

Fondamenti di Informatica T-1 (A.A. 2015/2016) - Ingegneria Informatica
Prof.ssa Mello
Prova Parziale d'Esame di Venerdì 12 Febbraio 2016 – durata 1h
Totale 12 punti, sufficienza con 7

Compito B

ESERCIZIO 1 (6 punti)

Sia data una lista di float `L1` e una variabile di tipo float `scartoQuadratico`. Si realizzi la funzione ITERATIVA

```
float media(list l);
```

che calcoli la media degli elementi di una lista `l`.

Si realizzi quindi una funzione RICORSIVA

```
list cerca(list l, float valore, float scartoQuadratico);
```

che ritorni una lista composta dagli elementi della lista `l` per i quali il quadrato della differenza rispetto a `valore` sia maggiore di `scartoQuadratico`.

Si realizzi infine una funzione `main()` che crei una lista `L1` ed invochi correttamente le due funzioni precedenti in modo tale da calcolare la lista `L2` composta dagli elementi appartenenti a `L1` per i quali il quadrato della differenza rispetto alla media degli elementi di `L1` sia maggiore di una variabile `scartoQuadratico` definita nel `main()`.

Per esempio, se `l1 = {12.57, 12.61, 12.49, 13.01, 12.44}` e `scartoQuadratico = 0.1` la media sarà `12.624` e quindi `l2 = {13.01}`.

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

ESERCIZIO 2 (2 punti)

Si consideri la seguente funzione:

```
int g(int x) {
    int y = 0;

    if (x >= y--)
    {
        return 1;
    }
    else
    {
        return 1 + g(++x);
    }
}
```

Mostrare la sequenza dei record di attivazione e il valore di ritorno nel caso in cui la funzione sia invocata con parametri attuali `(-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>

typedef struct {
    int x[3], y[3];
} Triangle;

void printV(int x, int y) {
    printf("<d, %d>\n", x, y);
}

int perp(int xA, int yA, int xB, int yB) {
    printV(xA, yA);
    printV(xB, yB);
    return xA * yB - yA * xB;
}

float area(Triangle t) {
    int* x = t.x;
    int* y = t.y;
    return perp(x[1]-x[0], y[1]-y[0], x[2]-x[0], y[2]-y[0]) / 2.0;
}

int main() {
    Triangle t = {{0,2,0},{0,1,2}};
    int i;
    for (i = 0; i < 3; ++i) {
        printV(t.x[i], t.y[i]);
    }
    printf("%f", area(t));
    return (0);
}
```

ESERCIZIO 4 (1 punto)

Si illustri il concetto di allocazione dinamica di un blocco di memoria (`malloc`) e se ne fornisca un breve esempio.

Soluzioni

ESERCIZIO 1

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

float media(list l)
{
    float somma = 0;
    int n = 0;

    if (empty(l)) return 0;

    while (l != NULL)
    {
        somma += head(l);
        l = tail(l);
        n++;
    }

    return somma/(float)n;
}

list cerca(list l1, float valore, float scartoQuadratico)
{
    if (empty(l1))
        return emptyList();
    else if (((head(l1) - valore)*(head(l1) - valore)) > scartoQuadratico)
        return cons(head(l1), cerca(tail(l1), valore, scartoQuadratico));
    else
        return cerca(tail(l1), valore, scartoQuadratico);
}

int main()
{
    list L1 = emptyList();
    list L2 = emptyList();
    float valore;
    float scartoQuadratico = 0.1;

    L1 = cons(12.44, L1);
    L1 = cons(13.01, L1);
    L1 = cons(12.49, L1);
    L1 = cons(12.61, L1);
    L1 = cons(12.57, L1);

    valore = media(L1);

    L2 = cerca(L1, valore, scartoQuadratico);

    return 0;
}
```

ESERCIZIO 2

RA	DL	3
x= -2 -1		
y= 0 -1		
RA	DL	2
x= -1 0		
y= 0 -1		
RA	DL	1
x=0		
y= 0 -1		

ESERCIZIO 3

L'output prodotto è

```
<0, 0>
<2, 1>
<0, 2>
<2, 1>
<0, 2>
2.000000
```

Una struttura di tipo `Triangle` contiene due array `x` ed `y` di interi.

La funzione `printV` stampa due interi `x` e `y` su standard output.

La funzione `perp` stampa l'input usando `printV`, esegue 3 operazioni e ritorna il risultato.

La funzione `area` riceve in ingresso un `Triangle`. Assegna a due puntatori a intero gli array contenuti nel `Triangle`. Esegue alcune sottrazioni tra gli elementi degli array e chiama `perp`. Divide per due il suo risultato e lo ritorna in formato `float`.

Il `main` crea una struttura di tipo `Triangle` e tramite un ciclo `for` ne stampa il contenuto. Successivamente chiama la funzione `area` e ne stampa il risultato.

In output si ottengono, nell'ordine: 3 stampe di `printV` con il contenuto della struttura `Triangle`, 2 stampe di `printV` con l'input di `perp`, ed infine il valore ritornato da `area`.