

Exercice N° :06

Algorithme exo 06 ;

Type Poly= ^cellule ;

Type cellule= enregistrement ;

Coef :entier ;

Expo : entier ;

Svt : Poly ;

Fin_enreg ;

Variable

T1, T2,T3,Q 3, P1,P2,Nouv : Poly

B : booleen ;

Debut

/* on suppose que les polynômes T1 et T2 existent déjà */

P1<-T1 ; P2<-T2 ; B<-Faux ; T3<-nil ;Q3<-nil ;

Tant que ((P1≠nil)et(P2≠nil))

Faire

Nouv<- Allouer(TaillDe(cellule)) ;

Si ((^P1).expo=(^P2).expo) alors Debut

(^nouv).expo<- (^P1).expo ;

(^nouv).coef<- (^P1). Coef+(^P2). Coef;

(^nouv).Svt<- nil;

Si (T3=nil) alors debut T3<-nouv ; Q3<-nouv ; fin ;

Sinon debut (^Q3).svt<-nouv ; Q3<-nouv ; fin si ;

P1<-(^P1).Svt ; P2<-(^P2).Svt ;

Fin ;

Sinon si ((^P1).expo >(^P2).expo) alors Debut

(^nouv).expo<- (^P1).expo ;

(^nouv).coef<- (^P1). Coef ;

(^nouv).Svt<- nil;

Si (T3=nil) alors debut T3<-nouv ; Q3<-nouv ; fin ;

Sinon debut (^Q3).svt<-nouv ; Q3<-nouv ; fin si ;

P1<-(^P1).Svt ;

Fin ;

Sinon Debut

(^nouv).expo<- (^P2).expo ;

(^nouv).coef<- (^P2). Coef ;

(^nouv).Svt<- nil;

Si (T3=nil) alors debut T3<-nouv ; Q3<-nouv ; fin ;

Sinon debut (^Q3).svt<-nouv ; Q3<-nouv ; fin si ;

P2<-(^P2).Svt ;

Fin ;

Fsi ;

Fsi ;

Fait ;

/*le polynôme 2 se termine et le polynôme 1 non */

Tant que (P1≠nil)

Faire

(^nouv).expo<- (^P1).expo ;

(^nouv).coef<- (^P1). Coef ;

(^nouv).Svt<- nil;

Si (T3=nil) alors debut T3<-nouv ; Q3<-nouv ; fin ;

Sinon debut (^Q3).svt<-nouv ; Q3<-nouv ; fin si ;

P1<-(^P1).Svt

Fait ;

/*le polynôme 1 se termine et le polynôme 2 non */

Tant que (P2≠nil)

Faire

(^nouv).expo<- (^P2).expo ;

(^nouv).coef<- (^P2). Coef ;

(^nouv).Svt<- nil;

Si (T3=nil) alors debut T3<-nouv ; Q3<-nouv ; fin ;

Sinon debut (^Q3).svt<-nouv ; Q3<-nouv ; fin si ;

P2<-(^P2).Svt

Fait ;

Fin ; /* fin algorithme*/

Exercice N° : 08

Type ST= enregistrement

PE : entier ;
 PD : entier ;
 Expo : entier ;
 Fin_enreg ;

1-

Type liste=^Cellule ;
 Type Cellule=enregistrement
 Nbre :ST ;
 Svt :Liste ;
 Fin_enreg ;

Procédure Construit (E/S Tete : Liste)

Variables

Nouv : Liste ;
 A :St ;

Debut

Lire(A.PE, A.PD, A.Expo) ; Tete<-nil ;

Tant que (A.Expo≠0)

Faire

 Nouv<-Allouer(TailleDe(Cellule)) ;
 (^Nouv).Nbre<-A ; /*affecter un enregistrement dans un enregistrement*/
 (^Nouv).Svt<-Tete ;
 Tete<-Nouv ;
 Lire(A.PE, A.PD, A.Expo) ;

Fait ;

Fin ;

2-

Fonction Calcul (E/ A :St ;) :réel

Variable

X :reel ;

Debut

X<-A.PD ;

Tant que (X>1) faire X<-X/10 ; Fait ;

Si (A.PE>0) alors X<-A.PE+X ; sinon X<-A.PE-X ; Fsi

Si (A.Expo>0) alors Tant que (Expo>0) faire X<-X*10 ; Expo<-Expo-1 ;Fait ;

 Sinon Tant que (Expo<0) faire X<-X /10 ; Expo<-Expo+1 ;Fait ;

Fsi ;

Retourner(X) ;

Fin ;

Algorithme Exo 8 ;

/* Déclaration des Types*/

Type ST= enregistrement

PE : entier ;
 PD : entier ;
 Expo : entier ;
 Fin_enreg ;

Type liste=^Cellule ;

Type Cellule=enregistrement
 Nbre :ST ;
 Svt :Liste ;
 Fin_enreg ;

/*Déclaration des variables*/

Variables

T,P :Liste ;

n,i :entier ;

V :^reel ; /* V est un pointeur sur les réel et qui représente un tableau de réel*/

/*Déclaration des fonctions et des procédures*/

Procédure Construit (E/S Tete : Liste) ;

Fonction Calcul (E/ A :St ;) :réel ;

Début

/* la construction de la liste qui contient des éléments de type St */

/* Appel à la procédure Construit */

Construit(T) ;

/* l'allocation dynamique du Tableau V, il faut calculer le nombre des éléments de la liste T */

P<-T ; n<-0 ; Tant que (P≠nil) faire n<-n+1 ; P<-(^P).Svt ; fait ;

V<-Allouer (n*TailleDe(reel)) ; /* n est la dimension du tableau */

/* remplissage du tableau V en utilisant la fonction calcul et en parcourant la liste des éléments st */

P<-T ; i<-1 ;

Pour (i<-1 à n) faire **V[i]<-calcul((^P).Nbre)** ; P<-(^P).Svt ; fait ;

/* Affichage du Tableau */

Pour (i<-1 à n) faire écrire (V[i]) ;

Fin ;