

TP - Les chaines de caractères

Objectif(s)

★ Découvrir les traitements sur les chaines de caractères et l'utilisation d'une pile.

Exercices obligatoires

Exercice 1

Rappelons qu'une chaîne de caractères est un tableau de caractères normal, dont la fin est marquée par le caractère spécial `'\0'`.

Question 1

Écrivez une fonction `longueur` qui prend en paramètre une chaîne de caractères (un tableau de caractères) et qui retourne sa longueur, en comptant le nombre de caractères qui précède le caractère `'\0'`.

L'objectif de cet exercice est de vérifier qu'une expression mathématique est correctement parenthésée. Les expressions `"((a + b) * (c + d))"`, `"((a)())"` et `"()"` sont correctement parenthésées. Les expressions suivantes ne le sont pas : `"((a + b) * (c + d))"`, `"(a + b)"`, `"()"`, `"(a(b))(c)"`.

Soit une chaîne de caractères contenant une expression arithmétique.

Question 2

Une première technique consiste à utiliser un compteur que l'on incrémente de 1 lorsque l'on rencontre une `'('` et que l'on décrémente de 1 lorsque l'on rencontre une `')'`. Si en parcourant la chaîne de caractères le compteur devient inférieur à 0 ou si à la fin du parcours le compteur est différent de 0, il y a une erreur de parenthésage.

Écrivez une fonction `verifie_expression_1` qui renvoie `true` si une expression est correctement parenthésée, `false` sinon.

Question 3

Pour faire la suite des questions, vous avez besoin d'une pile. Sur Ecampus, vous trouverez 3 fichiers : `pile.h`, `pile.c`, `testpile.c`. Chargez ces 3 fichiers, compilez les, lisez le code et vérifiez que tout fonctionne correctement.

Question 4

Utilisez une pile pour vérifier qu'une expression est correctement parenthésée. Lorsque l'on rencontre une `'('` on la met dans la pile, et lorsque l'on rencontre une `)` on dépile. Si en parcourant la chaîne de caractères, on ne peut pas dépiler car la pile est vide ou si à la fin du parcours la pile n'est pas vide, il y a une erreur de parenthésage.

Écrivez une fonction `verifie_expression_2` qui renvoie `true` si une expression est correctement parenthésée, `false` sinon.

Question 5

Nous allons maintenant complexifier un peu le problème en ajoutant des `'['` et des `']'` dans les expressions.

Les expressions `"[(a + b) * [(c + d)]]"` et `"((a)())"` sont correctement parenthésées. Les expressions suivantes ne le sont pas : `"[(a + b) * (c + d)]"`, `"(a + b)[]"`, `"([()])"`.

Quelle technique permet de vérifier une expression avec des parenthèses et des crochets ?

Écrivez une fonction `verifie_expression_3` qui renvoie `true` si une expression est correcte avec des parenthèses et des crochets, `false` sinon.

Exercice 2

Un palindrome est une figure de style désignant un texte qui peut être lu de gauche à droite et de droite à gauche en ayant le même sens. Le mot *kayak* est donc un mot palindromique. Pour les phrases, il est nécessaire de faire

abstraction des espaces, de la ponctuation et des accents. Deux exemples : *Esope reste ici et se repose* et *La mariée ira mal*.

Question 1

Écrivez le corps de la fonction `int est_un_mot_palindromique(char s[])` qui vérifie que la chaîne de caractères `s` passée en paramètre est un palindrome. Cette version ne s'intéressera qu'aux mots palindromiques écrits soit en minuscule soit en majuscule. Exemple de mot : LAVAL, kayak. On ne prendra pas en compte les caractères accentués.

Question 2

Modifiez la version précédente pour que les mots *Laval* et *Kayak* soient considérés comme des palindromes.

Renforcement

Exercice 3

Question 1

On veut maintenant traiter tout type de palindrome.

Écrivez le corps de la fonction `int est_un_palindrome(char s[])` pour vérifier que la chaîne de caractères `s` passée en paramètre est un palindrome.

La fonction doit rendre `true` avec par exemple la phrase suivante : *“Tu l’as trop ecrase, Cesar, ce Port-Salut!”* et `false` avec toutes les séquences non palindromiques.

On considère que :

- la phrase ne comporte pas de caractères accentués.
- la ponctuation et les espaces ne sont pas à prendre en compte.
- les majuscules et les minuscules sont équivalents.

Question 2

Faites maintenant en sorte que les mots accentués soient aussi reconnus.