



# Alignement

## Dans les programmes

1. Géométrie – Coordonnées d'un point dans le plan. Droite dans le plan repéré. Établir que trois points sont alignés, non alignés. Colinéarité de vecteurs.
2. Algorithmique – Instruction conditionnelle.

**Entrée :** Trois couples de nombres réels  $(x_A, y_A)$ ,  $(x_B, y_B)$ ,  $(x_C, y_C)$

**début**

$$m_1 \leftarrow \frac{y_A - y_B}{x_A - x_B}$$
$$m_2 \leftarrow \frac{y_A - y_C}{x_A - x_C}$$

**si**  $m_1 = m_2$  **alors**  
| Afficher "oui"

**sinon**  
| Afficher "non"

**fin**

1. Quelle sera la sortie de cet algorithme avec les entrées  $(0; 0)$ ,  $(2; 3)$  ;  $(4; 6)$  ?
2. Quelle sera la sortie de cet algorithme avec les entrées  $(0; 0)$ ,  $(2; 3)$  ;  $(4; 7)$  ?
3. Décrire de façon générale le rôle de cet algorithme.
4. Traduire l'algorithme pour une machine.
5. Avec quel type d'entrées l'algorithme donnera-t-il une erreur ?
6. Compléter l'algorithme pour tenir compte de toutes les entrées possibles. On utilisera des instructions SI pour tester les cas qui posent problème dans la version précédente.
7. Comparer les sorties de l'algorithme ci-dessous avec les sorties obtenues par le programme précédemment complété puis expliquer le fonctionnement de ce nouveau programme.

**Entrée :** Trois couples de nombres réels  $(x_A, y_A)$ ,  $(x_B, y_B)$ ,  $(x_C, y_C)$

**début**

**si**  $(y_B - y_A) \times (x_C - x_A) = (x_B - x_A) \times (y_C - y_A)$  **alors**  
| Afficher "oui"

**sinon**  
| Afficher "non"

**fin**

## Éléments de réponses – XCAS

Ce type de programme très élémentaire contraint l'élève à lire un texte à la syntaxe codifiée. On peut donc certainement le voir comme un bon exercice de préparation à la démonstration.

Commentaires

1. oui
2. non
3. Le programme teste l'alignement de trois points donnés par leurs coordonnées.
4. Sur machine :

### Xcas

```
sontalignes (x_A, y_A, x_B, y_B, x_C, y_C) := {  
  local m_1, m_2;  
  m_1 := (y_B - y_A) / (x_B - x_A);  
  m_2 := (y_A - y_C) / (x_A - x_C);  
  si m_1 == m_2 alors  
  return "les points sont alignés";  
  sinon return "les points ne sont pas alignés"; fsi;  
};
```

5. Dans le cas d'une division par 0, c'est à dire dans le cas où deux points ont même abscisse.
6. Traiter le cas des droites verticales contraint à un travail de logique élémentaire (et, ou, test conditionnel).

**Entrée :** Trois couples de nombres réels  $(x_A, y_A)$ ,  $(x_B, y_B)$ ,  $(x_C, y_C)$

**début**

```
si  $x_A = x_B$  ou  $x_A = x_C$  alors  
  si  $x_A = x_B$  et  $x_A = x_C$  alors  
  | Afficher "points alignés"  
  sinon  
  | Afficher "points non alignés"  
sinon  
   $m_1 \leftarrow \frac{y_A - y_B}{x_A - x_B}$   
   $m_2 \leftarrow \frac{y_A - y_C}{x_A - x_C}$   
  si  $m_1 = m_2$  alors  
  | Afficher "points alignés"  
  sinon  
  | Afficher "points non alignés"
```

**fin**

 **Xcas**

```
sontalignes(x_A,y_A,x_B,y_B,x_C,y_C) := {  
  local m_1,m_2;  
  // cas d'une droite "verticale" :  
  si x_A==x_B or x_A==x_C alors  
    si x_A==x_C and x_A==x_B alors  
      return "les points sont alignés";  
    sinon return "les points ne sont pas alignés";  
  fsi;  
  fsi;  
  // cas des abscisses toutes distinctes :  
  m_1:=(y_B-y_A)/(x_B-x_A);m_2:=(y_A-y_C)/(x_A-x_C);  
  si m_1==m_2 alors  
    return "les points sont alignés";  
  sinon return "les points ne sont pas alignés"; fsi;  
};;
```

7. Il s'agit ici de résumer les différents cas à l'aide d'une proportionnalité et de faire le lien avec la colinéarité de vecteurs.

 **Xcas**

```
sontalignes2(x_A,y_A,x_B,y_B,x_C,y_C) := {  
  si (y_B-y_A)*(x_C-x_A)==(x_B-x_A)*(y_C-y_A) alors  
    return "les points sont alignés";  
  sinon return "les points ne sont pas alignés"; fsi;  
};;
```