

1. Les commentaires

Il s'agit de textes explicatifs placés entre `/*` et `*/` destinés aux lecteurs du programme et qui n'ont aucune incidence sur sa compilation.

```
float informatique ; /* note du module*/  
int num ; /* numéro d'inscription */
```

2. Les types prédéfinis

char short int long float double long double

3. La déclaration des objets

- Les constantes

```
const type nom = valeur ;
```

- Les variables

```
type nom ;
```

4. Les Opérateurs

- Opérateurs arithmétiques

+ - * / division % reste de la

division

- Les opérateurs relationnels

> >= < <= == égal !=

différent

- Les opérateurs logiques

&& et || ou !

négation

- L'affectation simple : symbolisée par le signe

=

- L'affectation multiple : symbolisée par le

signe =

Variable1 =variable2=.....=variable n = val

expression ;

- L'affectation composée :

+ = - = * = / = %=

5. Les fonctions d'entrées/sorties

```
scanf("formats",&a,&b,...);
```

```
printf("formats ",expression-1, ... expression-n);
```

```
gets(var) : lire une chaîne de caractères var
```

6. Les formats d'impression en C

%d type entier %o octal d'un entier %x

hexadécimal d'un entier %f type %c

caractère %s chaîne de caractères \n

retour à la ligne

7. Structure d'un programme C

directives au préprocesseur

déclarations de constantes/variables externes

fonctions secondaires

```
main()
```

```
{
```

déclarations de constantes/variables

internes

instruction1

instruction 2

.....

instruction n

```
}
```

8. La bibliothèque <math.h>

Le résultat et les paramètres de toutes ces

fonctions sont de type double. Si les paramètres

sont d'un autre type,

ils seront convertis en double par le compilateur

9. Instruction IF

```
if (expression1)
{
    Bloc_1
}
Else
    if (expression2)
    {
        Bloc_2
    }
    Else
        if (expression3)
        {
            Bloc_3
        }
```

9. Instruction SWITCH

```
switch(i) {
case 1 : instruction 1;
    break;
case 2 : instruction 2 ;
    break;
.....
.....
case n : instruction n;
    break;
default: .....;
}
```

10. les boucles

- for

```
for (l = v1; i < v2; i++)
{
    Bloc;
}
```

- while

while (Condition)

```
{
Bloc;
}
• do ...while
do
{
Bloc;
}
while (condition);
```

11. Les Tableaux à une dimension

Déclaration : type nom[taille];

Exemple int tab[20];

12 Les Tableaux à 2 dimensions

Déclaration : type nom[lignes][colonnes];

int matrice[5][3];

13. les fonctions

- La déclaration d'une fonction se fait selon la syntaxe suivante :

```
type nom_fonction (type arg-1,.,type arg-n)
{
    Déclarations des variables locales ;
    Liste d'instructions ;
return expression ;
}
```

- appel d'une fonction

Nom_fonction(para-1, para-2 ,...,para-n)