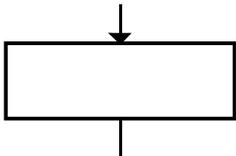
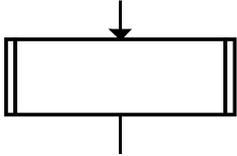
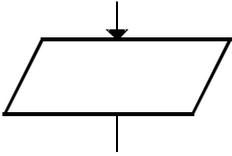
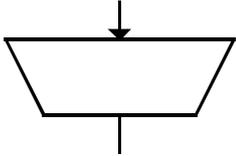
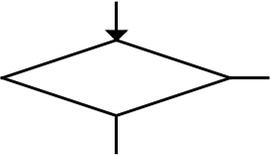
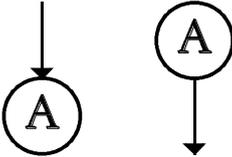


Cours C03_Réaliser un ALGORITHME – ALGORIGRAMME pour assurer la sécurité des biens et des personnes

1- Les définitions

Algorithme : c'est un ensemble de règles opératoires rigoureuses, ordonnant à un processeur d'exécuter dans un ordre déterminé un nombre fini d'opérations élémentaires ; il oblige à une programmation structurée.

Algorigamme : c'est une représentation graphique de l'algorithme. Pour le construire, on utilise des symboles normalisés. Ci-dessous, quelques exemples.

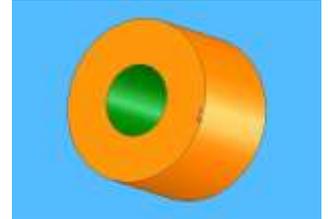
| | | |
|---|---|--|
|  <p>Début ou fin</p> |  <p>Liaison orientée</p> |  <p>Traitement</p> |
|  <p>Processus prédéterminé (sous-programme)</p> |  <p>Entrée / Sortie</p> |  <p>Opération manuelle</p> |
|  <p>Branchement conditionnel</p> |  <p>Renvoi</p> |  <p>Commentaire</p> |

2- Exemple

Enoncé du problème : une machine découpe des disques circulaires de diamètre D , percés d'un trou de diamètre $d < D$ dans une plaque. Des palpeurs mesurent D et d et transmettent ces informations à un calculateur. Celui-ci doit déterminer la surface S du disque et l'afficher.

Phase 1 : analyse du problème

- 1- Préparation du traitement : saisie des diamètres D et d.
- 2- Traitement : calcul de la surface du disque $S = \pi(D^2-d^2)/4$.
- 3- Edition des résultats : Affichage de la surface du disque S.



Phase 2 : expression du problème en langage algorithmique

Calcul_Surface_disque;
 (* Diamètres D et d existent *)
 (* Affichage de la surface du disque *)

Constantes Pi = 3,1416;
Variables D,d : nombres réels; (* données *)
 S : nombre réel; (*résultat*)

Début
 Lire (D,d);
 $S \leftarrow \text{Pi} * (D^2 - d^2) / 4$;
 Ecrire (La surface du disque est : S);
fin.

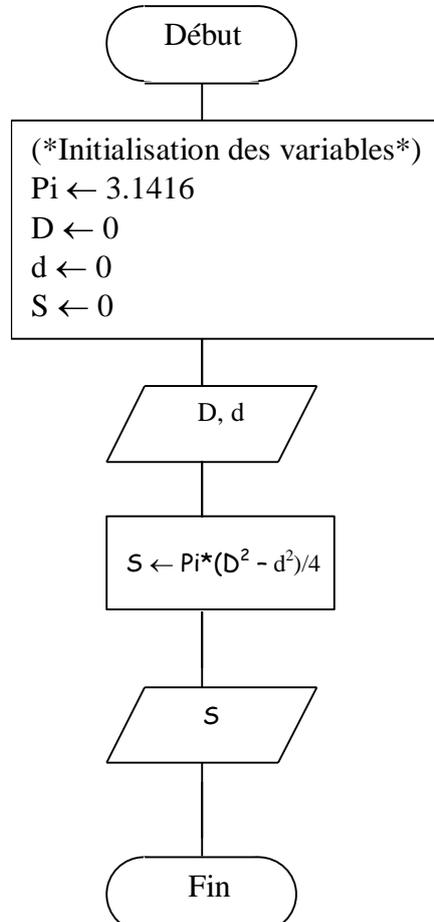
En-tête

Partie déclarative

Partie exécutive
 Préparation du traitement
 Traitement algorithmique
 Edition des résultats

Phase 3 : Traduction en langage C (voir cours suivant)

Le problème précédent peut également se traduire en algorithme, cela donnerait :



3- Organisation d'un algorithme et structure algorithme correspondante

L'en-tête : dans cette partie le concepteur donne un nom à l'algorithme, définit le traitement effectué et les données auxquelles il se rapporte.

La partie déclarative : dans cette partie, le concepteur décrit les différents « objets » que l'algorithme utilise. On y retrouve les constantes et les variables.

Les constantes

Ce sont des « objets » constants dans tout l'algorithme.

Déclaration : identificateur = valeur ;

Exemple : Constantes $\pi = 3,1416$;

Les variables

Ce sont des « objets » dont la valeur peut changer au cours de l'exécution de l'algorithme.

Déclaration : identificateur : type ;

Exemple : R, r : nombres réels;

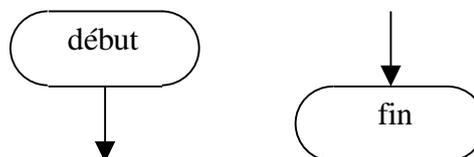
La partie exécutive : elle délimitée par les mots

début

...

...

fin

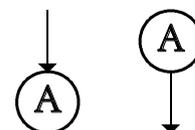


Les commentaires : des commentaires **doivent** être insérés dans le programme afin d'en faciliter la relecture.

(* données *)



Le renvoi : symbole utilisé deux fois pour assurer la continuité lorsqu'une partie de ligne de liaison n'est pas représentée.

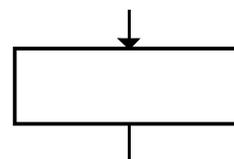


4- Les actions de base

L'affectation : c'est l'action essentielle de l'algorithme. Elle attribue une valeur (constante) à une variable. On notera cette action par le symbole \leftarrow .

identificateur \leftarrow valeur

Exemple : valeur_initiale \leftarrow 10

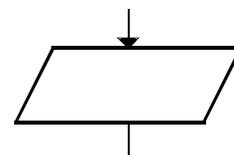


L'affectation n'a de sens que si les deux objets de part et d'autre du signe \leftarrow sont de même type.

La saisie (LIRE) ou l'écriture (ECRIRE) d'une valeur :

LIRE();

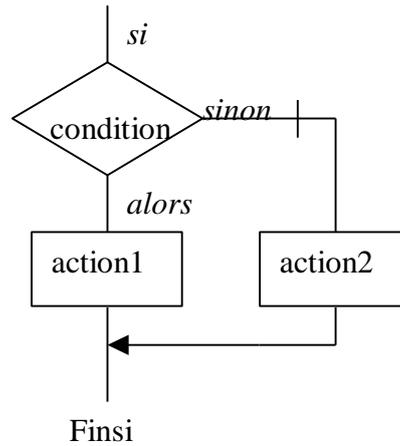
ECRIRE ();



5- Les structures alternatives

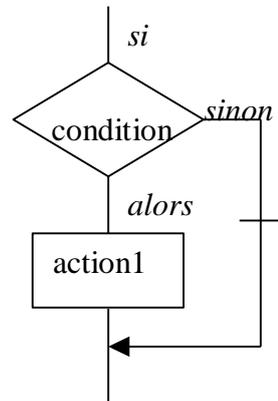
La structure alternative de base :

```
Si condition
  ALORS action1
  SINON action2
Fin si
```



Remarque : si l'action2 n'existe pas, on obtiendra :

```
Si condition
  ALORS action1
  SINON rien
Fin si
```

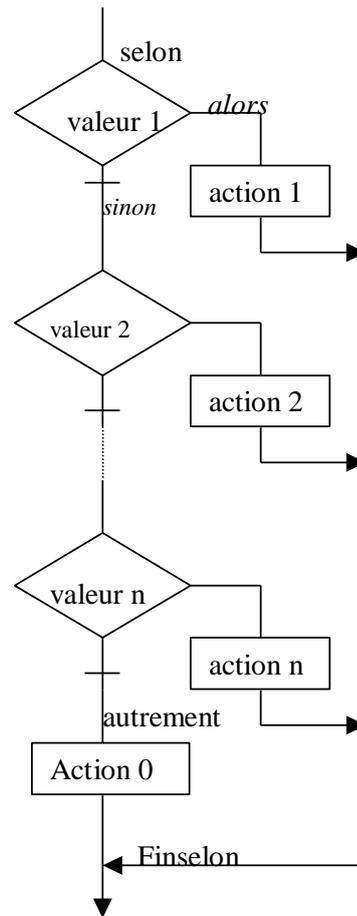


Les structures alternatives imbriquées :

```
Si condition1
  ALORS
    Si condition2
      ALORS action1
      SINON action2
    Fin si
  SINON
    Si condition3
      ALORS action3
      SINON action4
    Fin si
  Fin si
```

La structure de choix multiple :

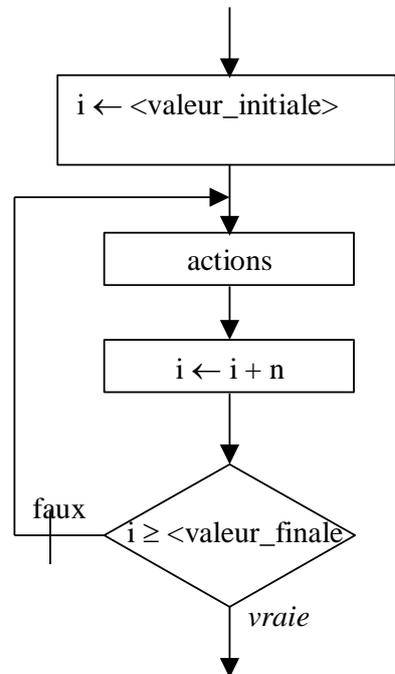
SELON expression
valeur1 : action1 ; **fin selon** ;
valeur2 : action2 ; **fin selon** ;
...
valeurn : actionn ; **fin selon**



6- Les structures itératives ou répétitives (elles permettent d'exécuter plusieurs fois une séquence d'instructions)

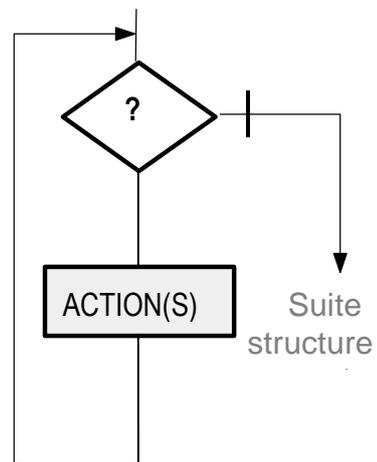
La boucle POUR (boucle de comptage) :

Initialiser i
POUR i variant de valeur_initiale jusqu'à valeur finale par pas de n **FAIRE**
 actions...
Fin Pour



La boucle TANT QUE ... FAIRE (test au début) :

TANT QUE condition **FAIRE**
 actions...
Fin TANT QUE



La boucle FAIRE ... TANT QUE (test à la fin) :

FAIRE
 actions..
TANT QUE condition

