

LE LANGAGE C

Objectifs:

- ① Introduire le langage C.

Pré requis:

- ① Algorithmique

Plan

I/ PRESENTATION DU LANGAGE C

II/ STRUCTURE D'UN PROGRAMME C

III/ LES CONCEPT FONDAMENTAUX DU LANGAGE C

1. Le jeu de caractères utilisé en C
2. Les identificateurs et les mots clés
3. Les types de données
4. Les séquences d'échappement
5. Les déclarations
6. Les expressions
7. Les instructions

IV/ Travaux pratiques (TP N° 0)

LE LANGAGE C

I/ PRESENTATION DU LANGAGE C

C est un langage structuré d'intérêt général, dont les instructions sont composées de termes ressemblant à des expressions algébriques et de mots clés réservés tels que if, else, for, do, while,...etc. Il ressemble de ce point de vue à d'autres langages structurés comme le pascal.

La caractéristique fondamentale du langage C est sa grande portabilité due à l'emploi de bibliothèques dans lesquelles sont regroupées les fonctionnalités liées à la machine. Chaque compilateur est ainsi livré avec les bibliothèques de fonctions dépendant de la machine. La plupart des programmes peuvent ainsi être exécutés sur des machines différentes avec un minimum de modification.

II/ STRUCTURE D'UN PROGRAMME C

Un programme en C est constitué d'une ou plusieurs fonctions dont une doit être appelée **main**. L'exécution du programme doit commencer toujours par cette fonction main.

Toute fonction, doit comprendre:

- * un en-tête de fonction, constitué du nom de la fonction suivi d'une liste d'argument optionnels entre parenthèses.

- * La liste des déclaration des variables à utiliser dans cette fonction

- * Les instructions à exécuter qui constituent le corps de la fonction.

Les arguments représentent la liste des paramètres de la fonction. Un bloc d'instructions est délimité par une accolade ouvrante "{" et une accolade fermante "}". Chaque deux instructions simples sont obligatoirement séparés par un point virgule ";". Les commentaires peuvent figurer n'importe où dans le programme. Ils représentent des ensembles de mots limités par "/*" et "*/" indiquant

respectivement le début du commentaire et la fin du commentaire. Par exemple
/* déclaration des variables */ est un commentaire.

Exemple de programmes C:

Exemple1 /* calcul de l'aire d'un cercle */

```
#include <stdio.h> /* inclusion de la bibliothèque */
main() /* en-tête */
{ /* début du bloc instruction */
    float rayon, surface; /* déclaration des variables */
    /* Les instructions */
    printf("rayon =");
    scanf("%f", &rayon);
    surface=3.14159*rayon*rayon;
    printf("surface =%f", surface);
} /* fin du bloc instruction */
```

Exemple2 /* calcul de l'aire d'un cercle */

```
#include <stdio.h> /* inclusion de la bibliothèque */
#define PI 3.14159 /* définition d'un symbole */
main() /* en-tête */
{ /* début du bloc instruction */
    float rayon, surface; /* déclaration des variables */
    float calcul(float rayon); /* déclaration d'une fonction */
    /* Les instructions */
    printf("rayon =");
    scanf("%f", &rayon);
    surface=calcul(rayon);
```

```
printf("surface =%f", surface);
}                               /* fin du bloc instruction */
/*-----*/
float calcul(float r) /* définition d'une fonction en-tête */
{                               /* début du bloc instruction */
    float a;                   /* déclaration des variables locales */
    /* Les instructions */
    a=PI*r*r;
    return(a);
}                               /* fin du bloc instruction */
```

III/ LES CONCEPT FONDAMENTAUX DU LANGAGE C

1. Le jeu de caractères utilisé en C

En langage C on trouve les 26 lettres de l'alphabet (majuscules et minuscules), les chiffres de 0 à 9 et un certain nombre de caractères spéciaux. La liste des caractères spéciaux est la suivante:

!	*	\	+	"	<
#	(=		{	>
%)	~	;]	/
^	-	[:	,	?
&	_	'	.	(espace)	

2. Les identificateurs et les mots clés

Les identificateurs sont les noms des variables, des fonctions ou des tableaux.

Un identificateur est une chaîne de caractères constituée des lettres de l'alphabet (majuscules et minuscules), des chiffres de 0 à 9 et du trait de soulignement '_'. Il doit toujours commencer par une lettre ou un trait de

soulignement mais jamais un chiffre. La longueur d'un identificateur dépend du compilateur utilisé, généralement 16 ou 32 caractères.

En C il existe un certain nombre de mots réservés et pré défini . ces mots réservés ne peuvent être employés que dans un rôle bien précis, donc, ils ne peuvent pas être utilisés comme identificateurs. La liste des mots réservés en C est la suivante:

auto	extern	sizeof
break	float	static
case	for	struct
char	goto	switch
const	if	typedef
continue	int	union
default	long	unsigned
do	register	void
double	return	volatile
else	short	while
enum	signed	

3. Les types de données

Le langage C offre plusieurs types de données. On trouve:

type	Description	Occupation mémoire
int	Valeur entier	2 octets ou 4 octets
char	Caractère simple	1 octet
float	Nombre en virgule flottante (réel)	4 octets
double	Nombre flottante en double précision	8 octets

4. Les séquences d'échappement

Les séquences d'échappement les plus fréquemment utilisées sont les suivantes:

Caractère	séquence d'échappement	code ASII
sonnerie (alarme)	\a	007
retour arrière	\b	008
tabulation horizontale	\t	009
tabulation verticale	\v	011
retour a la ligne (LF)	\n	010
nouvelle page (FF)	\f	012
retour chariot	\r	013
guillemet (")	\"	034
apostrophe (.)	\'	039
point d'interrogation	\?	063
slash (\)	\\	092
caractère nul	\0	000

5. Les déclarations

La déclaration consiste à associer un groupe de variables à un type de donnée particulier. La déclaration en C prend la forme d'un type, suivi du nom d'une ou plusieurs variables, puis un point virgule. Chaque variable tableau doit être suivie de crochets encadrant un entier positif spécifiant sa taille.

Exemples:

déclaration	signification
int x, y, z;	déclaration de trois variables entières
float valeur1, valeur2;	déclaration de deux variable réelle

char c;	déclaration d'une variable caractère
char t[100];	déclaration d'une variable de type tableau de 100 caractères

6. Les expressions

Une expression représente une donnée élémentaire, nombre ou caractère. Elle peut être constituée d'une seule entité ou d'une combinaison d'entités reliées par des opérateurs. Les expressions peuvent aussi être des conditions logiques prenant la valeur vrai ou faux. Les valeurs de vérité **true** et **false** sont représentés en C par 1 et 0.

Exemples:

Expression	Signification
a+b	addition
a=b	affectation
X=a+b	addition + affectation
Y<=Z	expression logique qui prend 1 si Y<=Z et 0 si Z<Y
X==N	test d'égalité
++K	incréméntation
i--	décréméntation

7. Les instructions

Il existe trois types d'instruction en C

- ✓ Les instructions simples
- ✓ Les instructions composées
- ✓ Les instructions de contrôle

✗ Les instructions simples

Une instruction simple consiste en une expression suivie d'un point virgule. Son exécution déclenche l'évaluation de cette expression.

Exemple:

```
a=3;
c==a+b;
++i;
printf("surface=%f",surface);
```

✂ Les instructions composées

Une instruction composée est constituée d'un ensemble d'instructions simple enfermées entre une accolade ouvrante et une accolade fermante. ({ et }).

Les instructions élémentaires contenues peuvent elles-mêmes être des instructions simples, des instructions composées ou des instructions de contrôle.

Exemple d'instructions composée:

```
{
    pi=3.14159;
    surface=pi*rayon*rayon;
    volume=(4/3)*surface*rayon;
}
```

✂ Les instructions de contrôle

Les instructions de contrôle sont utilisées pour réaliser certaines fonctionnalités des programmes comme les tests logiques, les boucles ou les branchements.

Exemple d'instruction de contrôle:

```
while(compteur<=n
{
    printf("X");
```



```
scanf("%f",&X);  
if(X>10)  
    somme+=X;  
else  
    somme-=X;  
++compteur;  
}
```

✂ Les constantes symboliques

Une constante symbolique est un nom qui peut être substitué à une chaîne de caractères pouvant représenter une constante numérique, une constante caractère ou une constante chaîne de caractères. Lors de la phase de compilation, chaque occurrence d'une variable symbolique est remplacée par la chaîne qui lui a été associée.

Les constantes symboliques sont définies au début du programme par la directive #define comme suit:

```
#define    nom    texte
```

Exemple:

```
#define    TRUE    1  
#define    FALSE    0  
#define    MAX    100  
#define    MIN    1
```

Il est conseillé que le nom soit toujours en majuscule pour le distinguer des identificateurs ordinaires.