

Fondamenti di Informatica T-1 (A.A. 2016/2017) - Ingegneria Informatica
Prof.ssa Mello
Prova Parziale d'Esame di Giovedì 12 Gennaio 2017 – durata 1h
Totale 12 punti, sufficienza con 7

Compito B

ESERCIZIO 1 (6 punti)

Siano date due liste di interi v e p . Si assuma che le due liste abbiano pari lunghezza. Si realizzi una funzione RICORSIVA che calcoli la sommatoria dei prodotti di ciascun elemento di v per il corrispondente di pari indice in p

```
int sommaProdotti(list v, list p)
```

La funzione deve pertanto restituire il risultato dell'espressione $S = \sum_{i=0}^n v_i * p_i$

Si realizzi inoltre una funzione ITERATIVA

```
int mediaPesata(int S, list p);
```

che calcoli la somma X degli elementi in p e ritorni il risultato S/X

Si realizzi infine una funzione `main()` che crei le due liste v e p ed utilizzi correttamente le due funzioni `sommaProdotti(v,p)` e `mediaPesata(S,p)` precedenti (dove S è il risultato di `sommaProdotti`), in modo tale da calcolare la media pesata intera. Ad esempio, date le liste $v = \{5,6,9,10,2\}$ e $p = \{1,1,2,3,1\}$ si dovrà ottenere come risultato finale $(5*1+6*1+9*2+10*3+2*1)/(1+1+2+3+1)=7$

Le funzioni dovranno essere implementate utilizzando le primitive dell'ADT lista, includendo "list.h".

ESERCIZIO 2 (2 punti)

Si consideri la seguente funzione

```
double P(int x){
    if ( x <= 0 )
        return -2.0;
    else{
        return P(x-1)*P(x-2);
    }
}
```

mostrare la sequenza dei record di attivazione ed il valore di ritorno nel caso in cui la funzione sia invocata con parametro attuale 2

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 mv(int a[], int m, char *b){
    int i=m-1, q;
    while(i>0){
        printf("%d %c\n", i, *b);
        q = a[i]/2;
        if (q != a[--i]){
            a[i]=q;
            (*b)++;
        }
        i--;
    }
    return q;
}

int main(){
    int a[]={2,4,8,17,32}, m = 5, q;
    char b='c';
    q = mv(a,m,&b);
    printf("%d %c\n",q,b);
    return 0;
}
```

ESERCIZIO 4 (1 punto)

Si introduca il concetto di variabile locale e se ne spieghi la visibilità, anche con l'aiuto di un esempio.

Soluzioni

ESERCIZIO 1

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include "list.h"

int sommaProdotti(list v, list p)
{
    if (empty(p))
        return 0;
    else
        return head(v)*head(p) + sommaProdotto(tail(v),tail(p));
}

int mediaPesata(list p, int S)
{
    //supponiamo che la funzione venga invocata sempre con S!=0
    int somma = 0;
    while (!empty(p))
    {
        somma += head(p);
        p = tail(p);
    }
    return S/somma;
}

int main()
{
    list v = emptyList();
    list p = emptyList();
    list tmp;
    int S;

    v = cons(5,v);
    v = cons(6,v);
    v = cons(9,v);
    v = cons(10,v);
    v = cons(2,v);

    p = cons(1,p);
    p = cons(1,p);
    p = cons(2,p);
    p = cons(3,p);
    p = cons(1,p);

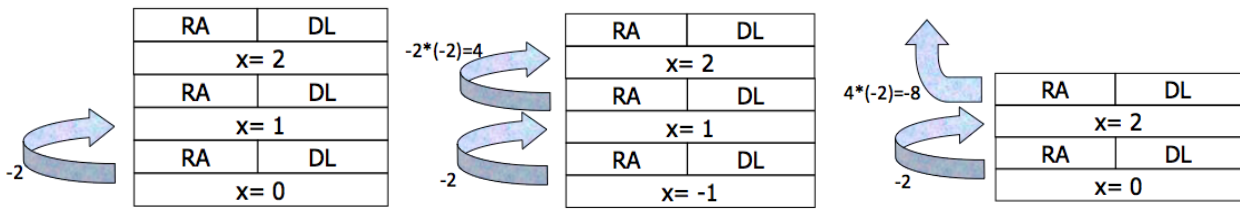
    S = sommaProdotti(v,p);
    M = mediaPesata(S,p);
    printf("Media pesata intera = %f",M);

    while(!empty(v)) { tmp = v; v = tail(v); free(tmp); }
    while(!empty(p)) { tmp = p; p = tail(p); free(tmp); }

    return 0;
}
```

}

ESERCIZIO 2



ESERCIZIO 3

L'output prodotto è

4 c
2 d
4 d

Il programma `main` crea un array `a` di interi ed invoca la funzione `mv`, utilizzando come parametri l'array stesso e la variabile `b='c'` (passata per riferimento). Il risultato è memorizzato in `q`.

La funzione `mv` esegue il ciclo `while` due volte scendendo l'array `a` dalla fine all'inizio.

Alla prima invocazione la funzione stampa "4 c" e `q` assume il valore di $a[4]/2$, ovvero 16. Tale valore viene confrontato con `a[3]` (pre-decremento di `i`). Poiché $16 \neq 17$, viene eseguito il codice all'interno del blocco `if`: `a[3]=16` e la cella puntata da `b` viene incrementata di un'unità. Poiché caratteri consecutivi sono memorizzati in posizioni consecutive della ASCII table, il valore in `b` diventa 'd'. Infine `i` viene nuovamente decrementato.

Alla seconda esecuzione del ciclo `while`, `i` vale 2 e viene stampato: "2 d". Poiché $8/2 == a[1]$ non viene eseguito il blocco `if`, ma solo il post-decremento di `i` che assume valore 0. A questo punto la condizione del `while` non è più soddisfatta e la funzione `mv` termina restituendo `q=4`.

Infine il `main` stampa "4 d". Il valore di `b` risulta modificato in seguito all'invocazione di `mv` poiché è stato passato per riferimento.