

Esame di LP1

16 Giugno 2016

Domande generali – Max 8 punti

Esercizio 1: [4 punti] Barrare tutte le frasi vere.

1. Il predicato Prolog *member* può enumerare i membri di una lista e generare liste [X]
2. Tra i compiti del supporto a run time c'è la gestione della memoria [X]
3. I linguaggi funzionali non sono computazionalmente completi []
4. In C++ l'ereditarietà multipla può causare un consumo esponenziale di memoria [X]

Esercizio 2: [2 punti] Calcolare la sommatoria di una lista di interi in stile funzionale puro supponendo che le funzioni *vuota*, *testa*, *coda* per manipolare la lista siano già implementate (usare uno dei linguaggi C, Java, Scheme o Lisp oppure pseudocodice)

Alcune soluzioni:

```
int sum( L ) {  
  if( vuota(L) ) return 0;  
  else return testa( L ) + sum( coda(L) );  
}
```

```
int sum( L ) {  
  return vuota(L) ? 0 : testa( L ) + sum( coda(L) );  
}
```

```
(defun sum (L)  
  (cond ((null L) 0)  
        (T (+ (testa L) (coda L)))))
```

Esercizio 3: [2 punti] Disegnare i data object relativi al seguente codice C:

```
int x = 3, *y, **z;  
y = &x;  
z = &y;
```

