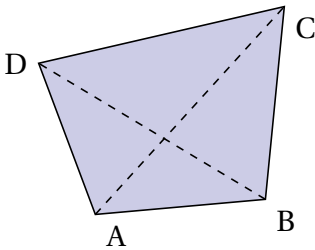
### Cú pháp

`\tkzDefGoldRectangle(pt1,pt2)` hoặc `\tkzDrawGoldRectangle[option](pt1,pt2)`

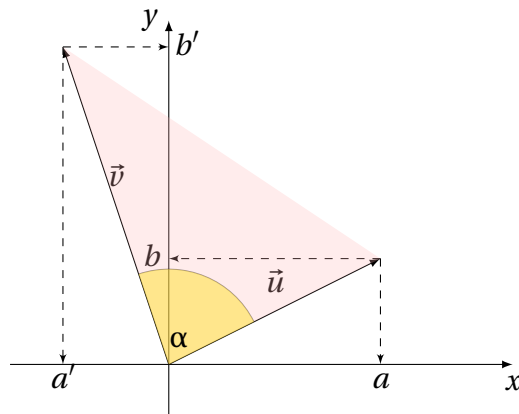
## 10.8 Vẽ đa giác đi qua các điểm

### Vẽ đa giác

`\tkzDrawPolygon[option](tên các điểm)`



Lệnh **`tkzClipPolygon`** cắt phần trong của đa giác.  
Lệnh **`tkzFillPolygon`** tô màu phần trong của đa giác.



## 11 Vẽ đường tròn

`\tkzDefCircle[option](A,B)` hoặc `(A,B,C)`

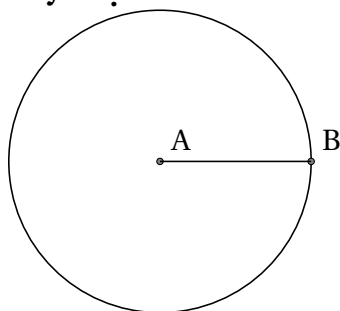
Các kết quả được lấy bằng các lệnh `tkzGetPoint` và `tkzGetLength`, kết quả mặc định được cho bởi `tkzPointResult` và `tkzLengthResult`.

Vẽ trực tiếp đường tròn bằng lệnh `tkzDrawCircle`

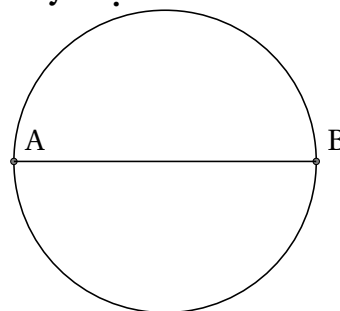
Các tùy chọn

- **radius**
- **diameter**
- **R**
- **circum**
- **in**
- **euler**
- **apollonius**
- **K**
- **orthogonal**
- **orthogonal through**

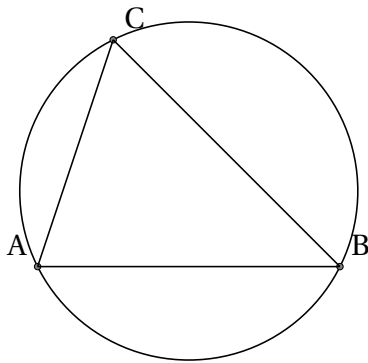
Tùy chọn **radius**



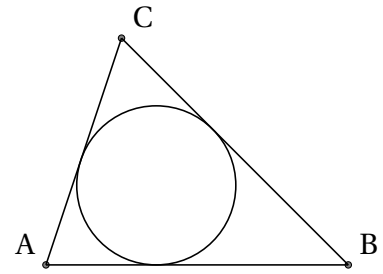
Tùy chọn **diameter**



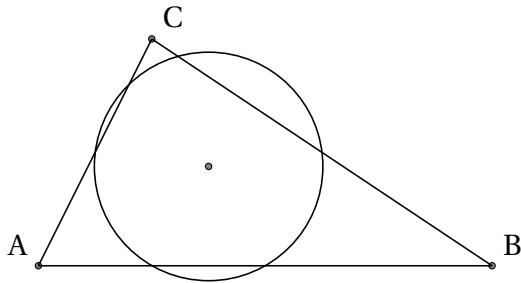
### Tùy chọn circum



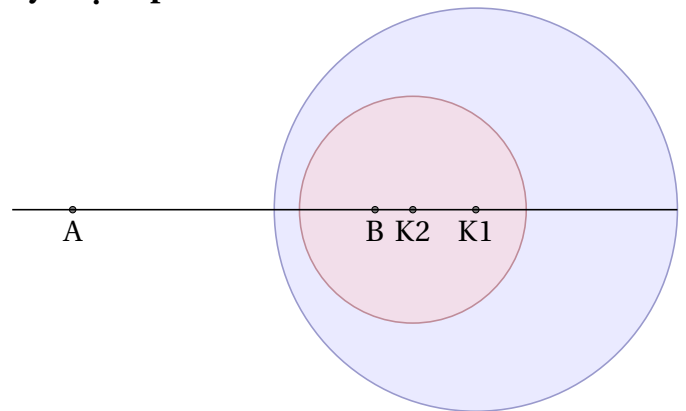
### Tùy chọn in



### Tùy chọn euler



### Tùy chọn apollonius



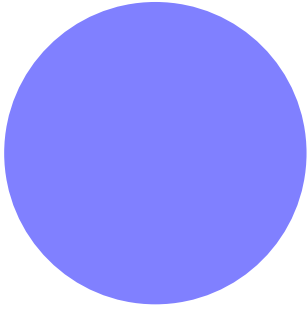
### Tùy chọn orthogonal: wait

## 11.1 Tô màu hình tròn

### Cú pháp

`\tkzFillCircle[option] (A,B)`

Các tùy chọn: **radius,R**

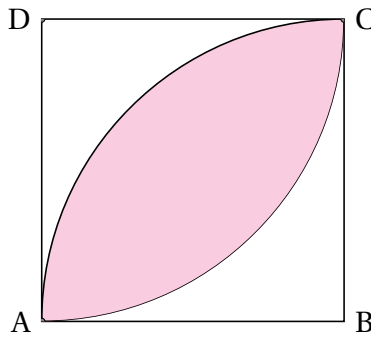


## 11.2 Cắt phần trong của hình tròn

### Cú pháp

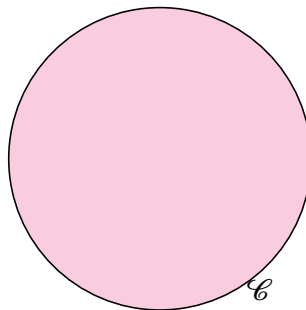
`\tkzClipCircle[option] (A,B)`

Các tùy chọn **radius,R**. Mặc định radius.



## 11.3 Gán nhãn cho đường tròn

`\tkzLabelCircle[option] (A,B) (angle){label}`



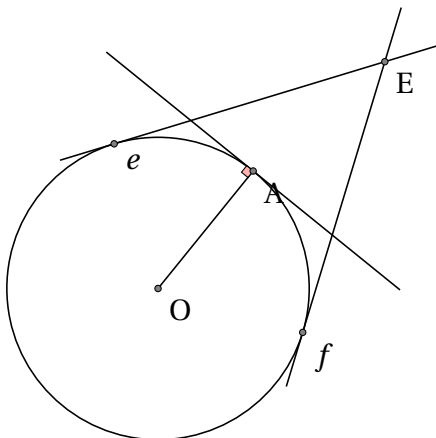
## 11.4 Vẽ tiếp tuyến của đường tròn

### Cú pháp

`\tkzTangent[option] (pt1,pt2) hoặc (pt1,dim)`

Các tùy chọn

- **at=I**: vẽ tiếp tuyến tại I.
- **from=I**: Vẽ tiếp tuyến đi qua I. Nhận tọa độ tiếp điểm bằng lệnh **tkzGetPoints**
- **from with R=I**: vẽ tiếp tuyến đi qua I, đường tròn cho bởi tâm và bán kính



## 12 Vẽ hình quạt

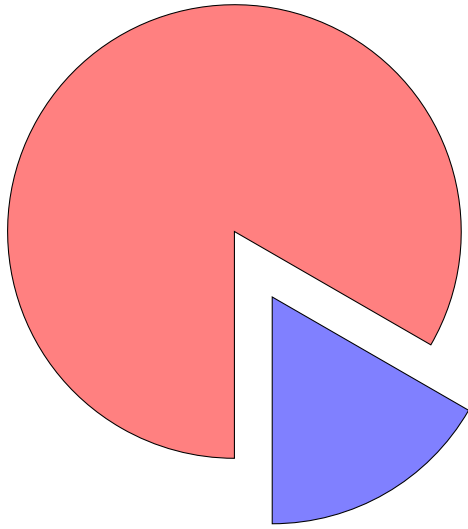
### Cú pháp

`\tkzDrawSector[option](O,...)(...)`

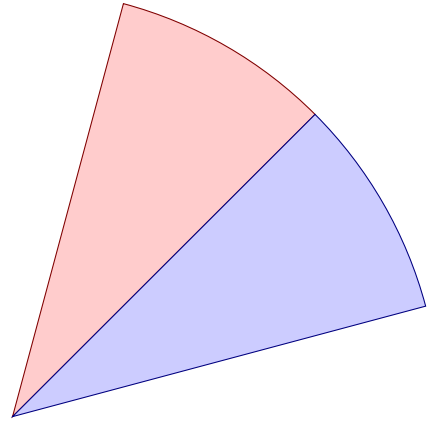
Tô màu bằng lệnh **tkzFillSector**. Tùy chọn mặc định là **towards**.

- **towards**
- **rotate**
- **R**
- **R with nodes**

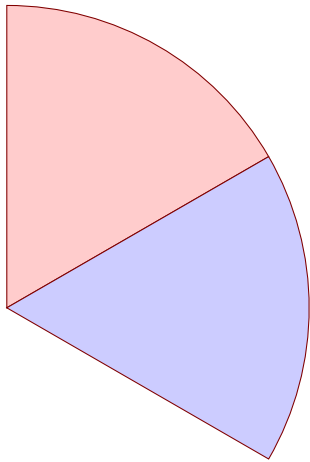




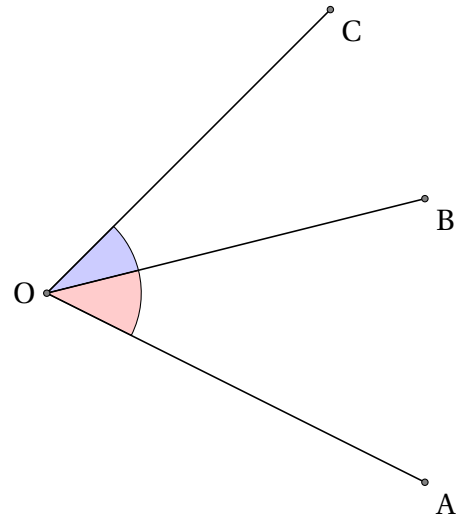
**Tùy chọn toward**



**Tùy chọn rotate**



**Tùy chọn R**



**Tùy chọn R at nodes**