
Listes Linéaires Chainées

Exercice 1 Soit une liste linéaire chaînée d'entiers dont l'adresse du premier maillon est rangé dans le pointeur *Tete*. Écrire les procédures suivantes, dans le cas d'une liste unidirectionnelles puis dans le cas d'une liste bidirectionnelle :

1. Suppression d'un entier donné de la liste.
2. Ajout d'un élément à une position donnée.
3. Inverser la liste.
4. Calcul du nombre d'occurrences d'un entier donné.

Exercice 2 *Liste de nombre premiers*

Nous considérons la construction d'une liste des nombres premiers inférieurs ou égaux à un entier n donnée. Pour construire cette liste, on commence, dans une première phase, par y ajouter tous les entiers de 2 à n en commençant par le plus grand et en terminant par le plus petit qui se trouvera à la tête de la liste. On considère ensuite successivement les éléments de la liste dans l'ordre croissant en on supprime tous leurs multiples stricts. Écrire cet algorithme.

Exercice 3 Différence de deux listes linéaires chaînées

Soient L_1 et L_2 deux listes linéaires chaînées unidirectionnelles. Écrire la procédure qui permet de construire la liste $L = L_1 - L_2$ contenant tous les éléments appartenant à L_1 et n'appartenant pas à L_2 .

Exercice 4 *Fonctions LISP*

Soit une liste linéaire chaînée contenant des nombres entiers et dont la tête est L :

1. Écrire la fonction $CAR(L)$ qui retourne la valeur du premier élément de la liste.
2. Écrire la fonction $CDR(L)$ qui retourne la liste sans le premier élément.
3. Écrire la fonction $CONS(x, L)$ qui retourne une liste dont le premier élément est x et le reste est la liste L .
4. Écrire la fonction $Triée(L)$ qui retourne vrai si la liste L est triée dans l'ordre croissant et faux sinon.
5. Écrire la fonction $Fusion(L1, L2)$ qui prend deux listes triées dans l'ordre croissant $L1$ et $L2$ et retourne une liste triée, dans le même ordre, contenant les deux listes et cela en utilisant les fonctions précédentes.

Exercice 5 Représentation d'un nombre binaire par une LLC

On convient de représenter un nombre binaire $b_1b_2..b_n$, où chaque b_i est un 0 ou un 1, par une liste linéaire chaînée où chaque élément contient un b_i .

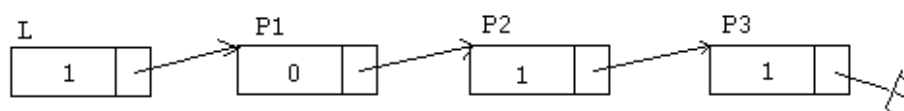
Considérons la procédure récursive suivante :

```
Procédure P( L : Pointeur(TMaiillon));
Var x : entier ;
Début
  Si ( L ≠ Nil) Alors
    P(Suivant(L));
    x ← Valeur(L) + R;
    Aff_val(L, x mod 2);
    Si (x=2) Alors
      | R := 1;
    Sinon
      | R := 0;
    Fin Si;
  Fin Si;
Fin;
```

Où

- L est une liste linéaire chaînée représentant un nombre binaire,
- R est une variable globale initialisée à 1,
- $A \bmod B$ désigne le reste de la division de A par B.

Donner les différentes valeurs de L et x, dans l'ordre, après l'appel récursif pour la liste suivante représentant le nombre 1011 :



- Que fait cette procédure ?
- Écrire un algorithme qui utilise cette procédure pour incrémenter un nombre binaire ainsi représenté.