Type nœud en enregistrement

Champ donnée en caractère

Champ gauche, droite en Pointeur de nœud

Fin d’enregistrement

Car = ‘ ‘

Ecrire (‘’Entrez une expression postfixée’’)

// Initialisation de la pile

Initialiser\_pile()

// Construction de l’arbre

// On boucle tant que la fin de ligne n’est pas atteinte

Tant que (non fin\_de\_ligne()) Faire

Lire(car)

Si (non fin\_de\_ligne()) lors

// création d’un nouveau nœud

Nouveau\_noeud = Allouer(noeud)

(\*Nouveau\_noeud).donnée = car

(\*Nouveau\_noeud).gauche = nil

(\*Nouveau\_noeud).droite = nil

Si ((car == ‘+’) OU (car == ‘-’) OU (car == ‘\*’) OU (car == ‘/’))

(\*Nouveau\_noeud).gauche = dépiler

(\*Nouveau\_noeud).droite = dépiler

FinSi

Si (car <> ‘ ‘) Alors

Empiler(Nouveau\_noeud)

FinSi

FinSi

FinTantQue

Racine\_arbre = dépiler()

Procédure ParcoursPrefixe(pt\_noeud)

Déclaration

Paramètre pt\_nœud en Pointeur de nœud

Début

Si (pt\_nœud <> Nil) Alors

Traiter(pt\_nœud)

ParcoursPrefixe((\*pt\_nœud).gauche)

ParcoursPrefixe((\*pt\_nœud).droit)

FinSi

Fin

Procédure ParcoursInfixe(pt\_noeud)

Déclaration

Paramètre pt\_nœud en Pointeur de nœud

Début

Si (pt\_nœud <> Nil) Alors

ParcoursInfixe((\*pt\_nœud).gauche)

Traiter(pt\_nœud)

ParcoursInfixe((\*pt\_nœud).droit)

FinSi

Fin

Procédure ParcoursPostfixe(pt\_noeud)

Déclaration

Paramètre pt\_nœud en Pointeur de nœud

Début

Si (pt\_nœud <> Nil) Alors

ParcoursPostfixe((\*pt\_nœud).gauche)

ParcoursPostfixe((\*pt\_nœud).droit)

Traiter(pt\_nœud)

FinSi

Fin

Procédure ParcoursParNiveaux(pt\_noeud)

Déclaration

Paramètre pt\_nœud en Pointeur de nœud

Début

Enfiler(pt\_nœud)

Tant que (non file\_vide()) Faire

pt\_nœud = défiler

traiter(pt\_nœud)

Si ((\*pt\_nœud).gauche <> nil) Alors

Enfiler((\*pt\_nœud).gauche)

FinSi

Si ((\*pt\_nœud).droit <> nil) Alors

Enfiler((\*pt\_nœud).droit)

FinSi

FinTantQue

Fin