

## Studente Filippo Callegari

**Attenzione:** una volta letto il presente testo è *obbligatorio* consegnare alla scadenza quanto elaborato, indipendentemente dal fatto che lo si consideri adeguato o meno.

Entro il termine stabilito si *deve* inviare via email una relazione in formato PDF dal nome

“ProgettoLC parte2b Filippo Callegari Relazione”.

La relazione può contenere immagini passate a scanner di grafici o figure fatte a mano.

**Per tutti gli esercizi** si richiede una descrizione dettagliata di tutte le tecniche **non**-standard impiegate (le tecniche standard imparate a lezione non vanno descritte).

**Per tutti gli esercizi** si richiede una descrizione delle assunzioni fatte riguardo alla specifica, sia relativamente a scelte non previste espressamente dalla specifica stessa, che a scelte in contrasto a quanto previsto (con relative motivazioni).

Non è assolutamente utile perdere tempo per includere nella relazione il testo dei vari esercizi o un suo riassunto o una qualsiasi rielaborazione, incluso descrizioni del problema da risolvere. Ci si deve concentrare solo sulla descrizione della soluzione e delle eventuali variazioni rispetto a quanto richiesto.

### Esercizio 1

---

Si consideri il seguente programma Haskell

```
fact :: Integer -> Integer
fact n = if n==0 then 1 else n * fact (n-1)

take :: Int -> [a] -> [a]
take _ [] = []
take n _ | n <= 0 = []
take n (x:xs) = x : take (n-1) xs

map f (x:xs) = f x : map f xs
map _ [] = []
```

Si rappresenti la sequenza di esecuzione di

```
let v=fact k; k=j-2; j=2*2 in take (5-k)(v : map fact [j-k, v, v, error "q",v])
```

Si assuma di essere in un contesto di esecuzione che forzi la valutazione *completa* del valore di suddetta query.

Nella soluzione, per rendere univoca l'applicazione delle occorrenze dei metodi di classe, si utilizzi la convenzione di denotare con **meth#type** una occorrenza del metodo **meth** ottenuta istanziando il parametro di tipo della classe con il tipo primitivo **type** (per esempio **+#Int** qualora il metodo **+ :: Num a=>a->a->a** venga usato nel caso **a = Int** per il contesto **Num a**). Si raccomanda di limitarsi alle istanze di metodo, *evitando di adornare le funzioni che non sono metodi* con il tipo dato dai parametri attuali.

Qualora si usasse una notazione lineare per rappresentare le espressioni dei passi di esecuzione, si abbia cura di usare una forma per cui non ci siano ambiguità relativamente alle (sotto)espressioni condivise.

### Esercizio 2

---

Caratterizzare tutti gli (eventuali) overlap delle regole dell'esercizio 1 e dell'esercizio 1 della parte 2a. Nello specifico, per ogni coppia di regole specificare se non sono in overlap oppure (in caso lo siano) si fornisca il testimone più generale (unico a meno di varianza).