

Un algorithme c'est quoi ?

- **Un algorithme** est un outil permettant de décrire un algorithme de façon simple et universelle. C'est une représentation graphique normalisée de l'enchaînement des opérations et des décisions effectuées par un programme d'ordinateur.



- **Un algorithme** est une suite d'étape à suivre pour résoudre un problème et obtenir un résultat.

On appelle cela aussi :

Organigramme

Logigramme

Ordinogramme

Pourquoi utiliser un algorithme ?

Il existe plusieurs métiers utilisant des algorithmes.



Technicien Alarme :
Utilise des algorithmes
et logiciels pour
programmer un
système d'alarme



Ingénieur domotique :
Utilise des algorithmes
pour contrôler des volets
roulants

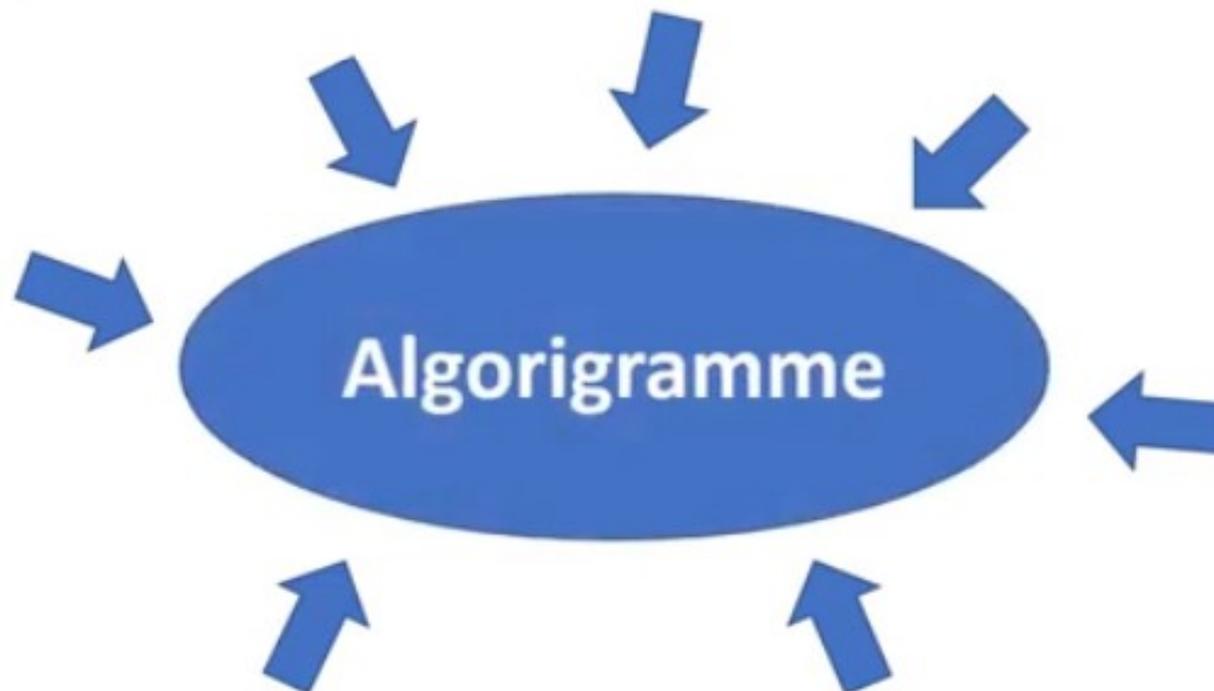


Trader :
Utilise des
algorithmes pour
des calculs
financiers

Ils utilisent tous des algorithmes mais avec des langages différents.

Il faut donc trouver un outil :

- Capable de décrire tous les algorithmes dans un langage commun simple et universel.



Pour réaliser un algorithme :

Nous aurons besoin de :



La forme ovale exprime le début ou la fin de l'algorithme



Le losange exprime une question.



Le rectangle exprime une action.

Débuts et fins (ovales)

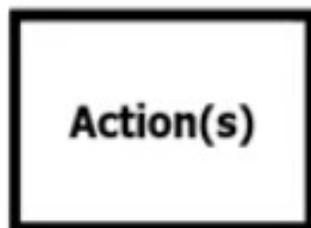


Il n'y a pas toujours de fin dans un algorithme car souvent, le système doit fonctionner en permanence. Dans ce cas, l'algorithme doit former une boucle.

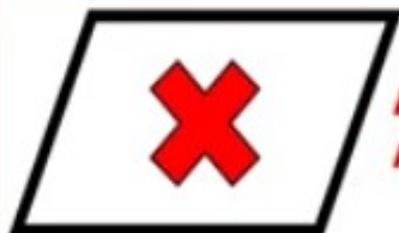


Le symbole doit être conforme et ne doit pas être autre qu'une forme ovale !

Les actions (rectangles)



L'action à réaliser par le système est définie par un ordre écrit à l'infinitif. Dans un rectangle, il est possible de placer plusieurs actions si elles doivent être réalisées en même temps.



Le symbole doit être conforme et ne doit pas être autre qu'un rectangle !

Les questions (losanges)

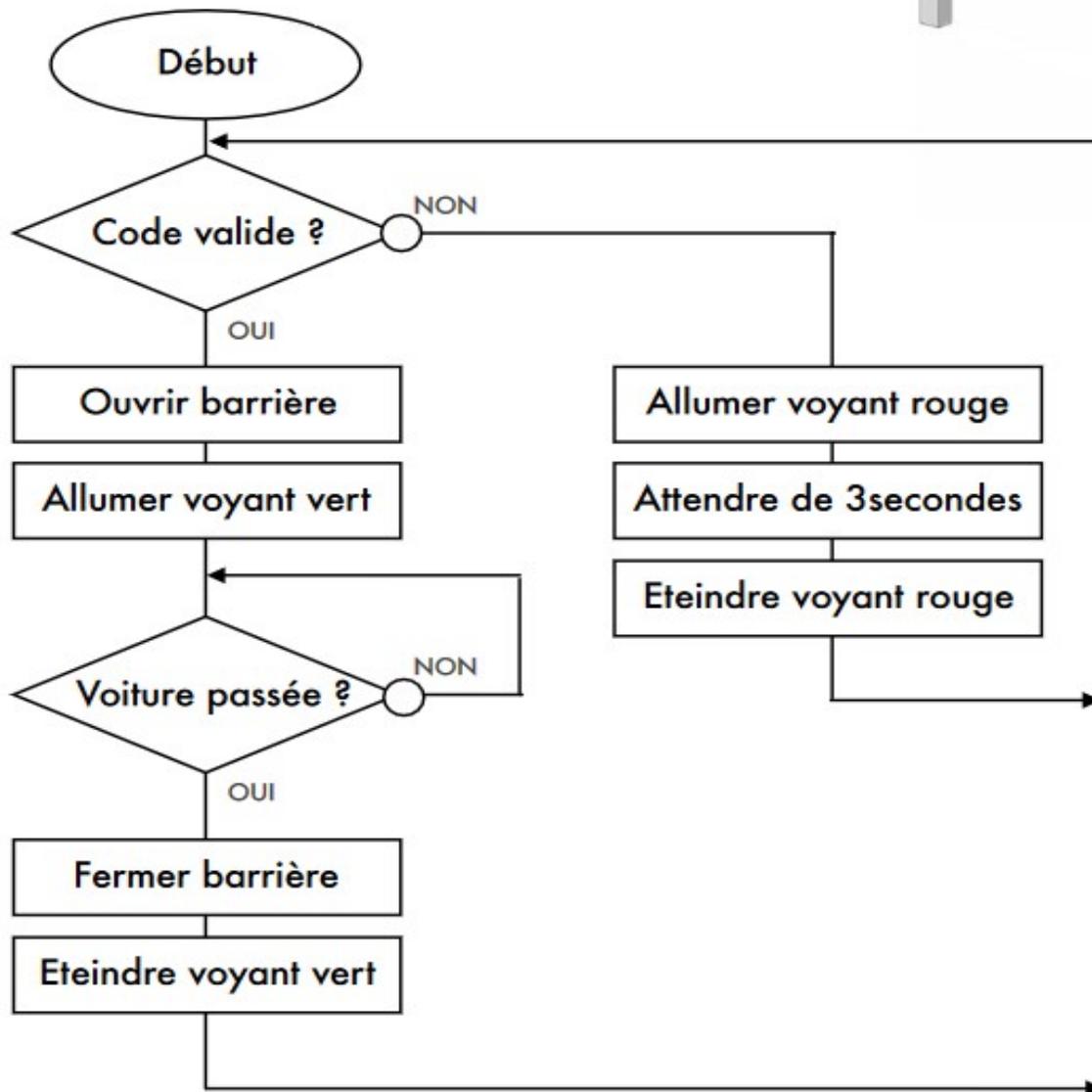
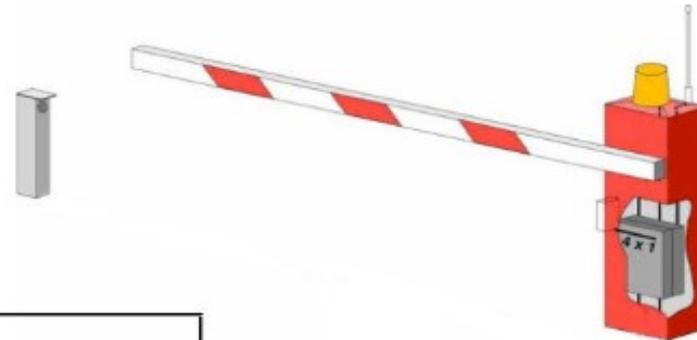


Le losange exprime une question.
La question se termine toujours par un point d'interrogation. La réponse à la question ne peut prendre que deux valeurs : **OUI** ou **NON**.

Chaque question à l'intérieur d'un losange doit obligatoirement se terminer par un point d'interrogation !

Exemple

Barrière automatisée



Une barrière de sécurité utilise un boîtier codé (digicode). Lorsqu'une voiture arrive, le conducteur doit saisir le bon code.

Si le code est bon, le système ouvre la barrière et allume un voyant vert.

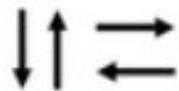
Si le code n'est pas bon, le système allume un voyant rouge pendant 3 secondes. Le conducteur doit ensuite ressaisir son code.

Lorsque le code est bon et après que la barrière se soit ouvert, un capteur indique au système si la voiture est passée.

Lorsque la voiture est passée, le système ferme la barrière et éteint le voyant vert.

Un autre conducteur peut alors utiliser la barrière automatisée.

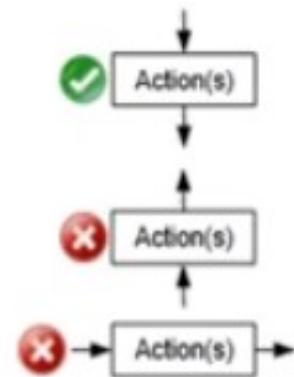
Et ensuite ?



On utilise des flèches (liaisons) pour relier les symboles entre eux. Les flèches sont toujours verticales ou horizontales. Les flèches sont de sens uniques.



Un algorithme se lit de haut en bas.



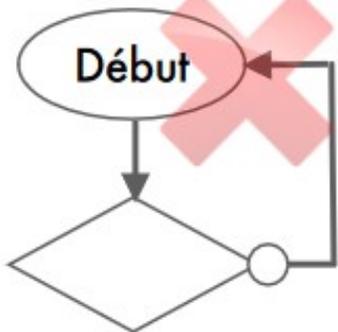
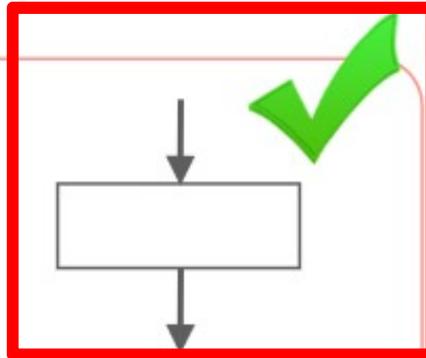
La mise en forme d'un algorithme doit être soignée. (*Tracé à la règle*)

Attention

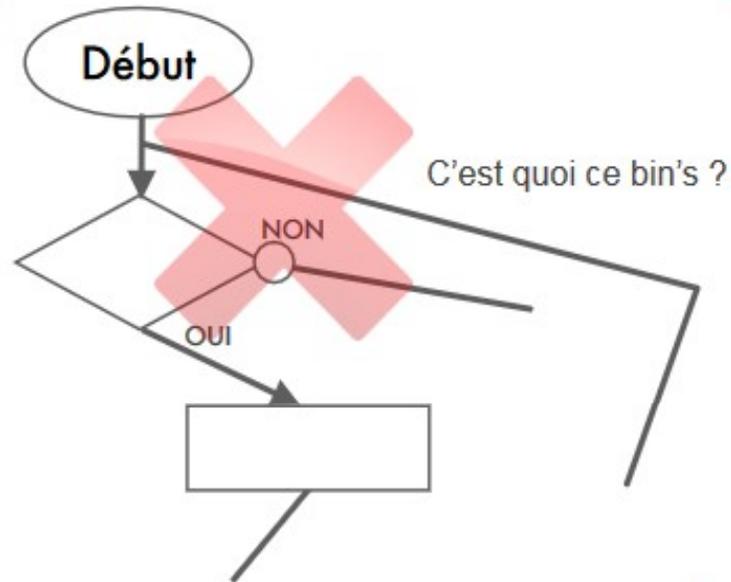
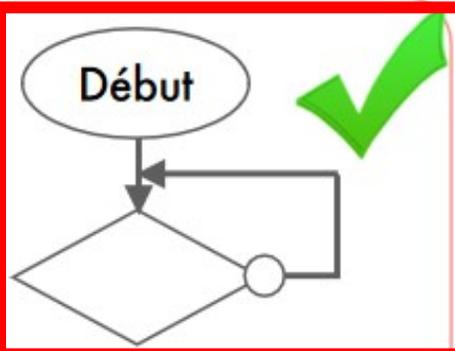
aux erreurs !



J'arrive de dessus, je repars par dessous.



D'une flèche on va ou on retourne à autre une flèche



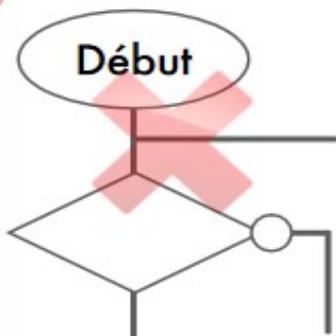
Utiliser uniquement des traits horizontaux ou verticaux !



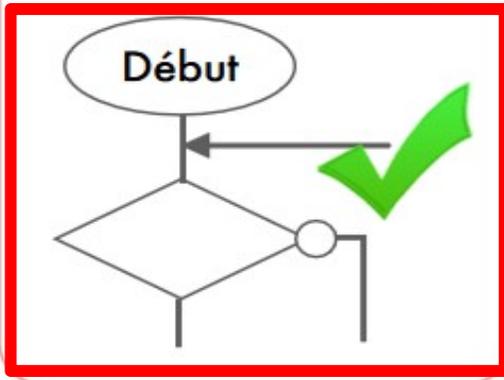
A vérifier

à chaque fois !

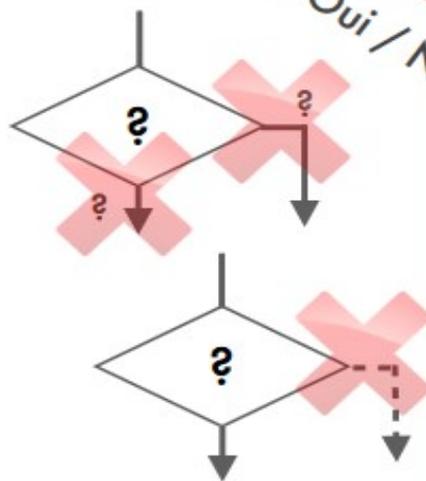
Les flèches



Ne pas oublier le sens des flèches. Sinon impossible de savoir dans quel sens on va ...
Les flèches jouent un rôle de «sens interdit».

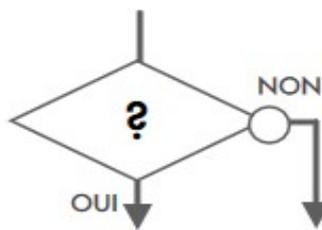


Les Oui / Non

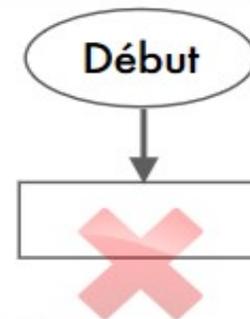


Ne pas oublier toutes les possibilités :

Si Oui alors ...
Sinon ...



Ultime vérif.



Impossible de ne pas finir !
La programmation doit obligatoirement se terminer pas «Fin» ou revenir sur une boucle !

