

**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 B**

**ESERCIZIO 1 (6 punti)**

Sono date due liste di interi  $l_1$  e  $l_2$  di pari lunghezza. Si realizzi una funzione RICORSIVA

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

che restituisca una lista di lunghezza pari alla lunghezza di  $l_1$  e  $l_2$ , in cui l'elemento in posizione  $k$  valga 1, -1, oppure 0, a seconda che i valori di  $l_1$  e  $l_2$  in posizione  $k$ , rispettivamente, abbiano lo stesso segno, abbiano segni diversi, oppure uno dei due sia uguale a 0. Per esempio, se  $l_1 = \{23, 12, -64, 0\}$ ,  $l_2 = \{15, -4, 11, 48\}$ , la funzione `compara()` deve restituire la lista  $\{1, -1, -1, 0\}$ .

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

**ESERCIZIO 2 (2 punti)**

Si consideri la seguente grammatica  $G$  con scopo  $S$ , simboli non terminali  $\{K, H, A, B\}$  e simboli terminali  $\{x, y, z, 0, 1\}$

```
S := K | H
K := KK | A | BH | HB
H := BB | BA | AB
A := x | y | z
B := 0 | 1
```

La stringa “**x0y1z**” 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 i=0,j=1,p=4;

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

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

    return ++j;
}

int main(){
    char s1[]="give";
    char s2[]="take";
    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 locale di una funzione e poi si spieghi il significato di STATIC.

# Soluzioni

## ESERCIZIO 1

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

list compara(list l1, list l2)
{
    if (empty(l1))
        return emptyList();
    else if (head(l1) * head(l2) > 0)
        return cons(1, compara(tail(l1),tail(l2)));
    else if (head(l1) * head(l2) < 0)
        return cons(-1, compara(tail(l1),tail(l2)));
    else
        return cons(0, compara(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 = compara(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:  
 $S \rightarrow K \rightarrow KK \rightarrow KKK \rightarrow AKK \rightarrow xKK (*) \rightarrow xHBK \rightarrow xBABK \rightarrow x0ABK \rightarrow x0yBK \rightarrow x0y1K \rightarrow x0y1A \rightarrow x0y1z$

A partire da (\*) si può utilizzare la seguente derivazione alternativa:  
 $xKK \rightarrow xBHK \rightarrow x0HK \rightarrow x0ABK \rightarrow x0yBK \rightarrow x0y1K \rightarrow x0y1A \rightarrow x0y1z$

Si può giungere a (\*) anche con la seguente derivazione alternativa:  
 $S \rightarrow K \rightarrow KK \rightarrow AK \rightarrow xK \rightarrow xKK$

## **ESERCIZIO 3**

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

```
2
tive
take
```

La funzione `main()` invoca la funzione `f()` con parametri di ingresso due stringhe con valore “give” e “take”. La funzione `f()` utilizza un ciclo `for` che esegue una sola iterazione, quando  $p=j=1$ . 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`, `p` vale 0 e il primo carattere della stringa `s2` 't' viene assegnato al primo carattere della stringa `s1` che si modifica quindi in “tive”, la variabile `a` viene modificata in modo da puntare alla stessa locazione di `b`, 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 `j` viene incrementata con un pre-incremento, così il valore restituito dalla funzione `f()` è 2. La funzione `main()` stampa quindi a video il valore 2, seguito dalle due stringhe `s1` e `s2` in cui la prima è stata modificata.