

Travaux Pratiques de Programmation II
[TP n°2 : Chaînes de Caractères & Piles]

Objectifs : - Introduire les chaînes de caractères
- Initier à l'implémentation de structures de données par la programmation modulaire
- Applications de la structure de données Pile.

Partie I : Cette première partie est une implémentation directe de certaines opérations sur *les chaînes de caractères* vues en TD. Pour cela, écrire un programme en C, qui lit une chaîne de caractères au clavier, ensuite, affiche à l'utilisateur un menu qui propose parmi d'autres les opérations suivantes :

- **affiche** : permet d'afficher la chaîne saisie au clavier ;
- **longueur** : donne le nombre de caractères de la chaîne ;
- **miroir** : permet d'inverser la chaîne de caractères (une 2^{ème} version à base de la structure de données Pile sera programmée dans la partie II ci-dessous) ;
- **est_palindrome** : vérifie si la chaîne courante est ou non un palindrome (voir TD) ;
- **en_majuscules** : transforme en majuscules toutes les lettres alphabétiques de la chaîne ;
- ...

Partie II : Il s'agit d'implémenter et de manipuler une pile de caractères, en utilisant la représentation contiguë d'une pile. *Une pile est représentée par une structure regroupant un entier, pour le sommet, et un tableau de MAX_PILE éléments (ici, des caractères).*

II.1- En utilisant la programmation modulaire, implémenter en C le type **Pile**.

Indications :

- Ouvrir un nouveau projet C, dans l'environnement de programmation **Dev-C++** ;
- Créer le fichier "**pile.h**", appelé *l'interface*, et l'ajouter au projet. Dans ce fichier, placer la déclaration complète, en C, de la structure de données représentant le type **Pile** ;
- Créer le fichier "**pile.c**", appelé *la réalisation*, et l'ajouter au projet. Dans ce fichier, définir en C les opérations fournies par la structure de données. Il est à noter que ce fichier contient au début l'inclusion du fichier "**pile.h**" ;

II.2- Après avoir implémenté en C la structure de données **Pile** pour les caractères, programmer une fonction **main** pour tester une pile caractères. Le programme affiche à l'utilisateur un menu avec les options suivantes :

- **empiler** un caractère dans la pile ;
- **dépiler** un caractère de la pile ;

- *taille de la pile* ;
- *sommet de la pile* ;
- *afficher* le contenu de la pile ;
- *quitter* le programme.

II.3- Applications de la structure de données Pile :

- a)- Ecrire un programme *C* pour *inverser une chaîne de caractères*.
- b)- Ecrire un programme *C* permettant de *vérifier si une chaîne de caractères* est un palindrome. Un palindrome se lit de la même manière de gauche à droite ou de droite à gauche. **Par exemple**, la chaîne "ressasser" est un palindrome
- c)- Ecrire un programme *C* permettant de *vérifier si une expression arithmétique contenant les délimiteurs* [,],), {, (et }, *est correcte ou non*. Une expression arithmétique est *correcte* si à chaque délimiteur ouvrant "[, (, {" correspond un délimiteur fermant "],), }" de même type. Le dernier symbole ouvert doit être le premier à fermer. Ce qui peut être simulé à l'aide d'une pile. A chaque rencontre d'un délimiteur ouvrant on l'empile et à chaque rencontre d'un délimiteur fermant on dépile.

Exemples :

{x + (100 - [a + b]) - d} est une expression correcte.

{x + (y - [a + b]) * 20 - [d + c} est une expression incorrecte.