



TP : CHAÎNES DE CARACTERES

Exercice 1 :

Ecrivez une fonction *compteCar(s,c)* qui prend en paramètres une chaîne *s* et un caractère *c*, puis compte et renvoie le nombre de caractères *c* dans la chaîne *s* (il est interdit d'utiliser la méthode *count*).

Testez cette fonction en demandant au préalable à l'utilisateur de saisir une chaîne de caractères et un caractère.

Indication :

Parcourir la chaîne de départ caractère par caractère et tester si chaque caractère est égal au caractère *c*

Exercice 2 :

Ecrivez une fonction *compteMots(s)* qui prend en paramètre une phrase *s*, puis compte et renvoie le nombre de mots de cette phrase. On considérera qu'une phrase commence par une lettre, que chaque mot est séparé par un espace et que la phrase se termine par un point.

Testez cette fonction.

Exercice 3 :

1. Ecrivez une fonction *inverse(s)* qui prend en paramètre une chaîne *s*, puis renvoie la chaîne inverse à celle donnée en paramètre. Par exemple, *inverse("zorglub")* renverra *"bulgroz"*. Testez.

Indications :

Créer une nouvelle chaîne vide. Parcourir la chaîne *s* caractère par caractère et ajouter chaque caractère à la nouvelle chaîne en utilisant l'opérateur *+* (attention à ajouter le nouveau caractère au bon endroit).

2. Ecrivez une fonction *palindrome(s)* qui prend en paramètre une chaîne *s*, puis affiche si la chaîne *s* est un palindrome (dans cette fonction, vous ferez appel à la fonction *inverse*). Testez.
Un palindrome est un mot (ou un texte) qui reste identique qu'on le lise de gauche à droite ou de droite à gauche. Par exemple, "gag", "radar" et "ressasser" sont des palindromes.

Exercice 4 :

1. Ecrivez une fonction *estCorrecte(s)* qui prend en paramètre une chaîne *s*, vérifie si la chaîne *s* contient au moins 8 caractères, vérifie si *s* contient au moins un chiffre, renvoie *True* si les deux conditions précédentes sont réalisées, *False* sinon. Testez.

Indication : on pourra créer une chaîne chiffres = "0123456789" et utiliser l'opérateur *in* qui permet de vérifier si un caractère est dans une chaîne.

2. Complétez la partie principale de votre programme en demandant à l'utilisateur de choisir un mot de passe avec au moins huit caractères dont au moins un chiffre et répétant la demande tant que le mot de passe donné ne vérifie pas ces conditions (vous utiliserez la fonction *estCorrecte(s)*). Testez.

Exercice 5 :

Ecrivez une fonction *sansEspace(s)* qui prend en paramètre une chaîne *s* puis renvoie la même chaîne sans les espaces. Par exemple, *sansEspace("Salut les amis !")* renverra *"Salutlesamis!"*.

Exercice 6 :

1. Ecrivez une fonction *codage(s,mot)* permettant de coder une chaîne de caractère *s* à l'aide d'un mot clé *mot* en intercalant entre chaque lettre de *s* les lettres de *mot*.

Par exemple, *codage("Marchandise livrée au point S", "chateau")* renverra la chaîne :
"Mcahractheaanudcihsaet elaiuvrheâet eaauu cphoaitneta uSc"

2. Ecrivez une fonction *décodage(s,mot)* permettant de décoder une chaîne de caractère *s* cryptée à l'aide du mot clé *mot*.

Exercice 7 :

1. Ecrivez une fonction *echange(s,i,j)* qui prend en paramètre une chaîne *s*, deux entiers *i* et *j* et qui renvoie une chaîne dans laquelle les caractères situés aux positions *i* et *j* ont été échangés.
Par exemple, *echange("informatique",1,4)* renverra "irfonmatique".
2. Ecrivez une fonction *melange(s,n)* qui prend en paramètre une chaîne *s* et un entier naturel *n* non nul, procède à *n* échanges aléatoires de lettres et renvoie une autre chaîne constituée avec les lettres mélangées. Par exemple, *melange("informatique")* pourra renvoyer "frimqneotuai".

Indications :

- Créer une nouvelle chaîne qui au début est égale à *s* (en effet, on ne veut pas que la chaîne de départ *s* soit directement modifiée)
 - Répéter *n* fois les instructions suivantes :
 - choisir deux entiers aléatoires *i* et *j* compris entre 0 et $\text{len}(s) - 1$ qui représenteront les positions des lettres à échanger ;
 - utiliser la fonction *echange* avec la nouvelle chaîne pour procéder à l'échange des lettres situées aux positions *i* et *j* et affecter le résultat à la nouvelle chaîne qui sera ainsi modifiée
3. Programmez le jeu suivant en utilisant la fonction précédente :
 - le joueur 1 saisit un mot ;
 - l'ordinateur mélange les lettres de ce mot et affiche le mot ainsi constitué ;
 - le joueur 2 doit retrouver le mot initial.