Chapter II : The Lists- Linked Lists

Introduction

Plus, the problem of the contiguous allocation in memory of elements of a table and the problem of size fixed that must be reserved in advance, the table-like structure poses other problems concerning inserting or removing an element because these actions require shifts in the contents of the boxes of the table that take time to run a program.

This type of value storage can therefore be costly in terms of execution time. There is another structure,

called chained list, to store values, this structure:

- Is a dynamic size variable (+/-)
- Makes it easier to insert and delete values in a linear list of items.
- Allows **non-contiguous allocation** of list items.

The Linked Lists

A linked list is a linear structure:

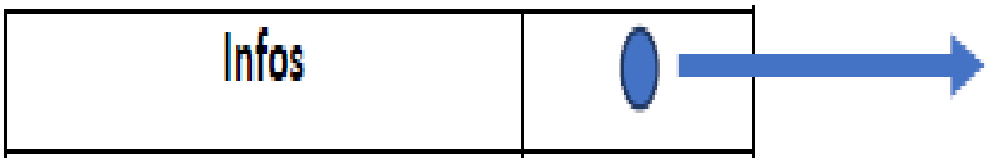
1) Which has no dimension fixed to its creation.
2) Its elements of the same type are scattered in memory and connected to each other by pointers.
3) Its size can be modified according to the space available in memory.

4) The list is accessible only by its head of list, that is to say its first element.

The definition of an Linear Linked List can be broken down into two (02) parts:

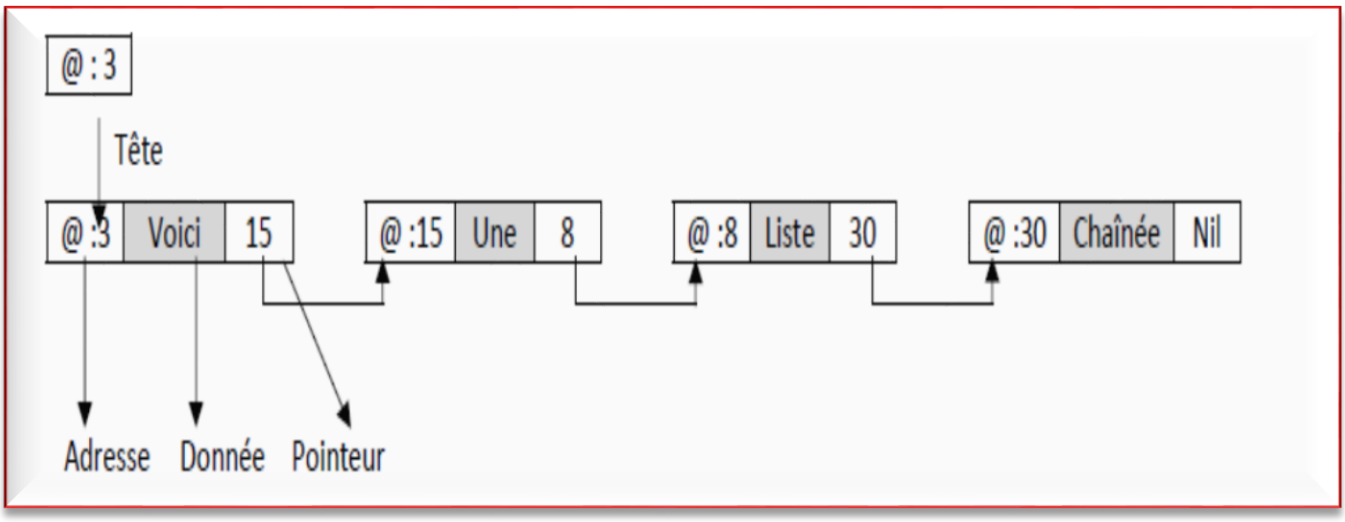
a) ***Part fields of an element*** of LLL (Info): which determines the type of elements of LLL (Ex: listof integers, students ...).

b) ***Element pointer part (Next):*** containing an address field (pointer) of the following element in the list. The sequence is then implemented by this pointer. The last item in the list points to nothing (Nil). You can access a list item by browsing through the items using their pointers.



Note: Each item in the list is a record.

The following chained list (@ indicates that the number after it represents an address):



- To access the third item in the list, you must always start reading the list by its **the first element** in the pointer where the position of the second element is indicated. In the pointer of the second item in the list we find the position of the third item...
- To add, delete or move an item you can simply allocate a space in the

update the pointers of the elements.