Auteur : Mohamed Messabihi
Matière : Programmation et structures de données
Mohamed.Messabihi@gmail.com

Université Abou Bakr Belkaïd - Tlemcen
Faculté des sciences
1ère L.M.D. MI

7 TP : Les chaînes de caractères

7.1 Initialisation des chaînes de caractères

Lesquelles des chaînes suivantes sont initialisées correctement ? Corrigez les déclarations fausses et indiquez pour chaque chaîne de caractères le nombre d'octets qui seront réservés en mémoire.

```c
int main(void)
{
    char a[ ] = "un\ndeux\ntrois\n";
    char b[12] = "un deux trois";
    char c[ ] = 'abcdefg';
    char d[4] = "cinq";
    char e[10] = 'x';
    char f[ ] = "Cette " " phrase " "est coupee";
    char g[2] = {'a', '\0'};
    char h[4] = {'a', 'b', 'c'};
    char i[4] = "'o'";
}
```

Affichez les chaines de caractères a, b, ..., i et analysez les résultats.

7.2 Fonctions utiles

1. Écrire la fonction chaine_Majuscule qui convertit toutes les lettres d'une chaîne en majuscules, sans utiliser de variable d'aide. **Exemple :** "Toto1 Loulou2" devient "TOTO1 LOULOU2"

2. Écrire une fonction uniquement_Lettres prenant pour paramètre une chaîne de caractères s et retournant une chaîne contenant uniquement les lettres de s. **Exemple :** "aB'3$ @kj " devient "aBkj".

3. Écrire une fonction iverse_Chaine qui inverse une chaîne de caractères et met le résultat dans une autre. Les chaînes sont données en argument. **Exemple :** "! looc" devient "cool !"

4. Écrire la fonction nombreMots qui retourne comme résultat le nombre de mots contenus dans une chaîne de caractères passée comme paramètre.

5. Écrire la fonction frequenceMot qui retourne le nombre d'occurrence d'un mot m dans une chaîne de caractère s passés comme paramètres.

6. Écrire un programme pour tester toutes les fonctions précédentes.

7.3 Cryptographie

Le codage des messages secrets selon Jules César consistait à choisir une clé entière k dans [1,25] pour construire à partir d'un message msg un nouveau message codé avec la technique suivante. Chaque lettre majuscule de msg est décalée de k positions vers la droite (l'alphabet est circulaire : après 'Z' on revient sur 'A'). Les autres caractères du message ne soint pas modifiés !

1. Écrire une fonction code_cesar(msg,k) qui retourne le message codé.
   Exemple : code_cesar('ENVOYEZ 36 HOMMES !',3)
2. Écrire une fonction decode_cesar(msg,k) qui prend un message codé par la fonction précédente et retourne le message en clair.
   Exemple : decode_cesar('HQYRBHC 36 KRPPHV !',3)
3. Si vous êtes curieux : décodez le message 'VETFIV KFKF VK CFLCFL ZTZ'. Désolé j'ai perdu la clé !

Une technique ultérieure de cryptographie consista à opérer non avec un décalage systématique, mais par une substitution aléatoire. Pour cela, on utilise un alphabet-clé, dans lequel les lettres se succèdent de manière désordonnée.

Exemple : HYLUJPVREAKBNDOFSQZCWMGITX. C'est cette clé qui va servir ensuite à coder le message. Selon notre exemple, les A deviendront des H, les B des Y, les C des L, etc.

4. Écrire un fonction qui effectue ce cryptage (l'alphabet-clé sera passé en paramètre).
5. Écrire un programme qui permet de tester toutes les fonctions précédente.

. . . . . .

7.4 Bio-informatique (Supplémentaire)

En biologie, un problème courant consiste à comprendre la structure des molécules d'ADN, et le rôle de structures spécifiques dans le fonctionnement de la molécule. Une séquence ADN est représentée par une suite $c_0c_1c_2...$ de caractères choisis parmi les quatre nucléotides : adenine (A), cytosine (C), guanine (G) et thymine (T). Par exemple, la chaîne de caractères 'AAACAACTTCGTAAGTATA' représente un brin d'ADN.

1. Écrire une fonction valide qui prend une chaine en paramètre, et qui renvoie 1 si la chaine contient une chaine ADN, et 0 sinon.
2. Écrire une fonction cherche_base qui retourne le rang où apparaît pour la première fois la base représentée par un caractère c dans la séquence ADN S, tous deux passés en paramètres. Si la base n'apparaît pas, la fonction devra renvoyer -1 pour indiquer une erreur.
   Exemples : cherche_base("ATTGCC", 'C') ; retourne 4 ; et cherche_base("GTTGCC", 'A') ; retourne -1.
3. Écrire une fonction proportion permettant de calculer les proportions de présence des bases A, C, G et T dans une sequence ADN.
4. Écrire une fonction complement qui retourne le complémentaire d'une séquence ADN : A devient T, T devient A, C devient G, G devient C.
5. Écrire une fonction sous-chaine qui prend deux séquences ADN en paramètres et retourne true si la première est une sous séquence de la deuxième.
6. Écrire une fonction laPlusLongueSousChaine permettant de retourner la plus longue sous-séquence commune à deux séquences ADN passées en paramètres.