

2 Chiffrement de César

2.1 Échauffement

Donnez la définition des fonctions

```
int maxi_tableau(int *tab,int taille) ;  
int min_tableau(float *tab,int taille) ;
```

qui retournent l'indice de la cellule contenant le maximum d'un tableau d'entiers et le minimum d'un tableau de réels.

Le principe du chiffrement de César.

Étant donné un entier positif n , ce codage consiste à remplacer chaque lettre d'un texte par la lettre qui se situe n places plus loin dans l'alphabet. Par exemple avec un décalage de 7, on obtient :

Clair	T	A	Q	U	E
Chiffré	[H	X	\	L

Ainsi, le texte clair T (code ASCII 84) est chiffré par le texte [(code ASCII 91).

2.2 Chiffrement.

Donnez la définition de la fonction de prototype :

```
char *chiffrer_cesar(const char *clair, int decalage) ;
```

qui prend en paramètre une chaîne de caractères non codée et retourne son chiffré (sans détruire le clair).

Remarque : on suppose disposer des fonctions classiques d'allocation (malloc).

2.3 Déchiffrement.

Utiliser la fonction de chiffrement pour implanter l'opération inverse dans la fonction de prototype :

```
char *dechiffrer_cesar(const char *chiffre, int decalage) ;
```

2.4 Premier pas vers le cassage du code.

Donnez la définition de la fonction de prototype :

```
char calcul_frequence(char *texte, float frequences_calculees[]) ;
```

qui calcule les fréquences de chaque lettre de la table des caractères ASCII (de 0 à 255) dans le texte et les stocke dans la table frequences_calculees. (On pourra déjà calculer l'occurrence de chaque lettre pour en déduire les fréquences d'apparitions). De plus, cette fonction doit retourner la lettre la plus fréquente.