Algo1

Structure d’un Algorithme

Structure d'un algorithme

L'Algorithme consiste en la décomposition d'un problème donné en sous problèmes élémentaires non décomposables selon un ordre de réalisation précis et selon une syntaxe et une présentation précise .

1.Formalisme d'un algorithme

L'algorithme est composé de trois parties :

1.1 La partie entete :

elle contient le le mot clé **Algorithme** , suivi du nom qui identifie le problème à résoudre .
ex : Algorithme Somme

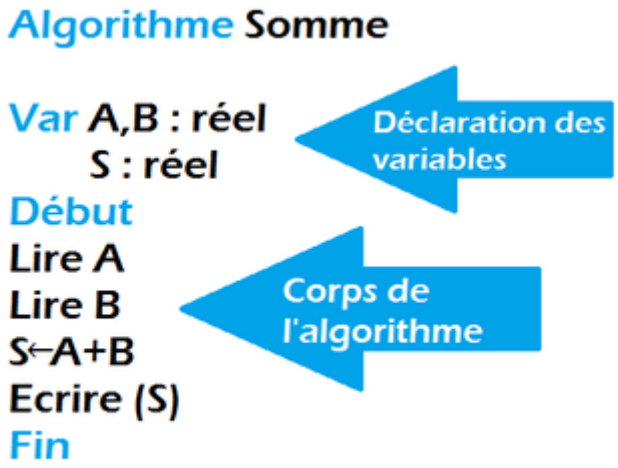
1.2 La partie déclarative :

elle contient toutes les données de l'algorithme , ce sont les **variables d'entrée** , les **variables de sorties** et les **variables intermédiaires** .
On commence par le mot clé **Var** .
ex : Var x , y : entier
c : caractère

1.3 La partie corps de l'Algorithme :

C'est la partie qui contient les traitements , ce sont toutes les instructions que doit exécuter la machine pour arriver au résultat .
Le corps de l'algorithme commence par le mot "Début" et se termine par "Fin" .

et voilà un exemple d'algorithme :



2. Eléments d'un algorithme

2.1 Le type

Le type d'un objet caractérise les valeurs que peut prendre cet objet , on distingue les types simples et les types structurés .
Un objet de type simple contient une seule information tandis qu'un objet structuré contient une collection d'information .
Nom_simple : Type_variable
ex: x,y : réel
Tab=Tableau[1..10] : entier

2.1.1 Les types simples

Ce sont les types numériques (réel,entier) ou logiques (booléen [vrai ou faux]) et le type caractère .
Les opérations définis sur les variables numériques sont les opérations arithmétiques **+** , **-** , **x** , **/** .
+ addition
- soustraction
x multiplication
/ division

et les opérations de comparaison :
< inférieur
> supérieur
= égal
<> différent
≤ inférieur ou égal
≥ supérieur ou égal

Les opérations prédéfinis sur le type caractère sont les opérations de comparaison **> , < , = ...**

Un objet de type logique (booléen) peut prendre 2 valeurs **VRAI** ou **FAUX**
Les opérations prédéfinis sur le type booléen sont **ET , OU , <> , =** .

ex : A et B sont de type booléen
les opérations suivantes sont valides :
A ET B
A OU B
A←VRAI

2.1.2 Les types énumérés

Le type énuméré est défini en donnant la liste de toutes les valeurs possibles que peut prendre un objet de ce type , on utilisera pour ca la formule suivante :
Nom_variable = (valeur1 , valeur2 , ..., valeurN)
ex :
Couleur=(Bleu , Jaune , Noir)
Jour=(Samedi , Dimanche , Mercredi)

2.1.3 Les types structurés

On distingue plusieurs types , nous allons les détailler plutard , dont nous citons :

- Les tableaux (les vecteurs et les matrices)
- Les arbres (piles et files)
- Les chaines de caractères

2.2 Les actions (Traitements)

2.2.1 L'affectation

C'est l'action d'attribuer à une variable une valeur numrique , ou une valeur résultant de l'évaluation d'une expression arithmétique ou logique . La variable doit etre du même type que la valeur .
La syntaxe d'affectation est comme suit :

<identificateur> := <expression> ou bien: **<identificateur> ← <expression>**
ex:

```
VAR X,Y : réel
    Bool : Bouléen
    C : caractère

X←20.25
Y←15.5
Bool←Faux
C←'B'
```

2.2.2 Lecture et écriture

C'est les actions qui expriment la communication entre l'algorithme et l'utilisateur .
Heureusement , Il éxiste des instructions pour permettre à la machine de dialoguer avec l'utilisateur .

Lecture
Elle ordonne à l'ordinateur de de lire ou prendre une donné (valeur) , que l'utilisateur va entrer au clavier et qu'elle va mettre dans sa

memoire .
C'est donc une instruction qui permet de stocker les valeurs , tapées au clavier , dans les cases mémoires corréspondants aux variables .
La syntaxe est donnée par :
LIRE <variable>
ex:
Var X,Y : entier
C : car
LIRE X,Y : Lecture des valeurs nmériques corréspondant aux variables , entrés par l'utilisateur.
LIRE C

Ecriture

Elle ordonne à la machine d'écrire ou d'afficher à l'écran le contenu d'une variable ou d'un méssage .
La syntaxe est comme suit :

Ecrire (<VARIABLE>)
Ecrire (méssage)

ex:
Algorithme Somme
Var S,A,B : réel
Début
Ecrire ("Saisir la valeur des deux nombres à sommer") ; *Le texte est toujours entre ""
Lire A,B ;
S←A+B ;
Ecrire("La somme est",S) ;
Fin

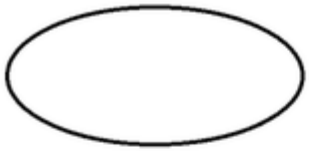
3. Déroulement d'un algorithme

Le déroulement d'un algorithme s'éffectue en appliquant des valeurs pour les variables , puis vérifier si l'on a le bon résutat qu'on éspère

avoir par l'algorithme , cela se fait en dréssant une table et présenter les variables d'entrée et celle de sortie

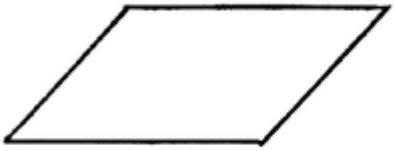
4. L'organigramme

L'organigramme utilise des symbles graphiques normalisés pour représenter le déroulement du traitement d'un problème , c'est donc une représentation graphique de l'enchainement d'une suite d'actions .
Quatres symboles sont utilisés :



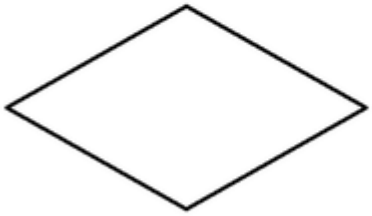
Ellipse

C'est pour marquer le début et la fin d'un organigramme



Parallélogramme

Sert à représenter les instructions d'entrée (lecture) et de sortie (écriture)



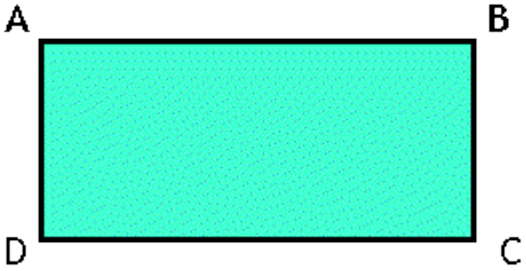
Losange

Pour la représentation des instructions conditionnelles

Flèche qui pointe la séquence d'instruction suivante

Exercice :

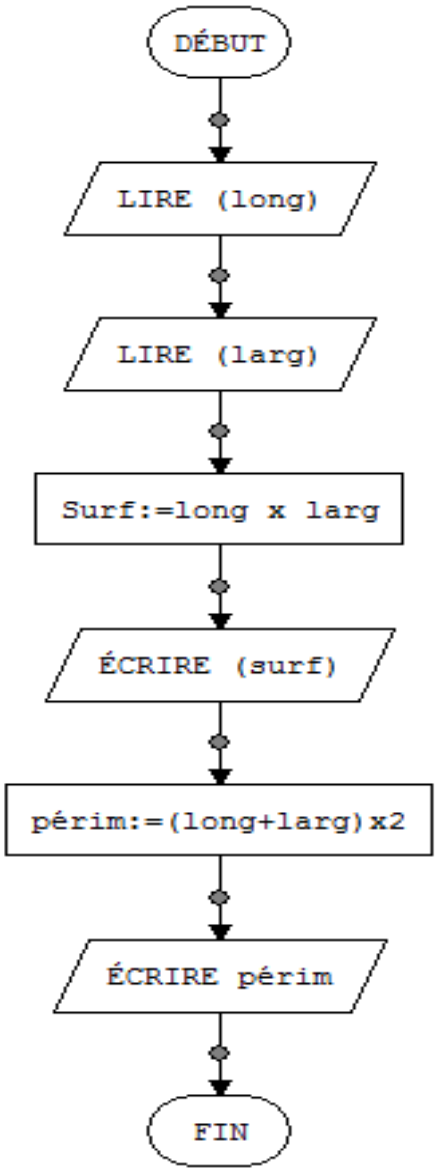
Soit un réctangle ABCD
Ecrire un Algorithme qui permet de calculer ma surface et le périmètre de ce réctangle .



Sollution :

Surface = Long x Larg
Périmètre = (Long + Larg)x2

Organigramme :



Algorithme :

Algorithme Surf_Périm_réctangle
Var long ,larg, périm, surf : réel

Début
Ecrire ("Donner la longueur");
Lire (long) ;
Ecrire ("Donner la largeur");
Lire (larg) ;
Surf←long x larg ;
Ecrire (surf) ;
Périm←(long+larg)x2 ;
Ecrire (Périm) ;
Fin.