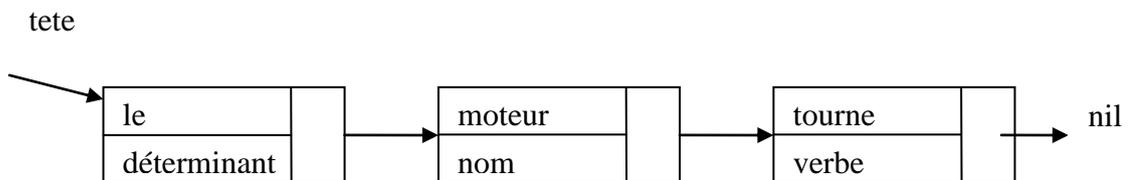


## Les listes chaînées – Modification d'éléments dans une liste chaînée

### Solution des exercices :

On désire créer une liste chaînée contenant les mots d'une phrase entrée par l'utilisateur. Pour les mots successivement entrés « le », « moteur », « tourne », elle prend la forme suivante :



### Question 1 :

Procéder aux déclarations de types nécessaires à la mise en place de cette structure.

**Type** ptelement = ^Element

**Type** Element = **Structure**

Mot : Chaîne

FonctionMot : Chaîne

Suivant : ptelement

**FinStructure**

### Question 2 :

Pour que les mots soient stockés dans la liste dans le même ordre que ceux de la phrase, quel type d'ajout d'élément dans la liste faut-il faire ?

Il s'agit d'un ajout en fin de liste. Chaque nouveau mot ajouté doit l'être à la fin pour respecter le sens de lecture de la phrase ensuite.

### Question 3 :

En admettant que la procédure d'ajout nécessaire existe sous la forme :

**Procédure** Ajout (E/S tete : ptelement, E mot : chaîne, E fonctionMot : chaîne)

où mot est le mot à entrer dans la liste et fonctionMot sa fonction dans la phrase (nom, verbe, etc.)

Ecrire le programme principal qui construit la liste correspondant à la phrase « le soleil brille » (déclarations de types, de variables et instructions)

**Type** ptElement = ^Element

**Type** Element = **Structure**

Mot : Chaîne

FonctionMot : Chaîne

Suivant : ptElement

**FinStructure**

**Déclaration**

tete : ptElement ;

Début

tete ← null

Ajout (tete, « le », « déterminant »)

Ajout (tete, « soleil », « nom »)

Ajout (tete, « brille », « verbe »)

Fin

**Question 4 :**

*Ecrire une procédure commentée chargée de mettre chaque verbe contenu dans la liste au pluriel.*

*On a pour hypothèse que les verbes entrés sont à la troisième personne du singulier au présent de l'indicatif...*

*On dispose pour cela d'une fonction déjà écrite et que l'on peut donc utiliser*

*conjuguer\_verbe ((E) mot : chaîne) : chaîne*

*qui conjugue un verbe à la troisième personne du pluriel et renvoie ce mot conjugué.*

**Procédure Conjuguer (E tete : ptElement)**

// Procédure chargée de conjuguer les verbes contenues dans la phrase pointée par tête au

// pluriel

// Pas de variable à déclarer, tête étant fournie en donnée uniquement ne sera pas affectée au

// retour de la procédure par les changements effectués sur cette variable dans le cadre de la

// procédure

**Début**

**Tant que** (tete <> null) **faire** // Tant que la tête de liste pointe sur un élément, on le teste

**Si** (tete^.FonctionMot = « verbe ») **alors**

        tete^.mot ← Conjuguer\_verbe (tete^.mot) // on fait appel à la fonction pour

**Finsi**

        // conjuguer notre mot

    tete ← tete^.suivant // on avance dans la liste

**Fin Tant que**

**Fin Procédure**