

CT 1.3 / CT 2.5 / CT 2.7 / DIC 1.5	Imaginer des solutions pour produire des objets et des éléments de programmes informatiques En réponse au besoin.
CT 3.1 / OTSCIS 2.1	Exprimer sa pensée à l'aide d'outils de description adaptés : croquis, schémas, graphes, diagrammes, tableaux.
CT 4.2 / CT 5.5 /IP 2.3	Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.

Algorithme et organigramme.

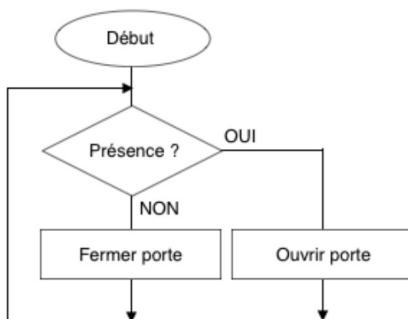
5^e 4^e 3^e

Un **algorithme** est une suite d'instructions précise et structurée qui décrit la manière dont on résout un problème. Cette description peut être textuelle (si, alors, sinon, tant que ...) ou graphique (appelé également organigramme, algorigramme ou logigramme).

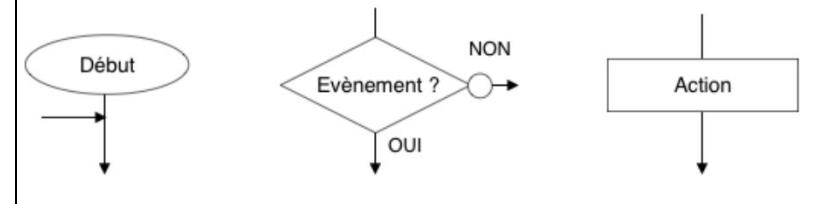
Algorithme graphique (organigramme) :

Algorithme textuel :

Si Présence
Alors ouvrir porte
Sinon fermer porte

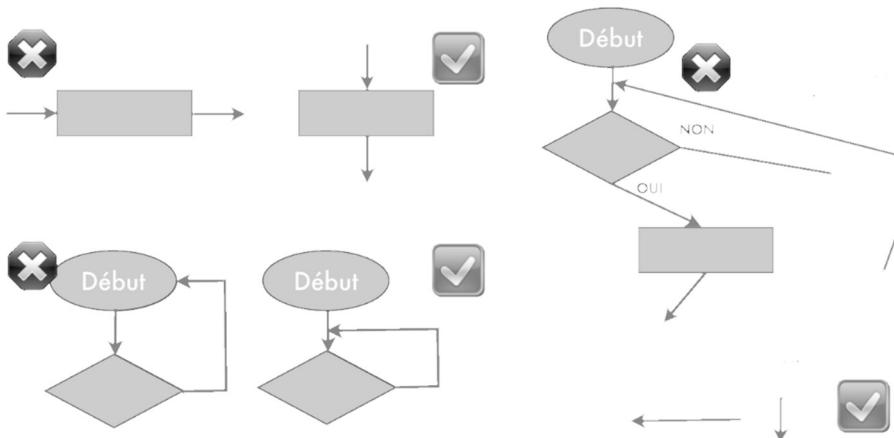


On utilise trois symboles de base :

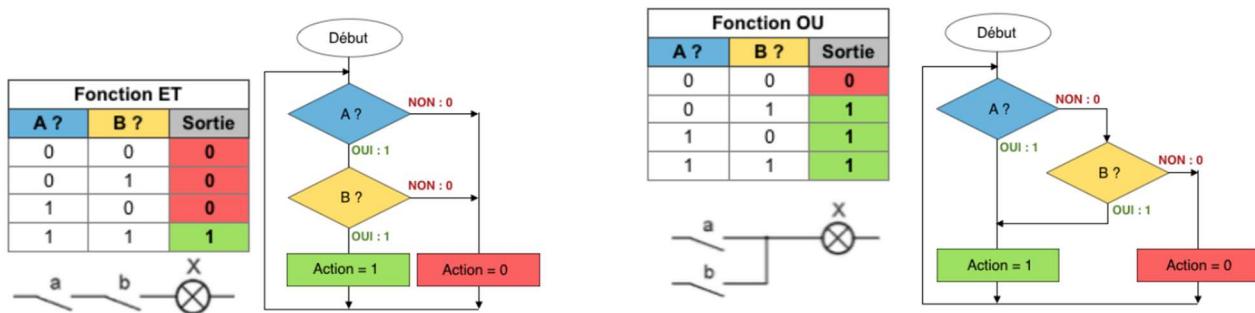


Dans ce cas des normes d'écriture sont à respecter :

- J'arrive de dessus, je repars par dessous.
- D'une flèche on va ou on retourne toujours à une autre flèche.
- Utiliser uniquement des traits horizontaux ou verticaux !
- Ne pas oublier le sens des flèches. Sinon impossible de savoir dans quel sens on va ...
- Ne pas oublier toutes les possibilités : Si Oui alors ... Si Non alors ...
- La programmation doit obligatoirement se terminer par «Fin» ou revenir sur une boucle !



L'utilisation des fonctions ET et OU sont essentielles pour présenter correctement une solution.



Algorithme et gestion des sous-problèmes

L'utilisation des sous-problèmes est idéale pour une meilleure lisibilité, pour alléger l'algorithme lors de succession d'actions identiques, pour faciliter le travail en collaboration, pour faciliter une recherche d'erreur (test individuel des sous-problèmes).

