LES VARIABLES, LES OPÉRATEURS ET LES OPÉRATIONS

.....< + - * / ^ >.....

VARIABLE

DÉFINITION 3:

Une variable est un espace mémoire nomme, de taille fixée prenant au cours du déroulement de l'algorithme un nombre indéfini de valeurs différentes.

- Un algorithme tourne généralement autours des variables;
- Le changement de valeur se fait par l'opération d'affectation.
- La variable diffère de la notion de constante qui, comme son nom l'indique, ne prend qu'une unique valeur au cours de l'exécution de l'algorithme.

VARIABLE

TYPE DE VARIABLE :

Dans la plupart des langages de programmation, avant de manipuler une variable, il faut préalablement déclarer son type. C'est à dire que la variable en question ne pourra changer de valeur que dans l'intervalle défini par le type qui lui est assigné.

Dans un algorithme, on se contente de 5 types de base, à savoir :

- Les entiers: qui sont des nombres sans virgule et qui peuvent être positifs ou négatifs. On parle alors de nombres entiers signés ;
- Les réels: qui sont des nombres avec virgule (dite virgule flottante) et qui peuvent être positifs ou négatifs aussi ;
- Les booléens: qui définissent deux valeurs (dites binaires) qui sont Vrai ou Faux (ou encore 1 ou 0) ;
- Les caractères: qui représentent tous les caractères alphanumériques ;
- Les chaînes de caractères: qui représentent des textes constitués de tout type de caractères comme les caractères alphabétique, numériques et symboles.

OPERATEURS

TYPES D'OPÉRATEURS :

Un opérateur est un outil qui permet d'agir sur une variable ou d'effectuer des calculs.

Il existe plusieurs types d'opérateurs:

- L'affectation: qui confère une valeur à une variable ou à une constante. Il est représenté par le symbole «←» (= en C/C++ …) ;
- Les opérateurs arithmétiques : qui permettent d'effectuer des opérations arithmétiques entre opérandes numériques :

+	addition
-	soustraction
*	multiplication
/	division
mod (% en C/C++)	modulo
^	puissance
div	division entière

OPERATEURS

TYPES D'OPÉRATEURS

- Les Opérateurs relationnels :

>	supérieur
<	inférieur
>=	Supérieur ou égal
=<	Inférieur ou égal
= (== en C/C++)	égal
≠ ou < > (!= en C/C++)	différent

- Les opérateurs logiques :
  - Opérateur unaire : «non» « ! en C/C++» (négation) ;
  - Opérateurs binaires : «et» «&& en C/C++» (conjonction), «ou» «| | en C/C++» (disjonction).
- La concaténation : qui permet de créer une chaine de caractères à partir de deux chaînes de caractère en les mettant bout à bout. Il est représenté par le symbole « + ».

- sur les entiers et les réels : addition, soustraction, multiplication, division, division entière, puissance, comparaisons, modulo ;
- sur les booléens : comparaisons, négation, conjonction, disjonction ;
- sur les caractères : comparaisons ;
- sur les chaînes de caractères : comparaisons, concaténation

OPÉRATIONS

PRIORITÉ DES OPÉRATIONS :

Lors de l'évaluation d'une expression, la priorité de chaque opérateur permet de définir l'ordre d'exécution des différentes opérations. Pour changer la priorité d'exécution, on utilise les parenthèses.

- Ordre de priorité décroissante des opérateurs arithmétiques et de concaténation :
  - «^» ;
  - «*», «/» et «div» ;
  - «mod» ;
  - «+» et «-» ;
  - «+» (concaténation).
- Ordre de priorité décroissante des opérateurs logiques :
  - «non» ;
  - «et» ;
  - «ou».

NOTION DE VARIABLE EN C

En C, lorsqu'on crée une variable, il faut toujours indiquer son type. Le tableau ci-dessous, donne les noms des types et leur plage.

type	Min	Max
Signed char	-127	127
int	-32 767	32 767
long	-2 147 483 647	2 147 483 647
float	$-1x10^{37}$	$1x10^{37}$
double	$-1x10^{37}$	$1x10^{37}$

Ces types sont signés (signed) mais il existe d'autre types non signés (unsigned) qui ne stockent que des nombres positifs.

- **Exemple** : signed int peut aller jusqu'à 32 767 alors que unsigned int va jusqu'à 65 535.

NOTION DE VARIABLE EN C

- type *nom_variable* permet de déclarer une variable. Mais le nommage des variables est régit par les règles suivantes :
  - elle commence par une lettre ;
  - les espaces sont interdits. On peut utiliser « underscore » pour cela ;
  - on ne peut utiliser des accents ;
  - on peut utiliser des minuscules, des majuscules et des chiffres.
- Pour déclarer une constante, il faut utiliser le mot const devant le type et il est obligatoire de lui donner une valeur au moment de sa déclaration.
- Pour afficher le contenu d'une variable avec printf, on utilise le format du type de la variable (ex : **%d** pour les **int** ou **%f** pour les **float**) à l'endroit ou l'on souhaite afficher la valeur de la variable.
- Il est possible d'afficher la valeur de plusieurs variables dans un seul printf. Il vous suffit pour cela d'indiquer des %d ou des %f là où vous voulez, puis d'indiquer les variables correspondantes dans le même ordre, séparées par des virgules.

NOTION DE VARIABLE EN C++

En C++, c'est presque pareil, sauf que coté initialisation d'une variable, on peut décider de faire comme le C ou avec une syntaxe propre à C++ qui est : TYPE NOM (VALEUR). Aussi, pour l'affichage du contenu d'une variable avec cout et les chevrons (<<), il suffit simplement de mettre le nom de la variable à l'endroit du texte à afficher.

- Opérateur ternaire: Il permet l'affectations du type.
  - Syntaxe: Si *condition* est vraie alors *variable* vaut *valeur*, sinon *variable* vaut *autre valeur*.
  - Exemple: int a = (b > 0) ? 10 :20;
- Les opérateurs de manipulation de bit :
  - & : ET bit à bit
  - | : OU bit à bit
  - ^ : OU Exclusif bit à bit
  - << : Décalage à gauche
  - >> : Décalage à droite
  - ~ : Complément à un (bit à bit)
- La fonction *sizeof* est utilisée pour connaitre la taille en mémoire d'une variable passé en paramètre.
  - int a = 1; sizeof(a) donne 4; double a = 3,14; sizeof(a) donne 8.

EXERCICES D'APPLICATIONS

Application 1 :

Ecrire un algorithme/programme permettant de déclarer deux variables de type réel, de saisir les valeurs, de calculer et d'afficher leur somme, produit et moyenne.

Application 2 :

Ecrire un algorithme/programme qui permet de permuter les valeurs de A et B sans utiliser de variable auxiliaire.

Application 3 :

Ecrire un algorithme/programme permettant de déclarer trois variables A, B, C de type réel, d'initialiser leurs valeurs et ensuite d'effectuer la permutation circulaire des trois variables.

Application 4 :

Ecrire un algorithme/programme qui permet de saisir les paramètres d'une équation du second degré et de calculer son discriminant delta.

Application 5 :

Ecrire un algorithme/programme qui à partir de la valeur saisie du côté d'un carré donné, permet de calculer son périmètre et sa surface et affiche les résultats à l'écran.