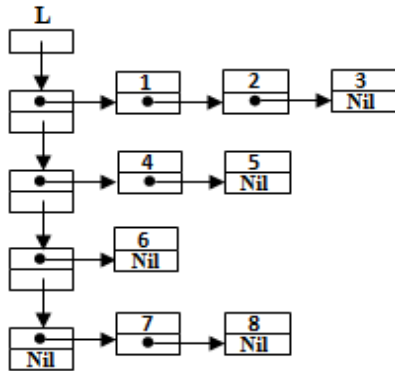


Durée : 1h30

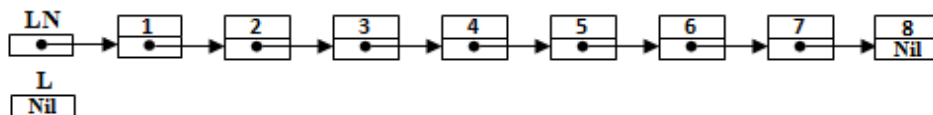
## EMD 2 ALGORITHMIQUE

### Exercice 1 : (12 points)

A) Soit L une liste chaînée dont les éléments sont des pointeurs vers des listes de nombres entiers.

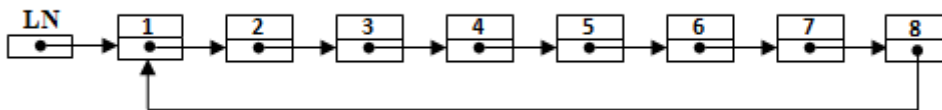


Après transformation :



1. Définir la structure de la liste L.
2. Ecrire une procédure **Liste\_Lineaire (L, LN)** permettant de transformer (sans allocation) la liste de tête L en une liste linéaire de tête LN.
3. Ecrire une fonction récursive **Rech\_rec** qui permet de rechercher une valeur V donnée, dans la liste LN.

B) Supposons maintenant que la liste LN est une liste circulaire, la dernière cellule est reliée à sa première cellule :



4. Ecrire une procédure **Supprimer\_K** qui permet de supprimer la cellule située à la  $k^{\text{ième}}$  position dans la liste circulaire.
5. Calculer la complexité de la procédure **Supprimer\_K**.
6. En utilisant la procédure **Supprimer\_K**, écrire une procédure **Supprimer\_tt\_K** qui, en commençant à partir de la première cellule, supprime successivement toutes les  $k^{\text{ièmes}}$  cellules de la liste ; en effectuant des parcours circulaires dans celle-ci jusqu'à ce que toutes les cellules soient supprimées.

**Exemple :** Si la liste circulaire contient 8 éléments dans cet ordre : 1, 2, 3, 4, 5, 6, 7, 8 et  $k = 3$  ; la suppression des cellules contenant les huit entiers se fait dans cet ordre : 3, 6, 1, 5, 2, 8, 4, 7.

**Exercice 2 : (08 points)**

Soit un fichier *T* contenant des mots de taille  $\leq 25$

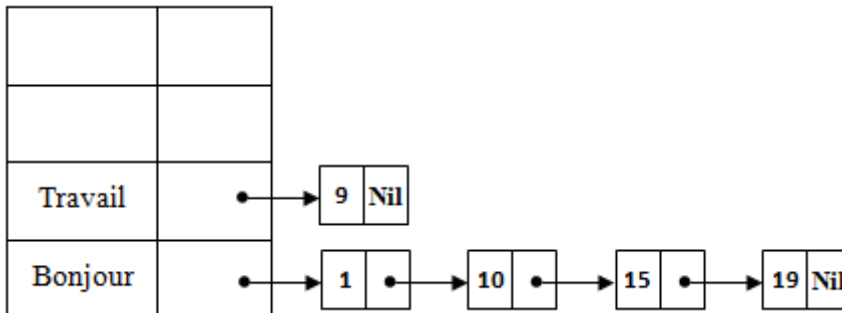
- a) Définir le type du *fichier T*.
- b) Un mot peut se répéter plusieurs fois dans le fichier T.

Ecrire une Procédure **PositionsMot** qui étant donnée un mot *M* retourne une **FILE** contenant toutes les positions du mot *M* dans le fichier *T*.

➤ **Remarques :** - Il faut donner la déclaration de la File.

- c) Ecrire une fonction **ExtraireMots** qui retourne une **Pile** contenant chaque mot du fichier T ainsi que la **File de ces positions**.

➤ **Remarques :** - Il faut donner la déclaration de la Pile selon le schéma suivant (avec élément qui est une structure contenant le mot (taille max=25) et un pointeur (TFile) qui pointe sur la File des positions).



*Pile de Mots avec File de positions*

**Remarque :** Vous pouvez utiliser sans réécrire

- La fonction **RechMotPile(TPile,Mot)** qui retourne vrai si le Mot existe dans la pile, Faux sinon.
- La procédure **Copy(F1,F2)** qui permet de copier tous les éléments du fichier F1 dans le fichier F2.

\*Toutes les fonctions et procédures de manipulation des piles et files sont supposées connues.

**Bon Courage !**