Declare Linked list

Structure = mode
    head : type
      Suivant : ^ mode

fin structure

type list = ^ mode

null in Algo for pointers : nil.

fonction search (L : mode , x : int) : ^ mode
    Var

==select : ^ mode==

        ⓵ to access to the object in
            Linked list.

    Debut.
      select ⟵ L
      tant que (select) faire
        si (select^. Val) = x Alors
          Retourner select.
        fin si
          select ⟵ select^. Suivant
      fin Tant que

      retourner select.

    Fin

procedure print L(L°, mode,)

Debut
    tant que L ≠ nil faire
        ecrie (L° Val)
    Fin tant que .

    Fin

Inserting in the beginning of the list :
given (x), and lists L :
The proc insertion head inserts an element in the list .
Procedure insert (Var L : Node , x : entier)
    Var
        tmp : ^ mode

    Debut
        Allouer (tmp)

        tmp^. Vol ← x .

        tmp° Suivante ← L

        L ← tmp .

    Fin
Delete the first element :
    procedure Delete (Var L : Node)
        Var
            tmp : ^ mode

        Debut
            Si (L ≠ nil) Alors .
                tmp ← L .
                L ← L° next .
                Liberer (tmp)

- the procedure Delete element eliminates on element x of linked list L .

This procedure needs a pointer on element x to be deleted .