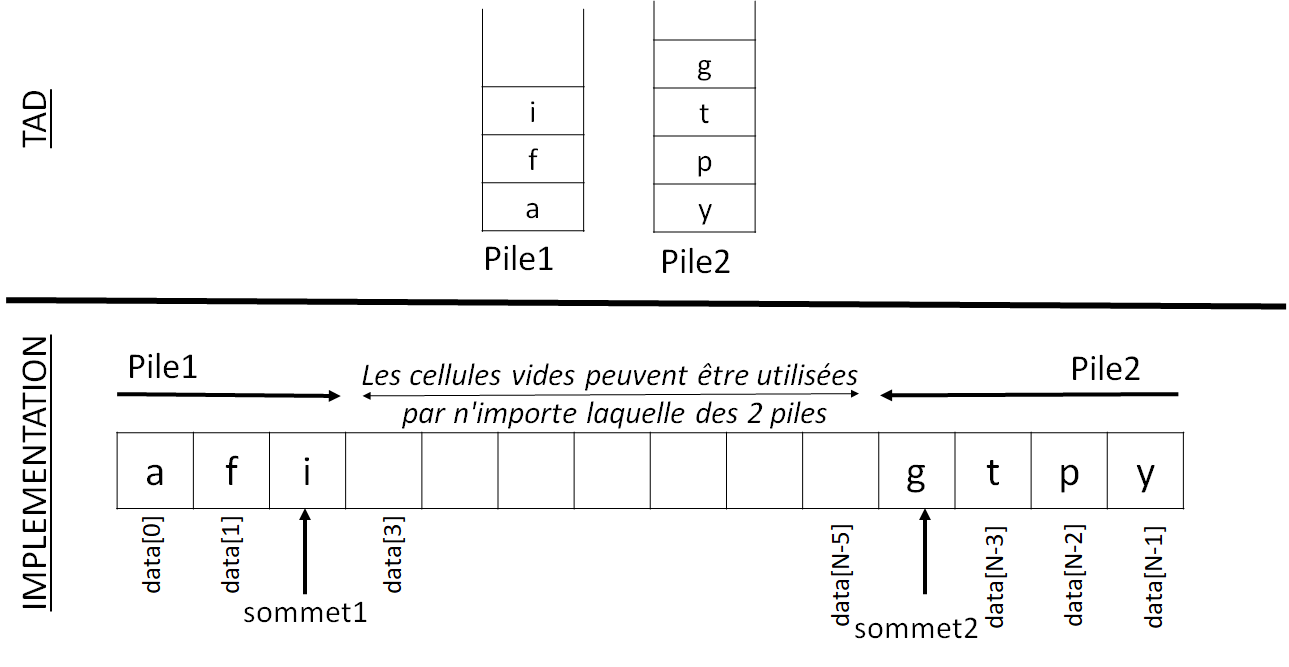
1. On peut définir une structure représentant 2 piles en utilisant un seul tableau linéaire de sorte qu'aucune des 2 piles ne sera pleine tant qu'il y a de la place dans le tableau et sans faire aucune opération de décalage des éléments du tableau.
   * Définir une structure Deux Piles qui implémente la notion des deux piles.
   * Implémentez les fonctions ***Empiler***, ***Depiler***, ***Tete*** et ***estPileVide.*** Les fonctions doivent prendre un paramètre entier supplémentaire pour identifier quelle pile sera utilisé.
   * Complexité ?



|  |
| --- |
| *One can define a structure representing 2 stacks using a single linear array so that neither of the 2 stacks will be full if there is space in the array and without doing any shift operation of the array elements.*   * + *Define a structure called TwoStacks which implements the notion of two stacks.*   + *Implement the functions Push, Pop, Top and isEmptyStack. Functions must take an additional integer parameter to identify which stack will be used.*   + Implement two stacks in one array – Ritambhara Technologies*Complexity?* |

1. On peut définir une structure représentant 3 piles en utilisant un seul tableau linéaire de sorte qu'aucune des 3 piles ne sera pleine tant qu'il y a de la place dans le tableau.
   * Définir une structure qui s’appelle troispiles qui implémente la notion des trois piles.
   * Implémentez les fonctions Empiler, Depiler, Tete et estPileVide. Les fonctions doivent prendre un paramètre entier supplémentaire pour identifier quelle pile sera utilisé.
   * Complexité ?

|  |
| --- |
| *A structure representing 3 stacks can be defined using a single linear array so that none of the 3 stacks will be full if there is room in the array.*   * + *Define a structure called ThreeStacks that implements the notion of three stacks.*   + *Implement the functions Push, Pop, Top and isEmptyStack. Functions must take an additional integer parameter to identify which stack will be used.*   *Complexity?* |

1. Write the following recursive functions:
   * Écrire une fonction récursive qui insère un élément e dans le bas d'une pile à l'aide d'opérations seulement la pile.

***Prototype: void InsererAuBas (pile \* p, élément e)***

* + Écrivez une fonction récursive qui inverse les éléments d'une pile en utilisant uniquement des opérations de pile.

***Prototype: void RenverserPile (pile \* s)***

* + Complexité des deux fonctions ?

|  |
| --- |
| *Write the following recursive functions:*   * + *Write a recursive function that inserts an element e into the bottom of a stack using only stack operations.*   ***Prototype: void InsertButtom (stack \* p, element e)***   * + *Write a recursive function that reverses the elements of a stack using only stack operations.*   ***Prototype: void ReverseStack (stack \* s)*** |