



# MINDS' ACADEMY

## Entrée et sortie en C :

Pour afficher des informations à l'écran, on utilise l'instruction de lecture **printf** :

***printf ("quantité %d, prix : %e", qte, prix) ;***

Elle comporte toujours un format, c'est-à-dire une chaîne de caractères (ici "quantité : %d, prix : %e") et, éventuellement, une ou plusieurs expressions (ici qte et prix). Le format comporte à la fois :

- Des codes de format, commençant par %, précisant comment est affichée une valeur de la liste
- Des libellés (caractères quelconque du type char) qui seront affichés tels quels.

Nous avons rencontré les codes de format suivants :

- **%d** pour une valeur de **type int**,
- **%t** pour une valeur de **type float**, affichée en notation flottante,
- **%e** pour une valeur de **type float**, affichée en notation exponentielle,
- **%c** pour le type char.

Dans un code de format, on peut spécifier un gabarit (comme dans %3d ou %100, au lieu de laisser printf utiliser le nombre exact de caractères nécessaires. De même, pour les types flottants, on peut spécifier la précision (comme dans %10.31 ou %12.4e), c'est-à-dire le nombre de chiffres souhaités après le point (par défaut, printf en prévoit 6).

Un caractère \n permet de provoquer un changement de ligne avant l'affichage suivant.

Si la liste comporte plus d'éléments qu'il n'y a de codes de format, les dernières valeurs ne seront pas affichées ; si, en revanche, il y a trop de codes de format, ces derniers provoquent l'affichage de valeurs imprévisibles.

Pour prélever des informations au clavier et les affecter à des variables, on utilise l'instruction de lecture **scanf** :

**`scanf("%d %f", &n, &x) ;`**

Elle comporte un format et une ou plusieurs adresses de variables ; une adresse de variable s'obtient en faisant précéder son nom de Nous avons rencontré les codes de format :

- **%d** pour une valeur destinée à une variable de type int,
- **%f** ou **%e** pour une valeur exprimée en notation entière, flottante ou exponentielle, destinée à une variable de type float,
- **%c** pour un caractère destiné à une variable de type char.

Dans la réponse fournie par l'utilisateur, les caractères espace et fin de ligne (ce dernier est automatiquement obtenu lors de la frappe de la touche de validation), jouent un rôle de séparateur. Les codes numériques sautent toujours les séparateurs, le code **%c**, quant à lui, ne le fait pas. Un espace dans le format demandé de sauter tous les séparateurs qui se présentent.

Les informations excédentaires fournies par l'utilisateur sont conservées pour une prochaine lecture.