

# PYTHON :

## Étudier l'alignement de trois points dans le plan

# Algorithmique /  
 Programmation  
 # Géométrie  
 # Python



### ÉNONCÉ

Ecrire, en langage Python, un programme permettant d'étudier l'alignement de trois points distincts du plan.

Plusieurs méthodes peuvent permettre d'étudier l'alignement de trois points :

- une méthode qui utilise la géométrie et les vecteurs
- une méthode qui utilise les fonctions affines et les équations de droite

Les deux méthodes nécessiteront de connaître les coordonnées des trois points.

Donc le début du programme récoltera ces informations.

Pour la méthode géométrique, il faut ensuite calculer les coordonnées de deux vecteurs différents formés à partir de ces trois points (ici les vecteurs  $\overrightarrow{AB}$  et  $\overrightarrow{AC}$ ).

Utilisons ensuite la formule de colinéarité issue du produit scalaire de deux vecteurs :

$$x_{\overrightarrow{AB}} \times y_{\overrightarrow{AC}} - x_{\overrightarrow{AC}} \times y_{\overrightarrow{AB}} = 0$$

```
AligFvec.py 001/001
def align(xA, yA, xB, yB
```

```
AligFvec.py 009/011
xAB=xB-xA
yAB=yB-yA
xAC=xC-xA
yAC=yC-yA

if xAB*yAC==xAC*yAB
return "Les 3 poi
```

Remarque :

Il sera possible d'indiquer à l'utilisateur comment utiliser la fonction "align" en ajoutant la ligne suivante en début de programme :

```
print ("align(xA,yA,xB,yB,xC,yC)")
```

Le programme complet :

```

AligFvec.py 001/011
def align(xA,yA,xB,yB,xC,yC):
    xAB=xB-xA
    yAB=yB-yA
    xAC=xC-xA
    yAC=yC-yA
    if xAB*yAC==xAC*yAB:
        return "Les 3 points sont alignes."
    else:
        return "Les 3 points ne sont pas alignes."

```

On peut tester la fonction en allant dans l'onglet {RUN} (on enregistre au passage) et on teste, par exemple les points de coordonnées (1;3), (2;5) et (3;7).

```

MicroPython v1.9.4
|CASIO COMPUTER CO.,
>>>from AligFdr impor
>>>align(1,3,2,5,3,7)
'Les 3 points sont al
>>>

```

Lorsque le texte est trop long pour être affiché sur l'écran, il est possible de se déplacer sur le texte à l'aide de la touche  de la calculatrice pour le lire.

```

MicroPython v1.9.4
|CASIO COMPUTER CO.,
>>>from AligFdr impor
>>>align(1,3,2,5,3,7)
'Les 3 points sont a
>>>

```

```

MicroPython v1.9.4
|CASIO COMPUTER CO.,
>>>from AligFdr impor
>>>align(1,3,2,5,3,7)
'oints sont alignes.'
>>>

```

Pour la méthode analytique, il faut ensuite calculer les coefficients  $a$  et  $b$  de l'équation de la droite passant par deux points en résolvant un système d'équations (ici pour les points  $A$  et  $B$ ) :

$$\begin{cases} y_A = a \times x_A + b \\ y_B = a \times x_B + b \end{cases}$$

Regardons ensuite si les coordonnées du troisième point vérifient cette équation.

```

AligFdr.py 009/009
a=(yA-yB)/(xA-xB)
b=yA-a*xA
if yC==a*xC+b:
    return "Les 3 poi
else:
    return "Les 3 poi

```

Le deuxième fonction align :

```
AligFdr.py 009/009
a=(yA-yB)/(xA-xB)
b=yA-a*xA

if yC==a*xC+b:
    return "Les 3 points sont alignes."
else:
    return "Les 3 points ne sont pas alignes."
FILE RUN SYMBOL CHAR A↔a ▶
```

On peut tester la fonction en allant dans l'onglet **{RUN}** (on enregistre au passage) et on teste, par exemple les points de coordonnées (1 ; 3), (2 ; 5) et (3 ; 7).

De la même manière, on lira le texte entier en utilisant la touche de déplacement .

```
MicroPython v1.9.4
|CASIO COMPUTER CO.,
>>>from AligFdr impor
>>>align(1,3,2,5,3,7)
'Les 3 points sont al
>>>
RUN A↔a CHAR
```

#### Remarques:

- Dans ce deuxième exemple il faudra faire attention au cas où les abscisses de A et B sont égales puisque si c'est le cas, il y aura une division par 0 pour le calcul du coefficient a. On pourra par exemple ajouter une condition du type `if xA==xB` et dans ce cas il suffira de vérifier si `xC` est égal à `xA`, auquel cas les trois points seront alignés sur une droite d'équation `x=xA`.
- Dans les deux cas on fera attention aux cas où les coordonnées sont des flottants puisqu'on utilise un test d'égalité.

Retrouvez toutes nos ressources pédagogiques sur [www.casio-education.fr](http://www.casio-education.fr)