Introduction à l'algorithmique et au langage C

2e partie - Structures de contrôle

L. Beaudoin & R. Erra & A. Gademer & L. Avanthey

2016 - 2017 – `http://learning.esiea.fr/`

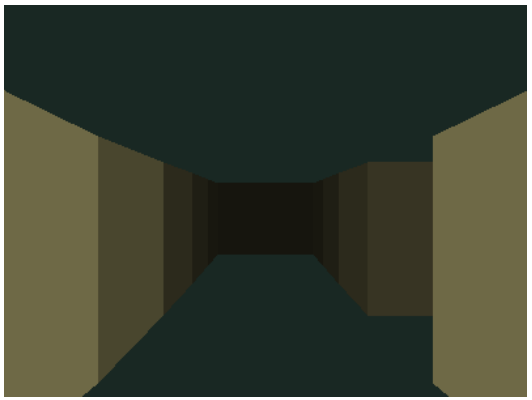
Plan

Plan: Introduction

Objectifs de cette présentation

Acquérir les bases théoriques qui vont nous permettre de faire réaliser une tâche par la machine !

- Qu'est-ce qu'une action élémentaire ? Comment enchaîner les actions ? Comment mémoriser une information ?
- Comment faire des choix ? Comment répéter une action ?

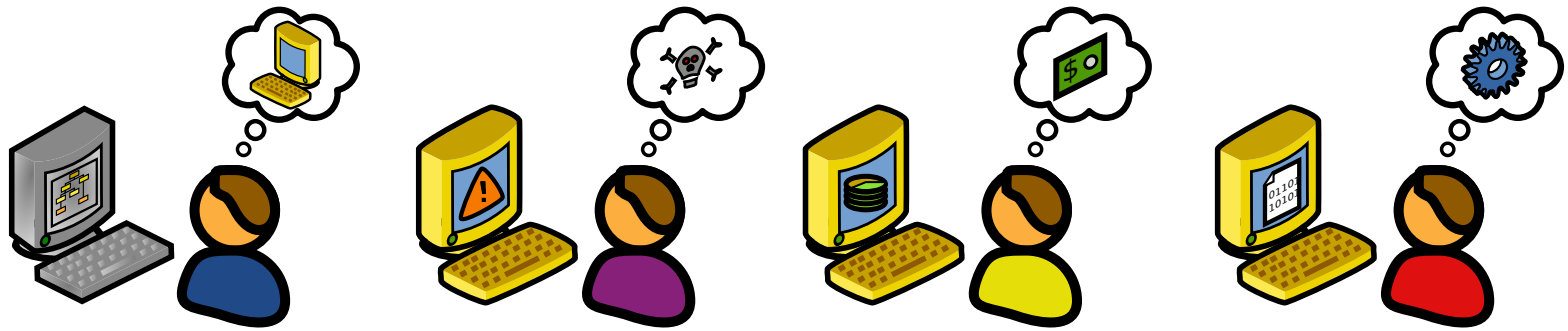


Plan: Définitions

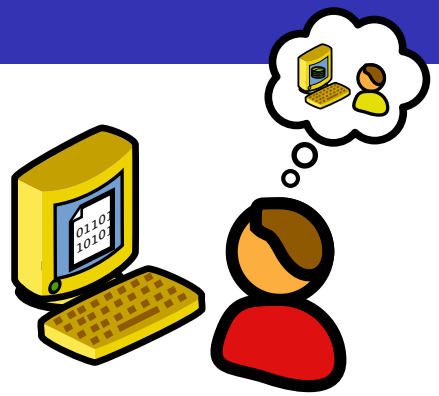
Définition : Informatique

Rappel : *« L'informatique est la science du traitement automatique de l'information. »*

- Cela regroupe de très nombreuses activités :
  - Architecte système (domaine fortement électronique)
  - Administrateur systèmes et réseaux, Webmaster, ...
  - Spécialistes des logiciels métiers (Infographistes, spécialistes des logiciels de gestion, géomaticiens...)
  - Programmeurs / Algorithmiciens
  - ...

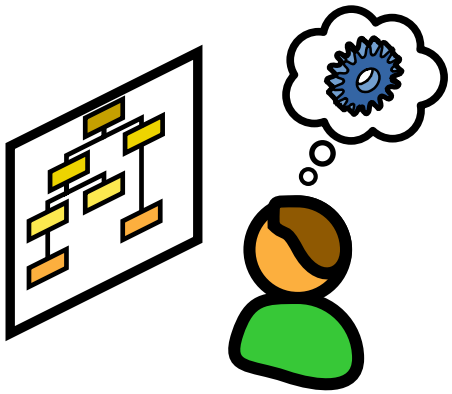


Programmeur ?



Qu'attend-t-on d'un programmeur ?

- Qu'il fasse réaliser des tâches à la machine
- Qu'il fasse des outils, utilisables par d'autres, qui réalisent ces tâches.



Qu'est-ce qu'un algorithmicien ?

- C'est un spécialiste de la meilleure manière de diviser un problème complexe en sous-problèmes simples et par conséquent faire résoudre ce problème par un ordinateur.

Les deux métiers sont fortement liés (voire inséparables) comme on le verra par la suite.

Définition : Algorithme

Algorithme

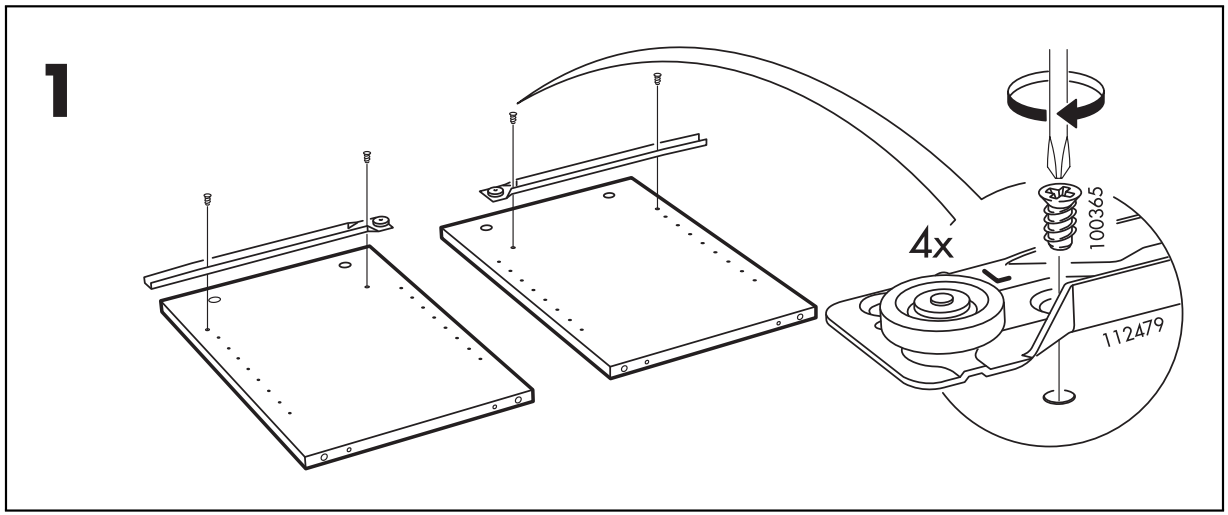
Ensemble des étapes simples nécessaires à la réalisation d'une tâche

⇒ **Par extension** : « Ensemble des instructions atomiques [1] qui, fournies dans l'ordre à un ordinateur, lui font réaliser une tâche. »

1. atomique : au sens de indivisible, qui ne peut être décomposée.

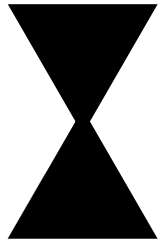
Définition : Algorithme

- Exemple : Plan de montage Ikea



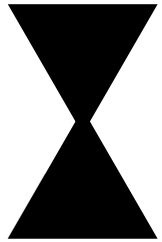
Exercice

- **Tâche** : *Mesurer 3 minutes avec un sablier de 1 minute et votre mémoire*.



Exercice

- **Tâche** : *Mesurer 3 minutes avec un sablier de 1 minute et votre mémoire*.



- Un algorithme possible :
  1. **Mémoriser** la valeur 0,
  2. **Retourner** le sablier
  3. **Attendre** que le sable s'écoule
  4. **Ajouter** 1 à la valeur et **mémoriser** la somme,
  5. **Si** la valeur est strictement inférieure à 3, **recommencer** à l'étape 2.

Plan: Déroulement d'un programme

Déroulement d'un programme

Rappel : « *Les ordinateurs sont des* ***machines de Turing****, ils passent d'un état à un autre* »

- Les ordinateurs ont donc un fonctionnement **séquentiel**.
- Les **programmes** sont exécutés par les ordinateurs, leur déroulement est donc lui aussi **séquentiel**

Exemple : la machine à café

1. **Faire tomber** le gobelet
2. **Faire tomber** la touillette
3. **Faire chauffer** l'eau
4. **Verser** le sucre
5. **Verser** le café
6. **Emmettre** un bip

Déroulement d'un programme

Séquentialité

Un ordinateur ne peut pas effectuer deux tâches simultanément, il doit d'abord en réaliser une et attendre qu'elle soit terminée pour pouvoir en commencer une autre.

Plan: Embranchements

Faire des choix

Si on veut de l'interaction, on doit pouvoir faire des choix conditionnels

- Première **structure de contrôle** : **les embranchements**

Exemple : la machine à café (bis)

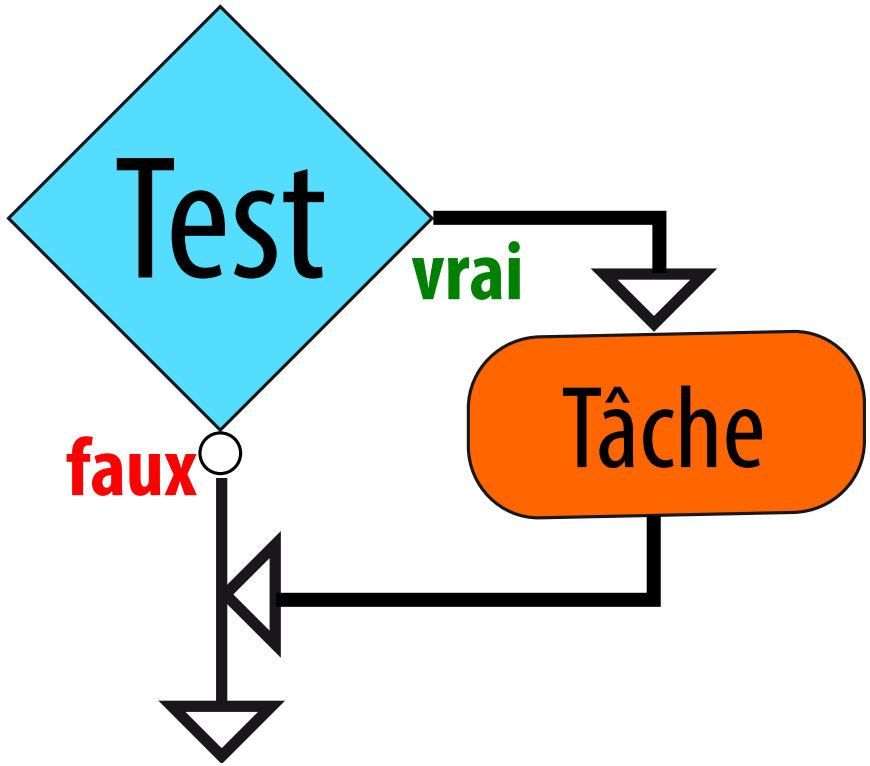
1. **Recevoir** une pièce
2. **Si** la pièce est vraie
   1. **Faire tomber** le gobelet
   2. etc.
3. **Sinon** (la pièce est fausse)
   1. **Rendre** la pièce

Faire des choix

- Action conditionnée :

  Si (...)

  alors {...}



Le schéma s'appelle un **diagramme de flux** et la version écrite un **pseudo-code**

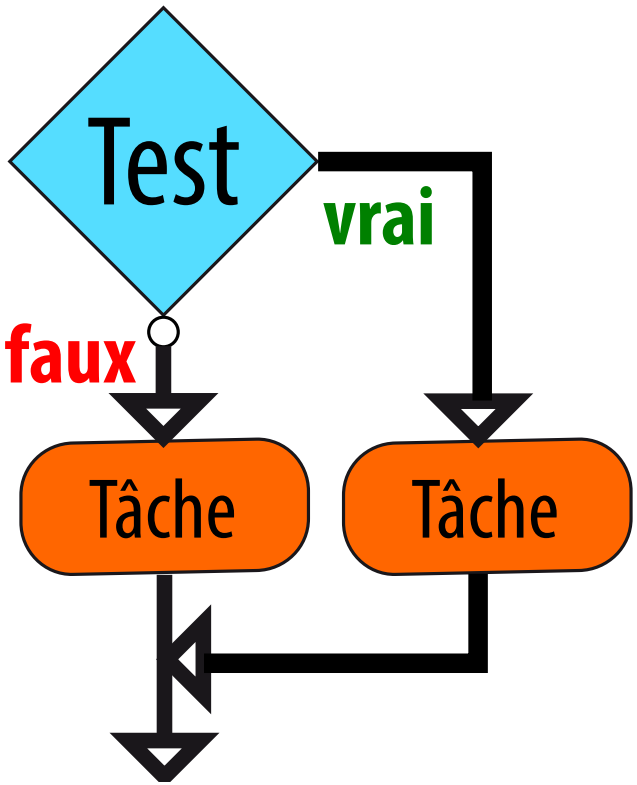
Faire des choix

- Choix :

  Si (...)

  alors {...}

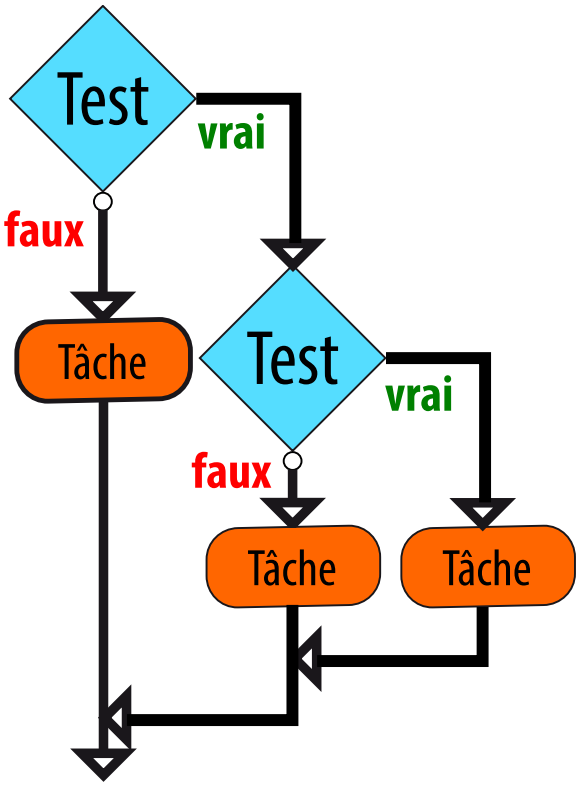
  Sinon {...}



Faire des choix

- Choix imbriqués :

```
Si (...) alors {...}
Sinon {
        Si (...) alors {...}
}
```

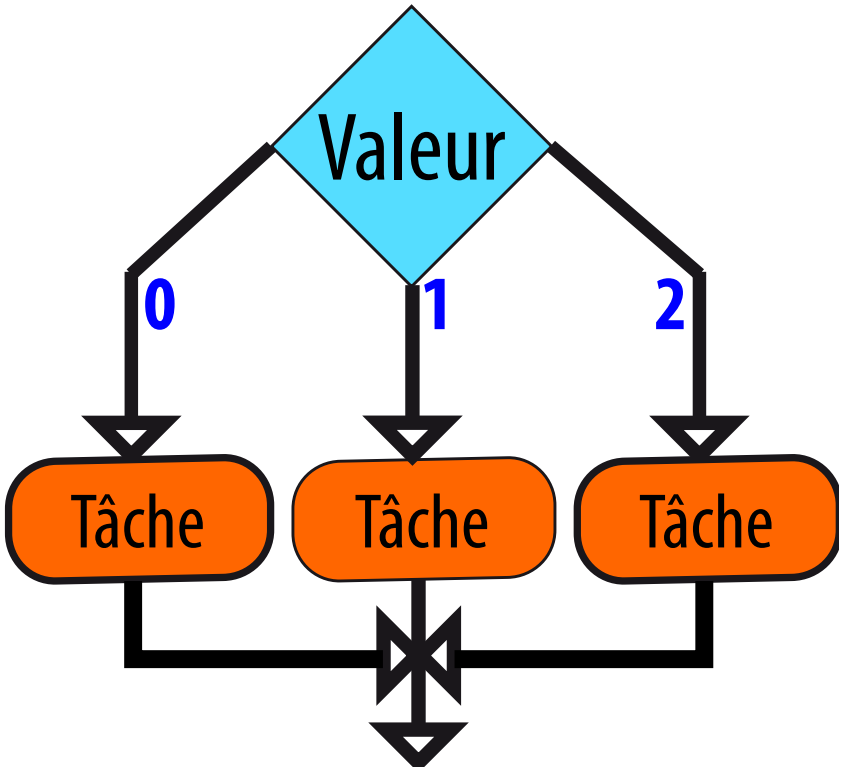


Faire des choix

- Choix multiple :

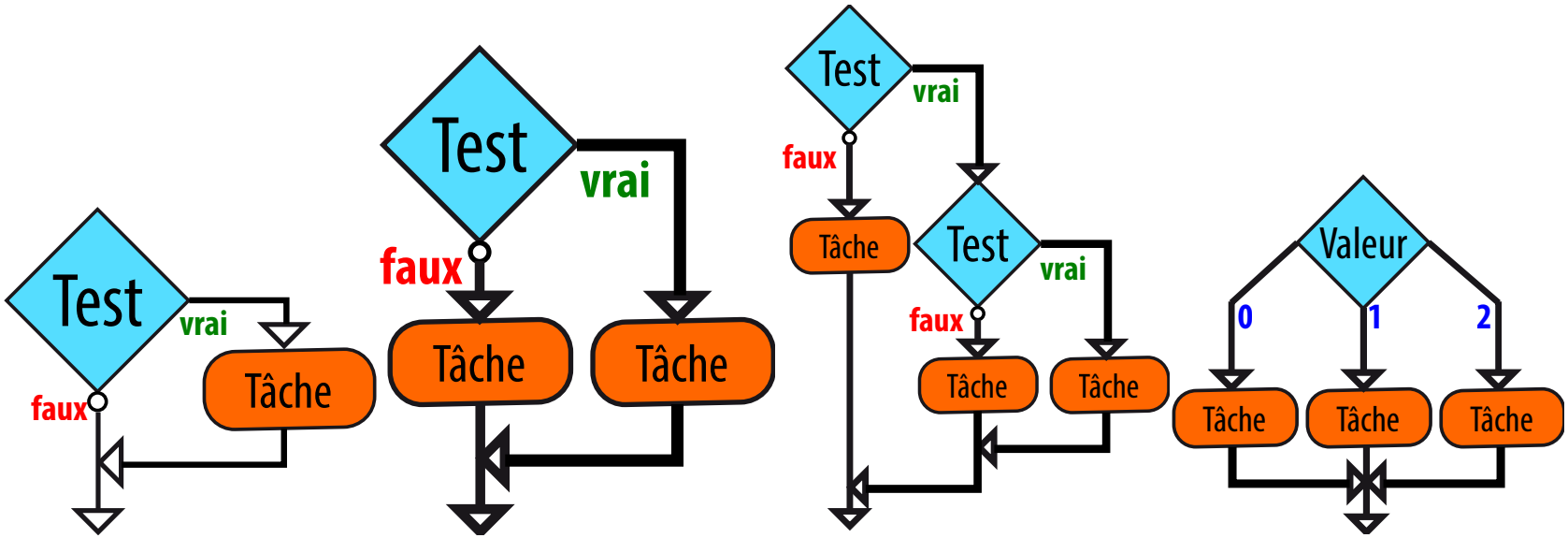
  Si *valeur* vaut :

  0 alors {...}
  1 alors {...}
  2 alors {...}



Faire des choix : récapitulatif

Il y a plusieurs types d'embranchements



Action conditionnée - Choix - Choix imbriqués - Choix multiples

Structure du programme

Dans les exemples précédents, on remarque deux éléments du programme qui n'ont pas le même sens

- (...) **les tests** qui sont des expressions dont la valeur est vraie ou fausse.
- {...} **les blocs** qui sont un groupe d'instructions correspondant à une action ou à une série d'actions.

Structure du programme

Exemples d'expressions :

- Il fait beau
- $2+2 = 5$
- Vous voulez du fromage OU vous voulez du dessert

Exemples de blocs :

- Je sors mon parapluie
- Je vais me plaindre à mon revendeur
- Je vous apporte la carte

Plan: Boucles

Répéter des actions

Si on veut automatiser, on doit pouvoir répéter des actions

- Automatisation : répéter des actions
- Deuxième **structure de contrôle** : **les boucles**

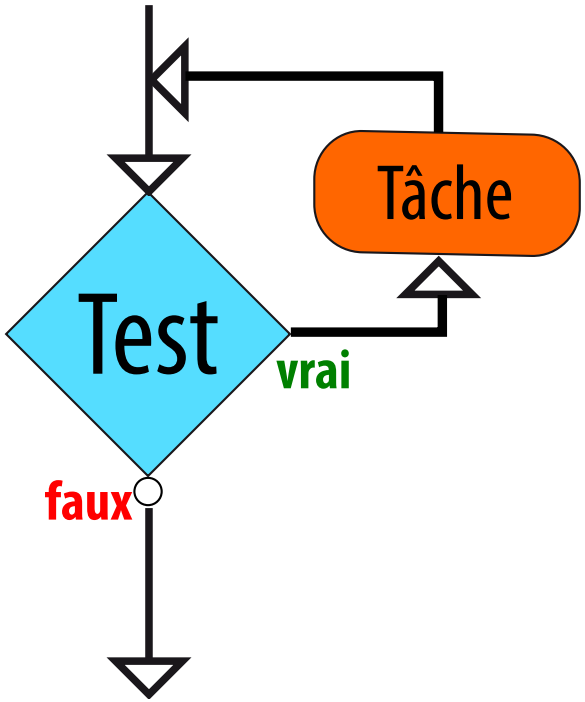
Exemple : la machine à café (ter)

1. **Tant que** je reçois une pièce
   1. **Si** la pièce est vraie
      1. **Faire tomber** le gobelet
      2. etc.
   2. **Sinon** (la pièce est fausse)
      1. **Rendre** la pièce

Répéter des actions

Répéter = revenir *en arrière* dans la séquence

- Tant que (test avant) :
  Tant que (...) fait {...}

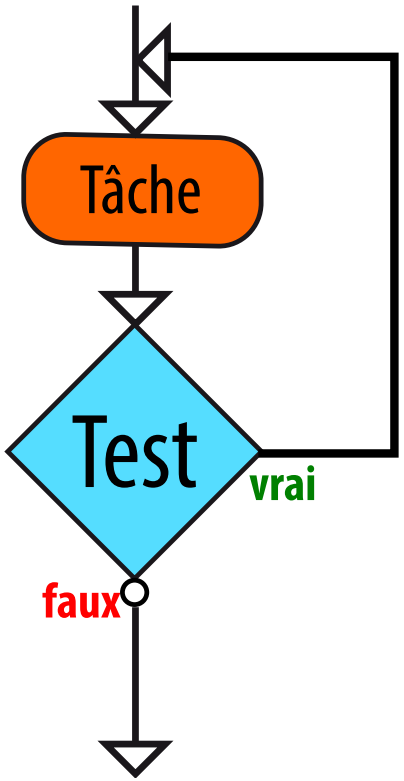


Boucles événementielles

On attend l'arrivée d'un événement pour arrêter la répétition

Répéter des actions

On peut aussi d'abord faire une action, puis tester le résultat pour savoir si on répète ou non



- Tant que (test après) :

  Fait {...} tant que (...)

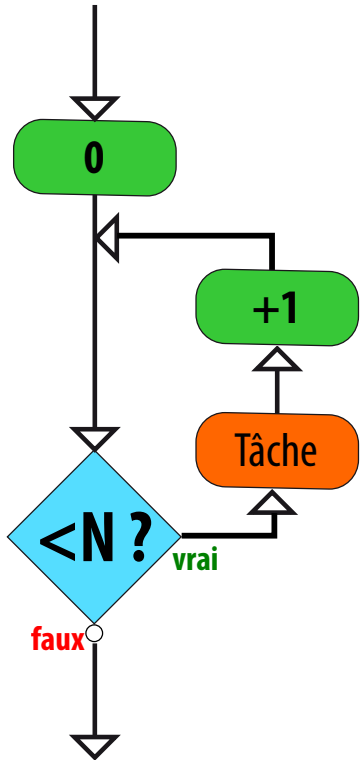
Si on ne sait pas **à l'avance** combien de fois on doit répéter l'action, on utilise une boucle événementielle

Répéter des actions

Si on sait à l'avance combien de fois on doit répéter l'action, alors c'est une **boucle itérative** : on utilise un **compteur** !

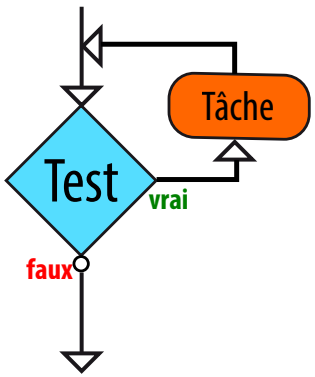
- Compteur :

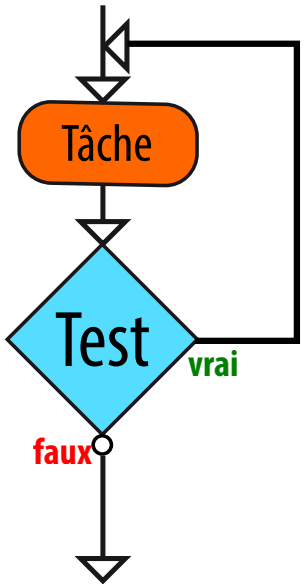
  Pour (*compteur* allant de 0 à $N-1$)
  fait {... Tâche ... Incrémente *compteur*}

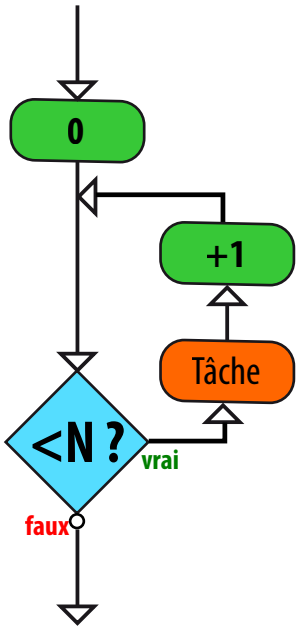


Répéter des actions : récapitulatif

Il y a également plusieurs types de boucles







Tant que *a priori* - Tant que *a posteriori* - Compteur

Plan: Variables

Mémoriser

Condition de Turing n°1 : *« connaitre son état »*

- Les ordinateurs ont une mémoire et on peut l'utiliser par l'intermédiaire des **variables** et des **constantes**.

Exemple : la machine à café (quater)

1. **Définir** "Prix" comme étant la valeur 5
2. **Définir** "Nombre de pièces" comme étant un "Entier" de valeur "0"
3. **Tant que** je reçois des pièces
   1. **Augmenter** "Nombre de pièces" d'une unité
4. **Si** "Nombre de pièces" $>$ "Prix"
   1. "Faire le café"

Mémoriser

- Pour le moment nous considèrerons qu'il existe des variables pouvant contenir :
  - des nombres **entiers relatifs**
  - des nombres à **virgule flottante** (aussi appelés flottants)
  - des **caractères** (texte)

1 2.5 c

Mémoriser

- Toutes les variables numériques peuvent être **manipulées** par des **opérateurs mathématiques** (+,-,*,/...) ou par des **opérateurs de tests** (<, <=, ==, !=, >,>=)
- On peut **affecter** une valeur ou un résultat à une variable avec l'**opérateur** =
- On **accède** généralement au contenu d'une variable ou d'une constante par son **nom**. ex : `prix = 5` ou `monnaie = pieces - prix`

Conclusion

Un programme se déroule **séquentiellement**

Nous avons vu **deux structures de contrôle** :

- Pour faire des **choix** : les **embranchements**
- Pour **répéter** des actions : les **boucles**

Pour **mémoriser** on utilise des **variables**

On peut représenter un algorithme par un **diagramme de flux** ou du **pseudo-code**

Conclusion

Algorithme

Ensemble des étapes simples nécessaires à la réalisation d'une tâche

Plan: Exercice

Exercice

QUESTION 1

 Écrire le diagramme de flux, puis le pseudo-code de l'algorithme répondant au problème suivant :

Je désire maintenir la température de la pièce à 20°. J'ai accès :

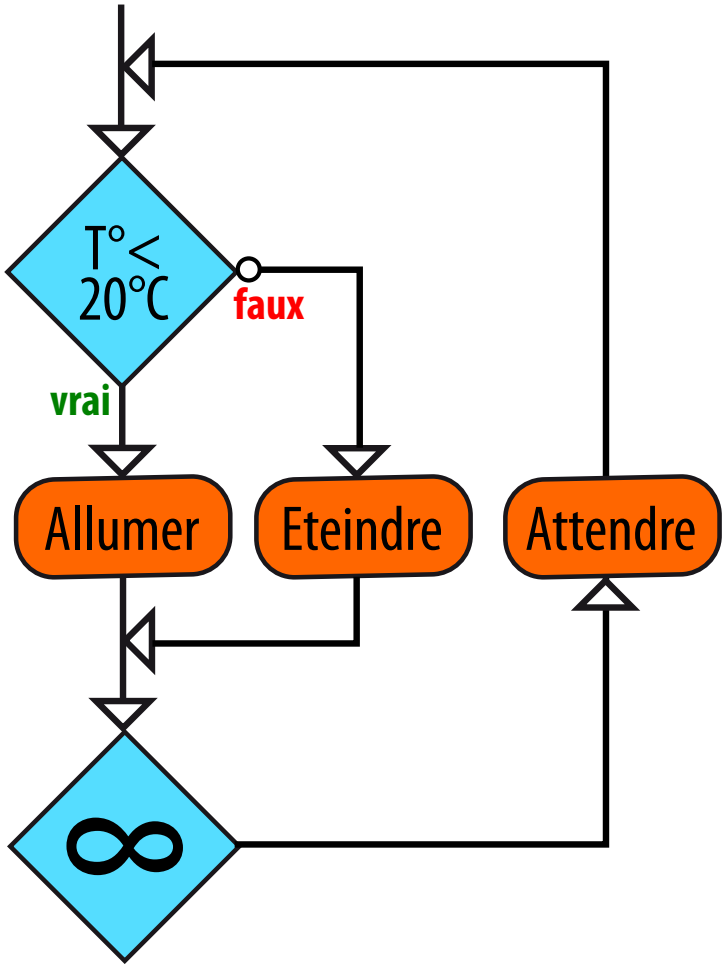
⇒ *aux tâches*

- « Allumer chauffage »
- « Éteindre chauffage »
- « Attendre »

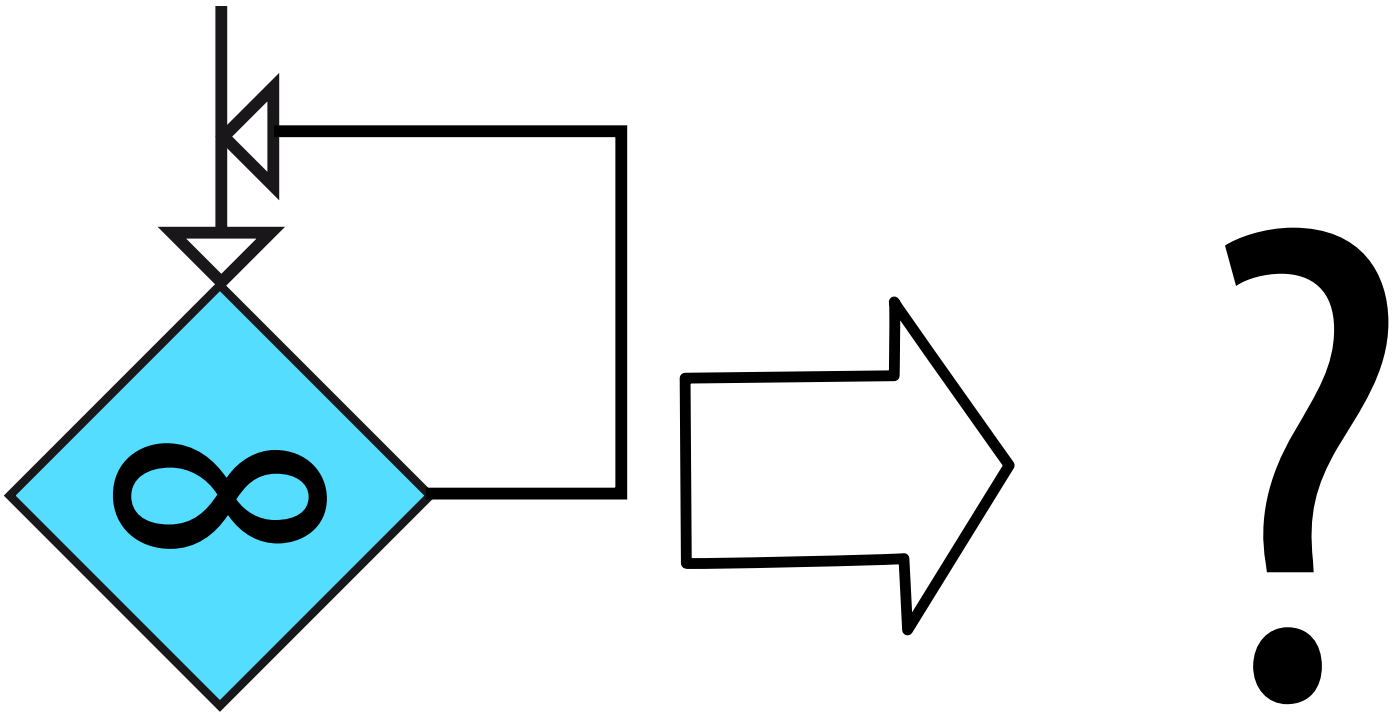
⇒ *aux tests*

- « Température $< 20°$? »
- « Température $> 20°$? »

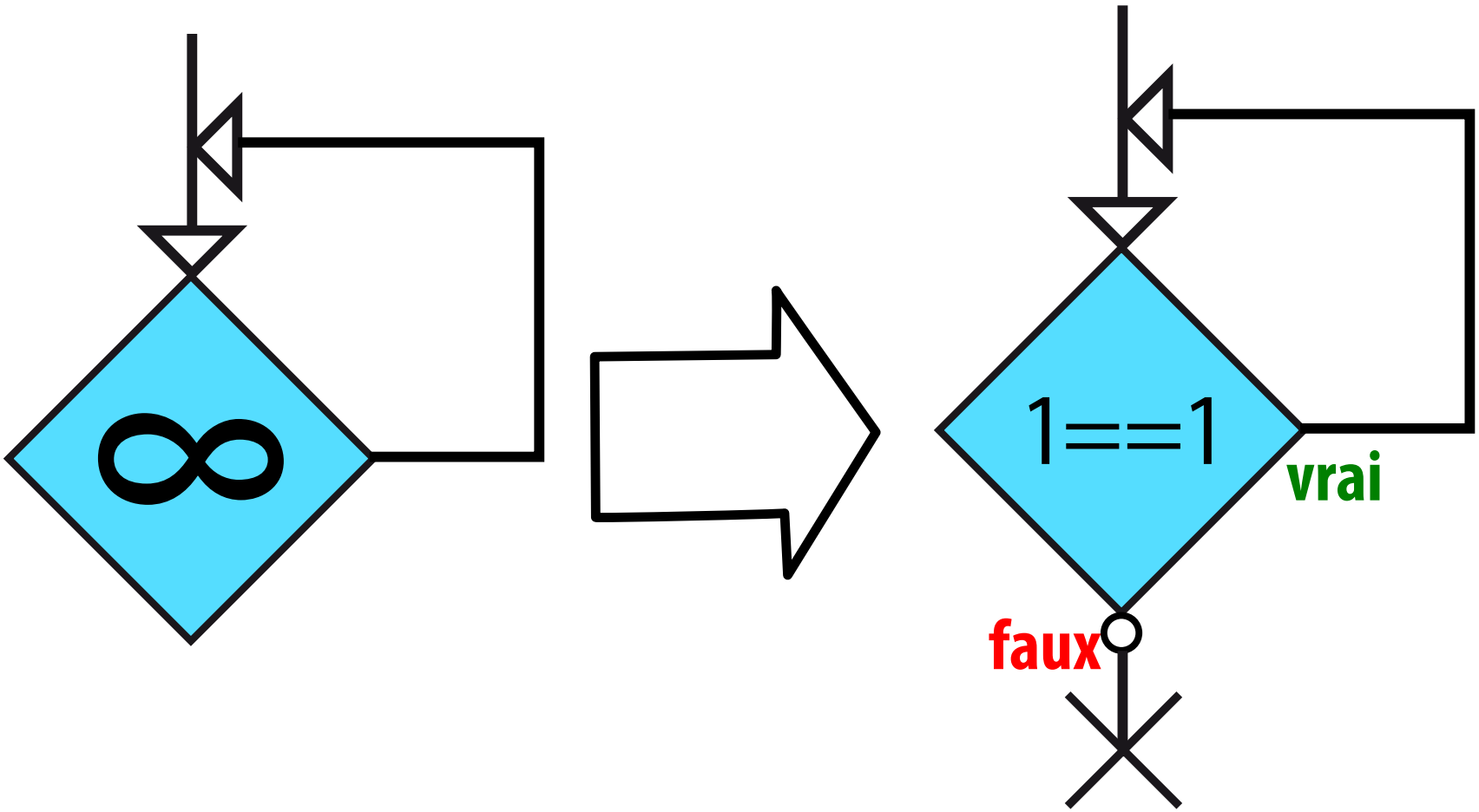
Diagramme de flux



Boucle infinie ?



Boucle infinie



Pseudo-code

répéter
  si $\mathtt{T} < 20°$ **alors**
    Allumer le chauffage
  sinon
    Éteindre le chauffage
  fin
  Attendre
tant que $1 == 1$