

Solution TD5 - ABD2 - les piles et les files
LGL811 / LIPIS1

Ex1:

3/ procédure Inverse($T: \text{TAB}; N: \text{Entier}$)

var

$P: \text{Pile}$

$i: \text{Entier}$

Début

Init_Pile($\&P$)

pour i de 1 à N faire

Empiler($\&P, T[i]$)

Fin pour

pour i de 1 à N faire

$T[i] \leftarrow \text{Dépiler}(\&P)$

Fin pour

Fin

Ex2:

1) procédure Inverse_Pile($P: \text{Pile}$)

var

$P_1, P_2: \text{Pile}$

Début

Init_Pile($\&P_1$)

Init_Pile($\&P_2$)

tant que ($\text{Vide}(\&P) = \text{faux}$) faire

Empiler($\&P_2, \text{Dépiler}(\&P)$)

Fin tant que

tant que ($\text{Vide}(P_1) = \text{faux}$) faire

Empiler($\&P_2, \text{Dépiler}(\&P_1)$)

Fin tant que

tant que ($\text{Vide}(P_2) = \text{faux}$) faire

Empiler($P, \text{Dépiler}(\&P_2)$)

Fin

2) fonction Copie_Pile (P: Pile): Pile

var

~~Q, Q1~~ : Pile

Début

Init_Pile(Q)

Init_Pile(Q1)

Tantque (Vide(P) = faux) faire

Empiler(&Q1, Dépiler(P))

fin_tq

Tantque (Vide(Q1) = faux) faire

Empiler(&Q, Dépiler(Q1))

fin_tq

Copie_Pile \leftarrow Q

fin

3) procédure Range (T: TAB; N: Entier

var

P, Q: Pile

i: Entier

Début

Init_Pile(&P)

Init_Pile(&Q)

pour i de 1 à N faire

si (T[i].Nod2 = 0) Alim

Empiler(&P, T[i])

sinon

Empiler(&Q, T[i])

finpour

$i^0 \leftarrow N$

tant que (vide(Q) = faux)

$T[i] \leftarrow \text{Dépiler}(\&Q)$

$i \leftarrow i - 1$

fin tq

tant que (vide(P) = faux) faire

$T[i] \leftarrow \text{Dépiler}(\&P)$

$i \leftarrow i - 1$

fin tq

fin

Ex 3:

procédure Trier_Fil (F: *File)

Var

P, Q: Pile

X: Entier

Début

Init_Pile(&P)

Init_Pile(&Q)

Tant que (vide(*F) = faux) faire

Empiler(&Q, Défiler(F))

fin tq

Tant que (vide(Q) = faux) faire

$X \leftarrow \text{Dépiler}(\&Q)$

Tant que (vide(P) = faux et sommet(P) \neq X) faire

Empiler(&Q, Dépiler(&P))

fin tq

Empiler(&P, X)

fin tq

Tant que (vide(P) = faux) faire

Enfiler(F, Dépiler(&P))

Fin

fin tq

(3)

Ex4:

fonction Evaluer(E: chaîne): Entier

var

P: Pile

i, X, Y, Z: Entier

Début

Init_Pile(&P)

pour i de 1 à long(E) faire

selon (E[i]) faire

+ : Y ← Dépile(&P)
X ← Dépile(&P)

- : Z ← X + Y (&Z)
Y ← Dépile(&P)
X ← Dépile(&P)

* : Z ← X - Y (&P, Z)
Y ← Dépile(&P)
X ← Dépile(&P)

/ : Z ← X * Y (&P, Z)
Y ← Dépile(&P)
X ← Dépile(&P)

^ : Z ← X / Y (&P, Z)
Y ← Dépile(&P)
X ← Dépile(&P)

Sinon : Empiler(&P, E[i])

fin selon

fin pour

Evaluer ← Dépile(&P)

fin

EX5

Algorithme Ranger

Var

P, Q, W : Pile

Début

Init_Pile (&Q)

Init_Pile (&W)

Tantque (Vide(P) = faux) faire

si (sommet(P) mod 2 = 1) Alon

Empiler(Q, Dépiler(P))

sinon

Empiler(W, Dépiler(P))

fin si

fin tantq

tantque (Vide(W) = faux

Empiler(&Q, Dépiler(&W))

fin tantq

fin