

Chaînes de caractères

En C, une chaîne de caractère est un tableau de type char. Pour identifier la fin de la chaîne de caractères, le caractère '\0' est utilisé. Exemple :

La chaîne "Le chat" est représentée par un tableau :

L	e		c	h	a	t	\0	x	x
---	---	--	---	---	---	---	----	---	---

Les caractères qui viennent après \0 ne sont pas pris en compte.

Exercice 1 : Ecrire une fonction `longCh` qui calcule la longueur (le nombre de caractères) d'une chaîne de caractères passée en paramètre

Exercice 2 : Ecrire une fonction `concatCh` qui prend en paramètre deux chaînes de caractères et qui ajoute la deuxième chaîne à la fin de la première (on parle de *concaténation*).

Exercice 3 : Ecrire une fonction `compCh` qui prend en paramètre deux chaînes de caractères et qui renvoie 1 si la première chaîne est avant la deuxième (dans l'ordre du dictionnaire) ; -1 si la deuxième est avant et 0 si les chaînes sont identiques.

Pointeurs

Exercice 1 : Ecrire une fonction `min_et_max` qui a comme paramètres `T` un tableau d'entiers de taille quelconque, `n` la taille du tableau, et `pmin` et `pmax` 2 pointeurs vers des entiers. Après exécution de la fonction, les pointeurs `pmin` et `pmax` doivent contenir les valeurs min et max du tableau.

Exercice 2 : Soit `P` un pointeur que l'on fait "pointer" sur un tableau `A`:

```
int A[] = {12, 23, 34, 45, 56, 67, 78, 89, 90};  
int *P;  
P = A;
```

Quelles valeurs ou adresses fournissent ces expressions:

- | | | | |
|----|--------------------------|----|-------------------------------|
| a) | <code>*P+2</code> | e) | <code>A+3</code> |
| b) | <code>*(P+2)</code> | f) | <code>&A[7]-P</code> |
| c) | <code>&P+1</code> | g) | <code>P+(*P-10)</code> |
| d) | <code>&A[4]-3</code> | h) | <code>*(P+*(P+8)-A[7])</code> |

Structures, pointeurs et fonctions

En repartant de l'exercice 3 du TD3 (Le carnet d'adresse).

Écrivez les fonctions suivantes:

- `modif_info_contact`** qui prend en paramètre un pointeur sur un `Contact` et modifie les informations de ce contact
- `supp_contact_n`** qui prend en paramètre un pointeur sur un `Carnet`, et un entier `n`, et qui supprime du carnet le contact numéro `n`.
- `supp_tous_contacts`** qui prend en paramètre un pointeur sur un `Carnet` et qui supprime tous les contacts du carnet