

## TD1- Programmation Impérative

### I- Règles de lisibilité - Structure d'un programme C

#### Exercice 1 :

##### Règles de visibilité :

- Une et une seule instruction par ligne
- L'accolade fermante d'un bloc est alignée sur l'accolade ouvrante correspondante
- Les instructions d'un bloc sont décalées par rapport aux accolades marquant le début et la fin du bloc

Exemple d'écritures autorisées :

<pre>int main() {     printf("bonjour ");     printf(" à tous \n");     return 1 ; }</pre>	<pre>int main(){     printf("bonjour ");     printf(" à tous \n");     return 1 ; }</pre>
--	---

Ecrire le programme suivant en respectant les règles d'écriture de manière à obtenir une meilleure lisibilité du code.

```
/* programme de calcul de circonférence */ #include<stdio.h> float circonference(float rayon){ float pi = 3.1415 ; float circon ; circon = 2 * pi* rayon ; return(circon) ;} int main() { float r ; int i, n= 5 ; /* saisie par l'utilisateur de n rayons et calcul des n circonférences correspondantes */ for(i=0 ; i<n ; i++){ printf(" Saisie du rayon n° %d \n", i) ; scanf("%f", &r) ; printf("La circonférence est de : %.2f\n", circonference(r)) ; } return 1 ;}
```

#### Exercice 2 :

Dans le programme suivant, relever les erreurs de structure et de syntaxe et corriger les.

```
int nombre ;
#include <stdio.h>

float main(){
    printf(" Bonjour\n ") ;
}
int main(){
    nombre = 4
    printf("au revoir \n") ;    réalise l'affichage d' au revoir à l'écran
    return 1 ;
}
```

### II-Identificateurs – Variables - Types de données

#### Exercice 1 :

Parmi les déclarations suivantes, indiquez celles qui sont correctes et celles qui ne le sont pas, expliquez votre réponse.

```
float r, r1, r2 ;
real x,y,z ;
int for,main ;
char rs-232 ;
double d1 ;d2 ;d3 ;
long int ____3 ;
unsigned char c ='c' ;
```

#### Exercice 2 :

On souhaite déclarer des variables dont l'étendue des valeurs est précisée. Ecrire les déclarations correspondant en choisissant les types les plus économiques sans perdre de précision. (On supposera que l'on dispose d'une architecture 16bits)

```
compteur [0 .. 300]
x, y [-120,100]
mesure [-10 .. 104]
surface [0.5 .. 150075]
nb1 [-1.. 1024]
nb2 [0 .. 70000]
trouve [vrai, faux]
```

#### Exercice 3 :

Quel est le type numérique le plus économique pour les groupes de nombres suivants (On supposera que l'on dispose d'une architecture 16bits):

(1)	1	12	4	0	-125
(2)	1	12	-4	0	250
(3)	1	12	4	0	250
(4)	1	12	-4	0.5	125
(5)	-220	32000	0		
(6)	-3000005.000000001				
(7)	410	50000	-2		