

**Fondamenti di Informatica T-1 (A.A. 2012/2013) - Ingegneria Informatica**  
**Prof.ssa Mello**  
**Prova Parziale d'Esame di Venerdì 15 Febbraio 2013 – durata 1h**  
**Totale 12 punti, sufficienza con 7**

**Compito B**

**ESERCIZIO 1 (6 punti)**

Data una lista di stringhe ben formate `l`, si realizzi una funzione RICORSIVA

```
list filterString(list l);
```

che restituisca una nuova lista contenente gli elementi presenti nella lista `l` che abbiano al più 2 caratteri numerici. Ad esempio, se `l = ["aaa12", " BbB", "123", " *+5544", "!*"]`, la funzione `filterString()` deve restituire la lista `["aaa12", " BbB", "!*"]` in quanto "aaa12" ha due caratteri numerici, "BbB" zero e "!" zero.

A tal scopo si implementi e si faccia uso della funzione ITERATIVA

```
int contaDigit(char *str);
```

che, data in ingresso una stringa ben formata, restituisce il numero di caratteri numerici presenti in tale stringa.

Le funzioni `filterString()` e `contaDigit()` dovranno essere implementate 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 `filterString()` 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):

+ 31 – 60

### **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>

int countB(char* stringa){
    if( *stringa == '\0' ){
        return 0;
    }
    else{
        return (*stringa - 'c') + countB(&stringa[1]);
    }
}

int main() {
    char* str = "edcd";
    int i = 0, res;
    char* new;

    new = (char*) malloc(sizeof(char)*10);

    while( *str != '\0' ){
        new[i] = *str - 1;
        i++;
        str++;
    }
    new[i] = '\0';

    res = countB(new);

    printf("%d %s\n",res, new);

    return 0;
}
```

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

Si descriva cosa succede durante il passaggio per riferimento di un parametro a una funzione utilizzando il linguaggio C; in particolare si analizzino le differenze principali tra il passaggio di una stringa ben formata e di una struct.

# Soluzioni

## ESERCIZIO 1

```
int contaDigit(char* str){
    int digit = 0;
    while( *str != '\0' ){
        if( *str >= '0' && *str <= '9' ){
            digit ++;
        }
        str++;
    }
    return digit;
}

list filterStrings(list l){
    int digit;
    if( empty(l) ){
        return l;
    }
    else {
        digit = contaDigit(head(l));
        if( digit < 3 ){
            return cons(head(l),filterStrings(tail(l)));
        }
        else {
            return filterStrings(tail(l));
        }
    }
}

int main(){
    list l, res;

    l = cons("aaa12", cons("BbB", cons("123", cons("*-+5544", cons("!*!",
emptylist())))));

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

    return 0;
}
```

## ESERCIZIO 2

+ 31 - 60 = - 29  
+ 31 ==> 00011111  
+ 60 ==> 00111100  
          11000011  
- 60 ==> 11000100  
- 29 ==> 11100011 = -128 + 64 + 32 + 2 + 1

## ESERCIZIO 3

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

0 dcbc

La funzione `main()` alloca spazio sufficiente a contenere 10 char e poi inserisce in tale spazio di memoria i caratteri presenti nella stringa ben formata `str` shiftati di meno uno. Successivamente la funzione `main()` invoca la funzione `countB()`.

La funzione ricorsiva `countB()` restituisce 0 se la stringa passata come parametro in ingresso è vuota. In caso contrario, la funzione `countB()` restituisce la distanza tra il primo carattere della stringa `str` e il char 'c' sommato a ciò che viene restituito dalla chiamata ricorsiva alla funzione stessa passando come parametro d'ingresso l'indirizzo del secondo carattere della stringa `str`.

La funzione `main()` stampa sullo standard output il valore restituito dalla funzione `countB()` e la stringa ben formata presente nell'area di memoria allocata dinamicamente.