Travaux dirigés

Langage C

Partie I : les chaines de caractères

Cet exercice est à résoudre de deux manières :

- Sans utilisation de pointeurs
- Avec utilisation des pointeurs.

Ecrire les fonctions/procédures suivantes :

1. Une fonction « dictionnaire » qui compare lexicographiquement deux chaînes de caractères CH1 et CH2, et affiche le résultat:

   Exemple:

   ```
   Introduisez la première chaîne: ABC

   Introduisez la deuxième chaîne: abc

   « ABC » précède « abc »
   ```

2. Une fonction « palindrome » qui permet de dire si une chaine de caractère est palindrome ou pas. Un mot palindrome est un mot qui, retourné, est identique à lui-même.
3. Une fonction « Concaténation » qui concatène deux chaines de caractère en une troisième.
4. Une fonction « conjugueur » qui lit un verbe régulier en "er" au clavier et qui en affiche la conjugaison au présent de l'indicatif de ce verbe. Contrôlez s'il s'agit bien d'un verbe en "er" avant de conjuguer.

   Exemple:

   ```
   Verbe : fêter
   je fête
   tu fêtes
   il fête
   nous fêtons
   vous fêtez
   ils fêtent
   ```

Partie II : les fonctions récursives

Ecrire les fonctions récursives suivantes :

1. Ecrire la fonction qui calcule la somme des n premiers carrés.
   Exemple : si n = 3 ; la fonction calculera $1^2 + 2^2 + 3^2$.
2. Ecrire la fonction qui permet de calculer, la multiplication de deux entiers par additions successives.
3. Ecrire la fonction qui permet de dire sin un entier est paire ou impaire, en supposant que les seules opérations possibles sont : la comparaison avec 0 et la comparaison avec 1.

Partie III : Le tri des tableaux

Exercice 1 : Tri par sélection

Le principe de cette méthode très intuitive consiste à :

- chercher le minimum dans un sous-tableau (au départ le tableau complet contenant les N valeurs non ordonnées)
- permuter ce minimum avec le premier élément du sous-tableau
- puis itérer ce traitement sur un nouveau sous-tableau de N-1 éléments (on ne tient plus compte du premier élément qui est maintenant à sa place)

Exemple :

Tableau initial = {3, 2, 9, 5}

1er tour : Considérer tout le tableau : {3, 2, 9, 5}

Identifier le minimum {3, 2, 9, 5} → Permutation avec le premier élément → {2, 3, 9, 5}.

Le premier élément du tableau est maintenant fixe (ne bouge plus)

2ème tour : Considérer le tableau à partir du 2ème élément : {**2**, 3, 9, 5}

Identifier le minimum {**2**, 3, 9, 5} → bien placé → {2, 3, 9, 5}

Les deux premiers éléments du tableau sont maintenant fixes.

3ème tour : Considérer le tableau à partir du 3ème élément : **{2, 3**, 9, 5}

Identifier le minimum : **{2, 3**, 9, 5} → Permutation avec le premier élément → {**2, 3, 5,** 9}

Les trois premiers éléments du tableau sont maintenant fixes.

4ème tour : nombre de tour est égal au nombre des éléments du tableau, donc fin de l'algorithme.

Ecrire la fonction *TriSel (int N, int tab[])*qui permet d'appliquer l'algorithme de tri par sélection sur le tableau tab de N éléments. Tester cette fonction.

Exercice 2 : Tri par bulle

Le principe de cette méthode consiste à comparer les couples de valeurs successives *Tab[i]* et *Tab[i+1]* pour `i` variant de `0` à `N-2`, et à les permuter si elles sont mal ordonnées. L'algorithme s'arrête lorsque l'on constate qu'aucune permutation n'a été effectuée lors du dernier "survol" du tableau.

Exemple :

Tableau initial = {3, 2, 9, 5}

1er tour :

{3, 2, 9, 5} → Permutation → {2, 3, 9, 5}

{2, 3, 9, 5} → {2, 3, 9, 5}

{2, 3, 9, 5}→ Permutation → {2, 3, 5, 9}

2ème tour :

{2, 3, 5, 9} → {2, 3, 5, 9}

{2, 3, 5, 9} → {2, 3, 5, 9}

{2, 3, 5, 9} → {2, 3, 5, 9}

Résultat : {2, 3, 5, 9}

Ecrire la fonction *TriBul (int N, int tab[])*qui permettra d'appliquer l'algorithme de tri par bulle. Tester cette fonction.