

**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 B**

**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 scelta(list l1);`

La funzione deve restituire una nuova lista contenente i valori di `l1` selezionati secondo un preciso criterio. In particolare, un valore è selezionato se è pari alla somma del valore che lo precede più il valore che lo segue.

Ad esempio, se `l1 = {6, 5, -1, 2, 3, 1}`, la funzione restituirà la lista `{ 5, 2, 3}`. Infatti il valore `5` è pari a `6+(-1)`; analogamente `2=(-1)+3` e `3=2+1`. Si noti che il primo e l'ultimo valore non possono essere comunque selezionati.

**ESERCIZIO 2 (3 punti)**

Data la funzione:

```
int exchange(float a, int b){
    if ( a==0)
        return b;
    else
        if (a>0)
            return exchange(a-b, b-2) + exchange(a-b+1, b) + 1;
        else
            return exchange(a+b, b) + exchange(a-b+2, b) + 1;
}
```

mostrare la sequenza dei record di attivazione nel caso in cui la funzione sia invocata con parametri attuali (2, 3) ed 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):

$$67 + (-37)$$

**ESERCIZIO 4 (2 punti)**

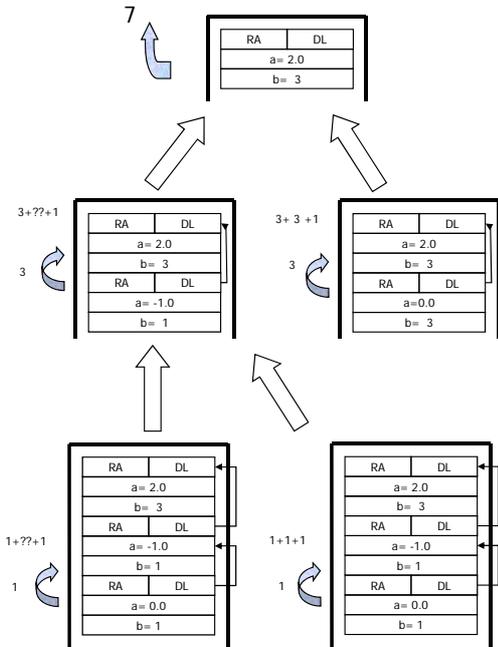
Si descriva rapidamente il concetto di tipo di dato astratto stack, si riportino la sua definizione in C mediante lista collegata ed i prototipi delle sue funzioni primitive.

**Soluzione**

**ESERCIZIO 1**

```
list scelta(list l1) {
  if (empty(l1) || empty(tail(l1)) || empty(tail(tail(l1))))
    return emptylist();
  else {
    if (head(tail(l1)) == head(l1) + head(tail(tail(l1))))
      return cons(head(tail(l1)), scelta(tail(l1)) );
    else
      return scelta(tail(l1));
  }
}
```

**ESERCIZIO 2**



**ESERCIZIO 3**

67	->	01000011	01000011 + (67)
+37	->	00100101	11011011 = (-37)
		11011010	-----
-37	->	11011011	00011110 (+30)