

Nom :

I/ Définition d'un algorithme :

Un algorithme est une suite de phrases relatant les actions et/ou opérations successives dans l'ordre chronologique nécessaire à la compréhension du fonctionnement d'un problème posé (le cahier des charges).

Un algorithme commence toujours par le mot début et se termine toujours par le mot fin.

Chaque action s'exprime au moyen d'un verbe.

Lors de l'écriture de l'algorithme, il faut respecter la mise en page, c'est à dire aligner les différentes actions ou opérations à effectuer dans leur chronologie.

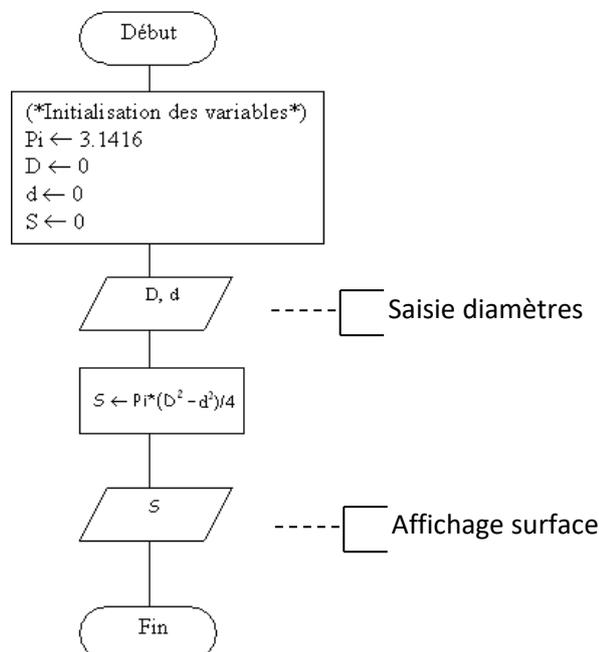
Exemple d'algorithme (calcul de la surface d'un disque) :

```
VARIABLES
  D EST_DU_TYPE NOMBRE
  d EST_DU_TYPE NOMBRE
  S EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  LIRE D
  LIRE d
  S ← PI*(D^2-d^2)/4
  AFFICHER "La surface du disque est:"
  AFFICHER S
FIN_ALGORITHME
```

II/ Définition d'un algorithme :

C'est une représentation graphique de l'algorithme. Pour le construire, on utilise des symboles normalisés.

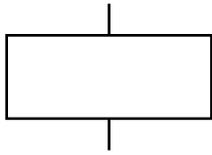
Exemple d'algorithme (calcul de la surface d'un disque) :



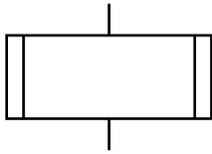
III/ Principaux symboles utilisés dans un algorithme



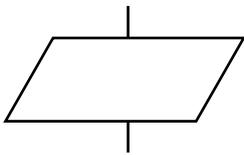
Début, fin, interruption.



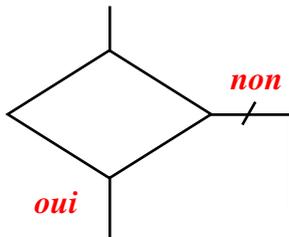
Symbole général : opération ou groupe d'opérations sur des données, instructions...



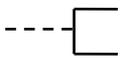
Sous programme (partie de programme considérée comme une simple opération) .



Entrée ou sortie : (afficher un texte, saisir une valeur ...)



Test → question impliquant un choix.



Commentaire.

IV/ Différentes structures

1) Structure linéaire

La structure linéaire est la structure de base de l'algorithmique. Elle se caractérise par une suite d'actions à exécuter successivement dans l'ordre de leur énoncé.

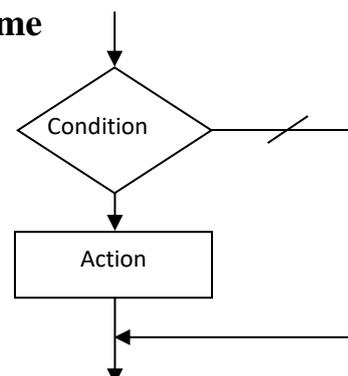
(On passe d'une ligne à la suivante.)

2) Structure conditionnelle.

Algorithme

Si condition
| Alors Action
Fin de si

Algorithme



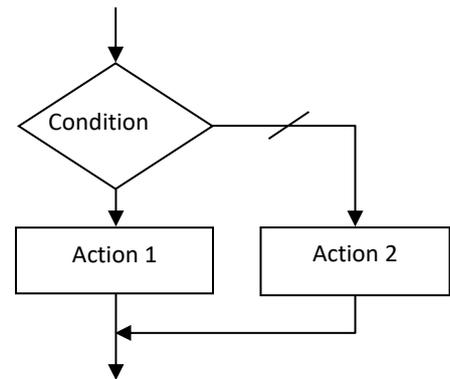
3) Structure alternative

Une structure alternative offre un choix, une condition conduit à telle ou telle réaction du système.

Algorithme

Si condition
| Alors Action 1
| Sinon Action 2
Fsi

Algorithme



4) Structure alternative multiple (suivant cas)

Une donnée est comparée successivement à des valeurs constantes :

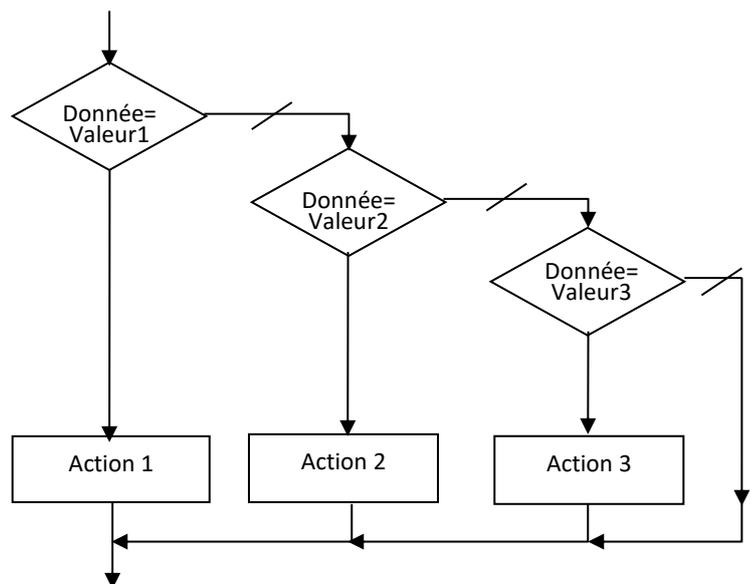
Cas ou donnée vaut

Valeur 1 : Action 1

Valeur 2 : Action 2

Valeur 3 : Action 3

fincas



Les cas permettent d'éviter un grand nombre d'imbrications

5) Structures itératives.

Une structure itérative, aussi appelée structure répétitive, permet de répéter une action jusqu'à ce qu'une condition soit validée.

La boucle « Tant que ... Fin de tant que ».

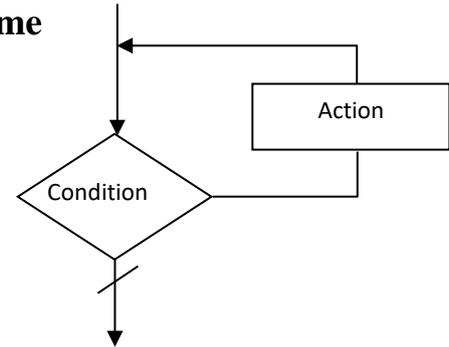
Algorithme

Tant que Condition vraie
 | Action
Fin de tant que

L'action n'est pas forcément exécutée.

Le nombre d'exécutions de la boucle est inconnu et variable.

Algorithme



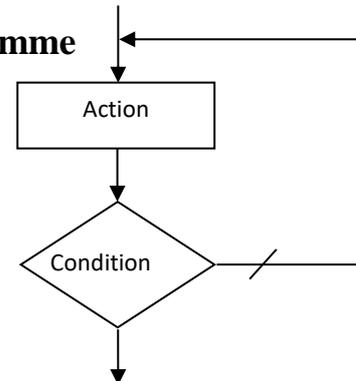
La boucle « Répéter ... Jusqu'à ».

Algorithme

Répéter
 | Action
Jusqu'à Condition vraie

Dans cette structure, le traitement est exécuté une première fois puis sa répétition se poursuit jusqu'à ce que la condition soit vraie.

Algorithme



Boucle « pour » (Nombre d'itérations connu)

Algorithme

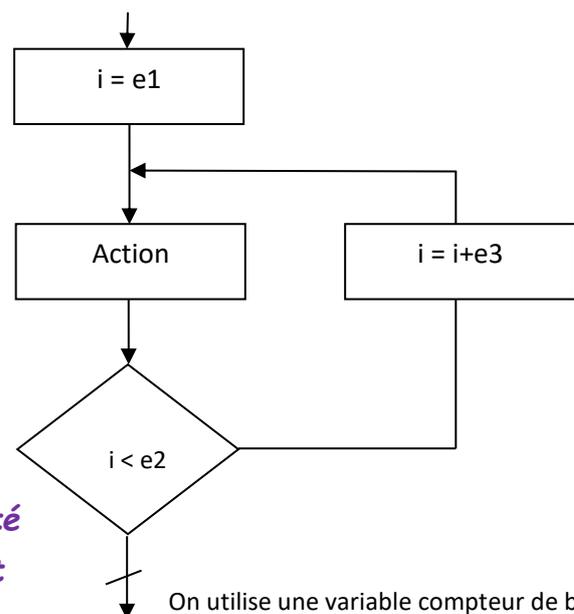
Pour (i de e1 à e2 par e3) faire
 Action
Fin pour

e1, e2, e3 : paramètres de boucle
e1 : valeur initiale
e2 : valeur finale
e3 : pas.

*Dans cette structure, le traitement est exécuté un nombre déterminé de fois, au départ i vaut e1, à chaque boucle on ajoute e3 à i, la boucle se termine lorsque e1 + (i - 1) * e3 > e2*

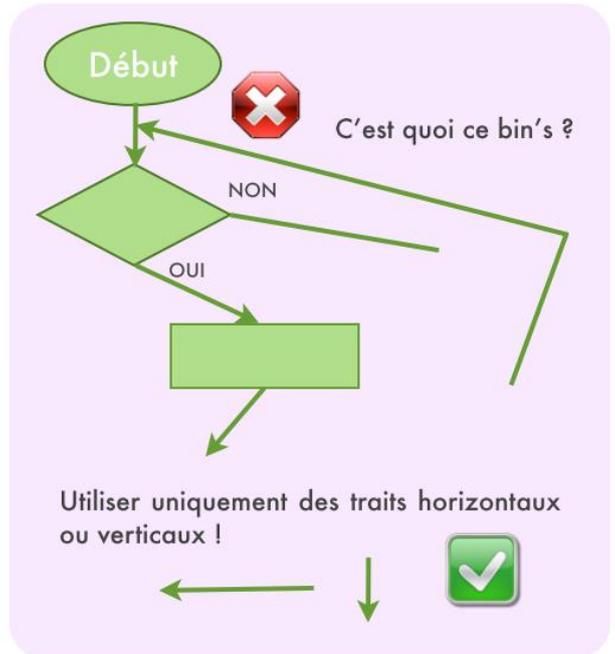
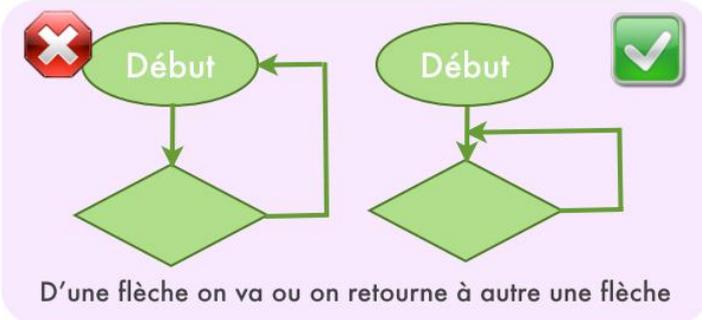
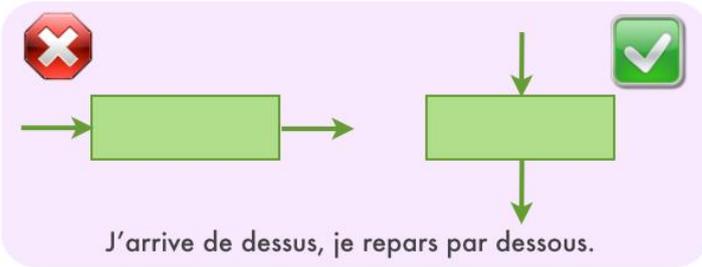
Principales règles à respecter

Algorithme

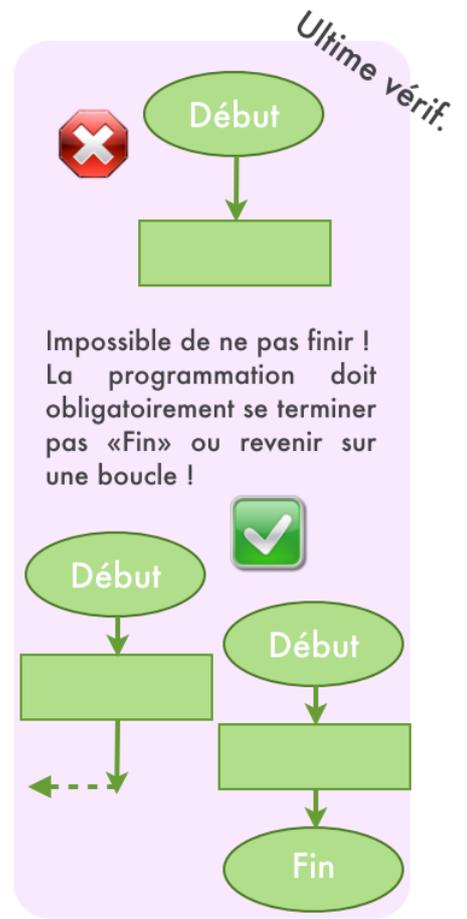
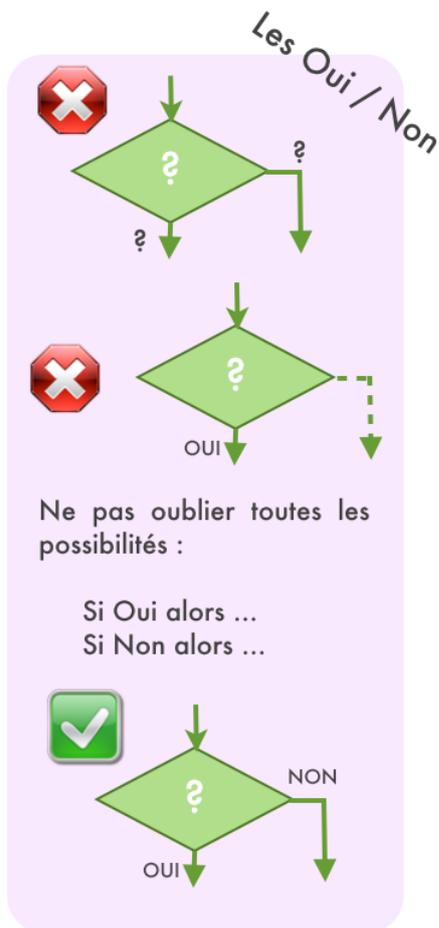


On utilise une variable compteur de boucle : « i »

Attention aux erreurs !



A vérifier à chaque fois !



VI/ Exercices :