

Programmazione funzionale (OCaml)

<https://taroccoesbrocco.github.io/progfunz.html>

Esonero 1 — 29 Aprile 2025

Esercizio 1 (15 punti)

1. (10 punti) Definire una funzione `reverse_to_list : string -> char list` tale che `reverse_to_list s` restituisca la lista dei caratteri nella stringa `s` letta in ordine inverso. Per esempio, `reverse_to_list "ciao" = ['o'; 'a'; 'i'; 'c']`. Si può utilizzare la funzione `String.length : string -> int` che restituisce la lunghezza di una stringa.
2. (3 punti) Definire una funzione `reverse : string -> string` tale che `reverse s` restituisca la stringa `s` in ordine inverso. Per esempio, `reverse "ciao" = "oaic"`. Anche se non sono state definite, si possono utilizzare:
 - la funzione `reverse_to_list` del punto precedente;
 - la funzione `implode : char list -> string` tale che `implode lst` restituisce la stringa composta dai caratteri della lista di caratteri `lst`, nello stesso ordine. Per esempio, `implode ['c'; 'i'; 'a'; 'o'] = "ciao"`.
3. (2 punti) Definire una funzione `palindrome : string -> bool` tale che `palindrome s` restituisca `true` se la stringa `s` è palindroma, `false` altrimenti. Una stringa è palindroma se, letta al contrario, rimane invariata. Per esempio, `palindrome "samas" = true` mentre `palindrome "saman" = false`. Si può utilizzare la funzione `reverse` del punto precedente anche se non è stata definita.

Esercizio 2 (10 punti)

Definire una funzione `sumOddSquares : int list -> int` tale che `sumOddSquares lst` calcoli la somma dei quadrati degli interi dispari nella lista `lst`. Per esempio, `sumOddSquares [1;2;3;4;5;6] = 35`. Si possono utilizzare le funzioni `List.map`, `List.filter`, `List.fold_left`, e `List.fold_right` viste a lezione.

È preferibile (per ottenere il punteggio massimo)

- usare funzioni anonime o locali invece di definire più funzioni,
- evitare di usare il pattern matching.

Esercizio 3 (10 punti)

Si supponga di aver immesso le seguenti dichiarazioni in OCaml, nell'ordine in cui sono scritte.

```
let x = 1 ;;
let f y =
  function z -> x+y+z ;;
let x = 3 ;;
```

1. (5 punti) Qual è il tipo di `f`?
2. (5 punti) Se si aggiunge infine la dichiarazione `let w = (f 4) 6 ;;`, quali sono il tipo e il valore di `w`?

Giustificare le risposte (ad esempio, rappresentando l'evoluzione dell'ambiente nel corso della sessione per determinare il valore di `w`).