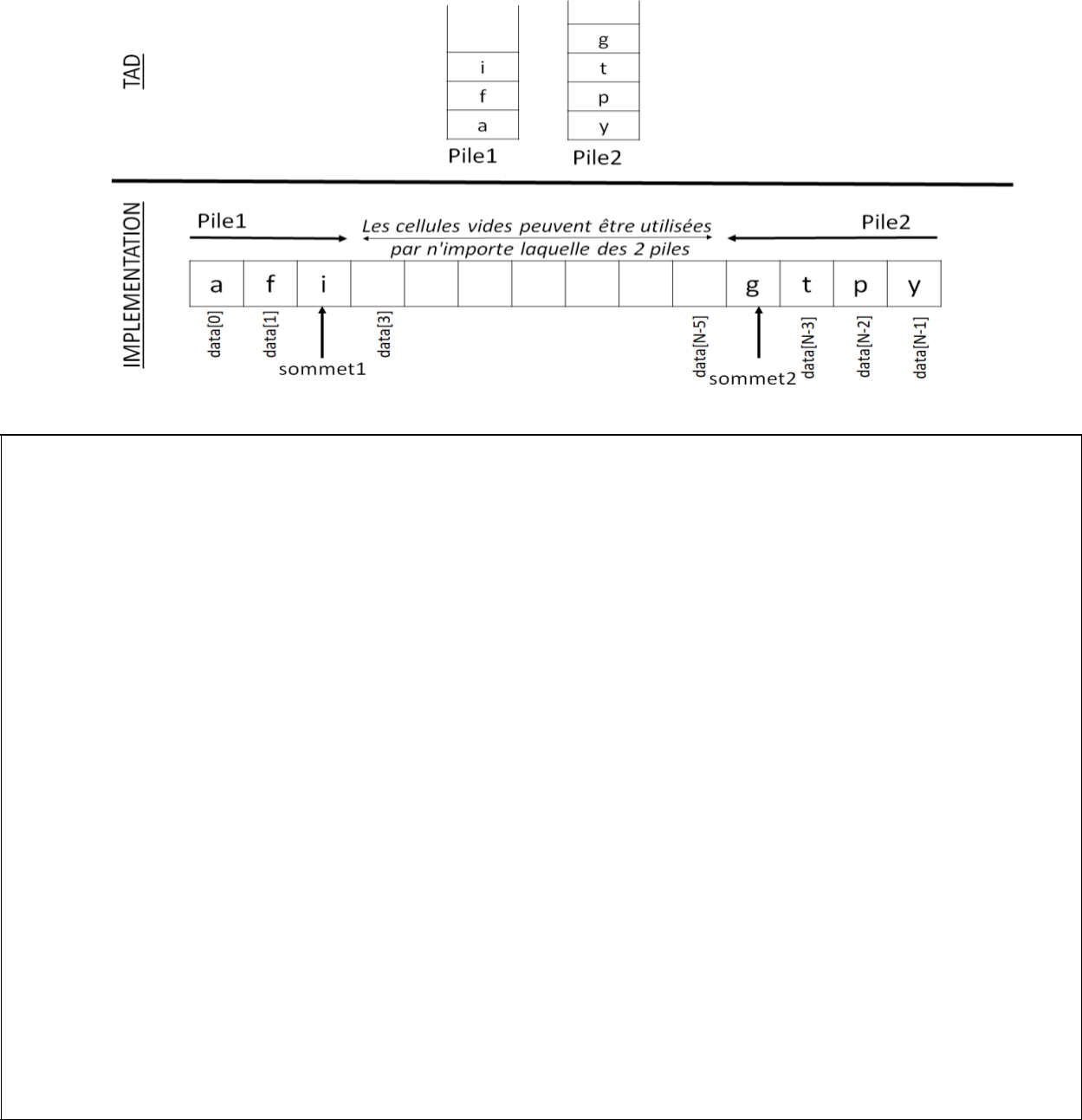
1. It is possible to define a structure representing 2 stacks by using a single linear table so that none of the 2 stacks will be full as long as there is space in the table and without performing any operation of shifting the elements of the table.
   * Define a TwoStacks structure that implements the notion of two stacks.
   * Implement the ***Stack***, ***Unstack***, ***Head*** and ***isEmptyStack*** functions***.*** Functions must take an additional integer parameter to identify which stack will be used.
   * Complexity?



*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.*
* *Complexity?*



Page **1** of **3**

1. A structure representing 3 stacks can be defined using a single linear array so that none of the 3 stacks will be full as long as there is space in the array.
   * Define a structure called three-stacks that implements the three-stacks concept.
   * Implement the Stack, Unstack, Head, and isEmptyStack functions. Functions must take an additional integer parameter to identify which stack will be used.
   * Complexity?

*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?*

3. Write the following recursive functions:

* Write a recursive function that inserts an e element at the bottom of a stack using only stack operations.

***Prototype: see InsertBottom (stack \* p, element e)***

* Write a recursive function that reverses the elements of a stack using only stack operations.

***Prototype: see ReverseStack (stack \* s)***

* Complexity of both functions?

Page **2** of **3**

*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 InsertButton (stack \* p, element e)***

* *A recursive function that reverses the elements of a stack using only stack operations.*

***Prototype: see ReverseStack (stack \* s)***

Page **3** of **3**