MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DE SOUSSE

Ecole Polytechnique de Sousse
Département Informatique

COURS D'ATELIER DE PROGRAMMATION 1
Section : 1ère Année Licence GL

ENSEIGNANTS :
Taoufik SAKKA ROUIS & Kaies BEN SALAH

---Année Universitaire 2022-2023---

https://github.com/srtaoufik/Atelier-Prog-C

PLAN
Semestre 1

CH1 : Introduction à la programmation C

CH2 : Les structures de contrôle conditionnelles

CH3 : Les structures de contrôle itératives

CH4 : Les tableaux à une et deux dimensions

CH5 : Les sous-programmes

CH6 : Les chaines de caractères

CH1
Introduction au langage C

1ère Année Licence GL

Kais ben Salah
&
Taoufik Sakka Rouis

Introduction au langage C

1- Généralités sur C

- Le langage C a été créé dans les années 1970 par Dennis Rithcie.
- Un programme en langage C est une suite d'instructions.
- Un programme en C est un ensemble de fonctions, parmi ces fonctions, il existe une fonction de démarrage dont le nom est "main".
- Le langage C possède un peu d'instructions. Il fait par contre appel à des bibliothèques, fournies en plus ou moins grand nombres avec le compilateur.

Introduction au langage C

2-Structure générale d'un programme C

#include……	Inclusion des bibliothèques
void main()	Fonction Principal
{	Début du programme
< Déclarations >;	Déclarations des variables
< Instructions >;	Les instructions du programme
}	Fin du programme

Remarque:

Chaque programme en C doit renfermer une fonction du nom de "main"(Fonction principale de chaque programme C).

Introduction au langage C

3. La directive « include »

- Se trouvent dans un répertoire nommé « include »
- Ils possèdent l'extension h,

Exemple

- #include<stdio.h> : (Fichiers d'entrée, sortie).
- #include<math.h> (Fichiers Mathématiques).
- #include <string.h> (Fonctions sur les chaines de caractères).

Exemple

```
#include<stdio.h>
void main()
{
        printf("Bonjour");
}
```

Introduction au langage C

4. Les Commentaires

Un commentaire est une suite de caractères placés entre les délimiteurs.

Exemple :

/*Les commentaires documentent les programmes */

Ajoutons un commentaire au programme précédant :

```
#include<stdio.h>
void main()
{
        printf("Bonjour");  /* Affichage d'un commentaire*/
}
```

Introduction au langage C

5. Les types de bases

Les différents types de bases sont :

- Les nombres entiers int
- Les nombres flottants float ou double
- Les caractères char

5.1. Le type entier (int)

Ils existent plusieurs types :

Type	Désignation	Occupation mémoire	Plage de valeurs
entier court	short int (short)	2 octets	-32768 à 32767
entier court non signé	unsigned short	2 octets	0 à 65535
Entier (type standard)	int	4 octets	-32768 à 32767
entier non signé	unsigned int	4 octets	0 à 65535
entier long	long int (long)	4 octets	-2 147 483 648 à 2 147 483 647
entier long non signé	unsigned long	4 octets	0 à 4 294 967 295

Introduction au langage C

5.2. Le type flottant (float)

Un nombre réel en C est dit nombre à virgule flottante. C'est un nombre dans lequel la position de la virgule en tant que séparateur entre partie entière et partie décimale n'est pas fixe.

Exemple : Nombre = 14.8

- Représentation 1 : $1.48\ \ 10^{1}$
- Représentation 2 : $0.148\ \ 10^{2}$
- Représentation 3 : $148.0\ \ 10^{-1}$

Type	Désignation	Nombre de chiffres significa-tifs de la mantisse	Occupation mémoire	Plage de valeurs
Réel simple précision	float	6	4 octet	$3.4 * 10^{-38}$ à $3.4 * 10^{38}$
Réel double précision	double	15	8 octet	$1.7*10^{-308}$ à $1.7*10^{308}$
Réel avec précision étendue	long double	19	10 octets	$3.4 * 10^{-4932}$ à$3.4 * 10^{4932}$

Introduction au langage C

5.3. Le type caractère (char)

- Il est utilisé pour représenter un caractère.
- Le type char (du mot anglais character, "caractère") et utilisé pour représenter un caractère, plus précisément la valeur entière d'un élément de l'ensemble des caractères représentables.

→ Ce nombre entier est le code ASCII du caractère.

Exemple :

Le caractère '**A**' est stocké sous forme char. L'ordinateur n'écrit pas le signe A, mais le nombre 65 (code ASCII de 'A') sur 1 octet = 01000001.

Ils existent 2 types différents :

Type	Désignation	Occupation mémoire	Plage de valeurs
caractère	char	1 octet	-128 à 127
caractère non signé	unsigned char	1 octet	0 à 255

Introduction au langage C

6. Les constantes

Les constantes sont utilisées pour calculer des valeurs, pour initialiser des variables, etc.

6.1. Les constantes entières

Exemples : -9, 0, 199

6.2. Les constantes à virgules flottantes

Exemples : 21E-4, -4005E3, 3.141, 314.1E-2

6.3. Les constantes de types caractères

Exemples : 'a', 'A', ' ?', '1'.

- code ASCII('A')=65 code ASCII('a')=97 code ASCII('0')=48

6.4. Les constantes de type chaîne de caractères

- Une chaîne peut ne comporter aucun caractère : "" (appelée la chaîne vide).
- Le compilateur insert automatiquement un caractère nul "\0" à la fin de chaque chaîne de caractères : une chaîne de caractères est ainsi une séquence ordonnée de caractères qui se termine par "\0".

Introduction au langage C

Remarque : Il existe plusieurs types de constantes, dont les plus importants sont les constantes non typées et typées.

a) Les constantes non typées

Avec la commande "#define", il est possible de donner un nom symbolique à une constante (de préférence au début).

Syntaxe : **#define <NomConstante> <Valeur>**

Exemple :

#define MAX 100
#define PI 3.14

b)Les constantes typées

Ces constantes sont déclarées avec le modificateur "const". Ce sont des variables dont la valeur n'est pas modifiable.

Syntaxe : **const <TypeVariable> < NomVariable > = <ValeurVariable>;**

Exemple :

const float e = 2.71 ;
const int MAX=100;

Introduction au langage C

6.5. Les caractères de contrôle

Séquence	Signification
\n	Génère une nouvelle ligne
\t	Pose une tabulation horizontale
\f	Provoque un saut de page
\a	Déclenche un signal sonore
\r	ramène le curseur au début de la ligne courante.

1. Exemple Saut de ligne : printf("Voici **\n** un **\n** programme C.") ;

Résultat :

Voici
un
programme C.

2. Exemple Tabulations : printf("Voici **\t** un **\t** programme C.") ;

Résultat :

Voici un programme C.

3. Exemple Retour-chariot :

printf("Voici un programme C.**\r** Voici un programme C.") ;

Résultat :

Voici un programme C.

Introduction au langage C

7. Les variables

Une variable :

- possède un nom,
- possède un type,
- possède un contenu,
- est modifiable,
- est rangée en mémoire à partir d'une certaine adresse.

Syntaxe :

Type NomVar1, NomVar2....., NomVarN ;

Type NomVar1= val1, NomVar2= Val2....., NomVarN= ValN ;

Exemple :

- char c ;
- int X, Y;
- float Z = 4.5 ;
- double W;
- int t_presse

Introduction au langage C

8. Les Opérateurs

8.1 Les opérateurs arithmétiques

- +, -, / , *
- % Modulo (MOD en algo)

Remarque :

- Si les deux arguments de / sont des entiers alors le résultat est un entier.
- Le résultat de x= 3/2 est 1, même si x est déclaré de type float.
- Le résultat de x= 3.0/2 (ou 3/2.0) est 1.5.
- Pas d'opérateur d'évaluation en puissance, (On utilise la fonction **pow(x, y)** = x^y).

Introduction au langage C

8.2. Les opérateurs de comparaison

- **<, <= , > , >=**
- **==** : égal à, **!=** : diffèrent de.
- Le résultat d'une comparaison est un entier
  - ✓ **0** (an algo : faux) si le résultat est faux
  - ✓ **1** (an algo : Vrai)si le résultat est vrai.
- La comparaison devient une expression de type entier.

8.3. Les opérateurs logiques

- **&&** ⟶ Et(en algorithme)
- **||** ⟶ Ou(en algorithme)
- **!** ⟶ Non(en algorithme)

Exemple:

(a < b)(c < d)

Prend la valeur 1(vrai) si les deux expressions a < b et c < d sont toutes deux vraies et la valeur 0 dans le cas contraire.

Introduction au langage C

8.4. L'opérateur d'affectation =

- Les opérateurs d'affectation mettent dans leur opérande de gauche la valeur de leur opérande de droite.
- L'opérande de gauche est appelé une Lvalue
- Les affectations sont évaluées de la droite vers la gauche.

a) Affectation simple

Exemple 1

```
int x ;
x =1 ;
```

La variable x du membre droit de l'affectation reçoit la valeur 1 .

Exemple 2

```
int x , y ;
x= y= 1 ;
```

Affecte la valeur 1 aux deux variables (x=1 et y=1).

Introduction au langage C

b) Affectation combinée :

Les affectations combinées mélangent une opération d'affectation avec une opération arithmétique ou logique.

Exemple

X+ = Y	⇔	X = X + Y
X− = Y	⇔	X = X − Y
X* = Y	⇔	X = X * Y
X/ = Y	⇔	X = X/Y
X% = Y	⇔	X = X%Y

Introduction au langage C

8.5.Les opérateurs d'incrémentation et de décrémentation

Les opérateurs d'incrémentation et de décrémentation unaires ++ et – augmentent (incrémentent) ou diminuent (décrémentent) la valeur d'une variable de la quantité 1.

- x**++** ⇔ **++**x ⇔ x = x + 1
- x**––** ⇔ **––**x ⇔ x = x – 1

8.5.1. Postfixe: Un opérateur de post-incrémentation ou de post-décrémentation entraîne que son opérande est d'abord utilisé, puis incrémenté ou décrémenté.

Exemple

```
int x = 1, y ;
y = x++;
```

Résultat :

1. La variable y est affectée la valeur de la variable x, avant que celle-ci ne soit incrémentée, donc **y=1**,
2. x augmente de 1 (**x=2**).

Introduction au langage C

8.5.2 Préfixe

Un opérateur de pré-incrémentation ou de pré-décrémentation entraîne que son opérande est incrémenté ou décrémenté avant d'être utilisé.

Exemple

```
int x =1, y;
y = ++x;
```

Résultat :

1. x augmente de 1 (**x = 2**),
2. y ne prend pas la valeur d'origine de x, mais la valeur accrue de 1 (**y= 2**).

Introduction au langage C

Remarque :

1- L'opérateur de ***cast*** qui permet de convertir explicitement le type d'une donnée en un autre type (conversion forcée).

Exemple :

```
int n = 5 , p=2;
float x ;
x = (float) n/p ;
```

- n/p donne un entier (2)
- le quotient réel de la division sera obtenu après conversion de n en réel par la conversion forcée, puis (n/p) sera affecté à x dont la valeur sera 2.5

2- l'opérateur **sizeof(type)** renvoie le nombre d'octets réservés en mémoire pour chaque type d'objet.

Exemple :

```
n = sizeof(int) ; /* n vaut 4 */
```

Introduction au langage C

9. Les entrées-sorties

La bibliothèque standard <stdio> contient un ensemble de fonctions qui assurent la communication de la machine avec le monde extérieur. Dans ce chapitre ; nous allons nous en discuter des plus importantes :

- **printf ()** : écriture formatée de données
- **scanf ()** : lecture formatée de données
- **putchar()** : écriture d'un caractère
- **getchar()** : lecture d'un caractère

Introduction au langage C

9.1 Écriture formatée de données

a) Syntaxe :

printf("<format>", <expr 1>, <expr 2>,..... , <expr n>)

b) Rôle :

Elle permet d'afficher sur l'écran, du texte, des valeurs de variables ou des résultats d'expressions dans le format choisi.

Le **format** est une chaîne de caractères qui peut contenir :

- des caractères de contrôles
- des codes de format, à raison d'un code par expression. Les codes de format commencent toujours par le symbole **%**
- Des commentaires

Introduction au langage C

c) Symboles de conversion :

Il indique la nature de la conversion de type à effectuer.

Symbole	Objet de donnée
d, i	Entier relatif
u	Entier naturel (unsigned)
o	Entier exprimé en octal
x , X	Entier exprimé en hexadécimal
c	caractère
f	Réel en notation décimale
e, E	Réel en notation exponentielle
s	Chaîne de caractères

Introduction au langage C

Exemple 1

1.

```
printf ("%d",5) ;
```

Résultat : 5

2.

```
 int i = 5 ;
printf ("%d " , i ) ;
```

Résultat : 5

3.

```
int i = 5 ;
printf ("%d plus %d donne : %d " , 1000 , i , i +1000);
```

Résultat : 1000 plus 5 donne : 1005.

Introduction au langage C

Remarque :

Par défaut, les nombres sont affichés avec le nombre de caractères nécessaires.

Format : **%nbd**

Si nb > nombre de caractères ⇔ Ajout des espaces à gauche

Exemple2

printf("%3d",N);

N = 3 → ~~3

N= -5200 → -5200

~ : représente un espace

Exemple 3

printf ("%f",x);

x = 1.2345 → 1.234500

x = 12.3456789 → 12.345679

→ 6 chiffres après la virgule

Introduction au langage C

Exemple 4

printf ("%10f",x);

x= 1.2345 → ~~1.234500

Exemple 5

printf("%e",x);

x = 1.2345 → 1.234500e+000

x = 123.45 → 1.234500e+002

1 chiffre avant la virgule, 6 chiffres après la virgule et 3 chiffres après le "e"

Introduction au langage C

9.2 Lecture formatée de données

a) Syntaxe :

```
scanf("<format>", <&var1>, <&var 2>, ..... , <&var n>)
```

b) Rôle :

Elle permet de lire des données à partir du clavier, dans le format spécifié. Les valeurs lues par scanf() sont rangées dans les adresses spécifiées. L'adresse (adr) d'une variable est indiquée par le nom de la variable précédé du signe (&).

c) Symbole de conversion :

Mêmes symboles que ceux de printf().

Introduction au langage C

Exemple 1 :

```
int x ;
printf(" Donner x :" );
scanf ("%d" ,&x ) ;
```

L'opérateur d'adressage & applique au nom de la variable, il informe donc la fonction « scanf » de l'emplacement de stockage de la valeur saisie.

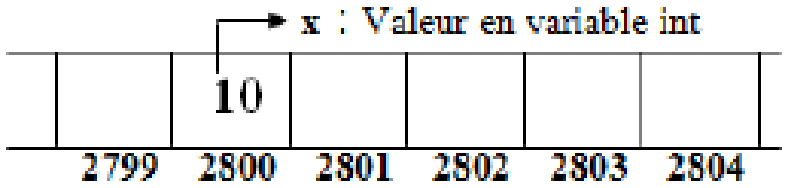

FIGURE 1 – Valeur et adresse d'une variable « int » en mémoire

Introduction au langage C

Exemple 2 :

Lors de l'entrée des données, une suite de signes d'espacement (espaces, tabulateurs, interlignes) est évaluée comme un seul espace.

Pour la suite d'instructions :

```
int jour , mois , annee ;
print(" donner le jour, le mois et l'année :" );
scanf ("%d%d%d" , &jour , &mois , &annee ) ;
```

Les entrées suivantes sont correctes et équivalentes :

Méthode 1

donner le jour, le mois et l'année : 22 10 2013

Méthode 2

donner le jour, le mois et l'année : 22 10 2013

Méthode 3

donner le jour, le mois et l'année : 22

10

2013

Introduction au langage C

9.3. La macro « getchar »

- Une macro est un nom qui représente une ou plusieurs instructions ou expressions.
- La macro « **getchar** » lit un caractère isolé depuis le clavier et le met à la disposition du programme.

Exemple :

```
# inc lude<stdio.h>
void main ( )
{
    char carlu ;
    printf ( "Entrez un caractère : " ) ;
    carlu= getchar () ;  /*Lecture d ' un caractère via «getchar» */
                         /* et affectation du caractère à la variable carlu */
    printf ( " \n le caractère saisi est =  %c  " , carlu ) ;
}
```

Introduction au langage C

9.4. La macro « putchar »

- La macro « **putchar** » affiche un caractère non formaté sur l'écran.
- La donnée à afficher est écrite sous forme de paramètre entre les parenthèses de la macro.

Exemple

```
# inc lude<stdio.h>
void main ( )
{
    char carlu ;
    printf ( "Entrez un caractère : " ) ;
    carlu= getchar() ;
    putchar ( carlu ) ;
}
```