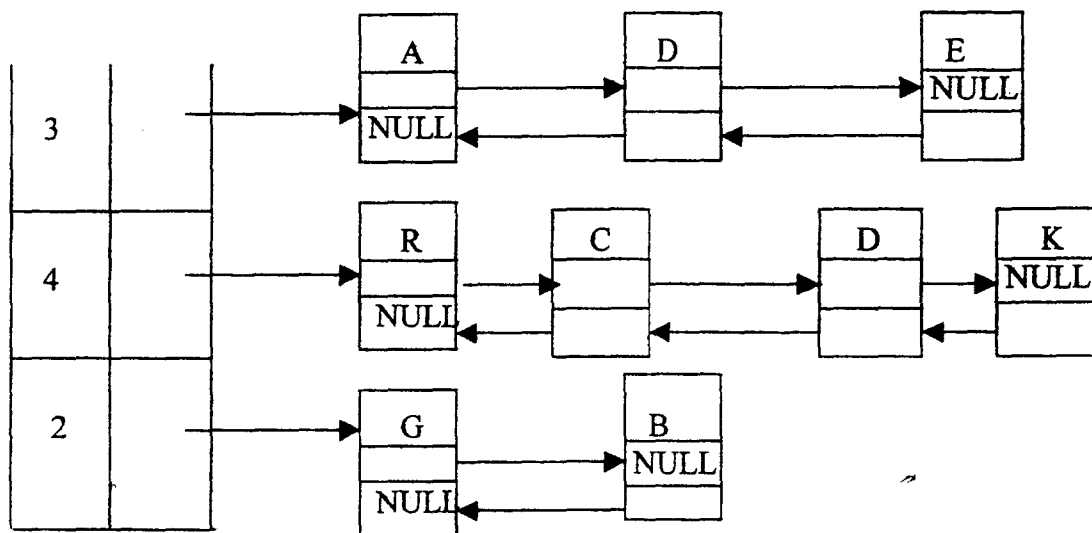


2^{ème} épreuve de moyenne durée**EXERCICE 1 (10pts)**

1/ Ecrire une fonction **DECOMPOSE** qui décompose un mot M dans une liste bidirectionnelle de points d'entrée tete et queue.

2/ Ecrire un programme qui introduit une suite de mots terminée par le mot vide, et construit une pile P contenant pour chaque mot, sa longueur et le lien vers sa décomposition.



Pile P

3/ Ecrire un programme qui affiche le mot le plus long et le mot le plus court, à partir de la pile.

EXERCICE 2 (5pts)

Soit un tableau T de N entiers ($N \leq 50$)

1/ Ecrire une fonction **POS_MIN** qui retourne la position du minimum dans le tableau T, à partir d'un indice d du tableau.

2/ Ecrire une fonction récursive **TRI**, qui réalise le tri du tableau T dans l'ordre croissant, en utilisant la fonction précédente.

EXERCICE 3 (5pts)

1/ Etant donné un mot M, écrire une fonction **VERIF** qui vérifie si le mot s'écrit sous la forme $A^i B^j$ avec $i \geq 0, j \geq 0$ (exemples : AAAABB, AAA, BB, ABBBB)

2/ Soit un fichier de caractères contenant des mots séparés par des blancs, écrire le programme qui compte le nombre de mots qui s'écrivent sous la forme $A^i B^j$.

N.B : * Les opérations de manipulation de piles sont supposées prédéfinies.

* On suppose que la longueur d'un mot ne dépasse pas 20.