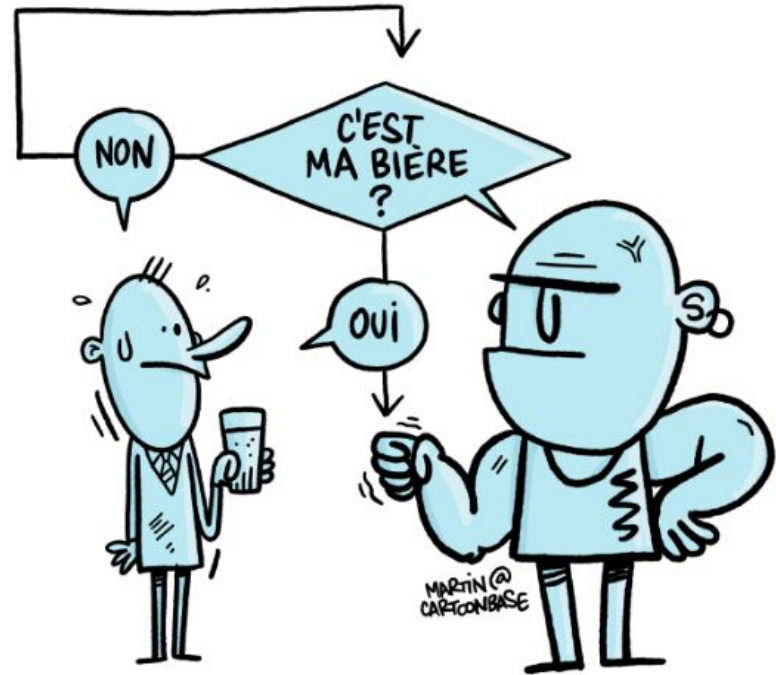


# Algorithmie

Session 2  
Algorigrammes  
Structure de données



UN ALGORITHME NE RÉSOUD PAS TOUT

# Algorithme :: Boucle Conditionnelle

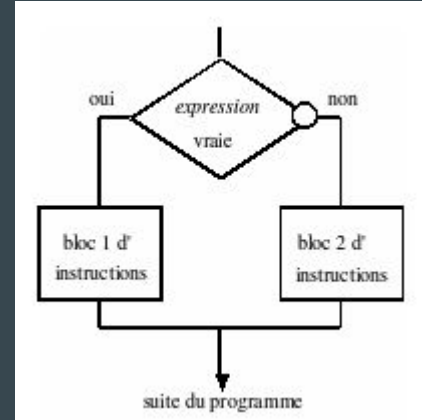
**SI** (expression vraie) **ALORS**

BLOC 1 D'INSTRUCTIONS

**SINON**

BLOC 2 D'INSTRUCTIONS

**FINSI**



# Algorithme :: Boucle SUIVANT CAS (SELON)

**SUIVANT CAS** variable

"C1" : BLOC 1 D'INSTRUCTIONS

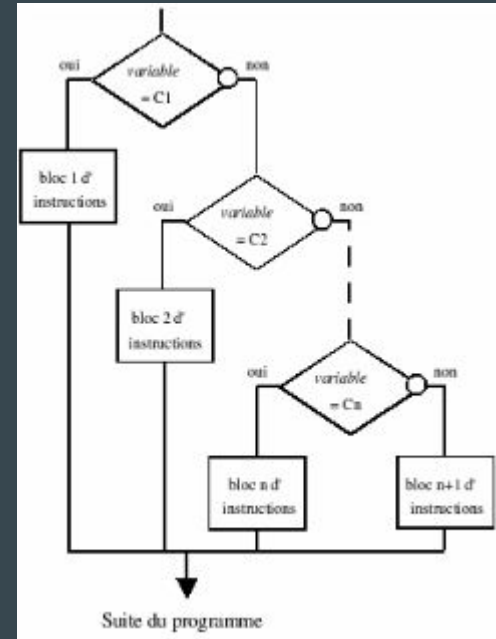
"C2" : BLOC 2 D'INSTRUCTIONS

"C3" : BLOC 3 D'INSTRUCTIONS

"C3" : AFFICHER ( " Mademoiselle " )

**AUTRE** : AFFICHER ( " Monsieur, Madame " )

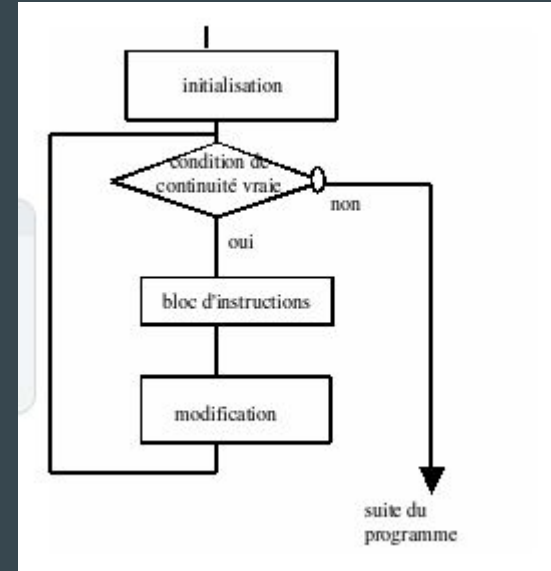
**FINSUIVANTCAS**



# Algorithme :: Boucle POUR

Répéter une suite d'instructions un certain nombre de fois

```
POUR <var> <- valInit A valFin PAR <pas>  
FAIRE  
    traitement {suite d'instructions}  
FINPOUR
```



# Algorithme :: Boucle TANTQUE ... FAIRE

Répéter un traitement a nombre d'itération inconnu

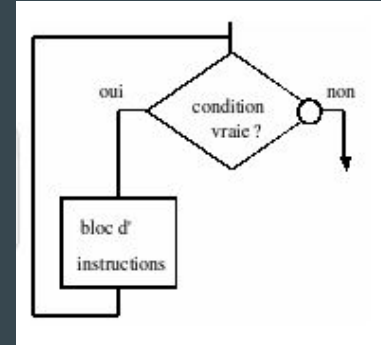
amorçage

**TANTQUE** <expression logique vraie> **FAIRE**

traitement {suite d'instructions}

relance

**FINTANTQUE**



# Algorithme :: Boucle FAIRE ... JUSQU'A (ici TANTQUE)

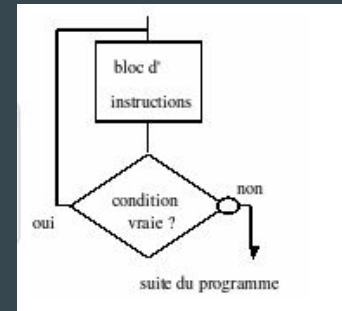
**FAIRE** cpt < 10

AFFICHER ("Donnez une valeur positive non nulle :")

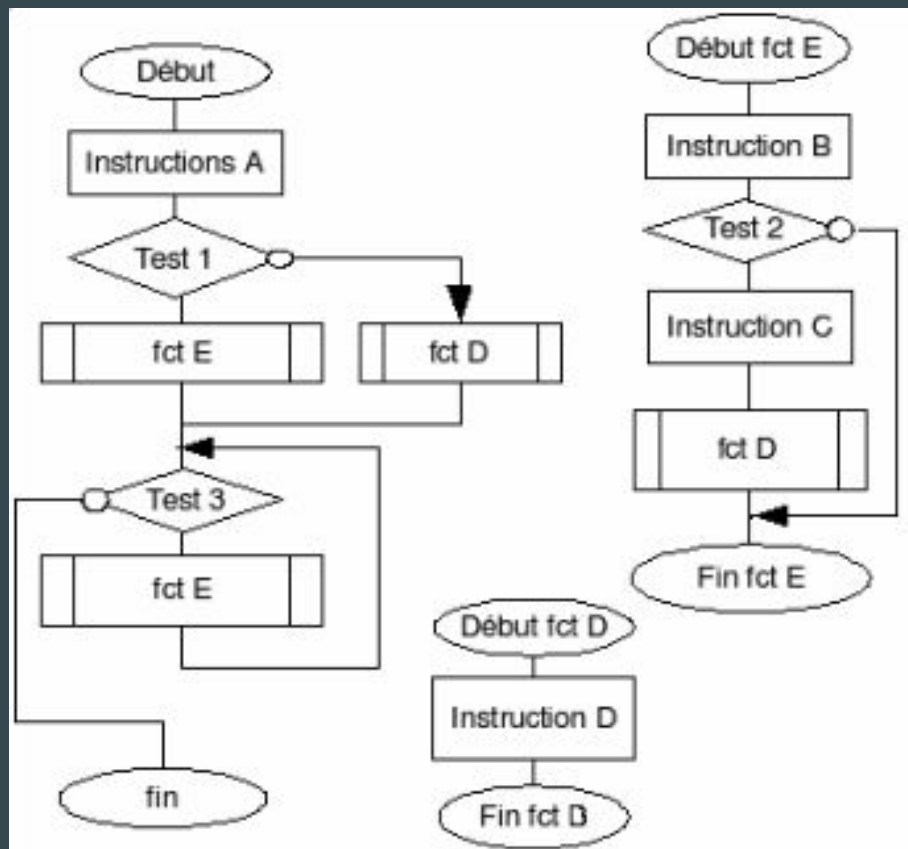
SAISIR(valeur)

**TANTQUE** valeur >= 0

AFFICHER ("La valeur positive non nulle que vous avez saisie est ", valeur)



# Algorithme :: Fonctions



# Algorithme :: Structure de données : Tableau

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
C	E	C	I		E	S	T		U	N		T	A	B	L	E	A	U



# Algorithme :: Structure de données : File (FIFO) Remplissage

⇒ U N E   F I L E

⇒   U N E   F I L E

⇒ T   U N E   F I L E

⇒ S T   U N E   F I L E

⇒ E S T   U N E   F I L E

⇒   E S T   U N E   F I L E

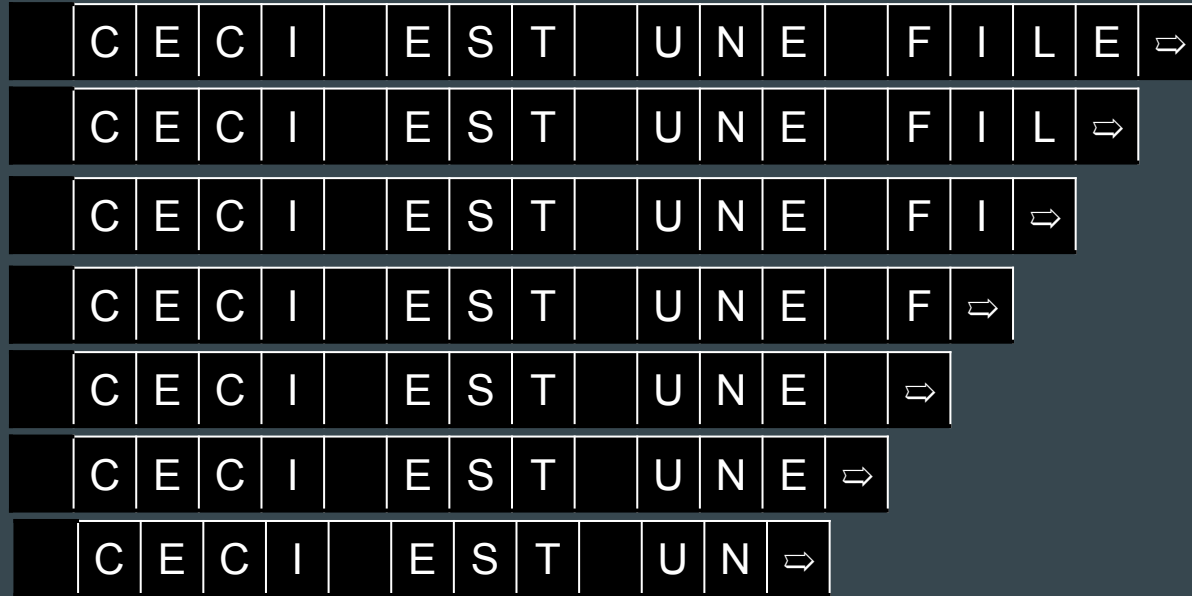
⇒ I   E S T   U N E   F I L E

⇒ C I   E S T   U N E   F I L E

⇒ E C I   E S T   U N E   F I L E

⇒ C E C I   E S T   U N E   F I L E

# Algorithme :: Structure de données : File (FIFO) Vidange



## Algorithme :: Structure de données : Pile (LIFO) Remplissage

[illegible]

## Algorithme :: Structure de données : Pile (LIFO) Videnge

[illegible]

# Algorithme: Exercice Présentiel ?

- ALGO : COPIER / COLLER

# Algorithme: Exercice COPIER / COLLER

Algorithme COPIER

{ Cet algorithme Copie et retourne la sous-partie (entre début et fin) du tableau passé en paramètre }

VARIABLE debut, fin : numérique {paramètres}

VARIABLE chaine : tableau {paramètre}

**DEBUT**

{ C'est ici que ca commence }

**FIN**

# Algorithme: Exercice COPIER / COLLER

Algorithme COLLER

{ Cet algorithme retourne le résultat de l'insert du contenu du tableau chaineAColler dans le tableau chaine, et à la position demandé }

VARIABLE position : numérique {paramètre}

VARIABLE chaine, chaineAColler : tableau {paramètres}

**DEBUT**

{ C'est ici que ca commence }

**FIN**

# Algorithme: Exercice COPIER / COLLER en php

```
<?php

/**
 * Fonction Copier
 * @param : array $source : chaine de caractère source
 * @param : int $debut : position de début de copier coller
 * @param : int $nb : nombre de caractères a copier
 * @return tableau de résultat
 */
function copier($source, $debut, $fin){
    // C'est ici que l'on copie
}

/**
 * Fonction Coller
 * @param : array $source : chaine de caractère source
 * @param : array $pressePapier : chaine de caractère a coller
 * @param : int $position : position a laquelle on "insert" le presse papier
 * @return tableau de résultat
 */
function coller($source, $pressePapier, $position){
    // C'est ici que l'on colle
}

$chaine = [
    "c","e","c","i"," ", "e","s","t"," ", "u","n","e"," ", "c","h","a","i","n","e"," ", "d","e"," ", "c","a","r","a","c","t","è","r","e","s"
];
$contentuCopie = copier($chaine, 13, 5);
$resultat = coller($chaine, $contentuCopie, 19);
echo "<pre>".print_r($resultat, true)."</pre>";
?>
```