Correction TD : Liste chainée

Exercice 02 :

1)

```
Joueur : Enregistrement
   nom : chaîne
   prenom : chaîne
   poids : réel
   num: entier
Fin Enreg

Cellule : Enregistrement
   val : Joueur
   suiv : *Cellule
Fin Enreg

Liste : Enregistrement
   Tête : *Cellule
Fin Enreg
```

2)

```
Fonction Recherche(L: Liste, x: entier) : Booléen
   var
      p : *Cellule
   début
      p <- L.Tête
      TantQue (p <> NULL et p->val.num <> x) faire
         p <- p->suiv
      FinTantQue
      Recherche <- (p <> NULL)
   fin
```

3)

```
Procedure Creer_list(L: *Liste)
   var
      p,q : *Cellule
      i : entier
   début
      L->Tête <- NULL
      Pour i de 1 à 15 faire
         q <- allouer(1)
         Écrire ("Donner les informations du joueur ", i, ":")
         Lire (q->val.nom)
```

```
        Lire (q->val.prenom)
        Lire (q->val.poids)
        Répéter
           Lire(q->val.num)
        jusqu'à ((q->val.num >= 1) et (q->val.num <= 15) et Recherche (*L, q->val.num) =
faux))
        q->suiv <- NULL
        Si (L.Tête = NULL) alors
           L->Tête <- q
           p <- L->Tête
        Sinon
           p->suiv <- q
           p <- p->suiv
        FinSi
     FinPour
  fin
```

4)

```
Fonction Plus_Lourds(L: Liste) : *Cellule
  var
     p,q : *Cellule
  début
     Si (L.Tête <> null) alors
        q <- L.teTêtete
        p <- L.Tête->suiv
        TantQue (p <> null) faire
           Si (p->val.poids > q->val.poids) alors
              q <- p
           FinSi
           p <- p->suiv
        FinTantQue
        Plus_Lourds <- q
     Sinon
        Plus_Lourds <- NULL
     FinSi
  fin
```

Exercice 03 :

1)

```
Processus : Enregistrement
  id: entier
  pr: entier
Fin Enreg
```

```
Cellule : Enregistrement
   val : Processus
   suiv : *Cellule
Fin Enreg
```

2)

```
Liste_proc : Enregistrement
   Tête : *Cellule
Fin Enreg
```

3)

```
Procedure Ajouter_processus(L: *Liste, p:Processus)
   var
      q,q1 : *Cellule

   début
      q->allouer(1)
      q->val <- p
      si((L ->tete = NULL) ou (p.pr < L->Tête->val.pr))alors
         q ->suiv <- L->Tête
         L ->Tête <- q
      Sinon
         q1<- L->Tête
         TantQue ((q1 ->suiv <> NULL) et (p.pr >= q1 ->val.pr)) faire
            q1 <- q1->suiv
         FinTantQue
         q->suiv <- q1->suiv
         q1->suiv <-q

      FinSi
   Fin
```

4)

```
Fonction Lancer_Processus(L: *Liste) : *Cellule
   var
      p: *Cellule
   début
      Si (L->Tête <> NULL) alors
         p <- L->Tête
         L->Tête->suiv <- L->Tête->suiv->suiv
         q->suiv <- NULL
      Sinon
         q <- NULL
      FinSi

      Lancer_Processus <- q
```

```
fin
```

Exercice 04 :

1)

```
Monome : Enregistrement
   coeff : réel
   deg : entier
Fin Enreg

Cellule : Enregistrement
      val : Monome
      suiv : *Cellule
Fin Enreg

Polynome : Enregistrement
   Tête: *Cellule
Fin Enreg
```

2)

```
Procédure Afficher(P : Polynome)
   var
      q:  *Cellule
   début
      q ← P.Tête
      Tant que (q <> NULL) faire
         Écrire(q->Val.Coeff, "x^", q->Val.deg,"+")
         q ← q.Suivant
      Fin Tant que
   Fin
```

3)

```
Fonction Copier_Polynome(A : Polynome) : Polynome
   var
      B : Polynome
      p,q,q1 :*Cellule

   B.Tête <- NULL
   P <- A.Tête
   début
      p <- A.Tête
      B.Tête <- NULL
```

```
    Tant que (P <> NULL) faire
       q <- Allouer(1)
       q->Val.coeff <- P->Val.coeff
       q->Val.deg <- P->Val.deg
       q->Suiv <- NULL

       Si (B.Tête = NULL) alors
          B.Tête ← q
          q1<- B.Tête
       Sinon
          q1->Suivant <- q
          q1 <- q
       Fin Si
       p <- p->Suivant
    Fin Tant que

    Copier_Polynome <- B
  Fin
```

4)

```
Procedure Ajouter_Monome(A : *Polynome; M : Monome)
  var
    p,q : *Cellule
  début
    q<- Allouer(1)
    q->Val <- M

    Si ((A->Tête = NULL) OU (M.deg > A->Tête.Val.deg) alors
       q->Suiv <- A->Tête
       A->Tête <- Q
    Sinon
       p<- A->Tête
       Tantque ((p->Suiv <> NULL) ET (p->Suiv->Val.deg > M.deg) faire
          p <- p->Suivant
       Fin Tantque
       q->Suiv <- p->Suivant
       P->Suiv <- q
    Fin Si
  Fin
```

5)

```
Procedure Multi_Monome(A : Polynome; M : Monome)
  var
    p:  *Cellule
```

```
   début

   Tantque (P <> NULL) faire
      p->Val.Coeff <- p.Val.coeff * M.coeff
      p->Val.deg <- p.Val.deg + M.deg
      p <- p->Suivant
   Fin Tantque
Fin
```

6)

```
Fonction Somme_Polynome(A : Polynome; 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) Alors
         q->Val.deg <- pa->Val.deg
         q->Val.coeff <- pa->Val.coeff + pb->Val.coeff
         pa <- pa->Suiv
         pb <- pb->Suiv
      Sinon
      Si (pa->Val.deg > pb->Val.deg) Alors
         q->Val.deg <- pa->Val.deg
         q->Val.Coeff ← pa->Val.Coeff
         pa <- pa->Suiv
      Sinon
         q->Val.deg <- pb->Val.deg
         q->Val.coeff <- pb->Val.coeff
         pb <- pb->Suiv
      Fin Si
      Fin Si

      Si (C.Tête = NULL) Alors
         q->Suiv <- NULL
         C.Tête ← q
```

```
        pc ← C.Tête
      Sinon
        pc->Suiv <- q
        q->Suiv <- NULL
        pc ← pc->Suivant
      Fin Si
   Fin Tant que

   Tant que (pa <> NULL) faire
      q<- Allouer(1)
      q->Val <- pa->Val
      q->Suiv <- NULL
      pa <- p->Suivant

      Si (C.Tête = NULL) Alors
        C.Tête <- q
        pc <- C.Tête
      Sinon
        pc->Suiv <- q
        pc <- pc->Suiv
      Fin Si
   Fin Tant que

   Tant que (pb <> NULL) faire
      q<- Allouer(1)
      q->Val <- pb->Val
      q->Suiv <- NULL
      pb <- pb->Suiv

      Si (C.Tête = NULL) Alors
        C.Tête <- q
        pc <- C.Tête
      Sinon
        pc->Suiv <- q
        pc <- pc->Suiv
      Fin Si
   Fin Tant que

   Somme_Polynome <- C
Fin
```

7)

```
Fonction Evaluer_Polynome(A : Polynome; X : réel) : réel
   var
```

```
    P : *Cellule
    R : réel

  R <- 0
  P <- A.Tête

  Tantque (P <> NULL) faire
    R <- R + P->Val.Coeff * (X^(P->Val.deg))
    P <- P->Suivant
  Fin Tantque

  Evaluer_Polynome <- R
Fin
```

8)

```
Fonction Deriver(A :Polynome) : Polynome
  var
    B : Polynome
    P : *Cellule
    M : Monome
  début

  B.Tête <- NULL
  P ← A.Tête

  Tantque ((P <> NULL) et (P->Val.deg > 0)) faire
    M.Coeff <- P->Val.Coeff * P->Val.deg
    M.deg <- P->Val.deg - 1
    Ajouter_Monome(&B, M)
    P <- P->Suivant
  Fin Tantque

  Deriver <- B
Fin
```