

Séance 3: Pointeurs et chaînes de caractères

Exercice 1. Recherche dichotomique

Ecrire une fonction `int chercheIndice(int* tab, int taille, int val)` qui prend en argument un pointeur vers un entier (qui contient l'adresse de la première variable d'un tableau d'entiers **trié par ordre croissant**), la taille du tableau, ainsi qu'une valeur entière, et qui retourne l'indice de la case dans laquelle se trouve la valeur (si la valeur n'est pas présente dans le tableau, cette fonction retourne -1). La recherche de cet indice se fera **par dichotomie**.

Exercice 2. Chaîne de caractères

En langage C, une chaîne de caractères est un tableau, comportant plusieurs données de type **char**, dont le dernier élément est le caractère nul `'\0'`, c'est-à-dire le premier caractère du code ASCII (dont la valeur est 0). Ce caractère est un caractère de contrôle (non affichable) qui permet d'indiquer une fin de chaîne de caractères. Ainsi une chaîne composée de n éléments sera en fait un tableau de $n+1$ éléments de type char.

Les fonctions permettant de manipuler les chaînes de caractères se trouvent dans l'entête `string.h`, ainsi pour les utiliser il faut ajouter la commande `#include <string.h>`.

Exemple de fonction :

`strlen(chaine)` : renvoie la longueur de chaîne sans compter le caractère `'\0'`.

`strcpy(chaine_destination, chaine_source)` : copie la chaîne source dans la chaîne destination.

-

1. Ecrire un programme qui lit au clavier une **chaîne de caractères** puis l'affiche dans l'ordre inverse (de droite à gauche).
On réservera par exemple de la mémoire pour 30 caractères (allocation **statique** de mémoire) en supposant que la chaîne saisie ne dépassera pas cette taille.

Exemple : si on saisit `abcdef`, on obtient : `fedcba`

2. Ecrire une fonction `inverser()` qui prend en arguments un tableau de caractères (un pointeur vers un caractère) et sa taille, ce tableau contenant une chaîne de caractères terminée par `'\0'`. La fonction `inverser()` doit allouer **dynamiquement** un tableau de caractères de taille suffisante, y stocker la chaîne dans l'ordre inverse, et renvoyer ce tableau. Le tableau renvoyé doit se terminer par le caractère `'\0'`.

Remarque : n'oubliez pas de libérer la mémoire quand elle ne sert plus.

Exercice 3.

1. Ecrire une procédure *afficheChaine(char* chaine)* qui prend en paramètre une chaîne de caractères (un tableau de caractères dont le dernier est '\0') et l'affiche en insérant un espace entre chacun de ses caractères.

Ecrire la fonction *main* dans laquelle il faut lire au clavier une chaîne de caractères et l'affichez en utilisant *afficheChaine*.

2. Ecrire une procédure *void remplacer1(char* chaine, char ancien, char nouveau)* qui prend en paramètre une chaîne de caractères, le caractère à remplacer et le caractère de remplacement, et qui remplace dans la chaîne de caractères tous les caractères *ancien* par le caractère *nouveau*.

Ecrire la fonction *main* permettant de tester *remplacer1*.

ATTENTION : pour la saisie au clavier d'un caractère on utilisera la fonction *getchar()* et non pas *scanf()* avec la syntaxe suivante (pour éviter des problèmes lors de l'appui sur la touche ENTREE qui correspond au caractère '\n') :

```
char car;
printf("Entrer un caractère :");
while ((car=getchar())!='\n');
```

3. Ecrire une procédure *void remplacer2(char* chaine, char ancien, char nouveau)* identique à la précédente, mais vous accèderez cette fois-ci aux caractères de la chaîne en utilisant **(chaine+i)*.
4. Reprendre les questions précédentes en manipulant des tableaux d'entiers au lieu de tableaux de caractères. Les fonctions *remplacerEntier1*, *remplacerEntier2*, et *remplacerEntier3* prendront un paramètre supplémentaire : la taille du tableau d'entiers passée en paramètre.