

Correction TD4-ASD2 - les listes chaînées  
LGLSIA / LIRIS1

Exercice 1

1) fonction nb\_occ (L: liste; x: entier): entier

var

nb: entier

p: \*cellule

Début

nb ← 0

p ← L.Tete

tantque (p <> NULL) faire

si (p → val = x) alors

nb ← nb + 1

fin si

p ← p → suiv

fin tq

nb\_occ ← nb

fin

2) procédure création (L: \*liste)

var

x, k: entier

Rep: caractère

Début

Init(L)

Répéter

Ecrire ("donner x: ")

lire (x)

Répéter

Ecrire ("donner la position k: ")

lire (k)

Jusqu'à (k >= 1 et k <= Taille(\*L) + 1)

Ajout\_Pos (L, x, k)

Ecrire ("Voulez vous ajouter un autre entier: ")

lire (Rep)

Jusqu'à (Rep = 'o' ou Rep = 'O')

Fin

3) fonction Déterminer (L: liste): Entier

Var

p: \* cellule

L1: liste

nb: Entier

Début

nb ← 0

p ← L.Tête

tantque (p <> NULL) faire

L1.Tête ← p → suiv

Si (Recherche (L1, p → val) = NULL) Alors

nb ← nb + 1

fin si

p ← p → suiv

fin tq

Déterminer ← nb

fin

4) vue en cours.

5) vue en cours.

6) vue en cours.

7) vue en cours.

8) N.B.: la fonction retourne l'adresse de la 1<sup>ère</sup> occurrence de x s'il y a  
procédure supp-prem-occ (L: \*liste; x: Entier)

Var

p, q: \* cellule

Début

q ← Recherche (\*L, x)

Si (q <> NULL) Alors

Si (q = L → Tête) Alors

supp-Tête (L)

Sinon

p ← L → Tête

tantque (p → suiv <> q) R

fin tq p ← p → suiv

p → suiv ← q → suiv

q → suiv ← NULL

libérer (q)

fin si fin si fin si

(2)

9) procedure Supp-dern-occ(L: \*liste; x: Entier)

Var  
q, q1: \*cellule  
L1: liste

Début

q ← Recherche(\*L, x)

Si (q <> NULL) Alors

L1.Tête ← q → suv

q1 ← Recherche(L1, x)

Si (q1 = NULL) Alors

~~insertion~~

p ← L1.Tête

tant que (p → suv <> q) f

p ← p → suv

fin tq

p → suv ← q → suv

q → suv ← NULL

liberer(q)

sinon

Supp-dern-occ(L1, x)

fin si

fin si

fin

10/ procédure  $\text{Supp-Tête-occ}(L: * \text{liste}; x: \text{Entier})$

Début

tant que  $(\text{Rech}(*L, x) \neq \text{NULL})$  faire

$\text{Supp-prem-occ}(L, x)$

fin tq

fin

11/ fonction  $\text{Inverser}(Ls: \text{liste}): \text{liste}$

var

$LS\_Inv: \text{liste}$

$p: * \text{cellule}$

Début

$\text{Init}(\&LS\_Inv)$

$p \leftarrow LS.Tête$

tant que  $(p \neq \text{NULL})$  f

$\text{Ajout-Tête}(\&LS\_Inv, p \rightarrow \text{val})$

$p \leftarrow p \rightarrow \text{suiv}$

fin tq

$\text{Inverser} \leftarrow LS\_Inv$

fin

12/ fonction  $\text{Intersection}(L1, L2: \text{liste}): \text{liste}$

var

$L3: \text{liste}$

$p: * \text{cellule}$

Début

$L3.Tête \leftarrow \text{NULL}$

$p \leftarrow L1.Tête$

tant que  $(p \neq \text{NULL})$  faire

Si (Rech(L2, p → val) <> NULL et Rech(L3, p → val) = NULL)

Ajout-Queue (&L3, p → val)

Finssi

P ← P → Suiv

fin tq

Intersection ← L3

fin

13/ fonction Union (L1, L2 : liste) : liste

var

L3 : liste

p : \* cellule

Début

L3.Tête ← NULL

p ← L1.Tête

tantque (p <> NULL) faire

Si (Rech(L3, p → val) = NULL) Alors

Ajout-Queue (&L3, p → val)

finssi

P ← P → Suiv

fin tq

p ← L2.Tête

tantque (p <> NULL) faire

Si (Rech(L3, p → val) = NULL) Alors

Ajout-Queue (&L3, p → val)

finssi

P ← P → Suiv

fin tq

Union ← L3

fin



14/ procédure Eclater ( $L1: \text{liste}; L2, L3: * \text{liste}$ )  
var  
     $p: * \text{cellule}$

Début

Init( $L2$ )

Init( $L3$ )

$p \leftarrow L1.\text{Tête}$

tantque ( $p \neq \text{NULL}$ ) faire

    si ( $p \rightarrow \text{val} \bmod 2 = 0$ ) alors

        Ajout-Queue ( $L2, p \rightarrow \text{val}$ )

    sinon

        Ajout-Queue ( $L3, p \rightarrow \text{val}$ )

    fin si

$p \leftarrow p \rightarrow \text{suiv}$

fin tantq

fin

15) procédure fusion ( $L1, L2: \text{liste}; L3: * \text{liste}$ )  
var

$p, q: * \text{cellule}$

Début

Init( $L3$ )

$p \leftarrow L1.\text{Tête}$

$q \leftarrow L2.\text{Tête}$

tantque (p <> NULL et q <> NULL) faire

si (p->val <= q->val) Alors

Ajout-Queue (L3, p->val)

p ← p->suiv

Sinon

Ajout-Queue (L3, q->val)

q ← q->suiv

fin si

fin tq

tantque (p <> NULL) f

Ajout-Queue (L3, p->val)

p ← p->suiv

fin tq

tantque (q <> NULL) f

Ajout-Queue (L3, q->val)

q ← q->suiv

fin tq

fin

17/ fonction compler-pres (L: liste): Entier

var

nb: Entier

p: \*cellule

Debut

nb ← 0

~~si (p->val <= p->val) Alors~~  
~~nb ← nb + 1~~  
~~p ← p->suiv~~

~~montrons par récurrence~~

~~#~~  
P ← L.Tête  
si (P <> NULL) Alors  
tant que (P → smv <> NULL) f  
si (P → smv → val > P → val) Alors  
nb ← nb + 1  
fin si  
P ← P → smv  
fin tq  
fin si  
compter\_pred ← nb

fin  
17/

procédure Trix (L: liste)

Var

p: \* cellule

Permut: booléen

x: Entier

Début si (L.Tête <> NULL et L.Tête → smv <> NULL) Alors

Repete

Permut ← faux

P ← L.Tête

tant que (P → smv ~~→ smv~~ <> NULL) f

si (P → val > P → smv → val) Alors

x ← P → val

P → val ← P → smv → val

P → smv → val ← x

Permut ← Vrai

fin si

P ← P → smv

tant que (Permut = faux)

(8)



## Ex 2 :

1) Joueur : Enregistrement

Nom : chaîne

Prénom : chaîne

Poids : Entier

Num : Entier

Fin Enreg

Cellule : Enregistrement

Val : Joueur

Smv : \*cellule

Fin Enreg

liste : Enregistrement

Tête : \*cellule

Fin Enreg

2) fonction Rech (L : liste ; x : Entier) : boolean

Var

p : \*cellule

Début

p ← L.Tête

tant que (p ≠ NULL et p → val.Num ≠ x) faire

p ← p → Smv

fin tq

Rech ← (p ≠ NULL)

fin

procédure Créer\_liste (L : \*liste)

Var

p, q : \*cellule

i : Entier

Début

L → tête ← NULL

Pour i de 1 à 15 faire

q ← Allouer

Ecrire ( "donner les info du joueur", i)

lire (q → val.Num)

lire (q → val.Prenom)

lire (q → val.Poids)

Repetér

line( $q \rightarrow \text{val} \rightarrow \text{Num}$ )  
Jusqu'à ( $q \rightarrow \text{val} \rightarrow \text{Num} \geq 1$  et  $p \rightarrow \text{val} \rightarrow \text{Num} \leq 15$  et  
Rech( $*L, q \rightarrow \text{val} \rightarrow \text{Num}$ ) = faux)

$q \rightarrow \text{suiv} \leftarrow \text{NULL}$

Si ( $L \rightarrow \text{Tête} = \text{NULL}$ ) Alors

$L \rightarrow \text{Tête} \leftarrow q$

$p \leftarrow L \rightarrow \text{Tête}$

Sinon

$p \rightarrow \text{suiv} \leftarrow q$

$p \leftarrow p \rightarrow \text{suiv}$

fin si

fin pour

3) fonction Plus\_bond( $L: \text{liste}$ ):  $* \text{cellule}$

Var

$p, q: * \text{cellule}$

Début ( $L \rightarrow \text{Tête} \neq \text{NULL}$ ) Alors

si  $q \leftarrow L \rightarrow \text{Tête}$

$p \leftarrow L \rightarrow \text{Tête} \rightarrow \text{suiv}$

tant que ( $p \neq \text{NULL}$ ) faire

si ( $p \rightarrow \text{val. Poide} > q \rightarrow \text{val. Poide}$ ) Alors

$q \leftarrow p$

fin si

fin tq

Plus\_bond  $\leftarrow q$

Sinon

Plus\_bond  $\leftarrow \text{NULL}$

fin si

fin

Ex 3:

1) processus : Enregistrement

Id : Entier

Pr : Entier

Fin Enreg

2) Cellule : ~~xxxx~~ Enregistrement

Val : processus

Smv : \* cellule

Fin Enreg

liste\_proc : Enregistrement

Tête : \* cellule

fin Enreg

3) procédure Ajouter-processus(L: \*liste\_proc; p: processus)

var

q, q1 : \* cellule

Début

q ← Allouer(1)

q → Val ← p

si (L → Tête = NULL ou p.pr < L → Tête → Val.pr) Alors

q → Smv ← L → Tête

L → Tête ← q

Sinon

q1 ← L → Tête

tantque (q1 → Smv <> NULL et p.pr > q1 → Val.pr) faire

fin tq q1 ← q1 → Smv

q → Smv ← q1 → Smv

q1 → Smv ← q

fin  
fin si

4) fonction lancer-processus ( $L: *liste\_proc$ ) :  $*cellule$

var

$q: *cellule$

Début

si ( $L \rightarrow Tête \neq NULL$ ) Alors

$q \leftarrow L \rightarrow Tête$

$L \rightarrow Tête \rightarrow Suiv \leftarrow L \rightarrow Tête \rightarrow Suiv \rightarrow Suiv$

$q \rightarrow Suiv \leftarrow NULL$

Sinon

$q \leftarrow NULL$

fin si

$Lancer\_processus \leftarrow q$

fin

Ex 4:

1) Monôme : Enregistrement

Coeff : réel

deg : Entier

fin Enreg

Cellule : Enregistrement

Val : Monôme

Suiv :  $*cellule$

fin Enreg

Polynôme : Enregistrement

Tête :  $*cellule$

fin Enreg

Rq: on peut ne pas utiliser l'enregistrement cellule et on ajoute le champ suiv dans l'enregistrement monôme et le champ tête dans le polynôme. Sera un pointeur sur monôme.



2/a) procédure Afficher (P: Polynôme)

Var

q: \* cellule

Début

q ← P.Tête

tantque (q ≠ NULL) faire

    Ecrire (q → val.coef, "x", q → val.deg, "+")

    q ← q → suiv

fin

fin

b) fonction CopiePolynome (A: Polynome) : Polynome

Var

B: Polynome

p, q, q1: \* cellule

Début

p ← A.Tête

B.Tête ← NULL

tantque (p ≠ NULL) faire

    q ← Allouer(1)

    q → val.coef ← p → val.coef

    q → val.deg ← p → val.deg

    q → suiv ← NULL

    si (B.Tête = NULL) Alors

        B.Tête ← q

        q1 ← B.Tête

    sinon

        q1 → suiv ← q

    fin si q1 ← q

p ← p → suiv

fin tq

CopierPolynome  $\leftarrow B$

fin

c/ procédure AjoutMonôme ( $A: \text{Polynôme}; M: \text{Monôme}$ )

var

$p, q: \text{cellule}$

Début

$q \leftarrow \text{Allouer}(1)$

$q \rightarrow \text{val} \leftarrow M$

si ( $A \rightarrow \text{Tête} = \text{NULL}$  ou  $M.\text{deg} > A \rightarrow \text{Tête}.\text{val}.\text{deg}$ ) Alors

$q \rightarrow \text{suiv} \leftarrow A \rightarrow \text{Tête};$

$A \rightarrow \text{Tête} \leftarrow q$

Sinon

$p \leftarrow A \rightarrow \text{Tête}$

tant que ( $p \rightarrow \text{suiv} \neq \text{NULL}$  et  $p \rightarrow \text{suiv} \rightarrow \text{val}.\text{deg} > M.\text{deg}$ ) faire

$p \leftarrow p \rightarrow \text{suiv}$

fin tq

$q \rightarrow \text{suiv} \leftarrow p \rightarrow \text{suiv}$

$p \rightarrow \text{suiv} \leftarrow q$

fin si

fin

d/ procédure Multiscalaire ( $A: \text{Polynôme}; S: \text{Entier}$ )

var

$p: \text{cellule}$

Début

$p \leftarrow A.\text{Tête}$

tant que ( $p \neq \text{NULL}$ ) faire

$p \rightarrow \text{val}.\text{coef} \leftarrow p \rightarrow \text{val}.\text{coef} * S$

$p \leftarrow p \rightarrow \text{suiv}$

fin tq

fin

e) procedure MultPolynome (A: Polynome, M: Polynome)

var

p: \*cellule

Début

tant que (p <> NULL) f

p → val.coef ← p → val.coef \* M.coef

p → val.deg ← p → val.deg + M.deg

p ← p → suiv

fin tant q

fin

f/ fonction SommePoly (A, B: Polynome) = Polynome

var

C: Polynome

pa, pb, pc, q: \*cellule

Début

C.Tête ← NULL

pa ← A.Tête

pb ← B.Tête

tant que (pa <> NULL et pb <> NULL) faire

q ← Allouer (1)

si (pa → val.deg = pb → val.deg) Alim

q → val.deg ← pa → val.deg

q → val.coef ← pa → val.coef + pb → val.coef

pa ← pa → suiv

pb ← pb → suiv

sinon

si (pa → val.deg > pb → val.deg) Alim

q → val.deg ← pa → val.deg

q → val.coef ← pa → val.coef

pa ← pa → suiv

Si non  
 $q \rightarrow \text{val.deg} \leftarrow p_b \rightarrow \text{val.deg}$   
 $q \rightarrow \text{val.coef} \leftarrow p_b \rightarrow \text{val.coef}$   
 $p_b \leftarrow p_b \rightarrow \text{suiv}$

fin si

fin si

Si  $(C.\text{Tête} = \text{NULL})$  Alors

$q \rightarrow \text{suiv} \leftarrow \text{NULL}$

$C.\text{Tête} \leftarrow q$

$p_c \leftarrow C.\text{Tête}$

Si non

$p_c \rightarrow \text{suiv} \leftarrow q$

$q \rightarrow \text{suiv} \leftarrow \text{NULL}$

$p_c \leftarrow p_c \rightarrow \text{suiv}$

fin si

fin tq.

tant que  $(p_a \neq \text{NULL})$  faire

$q \leftarrow \text{Allouer}(1)$

$q \rightarrow \text{val} \leftarrow p_a \rightarrow \text{val}$

$q \rightarrow \text{suiv} \leftarrow \text{NULL}$

$p_a \leftarrow p_a \rightarrow \text{suiv}$

Si  $(C.\text{Tête} = \text{NULL})$  Alors

$C.\text{Tête} \leftarrow q$

$p_c \leftarrow C.\text{Tête}$

Si non

$p_c \rightarrow \text{suiv} \leftarrow q$

$p_c \leftarrow p_c \rightarrow \text{suiv}$

fin si

fin tq.



tantque ( $p \neq \text{NULL}$ ) f

$q \leftarrow \text{Allouer}(2)$

$q \rightarrow \text{val} \leftarrow p \rightarrow \text{val}$

$q \rightarrow \text{suiv} \leftarrow \text{NULL}$

$p \leftarrow p \rightarrow \text{suiv}$

si ( $c.\text{Tête} = \text{NULL}$ ) alors

$c.\text{Tête} \leftarrow q$

$p \leftarrow c.\text{Tête}$

sinon

~~$c.\text{Tête} \rightarrow \text{suiv} \leftarrow q$~~

$p \leftarrow p \rightarrow \text{suiv}$

fin si  
fin tantq

Somme Poly  $\leftarrow c$

fin

g) fonction EvaluerPoly ( $A : \text{Polynôme}; X : \text{réel}$ ) : réel

var

$p : \text{*cellule}$

$R : \text{réel}$

Début

$R \leftarrow 0$

$p \leftarrow A.\text{Tête}$

tantque ( $p \neq \text{NULL}$ ) faire

$R \leftarrow R + p \rightarrow \text{val} \cdot \text{coef} * (X^{p \rightarrow \text{val} \cdot \text{deg}})$

$p \leftarrow p \rightarrow \text{suiv}$

fin tantq

EvaluerPoly  $\leftarrow R$

fin

h / fonction Dérivée (A : Polynôme) : Polynôme

var

B : Polynôme

P : \*cellule

M : Monôme

Début

B.Tête ← NULL

P ← A.Tête

tant que (P ≠ NULL et P → val.deg > 0) f

M.Coeff ← P → val.Coeff \* P → val.deg

M.deg ← P → val.deg - 1

AjouterMonôme (B, M)

P ← P → Suiv

fin

Dérivée ← B

fin