

Celui qui étudiait l'affichage formaté

1) Les formats :

a) Séquences d'échappement :

Ce sont des séquences spéciales précédées d'un \ :

Sauter une ligne :	\n
Poser une tabulation horizontale :	\t
Poser une tabulation verticale :	\v
Retour d'un caractère en arrière :	\b
Retour chariot (idem \n) :	\r
Sauter une page :	\f
Signal sonore :	\a
Afficher une apostrophe :	\'
Afficher un guillemet :	\«
Afficher un backslash (\) :	\\

A noter : pour afficher un % il faut taper %% et non \% !!!

b) Plages de valeurs des types élémentaires :

On supposera ici que le codage des int se fait sur 2 bits.

Char	-128 à 127
Unsigned char	0 à 255
Int	-32768 à 32767
Unsigned int	0 à 65535
Long	-2 147 483 647 à 2 147 483 648
Unsigned long	0 à 4 294 967 295
Float	$3,4 \cdot 10^{-38}$ à $3,4 \cdot 10^{38}$
Double	$1,7 \cdot 10^{-308}$ à $1,7 \cdot 10^{308}$
Long double	$3,4 \cdot 10^{-4932}$ à $3,4 \cdot 10^{4932}$

c) Les formats :

Nombre entier	%d et %i
Nombre entier non signé	%u
Caractère	%c
Chaîne de caractères	%s
Nombre à virgule flottante	%f et %e (format exponentiel)
Nombre entier octal	%o
Nombre entier hexadécimal	%x et %X

En ajoutant un l devant la lettre du format (ex : %ld) on obtient un long du type spécifié.

En ajoutant un L devant la lettre du format (ex : %Ld) on obtient un long double du type spécifié.

d) Quelques exemples (1) :

On supposera que les variables suivantes ont été définies :

```
float v=12.5436; int k=8; char c='A';
```

Et voici quelques exemples d'affichages (le « > » indique le résultat apparaissant à l'écran) :

1. `printf(«Affichage flottant : %f\nAffichage exponentiel : %e\n», v, v);`
> Affichage flottant : 12.543600
> Affichage exponentiel : 1.25436e+01
2. `printf(«Entier : %d -> conversion en octal : %o\n», k, k);`
> Entier : 8 -> conversion en octal : 10
3. `printf(«Caractère : %c -> valeur ASCII : %i\n», c, c);`
> Caractère : A -> valeur ASCII : 65

e) Les modificateurs de format :

Ces modificateurs devront être placés entre le % et la lettre du format. Certains modificateurs peuvent être combinés entre eux ...

Forcer l'affichage sur n caractères	n
Affichage justifié à gauche	-
Forcer l'affichage du signe	+
Forcer l'affichage de 0 avant	0
Chiffres limités après la virgule	.n
Forcer l'affichage du type	#

Note : # est utilisé lors des conversions en octal ou en hexadécimal, le symbole indiquant un nombre en octal est alors 0 et pour un nombre en hexadécimal il s'agit de 0x ou de 0X suivant le type choisi (x ou X).

f) Quelques exemples (2) :

1. `printf(«*%4d*\n», k);`
> * 8*
2. `printf(«%-4d*\n», k);`
> *8 *
3. `printf(«%04d\n», k);`
> 0008
4. `printf(«%.2f\n», v);`
> 12.54
5. `printf(«Entier : %d -> conversion en octal : %#o\n», k, k);`
> Entier : 8 -> conversion en octal : 010
6. `printf(«Entier : %d -> conversion en hexadécimal : %#x\n», -4, -4);`
> Entier : -4 -> conversion en hexadécimal : 0xffffc

2) Les formats spéciaux (appels système) pour les heureux possesseurs de Linux®™ :

Les appels systèmes suivants ne sont compatibles qu'avec les systèmes respectant les normes ANSI (*American National Standardization Institute*). Sur les anciennes versions de DOS®™ il existait un fichier ANSI.SYS qui gérait les séquences d'échappement (pilote clavier et écran). Sous Windows ces séquences ne fonctionneront pas bien évidemment ...

- Pour effacer l'écran (équivalent du `ClrScr` du Turbo Pascal avec `Uses Crt`) :

```
printf(«\033[2J»);
```

- Dans les 2 paragraphes suivant, le x pourra prendre ses valeurs dans l'intervalle 0..9.
(On aura par exemple : 0=noir, 3=jaune, 6=cyan, 7=gris, ...).

- Pour changer la couleur du texte :

```
printf(«\033[3xm»);
```

- Pour changer la couleur du fond :

```
printf(«\033[4xm»);
```