

Fondamenti di Informatica T-1 (A.A. 2008/2009) - Ingegneria Informatica
Prof.ssa Mello
Prova d'Esame di Venerdì 19 Giugno 2009 – durata 1h
Totale 12 punti, sufficienza con 7

ESERCIZIO 1 (5 punti)

Si scriva una funzione

```
float mediaPesata(list voti, list crediti);
```

che riceve in ingresso i puntatori **voti** e **crediti** alle radici di due liste contenenti una serie di interi. La funzione deve innanzitutto verificare che le due liste abbiano la stessa lunghezza; a tal scopo, si supponga di avere a disposizione una funzione **int lunghezza(list l)** che calcoli la lunghezza di una lista. Nel caso in cui le due liste abbiano la stessa lunghezza, la funzione deve restituire la media dei voti pesata rispetto ai crediti; in caso contrario, la funzione deve restituire un valore di errore. Si faccia l'ipotesi che ogni coppia voto/crediti sia costituita da elementi presenti nelle due liste alla stessa posizione.

Ad esempio, supponendo di avere una lista di voti [30,26,28] e una lista di crediti [6,6,3] la funzione deve restituire il valore:

$$28.4 = \frac{30 \cdot 6 + 28 \cdot 6 + 26 \cdot 3}{6 + 6 + 3}$$

La realizzazione della funzione si deve appoggiare su due ulteriori funzioni ausiliarie:

1. Una funzione iterativa

```
int somma(list l)
```

che riceve in ingresso una lista di interi e ne restituisce la somma. Per questa funzione non viene data a disposizione nessuna libreria predefinita sulle liste.

2. Una funzione ricorsiva

```
float sommaPesata(list valori, list pesi);
```

che riceve in ingresso una lista di valori e una lista di pesi e restituisce la somma dei valori, ognuno moltiplicato per il corrispondente peso. Per questa funzione si utilizzino le funzioni di libreria predefinite sulle liste di interi.

ESERCIZIO 2 (3 punti)

Si consideri la seguente funzione:

```
int tar(int r, int c)
{
    if(r < 1 || c < 1 || r < c)
        return 0;
    else {
        if(r == 1)
            return 1;
        else
            return tar(r-1,c-1) + tar(r-1,c);
    }
}
```

Mostrare la sequenza dei record di attivazione nel caso in cui la funzione sia invocata con parametri attuali (3,2) e il valore di ritorno.

ESERCIZIO 3 (2 punti)

Si rappresenti schematicamente l'architettura di una macchina di Von Neumann e se ne descrivano i componenti.

ESERCIZIO 4 (2 punti)

Si consideri la grammatica G con scopo S, simboli non terminali {A, B, C, D, X} e simboli terminali {f, g, m, n, 1, 2, 3}:

S ::= ABA | BAB
A ::= X | XC
B ::= X | DXD
C ::= f | g
D ::= m | n
X ::= XB | 1 | 2 | 3

La stringa "**mn11f3g**" appartiene al linguaggio generato da tale grammatica?
In caso affermativo, se ne mostri la derivazione left-most.

Soluzioni

ESERCIZIO 1

```
int somma(list l)
```

```
{
    int s = 0;
    while(l != NULL)
    {
        s += l->value;
        l = l->next;
    }
    return s;
}
```

```
int sommaPesata(list valori, list pesi)
```

```
{
    if(empty(valori))
        return 0;
    else return
        head(valori)*head(pesi) + sommaPesata(tail(valori),tail(pesi));
}
```

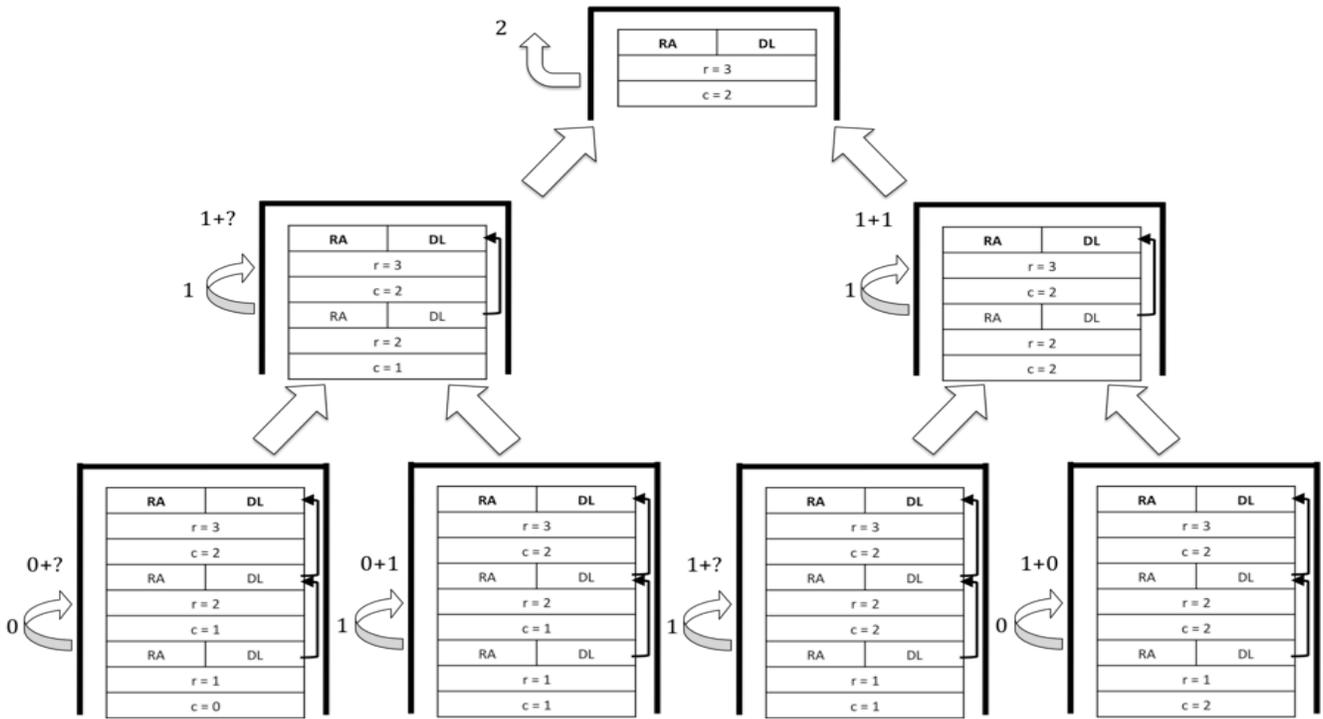
```
float mediaPesata(list voti, list crediti)
```

```
{
    if(lunghezza(voti) == lunghezza(crediti))
        return ((float) sommaPesata(voti,crediti)) / somma(crediti);
    else return -1;
}
```

Il cast a **float** è necessario per evitare perdita di informazione nel rapporto tra la somma pesata e il numero totale di crediti.

ESERCIZIO 2

La funzione calcola il valore dell'elemento, all'interno del triangolo di Tartaglia, in riga **r** e colonna **c**.



ESERCIZIO 4

La stringa **mn11f3g** non appartiene alla grammatica data. Il problema si verifica subito all'inizio della frase, con la sequenza **mn**. Le lettere **m** ed **n** possono essere generate solo dal non terminale D; a sua volta, D compare solo nell'espansione del non terminale B. Il problema sorge dal fatto che, stando alla regola di produzione, il non terminale B può essere riscritto solo come DXD. Quindi tra un carattere **m** ed un carattere **n** deve sempre esserci qualcosa (ottenuto sostituendo il non terminale X).

Si noti inoltre che la grammatica per alcuni frasi può essere ambigua, poiché sia il terminale A che il terminale B possono essere riscritti come X, e lo scopo iniziale può essere riscritto come ABA o BAB.