

PROGRAMMAZIONE I (A,B) - a.a. 2019-20

Seconda verifica intermedia - 16/12/2019

Esercizio 1

Si definisca in CAML, **usando foldr**, una funzione

```
loneq : 'a list -> int
```

che, data una lista, restituisce la **lunghezza** della sequenza più lunga di elementi contigui uguali.

Esempi: `loneq [-2; 2; 2; 2; 3; -10; -10; 2; 14] = 3`,

`loneq [12; 3; 4; -10; 14] = 1`

Esercizio 2

Si definisca in CAML, **usando la ricorsione esplicita**, la funzione

```
split : int list -> int list * int list
```

che data una lista di interi restituisce due liste. La prima con i valori che precedono la prima occorrenza del valore 0, la seconda con gli altri valori. Se il valore 0 non occorre, non ci sono valori che precedono la sua occorrenza.

Esempi: `split [-3; 4; 6; 0; 7; 0; 10; 3] = ([-3; 4; 6], [0; 7; 0; 10; 3])`

`split [-3; 4; 6; 7;10; 3] = ([], [-3; 4; 6; 7;10; 3])`

`split [0; -3; 4; 6; 7;10; 3] = ([], [0; -3; 4; 6; 7;10; 3])`

Esercizio 3

Si definisca in CAML, **usando la ricorsione esplicita**, la funzione

```
two : 'a list -> 'a list
```

che date una lista `l`, restituisce la lista dei valori che compaiono in `l` esattamente due volte.

Esempi: `two [3; 4; 6; 0; 7; 0; 10; 3] = [3; 0; 0; 3]`

`split [-3; 4; 6; 7;10] = []`