

Fondamenti di Informatica T-1 (A.A. 2016/2017) - Ingegneria Informatica
Prof.ssa Mello
Prova Parziale d'Esame di Giovedì 14 Settembre 2017 – durata 1h
Totale 12 punti, sufficienza con 7

Compito A

ESERCIZIO 1 (6 punti)

Siano date due liste `l1` e `l2` di caratteri. Si assuma che le due liste abbiano pari lunghezza. Si scriva una funzione RICORSIVA

```
list verificaDistanza (list l1, list l2, int d)
```

che ritorni una nuova lista `l3` avente la stessa lunghezza di `l1` e `l2`, dove l'elemento `k`-esimo vale 'Y' se la distanza alfabetica tra il `k`-esimo carattere di `l1` ed il `k`-esimo di `l2` è uguale a `d`, e 'N' altrimenti.

Si noti che la funzione `verificaDistanza` deve inserire nella lista risultante un elemento 'Y' sia nel caso in cui il `k`-esimo elemento di `l1` preceda in ordine alfabetico il `k`-esimo elemento di `l2` di `d` posizioni, sia nel caso in cui lo segua di `d` posizioni.

Si realizzi una funzione `main()` che crei le liste `l1={'a','e','v'}`, `l2={'d','b','z'}` ed utilizzi la funzione `verificaDistanza` per identificare quali coppie di caratteri distano 3 posizioni nell'alfabeto. La lista risultante sarà quindi `l3={'Y','Y','N'}`.

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

ESERCIZIO 2 (2 punti)

Si consideri la seguente funzione

```
double F(int a){
    if ( a <= 1 )
        return 10.0;
    else{
        return F(a/2)+F(a/3);
    }
}
```

mostrare la sequenza dei record di attivazione ed il valore di ritorno nel caso in cui la funzione sia invocata con parametro attuale 5

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 func(float* fvec, int val){
    int j=2;
    float i=4.0;
    while (j>0){
        if (i==fvec[--j]){
            val=3;
            fvec[j]=6.0;
        }else{
            (*fvec)+=4.0;
        }
        printf("%d %f\n", j, fvec[j]);
    }
    return val;
}

int main(){
    int val=8;
    float vec[]={3.0,4.0,5.0,6.0,7.0};
    int res=func(vec,val);
    printf("%d %d\n",val,res);
    return 0;
}
```

ESERCIZIO 4 (1 punto)

Si illustri brevemente il funzionamento del preprocessore C.

Soluzioni

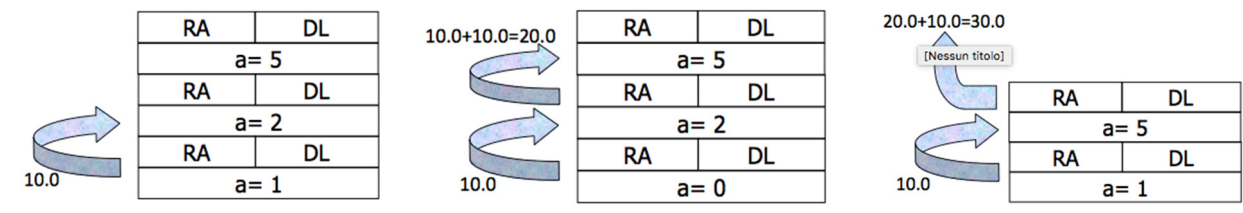
ESERCIZIO 1

```
list verificaDistanza(list l1, list l2, int d) {
    if (empty(l1) ) {
        return emptylist();
    } else {
        if ( (head(l2)-head(l1))==d || (head(l1)-head(l2))==d )
            return cons('Y', verificaDistanza(tail(l1),tail(l2),d));
        else
            return cons('N', verificaDistanza(tail(l1),tail(l2),d));
    }
}

int main() {
    list x;
    list l1 = emptylist();
    list l2 = emptylist();
    l1 = cons('v',l1);
    l1 = cons('e',l1);
    l1 = cons('a',l1);

    l2 = cons('z',l2);
    l2 = cons('b',l2);
    l2 = cons('d',l2);
    x = verificaDistanza(l1,l2,3);
    showlist(x);
    return 0;
}
```

ESERCIZIO 2



ESERCIZIO 3

L'output prodotto è

```
1 6.0
0 7.0
8 3
```

Il main crea un array di float c, inizializza v=8 e poi invoca la funzione func con tali parametri. La variabile v viene passata per valore.

All'interno della funzione, poichè j è inizializzato a 2, viene eseguito il codice nel while. La condizione nell'if viene testata dopo aver decrementato j di una unità (j=1). Poiché i=4.0 e c[1]=4.0, viene

eseguito il codice dell'`if`. La variabile `v` assume quindi valore 3 e `c[1]` viene modificato in 6.0. La prima `printf` stampa perciò i valori 1 e 6.0.

Alla seconda esecuzione del `while` `j=1`, per cui si entra nuovamente nel ciclo e viene testata la condizione dell'`if` dopo aver decrementato `j` di una unità (`j=0`). Poiché la condizione dell'`if` non è più soddisfatta (`i=4.0` e `c[0]=3.0`), viene eseguito l'`else`. L'istruzione `(*c) +=4.0` incrementa di 4.0 unità il valore contenuto nella cella che ha indirizzo `c`, ovvero tramuta `c[0]` in 7.0. La seconda `printf` stampa quindi 0 7.0.

Poiché ora `j=0`, il ciclo `while` si interrompe e la funzione termina restituendo il valore `v=3`. Il `main` associa tale valore alla sua variabile `r`, mentre la sua `v` (essendo stata passata per valore) continua a mantenere valore 8. L'ultima stampa risulta essere quindi 8 3.