

Esame di Programmazione – 13 Settembre 2011
Parte 1: Programmazione Funzionale

Cognome Nome Matricola

Esercizio 1 Si consideri la seguente funzione che implementa l'algoritmo di mergesort :

```
let rec mergesort lst =  
  match lst with  
  | [] -> []  
  | [x] -> [x]  
  | _ -> let (left, right) = split list in  
         merge(mergesort left, mergesort right);;
```

Tale algoritmo ordina una lista nel seguente modo:

- la lista (se di almeno due elementi) viene suddivisa in due liste, per mezzo della funzione `split`;
- queste due liste vengono (ricorsivamente) ordinate;
- infine a partire dalle due liste ordinate si ottiene un'unica lista ordinata confrontando via via i primi elementi di queste, per mezzo della funzione `merge`.

Scrivere i tipi delle funzioni `mergesort`, `split` e `merge`.

Esercizio 2 Si definisca la funzione `split` dell'Esercizio 1 in modo tale da ottenere una qualunque suddivisione equilibrata, per esempio mettendo alternativamente gli elementi da una parte o dall'altra.

Esercizio 3 Si definisca la funzione `merge` dell'Esercizio 1.

Esercizio 4 Si definisca, precisando il tipo, una funzione polimorfa `filtermap` che data una lista `l` e un predicato `p`, restituisca la liste dei valori ottenuta applicando la funzione `f` agli elementi di `l` su cui vale `p`.

Esercizio 5 Si utilizzi la funzione `filtermap` dell'esercizio precedente per ottenere una funzione `firsts` che data una lista di liste restituisce la liste dei primi elementi delle liste non vuote.

Per esempio:

```
firsts [[1;2];[5;6;7];[];[];[9;8;0]] = [1;5;9];;
```