

2.5 Calcul du décalage moyen

On suppose disposer d'une table codée par le tableau `float french[255]` qui contient la probabilité d'occurrences des lettres de la table ASCII en français. Par exemple, `french['e']` contient 0.1486, `french[' ']` contient 0.1835 et `french['u']` contient 0.0506. Ainsi, pour un texte de 1000 caractères, on s'attend à ce qu'il y ait 148 occurrences de la lettre `e`.

Écrire une fonction de prototype :

```
int decalage(char *texte, float *langue) ;
```

qui analyse le contenu du fichier crypté et retourne une proposition de décalage employé par le chiffre de César. (On cherche la lettre la plus courante dans le texte, on suppose que c'est la lettre ayant le plus de probabilités d'apparaître ; on obtient ainsi un décalage probable).

3 Une implantation sommaire de la fonction `scanf`

Dans cette section, on se propose d'implanter la fonction `scanf` dans une architecture dans laquelle le passage des paramètres se fait par une pile (cf. fin de la section pour une description de l'architecture).

Par ailleurs, on suppose ne disposer que d'une seule fonction externe dont le prototype est `int getchar(void)` ; et qui lit depuis l'entrée standard un caractère dont le code ASCII est retourné après conversion en un entier ; en cas d'erreur ou si l'entrée est un fichier dont on a atteint la fin, cette fonction retourne l'entier défini par la macro `EOF`.

Par exemple, pour extraire de l'entrée standard un caractère et le stocker dans une variable de type `unsigned char`, on peut s'inspirer du code suivant :

```
#include<stdio.h>

int main(){
    unsigned char c ;
    c = (unsigned char) getchar() ;
    return 0 ;
}
```

L'entrée standard est considérée comme une suite finie ou vide de n octets o_1, \dots, o_n ; si elle est vide la fonction `getchar` attend qu'un octet lui soit transmis, sinon après un appel à `getchar`, l'entrée standard est constituée de la suite finie de $n - 1$ octets o_2, \dots, o_n .

Objectif.

L'objectif est d'implanter la fonction de prototype :

```
int mscanf(const char *format, ...) ;
```

permettant de saisir depuis l'entrée standard :

- des caractères classiques ASCII de type `char`.
- des entiers machines positifs dans les bases décimale et binaire.

Cette fonction retourne 0 en cas de problème (mauvaise directive ou problème lors de la saisie i.e. le codage de la suite de caractères de l'entrée standard ne correspond pas aux conventions, cf. la suite) et 1 sinon.