

Examen de rattrapage

Exercice 1 Listes linéaires chaînées (1 + 2 + 2 + 2 + 3 + 4)

La représentation des nombres entiers dans les langages de programmation est toujours limitée à quelques octets. On souhaite surmonter ce problème en proposant une représentation des entiers quelque soit leur taille en utilisant les listes linéaires chaînées. Un nombre entier est représenté par une liste linéaire chaînée où chaque chiffre du nombre est représentée par un maillon de la liste. Le chiffre du poids faible est rangé à la tête.

Exemple : Le nombre 236799 est représenté par la liste linéaire chaînée suivante :

Nil \leftarrow [2] \leftarrow [3] \leftarrow [6] \leftarrow [7] \leftarrow [9] \leftarrow [9] \leftarrow Tete

1. Donner la déclaration permettant de représenter en mémoire de telles listes.
2. Soit la procédure f suivante :

```
Procédure  $f(L1, L2 : \text{Poniteur}(\text{TMaillon}); R : \text{entier}; \text{var } L : \text{Poniteur}(\text{TMaillon});$   
var  $p : \text{Poniteur}(\text{TMaillon});$   
       $S : \text{entier};$   
Début  
   $S \leftarrow R;$   
  Si ( $L1 \neq \text{Nil}$ ) Alors  
     $S \leftarrow S + \text{Valeur}(L1);$   
     $L1 \leftarrow \text{Suivant}(L1);$   
  Fin Si;  
  Si ( $L2 \neq \text{Nil}$ ) Alors  
     $S \leftarrow S + \text{Valeur}(L2);$   
     $L2 \leftarrow \text{Suivant}(L2);$   
  Fin Si;  
  Si ( $S \neq 0$ ) Alors  
    Allouer( $p$ );  
    Aff_val( $p, S \bmod 10$ );  
    Aff_adr( $p, \text{Nil}$ );  
     $L \leftarrow p;$   
     $f(L1, L2, S \text{ div } 10, \text{Suivant}(L));$   
  Sinon  
     $L \leftarrow \text{Nil};$   
  Fin Si;  
Fin;
```

(a) Soit les trois listes suivantes :

$$\begin{aligned} \text{Nil} &\leftarrow [1] \leftarrow \text{L1} \\ \text{Nil} &\leftarrow [2] \leftarrow \text{L2} \\ \text{Nil} &\leftarrow \text{L} \end{aligned}$$

Donner la liste L après l'appel de la procédure $f(\text{L1}, \text{L2}, 0, \text{L})$.

(b) Soit les trois listes suivantes :

$$\begin{aligned} \text{Nil} &\leftarrow [9] \leftarrow [9] \leftarrow [8] \leftarrow \text{L1} \\ \text{Nil} &\leftarrow [1] \leftarrow [1] \leftarrow \text{L2} \\ \text{Nil} &\leftarrow \text{L} \end{aligned}$$

Donner la liste L après l'appel de la procédure $f(\text{L1}, \text{L2}, 0, \text{L})$.

(c) Que fait la procédure f ?

3. En utilisant la procédure f précédente, écrire la procédure :

Procédure Mult2(L1 : pointeur(TMaiillon)); var L : pointeur(TMaiillon);

Permettant de multiplier le nombre rangé dans L1 par deux et ranger le résultat dans L.

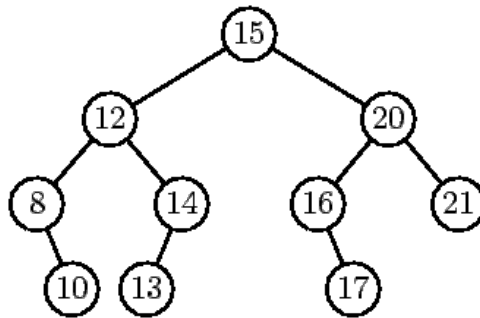
4. En utilisant la procédure f précédente, écrire la procédure :

Procédure MultN(L1 : pointeur(TMaiillon)); n : entier; var L : pointeur(TMaiillon);

Permettant de multiplier le nombre rangé dans L1 par n et ranger le résultat dans L.

Exercice 2 Arbres (2 + 2 + 2 + 1)

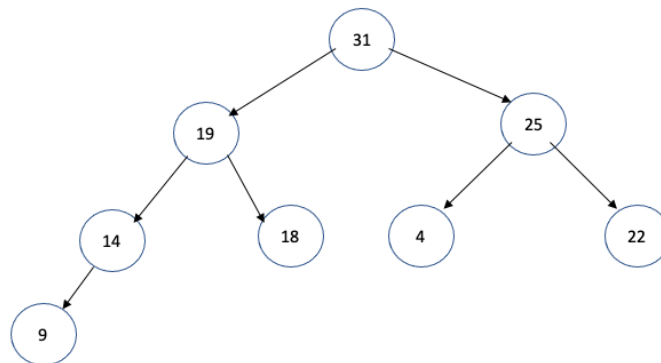
Soit l'arbre binaire de recherche suivant :



1. Donner l'arbre après l'insertion de la valeur 9

2. Donner l'arbre après la suppression de la valeur 15.

Soit le tas suivant :



1. Donner le tas après l'enfilement de la valeur 20

2. Donner le tas après un retrait.

★★★ Bonne chance ★★★

Corrigé type

Exercice 1 Listes linéaires chaînées (1 + 2 + 2 + 2 + 3 + 4)

$\text{Nil} \leftarrow [2] \leftarrow [3] \leftarrow [6] \leftarrow [7] \leftarrow [9] \leftarrow [9] \leftarrow \text{Tete}$

1. Déclarations permettant d'utiliser de telles listes.

```
Type TMaillon = Structure  
  Valeur : entier ;  
  Suivant : Pointeur(TMaillon) ;  
Fin ;  
Var Tete : Pointeur(TMaillon ;
```

1 pt

2. Soit la procédure f suivante :

- (a) La liste L après l'appel de la procédure $f(L1, L2, 0, L)$:

$\text{Nil} \leftarrow [3] \leftarrow \mathbf{L}$

- (b) La liste L après l'appel de la procédure $f(L1, L2, 0, L)$:

$\text{Nil} \leftarrow [1] \leftarrow [0] \leftarrow [0] \leftarrow [9] \leftarrow \mathbf{L1}$

- (c) La procédure f retourne la liste L représentant la somme des deux nombres représentés par L1 et L2

3. Procédure **Mult2(L1 : pointeur(TMaillon)) ; var L : pointeur(TMaillon) ;**

Pour multiplier L1 par deux, il faut juste l'additionner avec elle même.

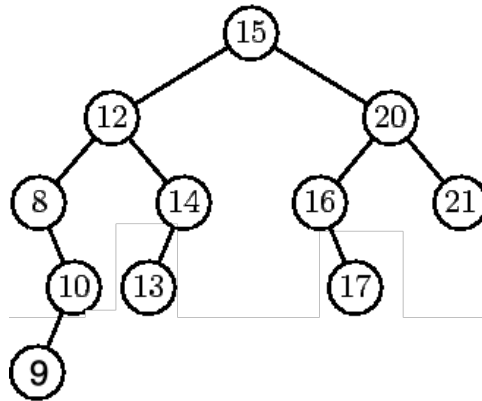
```
Procédure Mult2( L1 : Poniteur(TMaillon) ; var L : Poniteur(TMaillon)) ;  
Début  
  |  $f(L1, L1, 0, L)$  ;  
Fin ;
```

4. Procédure **MultN(L1 : pointeur(TMaillon)) ; n : entier ; var L : pointeur(TMaillon) ;**

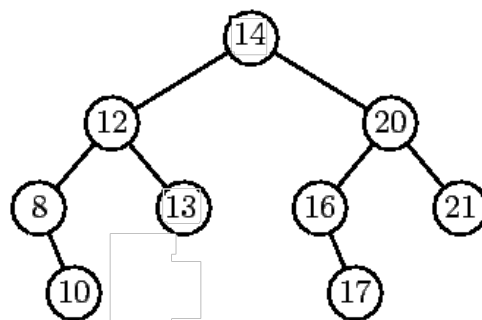
```
Procédure MultN( L1 : Poniteur(TMaillon) ; n : entier ; var i : Poniteu  
teur(TMaillon)) ;  
var i : entier  
Début  
  |  $L \leftarrow \text{Nil}$  ;  
  | Pour i de 1 à n-1 faire  
  | |  $f(L, L1, 0, L)$  ;  
  | Fin Pour ;  
Fin ;
```

Exercice 2 Arbres (2 + 2 + 2 + 1)

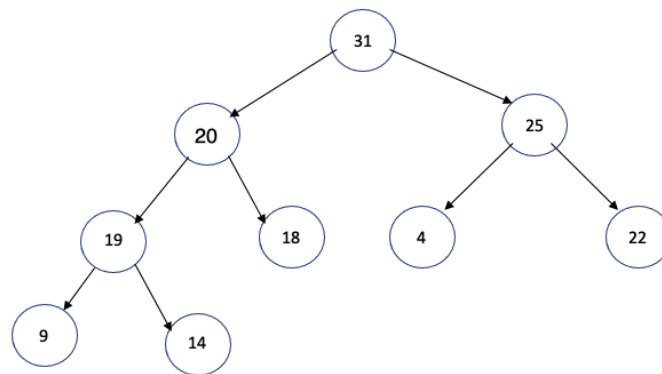
1. Donner l'arbre après l'insertion de la valeur 9



2. Donner l'arbre après la suppression de la valeur 15.



1. Donner le tas après l'enfilement de la valeur 20



2. Donner le tas après un retrait.

