

**Fondamenti di Informatica T-1 (A.A. 2012/2013) - Ingegneria Informatica**  
**Prof.ssa Mello**  
**Prova Parziale d'Esame di Giovedì 10 Gennaio 2013 – durata 1h**  
**Totale 12 punti, sufficienza con 7**

**Compito A**

**ESERCIZIO 1 (6 punti)**

Date due liste di interi `l1` e `l2` ordinate in ordine crescente senza ripetizioni, si realizzi una funzione RICORSIVA

```
list remove_dup(list l1, list l2);
```

che restituisca una nuova lista anch'essa ordinata in ordine crescente contenente gli elementi presenti nelle liste `l1` e `l2` escludendo gli elementi presenti in entrambe le liste. Ad esempio, se `l1 = [1, 3, 4, 7, 9, 12]` e `l2 = [0, 2, 4, 5, 9]`, la funzione `remove_dup()` deve restituire la lista `[0, 1, 2, 3, 5, 7, 12]` in quanto gli interi 4 e 9 sono presenti sia in `l1` che in `l2`.

La funzione `remove_dup()` dovrà essere implementata utilizzando le sole primitive dell'ADT lista; ogni altra funzione dovrà essere opportunamente specificata dal candidato. Si realizzi inoltre una semplice funzione `main()` di prova che invochi correttamente la funzione `remove_dup()` creata.

**ESERCIZIO 2 (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):

### **ESERCIZIO 3 (3 punti)**

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (si motivi opportunamente la risposta data)

```
#include <stdio.h>
#include <stdlib.h>

#define DIM 15

int merge(char* str1, char* str2, char *out){
    int res = 0, i = 0, diff;

    while( *str1 != '\0' && *str2 != '\0' ){
        diff = *str1 - 'a';
        res += diff;
        *(out+i) = *str2 + diff;
        i++;
        str1++;
        str2++;
    }

    out[i] = '\0';

    return res;
}

int main() {
    char* final;
    int res;

    final = (char*)malloc(sizeof(char)*DIM);

    res = merge("cbbcab", "colore", final);

    printf("%s %d\n", final, res);
    return 0;
}
```

### **ESERCIZIO 4 (1 punto)**

Relativamente al linguaggio C, il candidato illustri le differenze tra file di testo e file binari, mostrando anche brevi esempi di codice.

# Soluzioni

## ESERCIZIO 1

```
list remove_dup(list l1, list l2){
    if( empty(l1) ){
        return l2;
    }
    else if( empty(l2) ){
        return l1;
    }
    else{
        if( head(l1) == head(l2) ){
            return remove_dup(tail(l1), tail(l2));
        }
        else if( head(l1) < head(l2) ){
            return cons(head(l1), remove_dup(tail(l1), l2));
        }
        else{
            return cons(head(l2), remove_dup(l1, tail(l2)));
        }
    }
}

int main(){
    list l1, l2, res;
    l1 = cons(1, cons(3, cons(4, cons(7, cons(9, cons(12, emptylist())))));
    l2 = cons(0, cons(2, cons(4, cons(5, cons(9, emptylist()))));

    res = remove_dup(l1, l2);

    while( ! empty(res) ){
        printf("%d ", head(res));
        res = tail(res);
    }
    printf("\n");

    return 0;
}
```

## ESERCIZIO 2

+ 33 - 92 = - 59  
+ 33 ==> 00100001  
+ 92 ==> 01011100  
          10100011  
- 92 ==> 10100100  
- 59 ==> 11000101 = -128 + 64 + 4 + 1

## ESERCIZIO 3

Il programma è corretto sintatticamente e la sua esecuzione produce la stampa:

**epmqr f 7**

La funzione `main()` alloca spazio sufficiente a contenere 15 char e poi invoca la funzione `merge()`.

La funzione `merge()` itera lungo le stringhe ben formate `str1` e `str2`. Ad ogni iterazione del ciclo `while` calcola la distanza tra il primo carattere di `str1` e il carattere `'a'` e somma tale valore all'interno della variabile `res`; inoltre inserisce nell'area di memoria puntata dalla variabile `out` il primo carattere della variabile `str2` shiftato di `diff` posizioni. Infine la funzione restituisce per valore la somma totale delle differenze e per riferimento la stringa ben formata ottenuta.

La funzione `main()` stampa sullo standard output la stringa ben formata presente nell'area di memoria restituita dalla funzione `merge()` e la somma delle operazioni di shift effettuate.