

LIFAP1 – TD 10 : Chaînes de caractères

Objectifs : Se familiariser avec les chaînes de caractères, apprendre les manipulations de base et appliquer les techniques et compétences acquises lors des TD sur les tableaux

Déclaration : mot chaine[100] de caractères

Lecture au clavier : lire (mot)

Affichage de la chaîne : afficher(mot)

Accès à la lettre de la case 4 : mot[4]

Remarques

Toutes les cases ne sont pas nécessairement remplies

Caractère de fin de chaîne : '\0'

Elles sont toujours données/résultats, mais on ne met pas "&" en C/C++

Le premier indice est 0

1. Écrire l'algorithme d'une fonction qui compte le nombre de caractères dans une chaîne (sans utiliser la fonction prédéfinie *strlen*)
Ex : longueur("bonjour") renvoie 7
2. Écrire l'algorithme d'une procédure qui affiche une chaîne de caractère en inscrivant un caractère par ligne
Ex : affiche("hello") →
h
e
l
l
o
3. Écrire l'algorithme d'une procédure qui construit dans une nouvelle chaîne le miroir d'une chaîne de caractères.
Ex : le miroir de "bonjour" est "ruojnob"
4. Écrire l'algorithme d'une fonction qui retourne le nombre d'occurrences d'une lettre dans une chaîne de caractères
Ex : nb_occurrence("bonjour", 'o') → 2 nb_occurrence("bonjour", 'z') → 0
5. Écrire l'algorithme d'une fonction qui teste si une chaîne passée en paramètre est un palindrome ou non.
Ex : palindrome("eluparcettecrapule") → Vrai palindrome("bonjour") → Faux
6. Écrire l'algorithme d'une procédure qui prend une chaîne donnée en minuscules et construit la chaîne équivalente en majuscules, sans changer les caractères non-alphabétiques.
Ex : min2maj("bonjour") → "BONJOUR" min2maj("a2mains") → "A2MAINS"

Pour s'entraîner

Codage / décodage de César : il s'agit de transformer une chaîne de caractères en remplaçant chaque lettre du texte original par une lettre à distance fixe, toujours du même côté, dans l'ordre de l'alphabet. Par exemple si on choisit un décalage de 3 le 'a' sera remplacé par le 'd' le 'b' par le 'e' et ainsi de suite jusqu'au 'z' qui sera quant à lui remplacé par le 'c'.

Écrire en langage algorithmique un sous-programme permettant d'effectuer le codage de César. Ex : CESAR → FHVDU