Fondamenti di Informatica T-1 (A.A. 2015/2016) - Ingegneria Informatica
Prof.ssa Mello
Prova Parziale d'Esame di Giovedì 14 Gennaio 2016 – durata 1h
Totale 12 punti, sufficienza con 7

Compito A

ESERCIZIO 1 (6 punti)

Sono date due liste di interi `l1` e `l2` di pari lunghezza. Si realizzi una funzione RICORSIVA

```
list confronta(list l1, list l2);
```

che restituisca una lista di lunghezza pari alla lunghezza di `l1` e `l2`, in cui l'elemento in posizione k valga `1`, `0`, oppure `-1`, a seconda che il valore di `l1` in posizione k sia maggiore, uguale o minore al corrispondente valore di `l2`. Per esempio, se `l1={23,12,-64,48}`, `l2={15,-4,11,48}`, la funzione `confronta()` deve restituire la lista `{1,1,-1,0}`.

La funzione `confronta()` dovrà essere implementate utilizzando le primitive dell'ADT lista. Si realizzi una funzione `main()` che crei due liste e che utilizzi la funzione `confronta()`.

ESERCIZIO 2 (2 punti)

Si consideri la seguente grammatica G con scopo S, simboli non terminali {D, E, L, R} e simboli terminali {p, q, r, 1, 2}

```
S := L | E
L := LL | R | DE | ED
D := 1 | 2
R := p | q | r
E := DD | DR | RD
```

La stringa "**p1q2r**" appartiene al linguaggio di tale grammatica?

In caso affermativo se ne mostri la derivazione left-most.

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 f(char *a, char* b){
  int q=4,i=0,j=1;

  for(q=j; q>(i/2) && j; q--){
    if( b[q] == *(a+(q-i)) )
      return q++;
  }

  b[q] = *(a+q);
  b = a;

  return ++i;
}


int main(){
  char s1[]="live";
  char s2[]="safe";
  int res;

  res = f(s1,s2);

  printf("%d\n%s\n%s\n",res,s1,s2);
  return 0;
}
```

ESERCIZIO 4 (1 punto)

Si illustri il concetto di variabile globale e poi si spieghi il significato di EXTERN.

Soluzioni

ESERCIZIO 1

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

list confronta(list l1, list l2)
{
        if (empty(l1))
                return emptyList();
        else if (head(l1) > head(l2))
                return cons(1, confronta(tail(l1),tail(l2)));
        else if (head(l1) < head(l2))
                return cons(-1, confronta(tail(l1),tail(l2)));
        else
                return cons(0, confronta(tail(l1),tail(l2)));
}

int main()
{
        list L1 = emptyList();
        list L2 = emptyList();
        list L3 = emptyList();

        L1 = cons(48,L1);
        L1 = cons(-64,L1);
        L1 = cons(12,L1);
        L1 = cons(23,L1);
        L2 = cons(48,L2);
        L2 = cons(11,L2);
        L2 = cons(-4,L2);
        L2 = cons(15,L2);

        L3 = confronta(L1,L2);

        while (!empty(L1))
        {
                printf("%d ",head(L1));
                L1 = tail(L1);
        }
        printf ("\n");

        while (!empty(L2))
        {
                printf("%d ",head(L2));
                L2 = tail(L2);
        }
        printf ("\n");

        while (!empty(L3))
        {
                printf("%d ",head(L3));
                L3 = tail(L3);
        }
        printf ("\n");
```

```
        return 0;
}
```

ESERCIZIO 2

La frase appartiene al linguaggio. In particolare, la si può ottenere tramite la seguente derivazione left-most:
S → L → LL → LLL → RLL → pLL (*) → pEDL → pDRDL → p1RDL → p1qDL → p1q2L → p1q2R → p1q2r

A partire da (*) si può utilizzare la seguente derivazione alternativa:
pLL → pDEL → p1EL → p1RDL → p1qDL → p1q2L → p1q2R → p1q2r

Si può giungere a (*) anche con la seguente derivazione alternativa:
S → L → LL → RL → pL → pLL

ESERCIZIO 3

Il programma è corretto sintatticamente e la sua esecuzione produce la stampa:

```
1
live
lafe
```

La funzione main() invoca la funzione f() con parametri di ingresso due stringhe con valore “live” e “safe”. La funzione f() utilizza un ciclo for che esegue una sola iterazione, quando `q=j=1`. L'istruzione if presente all'interno del ciclo for non restituisce valore true, in quanto `b[1]='i'` mentre `*(a+(1-0))='a'`. All'uscita del ciclo for, `q` vale 0 e il primo carattere della stringa s1 ‘l’ viene assegnato al primo carattere della stringa s2 che si modifica quindi in “lafe”, la variabile `b` viene modificata in modo da puntare alla stessa locazione di `a`, ma si tratta soltanto delle copie locali dei puntatori, e non delle variabili `s1` e `s2` della funzione main. Nella successiva istruzione di `return`, la variabile `i` viene incrementata con un pre-incremento, così il valore restituito dalla funzione `f()` è 1. La funzione `main()` stampa quindi a video il valore 1, seguito dalle due stringhe s1 e s2 in cui la prima è stata modificata.