

Primo Compitino di Programmazione - 16 Novembre 2015

Cognome Nome Matricola

Esercizio 1. Definire una funzione ricorsiva `zip: int list -> int list` che data una lista di interi `l` restituisca la lista ottenuta da `l` sostituendo ogni sequenza di elementi uguali e contigui in `l` con un unico elemento uguale alla somma degli elementi nella sequenza.

Per esempio:

```
# zip [1;1;2;2;2;2;3;3;4;3;3;3];;  
- : int list = [2;8;6;4;9].  
  
# zip [];;  
- : int list = []
```

Esercizio 2. Scrivere una funzione `count_even` che data una lista di liste di interi `ls` restituisca la lista di interi che indica il numero di elementi pari in ciascuna lista appartenente a `ls`.

Per esempio:

```
# count_even [[1; 2]; [1; 2; 3; 4]; []];;  
- : int list = [1; 2; 0]  
  
# count_even [[]];;  
- : int list = [0]  
  
# count_even [];;  
- : int list = []
```

Infine si scriva il tipo della funzione `count_even`.

Esercizio 3. Definire una funzione `sum_factors: int -> int` che dato un numero intero non negativo `n` restituisca la somma dei fattori primi di `n` (escluso 1). Per esempio:

```
# sum_factors 6;;  
- : int = 5  
  
# sum_factors 7  
- : int = 7  
  
# sum_factors 0;;  
- : int = 0  
  
# sum_factors 1;;  
- : int = 0  
  
# sum_factors 2;;  
- : int = 2
```