

Fondamenti di Informatica T-1 (A.A. 2008/2009) - Ingegneria Informatica
Prof.ssa Mello
Prova Parziale d'Esame di Giovedì 11 Dicembre 2008 – durata 1h
Totale 12 punti, sufficienza con 7 - Compito A

ESERCIZIO 1 (5 punti)

Si supponga di avere a disposizione, già definito, l'ADT lista per interi (denominato `list`, con relative primitive). Il candidato definisca una funzione ricorsiva

```
list salta(list l1);
```

La funzione deve restituire una nuova lista contenente i valori di `l1` selezionati secondo un preciso criterio. In particolare, nella nuova lista dovrà essere inserito ogni elemento `e1` di `l1` che è seguito immediatamente da un valore multiplo esatto di `e1`. Inoltre, il valore multiplo esatto immediatamente consecutivo ad `e1` non dovrà essere inserito nella nuova lista.

Ad esempio, se `l1 = {5, 2, 6, 18, 3, 4, 8}`, la funzione restituirà la lista `{2, 4}`. Infatti, il valore `2` è seguito dal valore `6` che è un suo multiplo esatto, e quindi `2` viene selezionato. A questo punto il valore `6` non viene più considerato (nonostante il valore successivo `18` sia un suo multiplo esatto). Infine, `4` è seguito da `8`, che è un multiplo esatto e va quindi incluso nella lista risultato.

ESERCIZIO 2 (3 punti)

Data la funzione:

```
int swap(float a, int b){
    if ( (a>b) && (a>0))
        return a;
    else
        return swap(a+0.5, b+0.5) + swap(a+1.5, b) + 1;
}
```

mostrare la sequenza dei record di attivazione nel caso in cui la funzione sia invocata con parametri attuali `(-0.5, -2.0)` e il valore di ritorno. Si ignorino eventuali warning che potrebbero essere generati in fase di compilazione.

ESERCIZIO 3 (2 punti)

Un elaboratore rappresenta i numeri interi su 8 bit tramite la notazione in complemento a 2. Indicare come viene svolta la seguente operazione aritmetica calcolandone il risultato secondo la rappresentazione binaria in complemento a 2 (si trasli anche il risultato in decimale per verificare la correttezza dell'operazione):

`99 + (-56)`

ESERCIZIO 4 (2 punti)

Si descriva rapidamente il concetto di tipo di dato astratto coda, si riportino la sua definizione in C mediante lista collegata ed i prototipi delle sue funzioni primitive.

Soluzioni

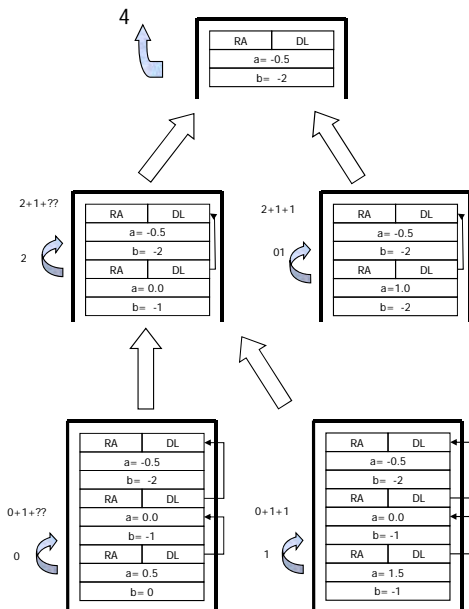
ESERCIZIO 1

```

list salta(list l1) {
  if (empty(l1) || empty(tail(l1)) )
    return emptylist();
  else {
    if (head(tail(l1)) % head(l1) == 0)
      return cons(head(l1), salta(tail(tail(l1)))) ;
    else
      return salta(tail(l1));
  }
}

```

ESERCIZIO 2



ESERCIZIO 3

```

99    ->    01100011
+56   ->    00111000
          11000111
-56   ->    11001000

```

```

01100011 + (99)
11001000 = (-56)
-----
00101011    (+43)

```