Notion d'algorithme et de programme.

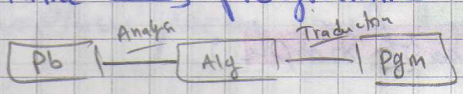
2.1 Concept d'un algorithme.

Le mot "Algorithme" est inventé par le mathématicien "AL-KHAWARISMI" (d'origine iranienne et de confession musulmane).

→ Un algorithme est un ensemble d'instructions séquentielles et logiquement ordonnées, permettant de transformer des données en entrée en données de sortie, afin de résoudre un problème.

2.2 La démarche et analyse d'un problème.

Problème → Modèle → Algorithme → programme → Résultats



2.3.1 Notion d'identificateur

Un identificateur est une chaine de caractères contenant uniquement des caractères alphanumériques (A→Z) et numériques [0-9] et tiré 8 '-8' et qui doit commencer soit par une lettre alphabétique ou "_".

2.3.2 Constantes et variables

Cste : un objet contenant une valeur qui ne peut jamais être modifiée "P=3.14"

variable : un objet contenant une valeur pouvant être modifiée.

2.3.3 Type de données

- Entier (-4, 2 ...)
- Réels (1,2 ; -1,3 ...)
- Caractères
- chaine de caractères ('Hello')
- Booléens (logique) (True - false)

Remarque : On écrit les commentaires entre les accolades { }.

2.4 Structure d'un algorithme / programme

Un algorithme est constitué de trois parties : ~~Entête~~

1. Entête ~~Déclarations~~ ~~Corps~~ : on déclare le nom de l'algorithme
2. Déclarations : → variable et constantes on déclare les données d'entrées et de sorties.
3. Corps : → Entrée, Traitement, Sorties est constitué d'un ensemble d'actions / instructions séquentiellement et logiquement ordonnées. (lecture - l'écriture - affectation - Structure de données)

Remarque : Dans le langage PASCAL, chaque instruction se termine par un point-virgule. Sauf à la fin du programme, on met un point.

(2) Type d'instructions

- les E/S , - l'affectation, - les structures de contrôle

① Entrées (lecture) : une instruction d'entrée nous permet dans un programme de donner un valeur quelconque à une variable.

② Sorties (écriture) : nous permet dans un programme d'afficher un résultat ou bien un message.

③ Instruction d'affectation : une affectation consiste à donner une valeur à une variable.

④ Structure de contrôle : - structures de cont conditionnelles
- " " répétitives (itératives)

① a- Test alternatif simple :

Si <condition> alors
<instructions>
Fin si
	if <condition> Then
	begin
	<instructions>;
	end;

(3) Remarque : Ds le langage pascale, un bloc est délimité par les deux mots clés begin et end

① b- Test alternatif double :

Si <condition> alors
<instruction 1>
Sinon
<instruction 2>
Fin si
	if <condition> Then
	begin
	<instruction 1>;
	end
	else
	begin
	<instruction 2>;
	end;

Remarque : - Dans le langage Pascal, il faut jamais mettre de ; avant else.
- On peut enlever begin end du if et ceux de else s'il y a une seule instruction dans les deux blocs.

② Structures de contrôle répétitives :

② a - Boucle Pour (For)

pour <indice> ← <vi> à <vf> faire
<instructions>
Fin pour
	for <indice> ← <vi> to <vf> do
	begin
	<instructions>;
	end;

Remarque : - Si le bloc contient une seule instruction, le begin et end sont facultatifs.

- Il ne faut pas jamais mettre de ; après le mot clé do

b. Boucle Tant-que (while)

tant-que <condition> faire
<instructions>
Fin Tantque
	while <condition> do
	begin
	<instructions>
	end ;

Remarque : comme la bouc for, il faut jamais mettre ; après do

- toute boucle pour peut être remplacée par une boucle tantque, cependant l'inverse n'est pas toujours possible.

c. Boucle répéter (Repeat)

répéter <instructions> jusqu'à <condition> ;	repeat <instructions> until <condition> ;

Remarques : - de la boucle repeat on utilise pas begin et end pour délimiter le bloc d'instructions

- La condition de répéter est l'inverse de la condition tant-que : pour répéter c'est la condition de sortie de la boucle, et pour tant que c'est la condition d'entrée

= Dans tant-que on teste la condition avant d'entrer à l'itération, et dans répéter on fait l'itération après on test la condition.

Structure de contrôle de branchements / sauts (l'instruction Goto)

permet de sauter à un endroit du programme et continuer l'exécution à partir de cet endroit.

aller à <num_etiq> ⋮ <num_etiq> :	goto <num_etiq> ; ⋮ <num_etiq> :

Remarque : Dans un programme Pascal, il faut déclarer les étiquettes dans la partie déclaration avec le mot clé label

Remarques importantes :

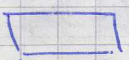
- Langage PASCAL est insensible à la casse (majuscule/minuscule)
- Lorsque l'action après Then, Else ou un Do comporte plusieurs instructions, on doit obligatoirement encadrer ces instructions entre BEGIN et END.
- Un ensemble d'instructions encadrées entre BEGIN et END, s'appelle un BLOC ou action composée

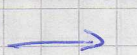
2.7 Représentation en organigramme.

Un organigramme est la représentation graphique de la résolution d'un problème.

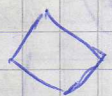
2.7.1 Les symboles d'organigramme.


début et Fin


calculs, traitement


ordre d'exécution


Els : lecture et écriture


Test et décision


Connecteur.

2.8 Les opérateurs :

nous permettent d'écrire des expressions qui seront évaluées par l'ordinateur.

2.8.1 Les
- opérateurs arithmétiques
- opérateurs relationnels
- opérateurs logiques.

2.8.1 Les opérateurs arithmétiques : +, -, x, /

div : (division entière, elle fournit la partie entière d'un division)

mod : (modulo ou reste de division)

2.8.2 Les opérateurs relationnels :

=, <> (différent), <, >, <=, >=

2.8.3 Les opérateurs logiques :

AND, OR et NOT.

2.8.4 Les fonctions

ABS, EXP, LN, LOG, SQRT, SQR, Arctan, Cos, Sin,

Round : retourne la valeur arrondie d'un nombre x

Trunc : retourne la partie entière d'un nombre x

2.8 Calcul d'expressions : priorités dans les opérateurs.

a. Les expressions arithmétiques :

Règle 1 : On évalue d'abord le contenu des parenthèses, en commençant par les parenthèses les plus internes

Règle 2 : On commence à effectuer les opérateurs de plus haute priorité. Dans le cas des opérateurs de même priorité, on commence par le plus à gauche.

b. Règles de priorités des opérateurs :

1- Les parenthèses 2- les fonctions 3- les moins unaire, le NOT

4- *, /, DIV, MOD, AND 5- +, -, OR 6- =, <>, <, >, <=, >=

→ Les opérateurs logiques :

opérande 1	opérande 2	op1 AND op2
True	True	True
True	False	False
False	True	False
False	False	False

op 1	op 2	op1 OR op2
True	True	True
True	False	True
False	True	True
False	False	False

Exemple d'identificateurs :

1E X ; Prix Unitaire ; Hauteur-Mur ; aA ; a?b

ne pas valide ne pas valide ne pas valide n'est pas valide

- Uses : permet d'utiliser une unité PASCAL (ensemble de fonctions prédéfinies). On utilise souvent la bibliothèque wincrt comme suit :

Uses wincrt

- un programme : est le résultat de la traduction d'un algorithme en choisissant un langage de programmation
- Le langage Pascal est un langage compilé : un code source, respectant la syntaxe du Pascal, écrit par un utilisateur (programmeur) est traduit à un code binaire exécutable par la machine.
- Le langage Pascal possède des mots clés (mots réservés) : Program, Uses, Var, Const, begin, end, ... ces mots ne peuvent pas être utilisés comme identificateur