

- récursion sur des `IntList`, ne pas oublier `IntCons(head, tail) =>` et pas `head :: tail -=>` comme dans une liste normale.
- on peut faire un `case IntCons(head, tail) if head > 0 =>` (ajouter un test supplémentaire dans le `case`).
- on utilise pas `Math.max` mais `scala.math.max`.
- ne pas oublier que les chaînes de caractères `String` disposent des propriétés `.head`, `.tail`, etc.
- on peut créer des contextes :

```
enum Context:
  case Empty
  case Cons(name: String, value: Int, rem: Context)
```

Une clef associée à une valeur dans un contexte est appelée un “Binding”.