(1)

(2)

(3)

(4)

(5)

(6)

Ex 2

Algo :

Procedure trieBoustrophedon(in\_out tab tableau[1..n]; deb,fin : 1..n )

Pour tout j de deb+1 a fin-1

Pour tout i de j a fin-j

Si tab[j - i] < tab[j – i - 1]

Alors échanger(tab[fin-i],tab[fin-i-1])

🡨(1)

FinSi

Si tab[i] > tab[i+1]

Alors échanger(tab[i],tab[i+1])

🡨(2)

FinSi

🡨(3)

FinPour

🡨(4)

Fin Pour

🡨 (5)

Semantique :

(1) : tab[fin-j-i-1]

(2) :

(3) :tab[fin-j-i-1] ⋀ // afaire

(4) : ⋀

(5) : ⋀ t[fin-d] > t[fin-d-1] 🡨 les deux assertions du ⋀ sont équivalentes

Ex 3