Série classe récursivité ex 7

Début
répéter
p ← pos(c, ch)
si p ≠ -1 alors
effacer(ch, p, p+1)
finsi
jusqu'à (p == -1)
Fin

ch = "aggregative", c = "g"
p ← pos(c, ch)
=1
p=-1 ? x → ret occ (effacer ↔ "aggregative")
↳ p ← 1 ≠ -1 → ret occ("aggregative")
↳ p ← 3 ≠ -1 → ret occ("areative")
↳ p ← -1 → "areative"

ex 6 : anagrammes :

ch1 = "arbre", ch2 = "arbre"

solution it :

fonction anagrammes(ch1, ch2 : ch) : bool
Début
i ← 0
tantque (i < long(ch1)) et (pos(ch1[i], ch2) ≠ -1)
i ← i + 1
ch2 ← effacer(ch2, pos(ch1[i], ch2), pos(ch1[i], ch2) + 1)
fin tq
retourner ch2 ↔ ""

ch1 = "star", ch2 = "rats"

pos(s, ch2) = -1 ? x

⇒ ch2 ← "rats"

pos(i, ch2) ≠ -1 ⇒ ch2 ← "ra"

pos(a, ch2) ≠ -1 ⇒ ch2 ← "ra" + a

pos(a, ch2) ≠ -1 ⇒ ch2 ← "ra" + a

⇒ ret ch2

---

sol it : fonction anagramme (ch1, ch2 : ....):

Début

i ← 0

tantque (ch2 ≠ "") et (pos(ch1[i], ch2) ≠ -1) ft

i ← i + 1

ch2 ← effacer (ch2, pos(ch1[i], ch2), pos(ch1[i], ch2) + 1)

ftq

ret ch2 ← ///