



Triangles rectangles

Dans les programmes

1. Géométrie : coordonnées d'un point du plan, distance de deux points du plan.
2. Fonctions : exemple de fonction à deux variables (fonction maximum).
3. Algorithmique : tests conditionnels.

On considère l'algorithme suivant :

Entrée : Trois couples de nombres réels (x_A, y_A) , (x_B, y_B) , (x_C, y_C) , coordonnées de points non alignés

début

$$c \leftarrow (x_B - x_A)^2 + (y_B - y_A)^2$$

$$a \leftarrow (x_B - x_C)^2 + (y_B - y_C)^2$$

$$b \leftarrow (x_C - x_A)^2 + (y_C - y_A)^2$$

si $a + b + c = 2 \times \max(a, b, c)$ **alors**

 | Afficher "oui"

sinon

 | Afficher "non"

fin

1. Quelle sera la sortie de cet algorithme avec les entrées (1; 1), (3;2), (0;3) ?
2. Quelle sera la sortie de cet algorithme avec les entrées (1; 1), (3;2), (0;4) ?
3. Quel est le rôle de ce programme ?
4. Réécrire le programme pour qu'il affiche le sommet de l'angle droit lorsque le triangle est rectangle.
5. On entre le programme de la façon suivante dans une calculatrice Ti (ti 82, ti 84...) :



Program : pyt	nom du programme
Disp "point P" Input A Input B	abscisse du point P ordonnée du point P
Disp "point Q" Input C Input D	abscisse du point Q ordonnée du point Q
Disp "point R" Input E Input F	abscisse du point R ordonnée du point R
$(C - A)^2 + (D - B)^2 \rightarrow G$ $(C - E)^2 + (D - F)^2 \rightarrow H$ $(E - A)^2 + (F - B)^2 \rightarrow I$	$G = PQ^2$ $H = QR^2$ $I = RP^2$
If G+H=I Then Disp "Rectangle en Q" Else If I+H=G Then Disp "Rectangle en R" Else If G+I=H Then Disp "Rectangle en P" Else Disp "non rectangle" end end end	

- (a) Entrez ce programme dans votre calculatrice. Testez le sur quelques exemples.
- (b) On donne en entrée les points $P(0;0)$, $Q(5;0)$ et $R(10^{-15};5)$. Quelle est la sortie? Commentez.

Éléments de réponses – XCAS

Xcas

```
estrectangle3 (x_A, y_A, x_B, y_B, x_C, y_C) := {  
  local AB2, AC2, CB2;  
  AB2 := (x_B - x_A) ^ 2 + (y_B - y_A) ^ 2;  
  AC2 := (x_C - x_A) ^ 2 + (y_C - y_A) ^ 2;  
  CB2 := (x_B - x_C) ^ 2 + (y_B - y_C) ^ 2;  
  si AB2 + AC2 + CB2 == 2 * max(AB2, max(AC2, CB2)) alors  
  return "rectangle";  
  sinon return "non rectangle"; fsi;  
};;
```

ou encore (syntaxe légèrement différente) :

Xcas

```
estrectangle4 (x_A, y_A, x_B, y_B, x_C, y_C) := {  
  local AB2, AC2, CB2;  
  AB2 := (x_B - x_A) ^ 2 + (y_B - y_A) ^ 2;  
  AC2 := (x_C - x_A) ^ 2 + (y_C - y_A) ^ 2;  
  CB2 := (x_B - x_C) ^ 2 + (y_B - y_C) ^ 2;  
  if (AB2 + AC2 + CB2 == 2 * max(AB2, max(AC2, CB2))) {return "rectangle";} else {return "  
  non rectangle "};  
};;
```

On peut aussi définir une fonction "maximum de trois nombres" :

Xcas

```
maxi (a, b, c) := {  
  local tmp;  
  tmp := a;  
  si b > tmp alors tmp := b; fsi;  
  si c > tmp alors tmp := c; fsi;  
  return tmp;  
};;
```

```
estrectangle3 (x_A, y_A, x_B, y_B, x_C, y_C) := {  
  local AB2, AC2, CB2;  
  AB2 := (x_B - x_A) ^ 2 + (y_B - y_A) ^ 2;  
  AC2 := (x_C - x_A) ^ 2 + (y_C - y_A) ^ 2;  
  CB2 := (x_B - x_C) ^ 2 + (y_B - y_C) ^ 2;  
  si AB2 + AC2 + CB2 == 2 * maxi (AB2, AC2, CB2) alors  
  return "rectangle";  
  sinon return "non rectangle"; fsi;  
};;
```



1. oui
2. non
3. Reconnaître si un triangle est rectangle (en "oubliant" les cas particuliers tels que l'entrée (0;0), (0;0), (1;1)).
4. Algorithme.

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

début

$c \leftarrow (x_B - x_A)^2 + (y_B - y_A)^2$

$a \leftarrow (x_B - x_C)^2 + (y_B - y_C)^2$

$b \leftarrow (x_C - x_A)^2 + (y_C - y_A)^2$

si $a + b = c$ **ou** $a + c = b$ **ou** $c + b = a$ **alors**

si $a + b = c$ **alors**

 └ afficher "rectangle en C"

si $a + c = b$ **alors**

 └ afficher "rectangle en B"

si $b + c = a$ **alors**

 └ afficher "rectangle en A"

sinon

 └ Afficher "non"

fin

Ou à l'aide de tests imbriqués :

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

début

$c \leftarrow (x_B - x_A)^2 + (y_B - y_A)^2$

$a \leftarrow (x_B - x_C)^2 + (y_B - y_C)^2$

$b \leftarrow (x_C - x_A)^2 + (y_C - y_A)^2$

si $a + b = c$ **alors**

 | afficher "rectangle en C"

sinon

si $a + c = b$ **alors**

 | afficher "rectangle en B"

sinon

si $b + c = a$ **alors**

 | afficher "rectangle en A"

sinon

 └ Afficher "triangle non rectangle"

fin

Avec Xcas (un return fait sortir immédiatement de l'exécution du programme, cela évite l'imbrication des tests) :

 **Xcas**

```
estrectangle (x_A, y_A, x_B, y_B, x_C, y_C) := {  
  local AB2, AC2, CB2;  
  AB2 := (x_B - x_A) ^ 2 + (y_B - y_A) ^ 2;  
  AC2 := (x_C - x_A) ^ 2 + (y_C - y_A) ^ 2;  
  CB2 := (x_B - x_C) ^ 2 + (y_B - y_C) ^ 2;  
  si AB2 + AC2 = CB2 alors return "triangle rectangle en A"; fsi;  
  si AB2 + CB2 = AC2 alors return "triangle rectangle en B"; fsi;  
  si CB2 + AC2 = AB2 alors return "triangle rectangle en C"; fsi;  
  return "triangle non rectangle";  
};;
```

5. Le résultat pour le triangle "presque rectangle" est l'occasion de parler des nombres en machine – et de rappeler qu'en mathématiques "presque égal" n'est pas "égal" (ce qui est loin d'être un acquis pour nos lycéens).