

# ALGORITHMES

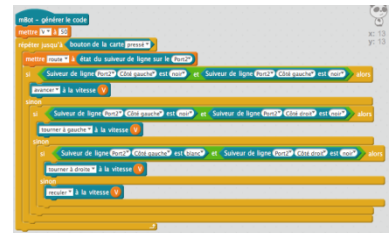
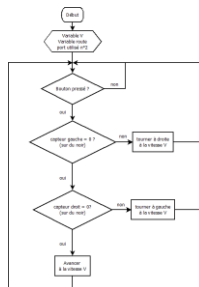
## Compétences du socle travaillées

- Exprimer sa pensée à l'aide d'outils de description adaptés : croquis, schémas, graphes, diagrammes, tableaux (représentations non normées).
- Appliquer les principes élémentaires de l'algorithmique et du codage à la résolution d'un problème simple.
- Piloter un système connecté localement ou à distance.

## Ce que vous devrez savoir faire :

1. Décomposez un problème en étapes, l'algorithme
2. Transposer l'algorithme en logigramme
3. Ecrire un programme informatique

- Créer deux variables, N pour la vitesse et Route pour le numéro du port de sortie utilisé (on prendra le port n°2)
- Répéter le programme jusqu'à ce qu'on appuie sur le bouton de la carte
- Si les deux capteurs gauche et droit sont sur du noir alors avancer à la vitesse V
- Sinon si le capteur gauche est sur du noir et le droit sur du blanc alors tourner à gauche, toujours à la vitesse V
- Et si le capteur gauche est sur du blanc et le capteur droit sur du noir, alors tourner à droite à la vitesse V
- Sinon reculer à la vitesse V



2. Algorithme

3. Logigramme

4. Programme

## Méthode de travail :

Par groupe de 2 ou 3 personnes maximum.

## Matériel et logiciels nécessaires :

- Une feuille de note
- Un robot
- Un logiciel pour créer des logigrammes
- Un logiciel pour programmer le robot

## 1. ALGORITHME

Plein de choses que nous faisons avec notre cerveau (compter, trier des objets, rechercher un mot dans un texte, faire une recette de cuisine, etc.) peuvent être faites par une machine, car cela peut se décomposer en une suite d'actions et de conditions. Par exemple quand on traverse la rue, pour éviter de se faire écraser, on regarde à gauche puis à droite puis à gauche. C'est très efficace et c'est toujours la même méthode ; c'est ça un algorithme ! Une méthode !

Cette méthode consiste donc à décomposer un problème pour obtenir le résultat escompté.

### Exercice1 :

On souhaite préparer des biscottes beurrées avec de la confiture. Trouvez et inscrivez les étapes.

1.....

2.....

3.....

Vous venez d'écrire, sans le savoir, ce qu'on appelle un algorithme !

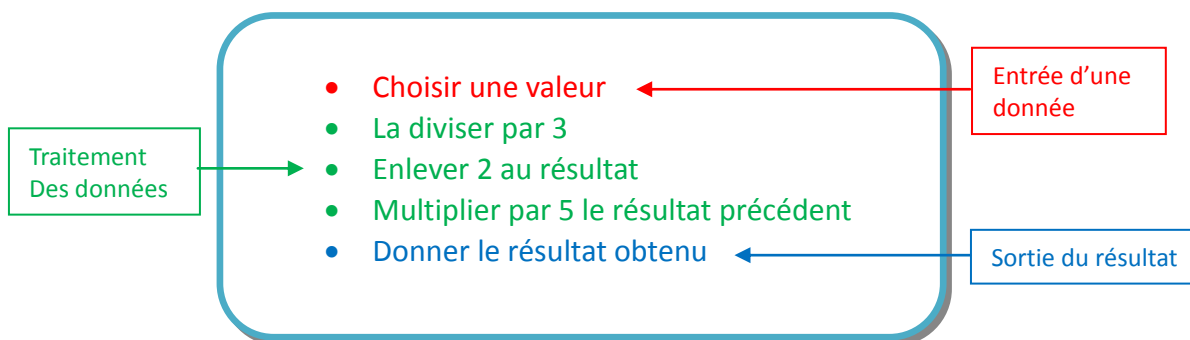
### Exercice 2 :

Voici un programme de calcul habituel fait au collège.

- Choisir une valeur
- La diviser par 3
- Enlever 2 au résultat
- Multiplier par 5 le résultat précédent
- Donner le résultat obtenu

Ce programme de calcul est un algorithme, puisqu'on a décrit les différentes étapes d'un calcul permettant d'obtenir un résultat. La description de cet algorithme permet d'automatiser le calcul proposé grâce à une calculatrice ou un logiciel.

Ce programme de calcul se décompose en **trois phases**. Chacune d'elles correspond à une ou plusieurs étapes de l'algorithme. La phase de traitement des données est constituée de 3 étapes...



Un algorithme possède donc la structure suivante :

- un **début**
- Une **déclaration des variables**
- Une ou plusieurs **entrées**
- Un **traitement** (calculs, comparaisons, conditions, boucles).
- Une ou des **sorties** (*afficher, avancer, émettre un son, tourner, s'arrêter...*)
- Une **fin**

Cette suite d'instructions est ensuite écrite en **langage informatique** (C, C++, Java, etc.  
Au collège nous utilisons les langages Scratch ou mBlock).

## Variables

Lors de l'exécution d'un algorithme, on va avoir besoin de **stocker** des données, voir des résultats. Pour cela, on utilise des **variables** auxquelles on attribue un nom.

Une variable, c'est comme une boîte, repérée par un nom, qui va contenir une information.

Dans l'écriture d'un algorithme, on prendra l'habitude de préciser dès le départ le nom des variables utilisées en indiquant leur type (nombre, chaîne de caractère, liste, etc.). Cette étape est appelée déclaration des variables.

## Tests conditionnels

La résolution des certains problèmes nécessite la mise en place d'un **test** pour savoir si l'on doit effectuer une tâche. **Si** la condition est remplie **alors** on effectue la tâche, **sinon** on effectue (éventuellement) une autre tâche.

Dans un algorithme, on code la structure « Si... Alors... Sinon » sous la forme suivante :

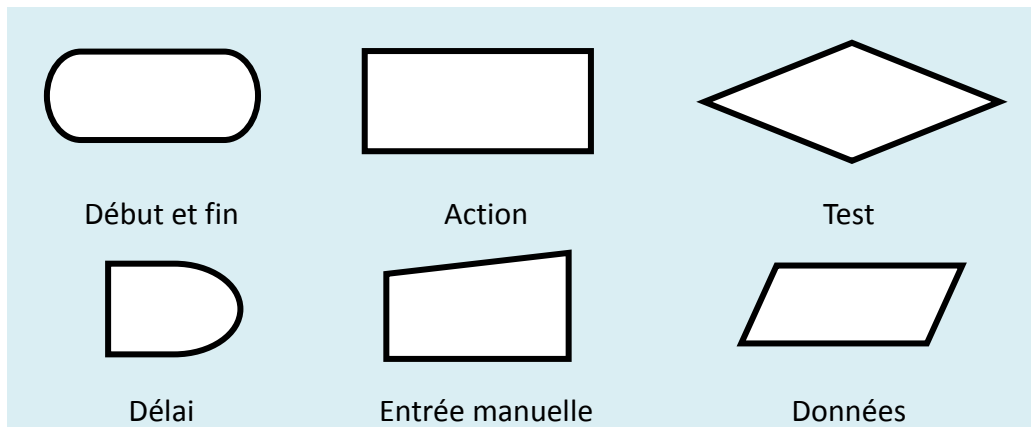
<b>Si</b> condition <b>Alors</b> Tâche 1 Tâche 2 ... <b>Sinon</b> Tâche 1bis Tâche 2bis ... <b>Fin Si</b>	<b>Début</b> <b>Déclaration des variables</b> : X, un nombre entier <b>Entrée</b> :     saisir X <b>Traitement</b> : Si X > 10 alors Affecter à X : 2X Sinon Affecter à X : X+1 Fin Si <b>Sortie</b> :     Afficher X <b>Fin</b>
Test conditionnel	Exemple d'un algorithme simple

## 2. LOGIGRAMME

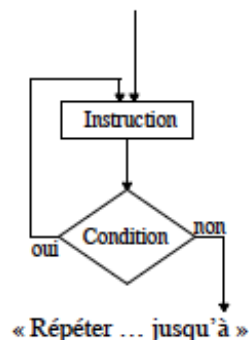
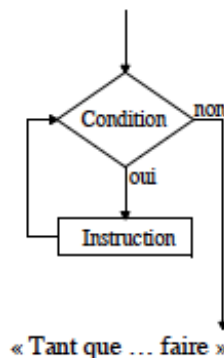
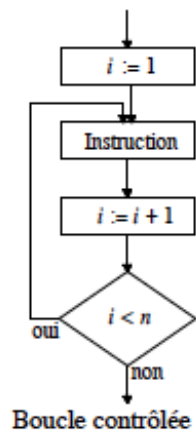
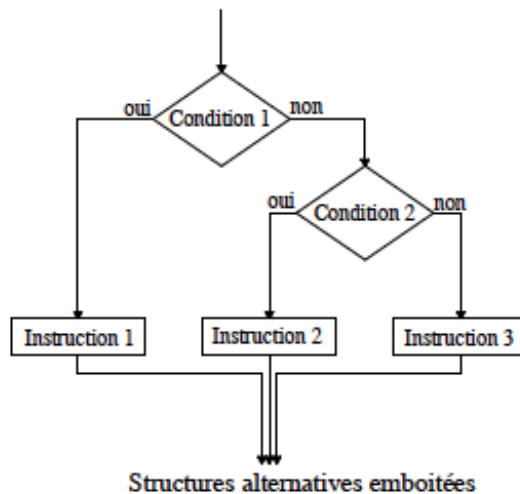
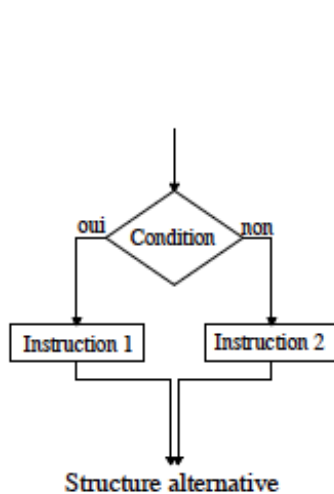
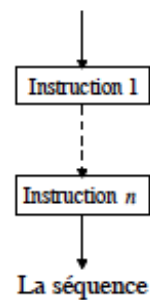
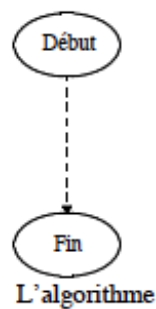
L'objectif du logigramme est de représenter graphiquement les étapes d'un algorithme dont le déroulement est chronologique.

Une fois que l'ensemble des données et informations est collecté, il s'agit de repérer les grandes étapes du processus et de les noter. La formulation doit être minimale : verbe d'action à l'infinitif (par ex. : tourner) + complément (par ex. : à gauche).

Le logigramme utilise principalement les symboles ci-dessous pour identifier les grandes phases du processus. Ils permettent d'obtenir une vision globale et partagée du processus :



## Quelques structures de logigramme



### 3. APPLICATION

#### Problématique

Nous sommes toujours étonnés de voir un robot se diriger seul... On dirait qu'ils ont des sens, un peu comme nous... Comment arrivent-ils à se diriger dans leur environnement ?

#### Approche :

Nous allons tenter de répondre à ce questionnement par l'étude de deux cas : un où le robot est guidé par un élément extérieur et l'autre où il « scanne » son environnement...

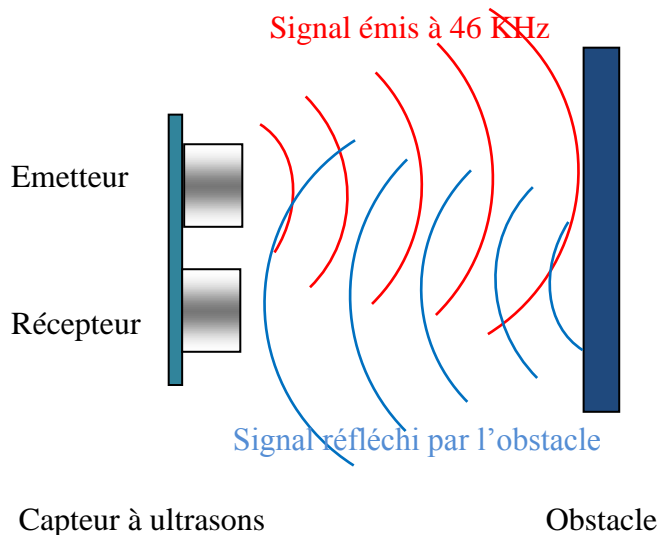
#### Premier cas :

Les moyens de guider un robot sont multiples : télécommande, clavier d'ordinateur, son, lumière, etc. L'élément extérieur, ici choisi, est une *ligne noire tracée sur un fond blanc*. Le robot devra la détecter et la suivre. Nous utiliserons un capteur infrarouge qui est sensible au contraste. Dans ce capteur, l'émetteur (une diode infrarouge, comme sur les télécommandes) envoie une lumière IR que le sol réfléchit en direction du récepteur (un phototransistor) qui capte la quantité de lumière en retour.



Comme les couleurs **foncées** réfléchissent moins bien la lumière que les couleurs **claires**, le capteur peut ainsi nous informer s'il se trouve sur du blanc (lorsqu'il reçoit beaucoup de lumière), il prend alors la valeur 1, ou s'il se trouve sur du noir (lorsqu'il reçoit peu de lumière), il prend alors la valeur 0.

Deuxième cas : le « scanneur » « balaiera » son environnement sur le principe de l'émission et de la réception d'ondes. Nous utiliserons un capteur à ultrasons qui possède un émetteur et un récepteur d'ondes.



Pour cette expérimentation nous utilisons le robot **mBot** car il possède les deux capteurs en question (infrarouge et ultrasons)

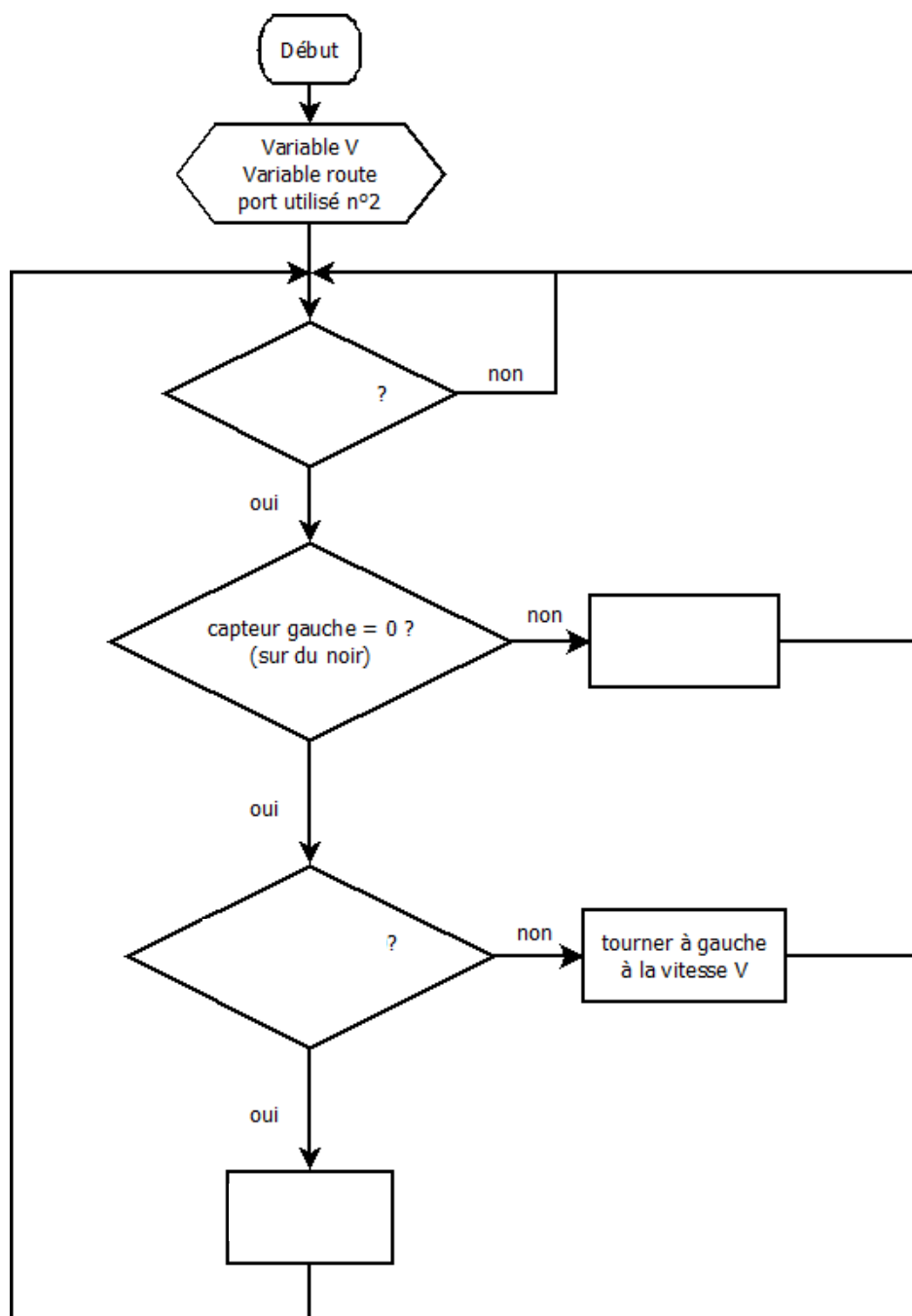
#### EXEMPLE : Programmation d'un suivi de ligne

### Algorithme du suivi de ligne

- On Crée deux variables : **V** pour la vitesse et **Route** pour le numéro du port de sortie utilisé (on prendra le port n°2)
- Répéter le programme jusqu'à ce qu'on appuie sur le bouton de la carte
- Si les deux capteurs gauche et droit sont sur du noir alors avancer à la vitesse V
- Sinon si le capteur gauche est sur du noir et le droit sur du blanc alors tourner à gauche, toujours à la vitesse V
- Sinon si le capteur gauche est sur du blanc et le capteur droit sur du noir, alors tourner à droite à la vitesse V
- Sinon reculer à la vitesse V

### Logigramme du suivi de ligne

Complétez les 4 symboles vides en vous aidant de l'algorithme



## 4. PROGRAMME INFORMATIQUE

Suite d'instructions écrite dans un langage informatique (C, C++, Python, Java, Ada, Fortran...) compréhensible par un *microprocesseur* (une puce comme l'*Intel i7* ou l'*Atmel 328* de la carte Arduino que nous utilisons dans le robot mBot). Le microprocesseur exécute cette suite d'instructions traduite en code binaire...

```
int ledPin = 8;
void setup()
{
  pinMode(ledPin, OUTPUT);
}
void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(1000);
  digitalWrite(ledPin, LOW);
  delay(1000);
}
```

Exemple de programme en C écrit à la main



Le même programme conçu avec Scratch

Programme suivi de ligne écrit avec mBlock

