

SW N°3 : LINEAR LINKED LISTS

Exercise 1: Creation and browsing

Let L be a linear chained list.

1. Write an algorithm that constructs the list L from **n** read value, and then displays its elements.
2. Write a **procedure** that displays the elements of list L.
3. Write a **function** that calculates the number of elements in the list L.

Exercise 02: Search

Let L be a linear chained list.

1. Write a **procedure** that search for the element that has the value **Val** in the list and returns its position (pointer).

Exercise 03: Insertion

Let L be a linear chained list.

1. Write a procedure *InsetHead(L, Val)* which allows you to insert the value Val at the start of the list L.
2. Write a procedure *InsetQueue(L, Val)* which allows you to insert the value Val at the end of the list L.

Let L be a linear chained list of *ordered* integer values.

3. Write a procedure *InsetOrdonnee(L, Val)* which allows you to insert the value Val into the list L.

Exercise 04: Deletion

Let L be a linear chained list.

1. Write a *DeleteVal(L, Val)* procedure that deletes an element that has the value **Val** in the list if it exists.
2. Write a *DeleteOcc(L, Val)* procedure which deletes all occurrences of the value **Val** in L.

Exercise 05: Reverse

Let L be a linear chained list. Write an algorithm that reverses the elements of the list:

1. By creating a new list
2. Without creating a new list (By Reversing the chaining of the existing pointers)

Exercise 06: Splitting

Let L be a linked linear list of real values.

1. Write a procedure *TwoParts (L, L1, L2)* which divide (or split) the list L – in the middle – into two lists: L1 that contains elements of the first part of the list L and L2 contains the rest of the elements.

Exercise 07: Merging

Let L1 and L1 be two ordered linear chained lists.

1. Write an algorithm that merges the two lists to construct a single list L.

Exercise 08: S2 exam - PSD 2013-2014

Let L be a linked linear list that is used to store the scores of the 1st year students of the CS Department.

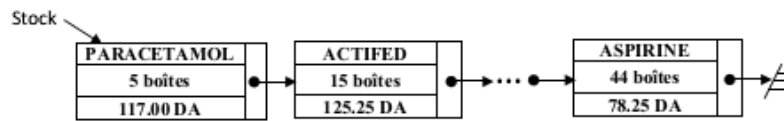
1. Write an algorithm that displays the number of students admitted to the second year.
2. Write an algorithm that removes all abundant students (who have an average of zero)

Exercise 09: the pharmacist

A pharmacist wishes to process information concerning his stock of medicines by computer.

We suggest representing this information in the form of a linear linked list or each node contains the name of a medication, the quantity available (number of boxes) and the unit price.

Example:



We ask you to:

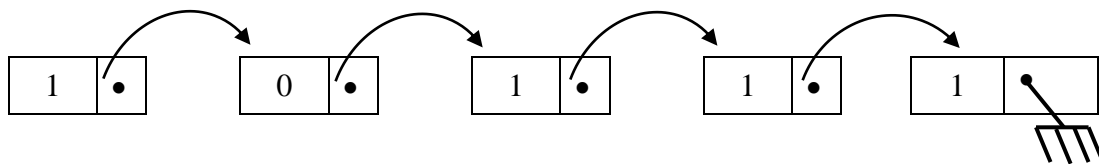
1. Give the data structures necessary for the representation of this stock.
2. Write the procedure **Sell**(*Stock*, *Med*, *NbBoxes*) allowing you to remove, if possible, '*NbBoxes*' from the medicine '*Med*' from '*Stock*'.
Warning: Medicines whose quantity reaches 0 must be removed from '*Stock*'.
3. Write the procedure **Buy**(*Stock*, *Med*, *Number of Boxes*, *Price*) allowing the pharmacist to refill his stock by '*NbBoxes*' of the medicine '*Med*' having the unit price '*Price*' DA.
4. Write the **StockPrice**(*Stock*) function to calculate the total price of medicines in '*Stock*'.

Exercise 10: S2 exam - PSD 2015-2016

Let N be an integer written under base 10.

1. Write an algorithm that converts N to binary and store it in a linear linked list digit by digit.

Example: If $N = 29$; then $(29)_{10} = (11101)_2$



2. Write a function that returns a number – in binary – from a given list L.

Exercise 11: PSD Review 2015-2016

Consider a L1 list containing the information of the 2nd year Mathematics students, where each element contains the following information (Registration number, name, average).

Write the Main algorithm which:

1. Construct the L1 list, where the number of students and the information of each student are read from the keyboard
2. Sort (Order) the list, by calling the procedure **Order**(L) which allows you to make the ordered list in order decreasing.
3. Display the average of the section, by calling the function **Moy_section**(L) which returns the general average of the section
4. Display the list of admitted students, by calling procedure **display_admitted**(L) which displays the list of admitted students

Exercise 12: Tables

Let T be a randomly filled array of integer values.

1. Write an algorithm that constructs a linear linked list L from the table T.
2. Write an algorithm which constructs a linear linked list L from the table T such that the elements of the list L are ordered.