

Parcial CAP - Possibles Solucions

Curs 2025-26 (3-XI-2025)

Duració: 2 hores

1.- Clojure - Seqüència Estrictament Creixent -- T53479

a/ (1.5 punts)

Solució: Una possible solució és aparellar els elements consecutius de la seqüència. N'hi havia prou amb escriure:

```
(defn creixent [s]
  (let [ss (map vector s (rest s))] ...))
```

b/ (1.5 punts)

Solució: Ara, una aplicació *directa* de la propietat universal del fold (analitzant l'estructura d'auxiliar) donava lloc a:

```
(defn creixent [s]
  (let [ss (map vector s (rest s))]
    (fold #(and (< (first %1) (second %1)) %2) true ss)))
```

2.- Clojure - Nombres romans -- T26671

En tots dos casos farem servir un diccionari i una funció auxiliar:

```
(def valors {\I 1 \V 5 \X 10 \L 50 \C 100 \D 500 \M 1000})

(defn addOrSub [c1 c2 x]
  (let [v1 (valors c1)
        v2 (valors c2)]
    (if (>= v1 v2)
      (+ x v1)
      (- x v1))))
```

- Recursiu (1.75 punts):

Solució: Una possible solució és:

```
(defn roman2int [s]
  (let [cnt (count s)]
    (cond
      (zero? cnt) 0
      (= 1 cnt) (valors (first s))
      :else
      (let [[c1 c2 & cs] s]
        (addOrSub c1 c2 (roman2int (apply str (conj cs c2))))))))
```

- Iteratiu (1.75 punts):

Solució: Una possible solució és:

```
(defn roman2int2 [s]
  (if (zero? (count s))
    0
    (loop [cadena s acc 0]
      (if (= 1 (count cadena)) (+ acc (valors (first cadena)))
        (let [[c1 c2 & cs] cadena]
          (recur (apply str (conj cs c2)) (addOrSub c1 c2 acc)))))))
```

3.- Clojure - A Base 10 -- U36688 (3.5 punts)

Solució: Una possible solució, seguint els problemes que us vam posar per a que practiquèssiu les *closures* sense estat mutable (l'assert en realitat NO cal):

```
(defn transforma_a_base_10 "1 <= base <= 9" [base]
  (letfn [(ff [y]
            (letfn [(gg [d]
                      (assert (or (= -1 d) (and (<= 0 d) (< d base))))
                      (if (= -1 d)
                        y
                        (ff (+ (* base y) d))))
              gg))]
    (ff 0)))
```

Fixeu-vos en que és ben bé com la solució del repetits que vam veure a classe, i de l'aplica i el filtra.