Fiche de révision : Programmation leçon 1

Inclure une bibliothèque

- Il est possible d'inclure dans un programme des bibliothèques avec des fonctions prédéfinies. "*stdio.h*" permettant des fonctions d'entrée (scanf...) et de sortie (printf...).
- Il en existe d'autre ("math.h","string.h",sdlib.h)

Règle n°1 :

Une instruction ou une déclaration de variable se termine toujours par **un point virgule (;)**

Règle n°2 :

Qu'est ce qu'une instruction ? **2 possibilité** :

- sois une affectation de variable (ex : nom_var= expression)
- sois un appel de fonction : (ex : nom_fn(param1,param2))

Règle n°3 :

Un commentaire est toujours entouré de : /*...*/ ou commence par //

Règle n°5 :

- Si la valeur peut changer au cours du programme, on utilise **une variable**
- Si la valeur est fixe au cours du programme, on utilise **une constante nommée**

Règle n°5 :

Un identifiant d'une information peut contenir lettre,chiffre ou espaces soulignés(_). Tout autre caractère est interdit, y compris les lettres accentuées.

Les différents types de bases :

Les différentes types de variables et la manière de les définir :

- *char c* : la variable c contient **un caractère**
- *int i* : la variable i contient **un nombre entier**
- *float x* : la variable x contient **un nombre réel**, codé sur 32 bits
- *double y* : la variable y contient **un nombre réel**, codé sur 64 bits
- *char texte [40]* : la variable texte contient **une chaîne de 40 caractère**

Point n°1

Le type booléen n'existe pas en algo C, donc on remplace par un entier :

- la valeur 0 représente **Faux**
- Toute autre valeur (inclue 1) représente **Vrai**.

Point n°2

- Les entier (int) sont des nombres **positifs ou négatifs** : 13,-87,0
- Les réels (float ou double) sont des nombres positifs ou négatifs à virgule ; mais attention, la virgule est représentée **par un point,** notation anglo-saxonne ; exemples : 12.75, -0.04, 131., .7777, -412

Point n°3

- Un caractère est noté **entre apostrophes** : 'c' 'C' '6' ' ' (espace)
- Une chaînes de caractères (char []) est notée **entre guillemets**: "du texte",
- '\n' représente **un changement de ligne**
- '\t' représente **une tabulation**, c'est-à-dire un saut vers une colonne plus à droite (8 caractères)
- '\0' représente **le marqueur de fin** d'une chaîne de caractères

Point n°4

- Un caractère est noté **entre apostrophes** : 'c' 'C' '6' ' ' (espace)
- Une chaînes de caractères (char []) est notée **entre guillemets**: "du texte",
- '\n' représente **un changement de ligne**
- '\t' représente **une tabulation**, c'est-à-dire un saut vers une colonne plus à droite (8 caractères)
- '\0' représente **le marqueur de fin** d'une chaîne de caractères

Point n°5

Toute expression arithmétique entre valeurs **entières** donne pour résultat **un entier** :

- note2 = 16/3 ; la variable note2 contient alors 5, la valeur arrondie de 5.333... car division entre **entiers**
- note2 = 16/3.0 ; // la variable note2 contient alors 5.333... car **le dénominateur est un réel**

Affichage (sorties) :

Si le message à afficher doit contenir des informations stockées dans des variables, il faut alors ajouter **des codes** dans la chaîne de caractères pour préciser l'emplacement de chacune de ces informations :

Code	Type
%c	caractère (char)
%d	entier (int)
%f	réel (float)
%lf	réel long (double)
%s	chaînes de caractères (char [])

Exemple :

printf("Mme %s a %d ans.\n", nom, age)

affiche le texte suivant si "nom" contient "Martin" et age contient 29.
On obtient donc : **MMe Martin a 29 ans**

Les codes peuvent également être enrichis :

code enrichi	type
%nb1c	caractère (*char*)
%nb1d	entier (*int*)
%nb1s	chaînes de caractères (*char []*)
%nb1f	réel (*float*)
%nb1.nb2f	réel (*float*)
%nb1lf	réel long (*double*)
%nb1.nb2lf	réel long (*double*)

Exemple :

printf("Le %3de étudiant a obtenu la note de %5.3f/20\n", rang, note) ;
affiche le texte suivant si la variable entière rang contient 4 et la variable réelle note contient 12.75. On obtient donc :
Le 4e étudiant a obtenu la note de 12.750/20

Lecture (entrée) :

- La fonction "scanf" est une fonction qui permet de lire une information saisie au clavier par **l'utilisateur** et de la stocker dans une variable. Elle a besoin de 2 paramètre : **Le code** du type de l'information à lire et **l'adresse** en mémoire dans laquelle stocker l'information. Pour indiquer l'adresse il faut rajouter "&" devant le nom de la variable.
- Ex : scanf("%c", &lettre) : si lettre est une variable de type char
  scanf("%d", &nb_etu) : si nb_etu est une variable de type int

Fiche faite par Boukhalfa Ayoub