

**Fondamenti di Informatica T-1 (A.A. 2009/2010) - Ingegneria Informatica**  
**Prof.ssa Mello**  
**Prova Parziale d'Esame di Giovedì 11 Febbraio 2010 – durata 1h**  
**Totale 12 punti, sufficienza con 7**  
**Compito A**

**ESERCIZIO 1 (6 punti)**

Si supponga di avere a disposizione, già definita, l'ADT lista per interi. Il candidato definisca una funzione:

`list filtraRip(list l1);`

che, ricevuta in ingresso una lista di interi ordinata, restituisca una lista ottenuta eliminando da **l1** tutti gli elementi che sono ripetuti. La funzione dovrà essere implementata utilizzando le sole primitive dell'ADT lista; ogni altra funzione dovrà essere opportunamente specificata dal candidato.

Ad esempio, se  $l1 = [1, 2, 2, 2, 3, 5, 5, 6]$ , la funzione `filtraRip(...)` restituirà la lista  $[1, 3, 6]$ .

**ESERCIZIO 2 (2 punti)**

Si consideri la grammatica G con scopo S, simboli non terminali {A, B, C, D, E} e simboli terminali { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0}:

**S ::= ABA | BAB**

**A ::= C | CA**

**B ::= CD | CDBE**

**C ::= 2 | 4 | 6 | 8**

**D ::= 1 | 3 | 5 | 7 | 9**

**E ::= 0**

La stringa “212467890” appartiene al linguaggio generato da tale grammatica? In caso affermativo, se ne mostri la derivazione left-most.

### 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 i = 0;

char * filter(char * v, char thres){
    int k;
    int i;
    char * res;
    for (k=0; *(v+k)!='\0'; k++);
    res = (char *) malloc(sizeof(char) * (k));
    i = 0;
    for (k=0;v[k]!='\0';k++) {
        if (v[k]<=thres) {
            *(res+i) = *(v+k);
            i++;
        }
        else
            v[k] = v[(i>0)? i: k];
    }
    res[i] = '\0';
    return res;
}

int main(){
    char * p;
    char str[] = "abcdefg";
    do {
        if (str[i])
            i++;
    } while (str[i]!='\0');
    p = filter(str, str[i/2]);
    printf("%s\n%s\n", str, p);
    free(p);
    return (0);
}
```

### ESERCIZIO 4 (1 punto)

Si descriva sinteticamente il tipo di dato astratto "stack" visto a lezione, introducendo le primitive e fornendo un esempio di uso di tale ADT.

## Soluzioni

### ESERCIZIO 1

```
list filtraRip(list l1) {
    element temp;
    int ripetuto;
    if (empty(l1))
        return emptylist();
    else {
        ripetuto = 0;
        temp = head(l1);
        l1 = tail(l1);
        while (!empty(l1) && temp == head(l1) ) {
            l1 = tail(l1);
            ripetuto = 1;
        }
        if (!ripetuto)
            return cons(temp, filtraRip(l1) );
        else
            return filtraRip(l1);
    }
}
```

### ESERCIZIO 2

La frase appartiene al linguaggio. In particolare, la si può ottenere tramite la seguente derivazione left-most:  
S -> BAB -> CDAB -> 2DAB -> 21AB -> 21CAB -> 212AB -> 212CB -> 2124B -> 2124CDBE ->  
21246DBE -> 212467BE -> 212467CDE ->2124678DE -> 21246789E -> 212467890.

### ESERCIZIO 3

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

```
abcdeee
abcd
```

La funzione `main()` dapprima calcola la lunghezza della stringa `str`, e poi invoca la funzione `filter(...)`. In particolare, i parametri attuali sono la stringa `str` e il carattere di `str` posizionato all'indice  $i/2$ : siccome `i` vale 7, viene passato il carattere all'indice 3, cioè il carattere 'd'.

La funzione dapprima calcola la dimensione della stringa `v`, e poi alloca memoria per un altro array di caratteri di pari dimensione. Quindi scorre la stringa `v` fino alla fine, e confronta ogni carattere con il carattere passato tramite il parametro `thres`. Se il carattere è "minore o uguale" a `thres`, allora viene copiato nel vettore `res`; altrimenti la funzione sovrascrive il carattere nel vettore `v` passato come parametro. Il valore che vi viene inserito dipende dall'indice `i`. Siccome `thres` è pari, nell'invocazione, al carattere 'd', vengono copiati in `res` i primi quattro caratteri ("abcd"). I caratteri successivi sono tutti maggiori di `thres`, e quindi nel vettore `v` vengono sovrascritti: l'indice `i` a quel punto dell'esecuzione vale 4, e quindi viene messo il valore `v[4]`, cioè il carattere 'e'.

La funzione `main()` infine stampa a video le stringhe `str` e `p`.