

TD6 I31

L'objectif de ce T.D. de définir des fonctions élémentaires sur des chaînes de caractères.

- Une chaîne de caractères sera représentée par un **tableau de caractères**. Rappelons que pour un tableau $A[N]$ la lettre A correspond à l'adresse de la première case du tableau.
- La fin de chaîne sera donnée par un caractère spécial correspondant au caractère `'\000'`

1. Faites la fonction `int compte(char *str)` qui compte le nombre de voyelles.
2. Faites la fonction `void asciii(char *str, int A[N])` qui transforme une chaîne en un tableau correspondant en leur code ascii.
3. Faites une fonction `majuscule(char *str1, char* str2)` qui fait passer le mot en majuscule. conseil la différence en code (`'A'`-`'a'`) est le même pour toutes les lettres.
4. cryptage `xor` (en $C \sim$) permet de faire un cryptage de clés difficilement violable assez facilement pour de petits messages. Pour cela on s'appuie sur la propriété suivante du `xor`. $A \hat{B} \hat{A} = B$. (vérifiez la propriété) Si s_1 est la clé et s_2 le message, la version cryptée correspond à la chaîne $s_3 = s_1 \hat{s}_2$; et le décryptage à $s_3 \hat{s}_1$. Faites la fonction qui permette de crypter et de décrypter le message. Si la clé est de la longueur du message ce cryptage est difficile (voire impossible) à décoder.

5. Faites une fonction `void prefix(int n, char *str, char* strp)` qui crée une nouvelle chaîne contenant les n premiers caractères de la chaîne `str`. N'oubliez pas les cas d'erreurs.

Exemple : `prefix(4, "anacoluthie", p)` retourne `p="anac"`.

6. Faites une fonction `void prefixcom(char *str1, char *str2, char *str)` qui crée une nouvelle chaîne contenant le plus grand préfixe commun au deux chaînes `str1` et `str2`.

Exemple : `prefixcomm("miser", "mime", p)` retourne `p="mi"` . `prefixcomm("miser", "tomate")` retourne la chaîne vide `p=""`.

7. Faites une fonction `bool_t hide(char *str1, char *str2)` qui détermine si un mot est «caché» dans un autre. Un mot `str1` est caché dans `str2` si chaque lettre de `str1` se retrouve dans l'ordre dans `str2` éventuellement intercalée entre d'autres lettres.

Exemple : `hide("mise", "moisissure")` retourne vrai (**MoISissurE**), `hide("mise", "maison")` retourne faux.

8. Faites une fonction `void javanais(char *c)` qui **modifie** la chaîne pour la transcrire en javanais. L'écriture en javanais consiste à ajouter après chaque voyelle `[v]` la chaîne `d[v]g[v]`. Pour cette question, **le traitement devra être uniquement fait sur le tableau de caractères passé en argument. Aucun tableau intermédiaire ne sera admis.**

Exemple : `javanais("cheval")` modifie la chaîne ainsi `"chedegevadagal"`.
`javanais("miser")` modifie la chaîne ainsi `"midigisedeger"`.