

LES ENTREES SORTIES

Objectif:

Connaître les fonctions nécessaires a la manipulations des Entrées / Sorties standards.

Pré requis:

Les Chapitres précédents du cours de langage C

PLAN

I/ LECTURE DE CARACTERE ISOLE (La fonction `getchar()`)

II/ ECRITURE DE CARACTERE ISOLE (La fonction `putchar()`)

III/ ENTREE DE DONNEES (la fonction `scanf()`)

IV/ ECRITURE DE DONNEES EN SORTIE (La fonction `printf()`)

V/ LES FONCTIONS `gets` ET `puts`

LES ENTREES SORTIES

I/ LECTURE DE CARACTERE ISOLE (La fonction `getchar()`)

La fonction **getchar()** fait partie de la bibliothèque "stdio.h", elle renvoie un caractère à partir du clavier.

Syntaxe :

```
caractère = getchar( );
```

Exemple

```
char c;  
c = getchar( );
```

II/ ECRITURE DE CARACTERE ISOLE (La fonction `putchar()`)

La fonction **putchar()** permet l'affichage d'un caractère isolé.

Syntaxe :

```
putchar(variable caractère)
```

Exemple

```
char c;  
.....  
putchar(c);
```

III/ ENTREE DE DONNEES (la fonction `scanf()`)

La fonction **scanf()** permet l'entrée de plusieurs données de types différents.

Syntaxe générale :

```
scanf(chaine_de_format, arg1, arg2, ....., argn)
```

- chaine_de_format est une chaîne qui contient les informations nécessaires à l'analyse des données lues sur l'unité standard de l'ordinateur.

- arg1, ..., argn représentent la liste des arguments représentant les diverses données; se sont des adresses des variables.

Exemple :

```
# include <stdio.h>

main()
{
  char pièce[20]
  int reference;
  float prix;
  .....
  scanf("%s%d%f",pièce,&reference,&prix);
  .....
}
```

Les types fréquemment utilisés pour la saisie des données sont :

Caractère de conversion	Signification
c	donnée de type caractère simple.
d	donnée de type entier décimal.
e	donnée de type virgule flottante (exposant).
f	donnée de type virgule flottante (virgule).
a	donnée de type entier court.
o	donnée de type entier octal.
x	donnée de type entier hexadécimal.
u	donnée de type entier non signé décimal.
s	donnée de type chaîne qui se termine par "\0".

IV/ ECRITURE DE DONNEES EN SORTIE (La fonction printf)

La fonction **printf** permet l'écriture de données en provenance de l'ordinateur sur l'unité standard de sortie. Ces données peuvent avoir des types différents.

De façon générale, la fonction printf s'écrit de la façon suivante :

```
printf(chaine_de_format, arg1, ....., argn).
```

Elle à le même esprit que scanf.

Les types fréquemment utilisés :

Caractère de contrôle	Signification
c	donnée affichée sous forme de caractère unique.
d	donnée affichée sous forme d'entier décimal signé.
e	donnée affichée sous forme de valeur réelle avec exposant.
f	donnée affichée sous forme de valeur réelle sans exposant.
o	donnée affichée sous forme d'entier octal.
u	donnée affichée sous forme d'entier décimal non signé.
x	donnée affichée sous forme d'entier hexadécimal.
s	donnée affichée sous forme de chaîne.

Exemple 1 :

```
# include <stdio.h>
# include <math.h>
main()          /* affichage des valeurs réelles */
{
  float i=2.0, j=3.0;
  printf("%f%f%f%f", i, j, i+j, sqrt(i+j));
}
```

Ce programme produit l'affichage des valeurs :

2.000000 3.000000 5.000000 2.236068

Exemple 2 : Lecture et réaffichage d'une ligne de texte :

```
# include <stdio.h>
main()
{
  char ligne[80];
  scanf("%s", ligne);
  printf("%s", ligne);
}
```

V/ LES FONCTIONS gets ET puts

Exemple : Lecture et réaffichage d'une ligne de texte.

```
# include <stdio.h>
main()
{
    char ligne[80];
    gets(ligne);
    puts(ligne);
}
```


