

Synthèse : analyse d'un système automatique

partie 3 : La programmation d'un système automatique : Notion d'algorithme

Nom :

prénom

A) Algorithme et Logigramme (organigramme)



Thème – L'INFORMATIQUE ET LA PROGRAMMATION

Compétence – Écrire, mettre au point et exécuter un programme

Compétence associée – Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.



Connaissance : Notions d'algorithme et de programme

Les objets connectés sont souvent programmés pour fonctionner **automatiquement**. Chaque fonction numérique de l'objet connecté peut être assimilée à un « problème » à résoudre. La **RÉSOLUTION D'UN PROBLÈME** par un programmeur peut s'effectuer en **TROIS ÉTAPES** :

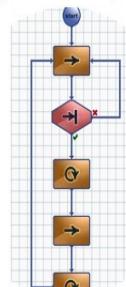
- 1- **Ecriture d'un algorithme** : Suites logique d'opérations ou d'instructions, souvent rédigées sur feuille de papier en utilisant le langage naturel et des mots clés : si, alors, tant que, jusqu'à ...

Exemple : Un robot évitant un obstacle.



- ~ Si le robot détecte un obstacle avec son capteur de pare-choc, alors tourner à gauche de 90° ; avancer de 10cm puis tourner à droite de 90°.
- ~ Sinon avancer indéfiniment.

- 2- **Construction à l'aide d'un logiciel d'une représentation graphique de l'algorithme**



- 3- **Traduction de la représentation graphique en langage de programmation** qui lui-même sera convertit en **langage machine** (code binaire) que le **microprocesseur** peut exécuter

```

    ...
    ...ile((!(analogReadA7>1,
    delay(1000);
    while(((analogReadA7)>1020:1),
    {
        distance = ultrasonic.distanceCm(),
        if((distance) < (100))
            motor.move(1,0);
        delay(1000);
        motor.move(4,100);
        delay(1000);
        motor.move(1,100);
        delay(1000);
        motor.move(1,0);
        delay(1000);
        motor.move(0,6);
        delay(1000);
        motor.move(3,100);
        delay(1000);
        motor.move(1,100);
        ...
    }
    motor.move(1,100);
}
)

```

Programme : lignes de codes en langage C

Pour résoudre un problème, le programmeur commence par écrire un algorithme dans lequel il donne des **ORDRES EN FONCTION DE CONDITIONS (ÉTAT DES CAPTEURS)**. Puis il construit sur un ordinateur une **REPRÉSENTATION GRAPHIQUE DE L'ALGORITHME** (Organigramme ou par bloc avec Scratch).

Le logiciel va ensuite traduire la représentation graphique en ligne de code (le programme) que le système va exécuter.

Fiche connaissance – Notions d'algorithme et de programme

IP-2-3-FE1 - Cycle 4

B) principes de l'expression d'un algorithme sous forme de Logigramme

document Académie de Toulouse

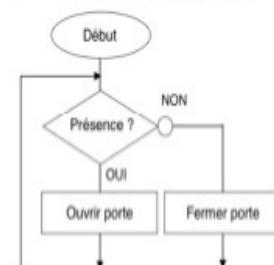
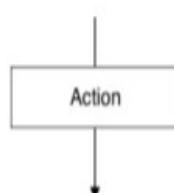
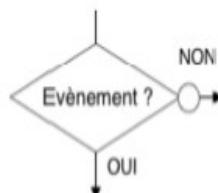
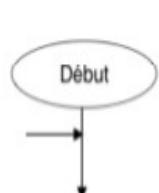
Symboles de base



Un algorithme est une suite d'instructions précises et structurées qui décrit la manière dont on résout un problème.

Cette description peut être textuelle (si, alors, sinon, tant que ...) ou graphique (appelé également organigramme ou logigramme).

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



Début de l'algorithme

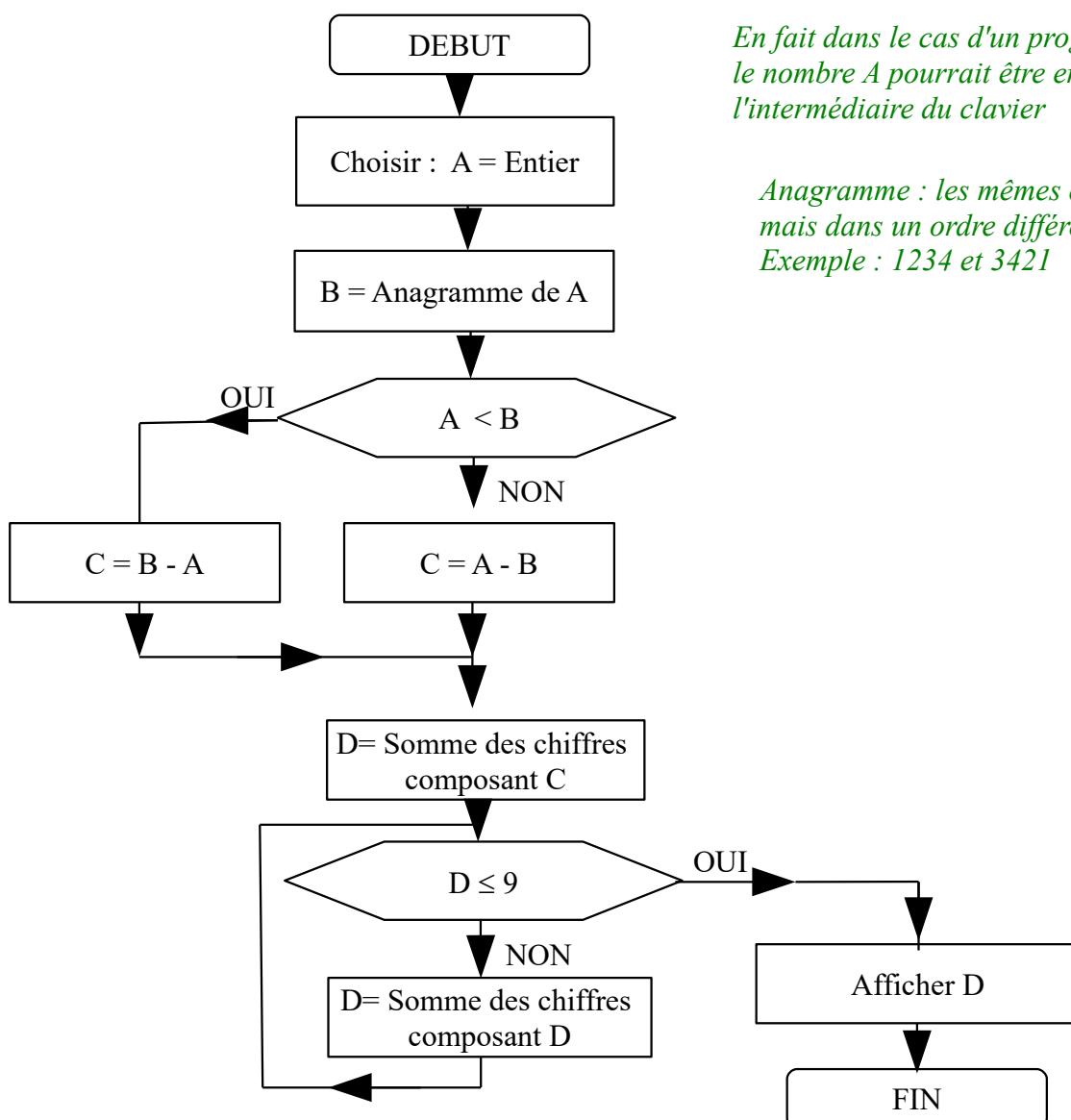
Question à 2 réponses possibles : OUI ou NON

Action : verbe à l'infinitif + complément.

Exemple de l'ouverture automatique d'une porte (magasin).

C) Un exemple de logigramme

Voici l'expression d'un algorithme, sous forme de logigramme qui pourrait être programmé dans un ordinateur



En fait dans le cas d'un programme, le nombre A pourrait être entré par l'intermédiaire du clavier

*Anagramme : les mêmes chiffres mais dans un ordre différent.
Exemple : 1234 et 3421*

vous testerez cet algorithme pour 5 nombres

Valeur de A choisie					
Valeur de B choisie					
D affichée par le programme					

Que remarque t-on ?