

## Tableaux

On utilisera les définitions suivantes :

```
#define LONGUEUR 100
typedef char chaine[LONGUEUR];
```

### Exercice 1

1. *Tester les fonctions suivantes :*

```
void saisir(chaine ch){
    scanf("%s",ch);
}
void afficher(chaine ch){
    printf("%s\n",ch);
}
```

2. *Écrire une fonction `int longueur(chaine ch)` qui retourne la longueur de la chaîne de caractères `ch`.*

3. *Écrire une fonction `char majuscule(char c)` qui retourne la majuscule correspondant à `c` si `c` est une lettre minuscule, et `c` autrement.*

*Utiliser `majuscule` pour écrire une fonction*

```
void EnMajuscule(chaine source, chaine destination)
```

*qui convertit en majuscules les caractères de `source` et place le résultat dans `destination`.*

4. *Écrire une fonction qui indique la précédence lexicographique (pour le code ASCII) de deux chaînes de caractères. Cette fonction retournera `-1`, `0` ou `1` selon que la première chaîne est inférieure, égale ou supérieure à la deuxième.*

## Exercice 2

Pour chiffrer un message, on utilisait (au temps de César) la technique dite du décalage. Elle consiste à associer à la  $x$ -ième lettre de l'alphabet, la  $(k+x)$ -ième (modulo 26). Par exemple

pour  $k = 1$ , 'A' est codé 'B', 'B' est codé 'C', ..., 'Z' est codé 'A' ;

pour  $k = 2$ , 'A' est codé 'C', 'B' est codé 'D', ..., 'Z' est codé 'B'.

On supposera que les chaînes manipulées ne contiennent que des lettres majuscules (sans espace).

1. Écrire une fonction

```
void encoder(chaine source, int k, chaine destination)
```

qui place dans `destination` la chaîne `source` codée avec un décalage de `k`.

2. Écrire une fonction

```
void decoder(chaine source, int k, chaine destination)
```

qui place dans `destination` le décodage de la chaîne `source`, supposée codée avec un décalage de `k`.

3. Faire un programme principal utilisant `encoder` et `decoder`. Montrer qu'il n'est pas très difficile de casser ce type de chiffrement (i.e. de savoir décoder sans connaître `k`).

## Exercice 3 Le jeu du pendu.

Programmer en C le jeu du pendu : le programme manipulera une chaîne de caractères pour conserver le mot secret et une seconde pour mémoriser les caractères que le joueur a réussi à identifier.

On pourra utiliser une fonction auxiliaire

```
int valide(char c, chaine source, chaine dest)
```

qui écrit dans `dest` le caractère `c` aux indices où il apparaît dans `source`. La fonction retournera le nombre d'occurrences de `c` dans `source`.

## Exercice 4

Remplir un tableau de taille  $n \times n$  par les entiers  $1, 2, \dots, n^2$  en "spirale".

Exemple :

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9