



**Exercici 2.** Assumiu que el tipus de dades `llista_simple` és una simplificació del concepte de llista tal que els elements són caràcters, la mida màxima està afitada i les seves operacions responen a la següent especificació:

- `llista_buida()` Retorna una llista buida. I.e.:  $llista\_buida() = @$ .
- `next(l)` Retorna una llista  $l$  en la que el cursor ha avançat cap a la dreta. I.e.: si  $l = e_1 \dots e_{i-1} @ e_i e_{i+1} \dots e_n$  llavors  $next(l) = e_1 \dots e_{i-1} e_i @ e_{i+1} \dots e_n$ ; singularment, si  $l = e_1 \dots e_n @$  llavors  $next(l) = e_1 \dots e_n @$ .
- `previous(l)` Retorna una llista en la que el cursor ha avançat cap a l'esquerra. I.e.: si  $l = e_1 \dots e_{i-1} e_i @ e_{i+1} \dots e_n$  llavors  $previous(l) = e_1 \dots e_{i-1} @ e_i e_{i+1} \dots e_n$ ; singularment, si  $l = @ e_1 \dots e_n$  llavors  $previous(l) = @ e_1 \dots e_n$ .
- `at_begining(l)` Retorna cert ssi  $l = @ e_1 \dots e_n$  o  $l = @$ .
- `at_end(l)` Retorna cert ssi  $l = e_1 \dots e_n @$  o  $l = @$ .
- `get(l)` Retorna l'element de la llista  $l$  indicat pel cursor. I.e.: si  $l = e_1 \dots @ e_i \dots e_n$ , llavors  $get(l) = e_i$ . Noteu que si  $l = e_1 \dots e_n @$  o  $l = @$ , llavors  $get(l)$  no es pot aplicar.
- `put(l,e)` Afegeix l'element  $e$  a la llista  $l$  a la posició indicada pel cursor. I.e.: si  $l = e_1 \dots e_{i-1} @ e_i \dots e_n$ , llavors  $put(l, x) = e_1 \dots e_{i-1} @ x e_i \dots e_n$ .

Proposeu el fitxer de `headers` `llista.h` que correspondria a un mòdul que implementa aquesta llista. En aquest header no us preocupeu per la definició de tipus que correspon a la llista, que s'aborda en el següent exercici.

**Exercici 3.** Completeu el mòdul anterior. Considereu que mai arribeu a la mida màxima de la llista. Cal que implementeu la llista assumint que teniu disponible un mòdul que ofereix el tipus pila. La idea és implementar una llista basant-se en dues piles. No és necessari que implementeu el mòdul pila però sí que n'heu d'escriure el corresponent `header`. Seguiu el següent esquema de treball:

1. Escriviu el `header` `pila.h` corresponent al mòdul pila.
2. Deduïu com es pot implementar una llista basant-se en dues piles.
3. Completeu el `header` `llista.h` de l'exercici anterior amb la definició de tipus que correspon a la llista.
4. Escriviu el fitxer d'implementació del mòdul llista, `llista.c`, implementant totes les operacions amb el significat que s'ha dit.