

Fondamenti di Informatica T-1 (A.A. 2016/2017) - Ingegneria Informatica
Prof.ssa Mello
Prova Parziale d'Esame di Giovedì 26 Gennaio 2017 – durata 1h
Totale 12 punti, sufficienza con 7

Compito A

ESERCIZIO 1 (6 punti)

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

```
float sommaDiv(list p, list v)
```

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

Si realizzi inoltre una funzione ITERATIVA

```
float armonicaPesata(list p, float S);
```

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

Si realizzi infine una funzione `main()` che crei le due liste p e v ed utilizzi correttamente le due funzioni `sommaDiv(p,v)` e `armonicaPesata(p,S)` precedenti (dove S è il risultato della funzione `sommaDiv`), in modo tale da calcolare la media armonica pesata. Ad esempio, date le liste $p = \{1.0, 1.1, 2.0\}$ e $v = \{5.0, 6.1, 9.8\}$ si dovrà ottenere come risultato finale $(1.0+1.1+2.0) / (1.0/5.0 + 1.1/6.1 + 2.0/9.8) \approx 7.02$

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

ESERCIZIO 2 (2 punti)

Si consideri la seguente funzione

```
double Q(int a){
    if ( a >= 0 )
        return 3.0;
    else{
        return Q(a+1)*Q(a+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 mp(int x[], int n, char *y){
    int i=n-1, k;
    while(i>0){
        printf("%d %c\n", i, *y);
        k = x[i]/2;
        if (k != x[--i]){
            x[i]=k;
            (*y)++;
        }
        i--;
    }
    return k;
}

int main(){
    int x[]={2,4,8,17,32}, n = 5, i;
    char y='u';
    int k = mp(x,n,&y);
    printf("%d %c\n",k,y);
    return 0;
}
```

ESERCIZIO 4 (1 punto)

Si introduca il concetto di variabile globale 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"

float sommaDiv(list p, list v)
{
    if (empty(p))
        return 0;
    else //supponiamo che i denominatori siano sempre !=0
        return head(p)/head(v) + sommaDiv(tail(p),tail(v));
}

float armonicaPesata(list p, float S)
{
//supponiamo che la funzione venga invocata sempre con S!=0
    float somma = 0.0;
    while (!empty(p))
    {
        somma += head(p);
        p = tail(p);
    }
    return somma/S;
}

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

    p = cons(1.0,p);
    p = cons(1.1,p);
    p = cons(2.0,p);

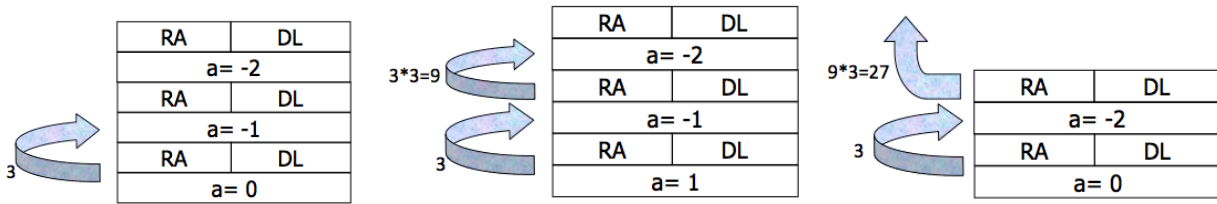
    v = cons(5.0,v);
    v = cons(6.1,v);
    v = cons(9.8,v);

    S = sommaDiv(p,v);
    A = armonicaPesata(p,S);
    printf("Media armonica pesata = %f",A);

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

    return 0;
}
```

ESERCIZIO 2



ESERCIZIO 3

L'output prodotto è

4 u
2 v
4 v

Il programma `main` crea un array `x` di interi ed invoca la funzione `mp`, utilizzando come parametri l'array stesso e la variabile `y = 'u'` (passata per riferimento). Il risultato è memorizzato in `k`.

La funzione `mp` esegue il ciclo `while` due volte scandendo l'array `x` dalla fine all'inizio.

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

Alla seconda esecuzione del ciclo `while`, `i` vale 2 e viene stampato: "2 v". Poiché $8/2 == x[1]$ 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 `mp` termina restituendo `k = 4`.

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