

Fondamenti di Informatica T-1 (A.A. 2009/2010) - Ingegneria Informatica
Prof.ssa Mello
Prova Parziale d'Esame di Giovedì 09 Settembre 2010 – durata 1h
Totale 12 punti, sufficienza con 7

ESERCIZIO 1 (6 punti)

Data una lista di float `seq`, si realizzi una funzione (ricorsiva)

```
list check(list seq);
```

che restituisca una nuova lista con gli elementi di `seq`, selezionati in relazione al valore delle successive celle della stessa lista `seq`. In particolare la lista restituita deve contenere i valori della lista `seq` solo se il valore dell'elemento attualmente in esame è maggiore della somma dei due elementi successivi (se esistono); in caso contrario la cella in esame non deve essere inclusa nella lista restituita.

Ad esempio se `seq = [5.3, -6.1, 1.9, -1.2]` la funzione `check` deve restituire la lista `[5.3, 1.9]` poiché 5.3 è maggiore di -4.2 (= -6.1 + 1.9) e 1.9 è maggiore di -1.2, mentre -6.1 è minore di 0.7 (= 1.9 - 1.2) e -1.2 è minore di 0 (essendo -1.2 l'ultimo elemento della lista, la somma degli elementi successivi è 0).

La funzione `check` dovrà essere implementata utilizzando le sole primitive dell'ADT lista; ogni altra funzione dovrà essere opportunamente specificata dal candidato. Si realizzi inoltre un semplice `main` di prova che invochi correttamente la funzione `check` creata.

Nota: l'ordine degli elementi della lista restituita dalla funzione `check` è ininfluente.

ESERCIZIO 2 (2 punti)

Si consideri la grammatica G con scopo S e simboli terminali $\{1, 2, 3, 4\}$

$$\begin{aligned} S &::= XY \mid WXY \\ Y &::= ZX \mid X \\ W &::= 1 \\ X &::= 1 \mid 2 \mid 3 \mid 4 \\ Z &::= 3 \end{aligned}$$

La stringa "1234" appartiene al linguaggio generato da 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>
#include <string.h>

void split(char* v1, char* v2){
    int i = 0;
    int j = 0;
    while( *(v1+i) != '\0' ){
        if( *(v1+i) >= 'A' && *(v1+i) <='Z' ){
            *v2 = *(v1+i);
            v2++;
        }
        else{
            *(v1+j) = *(v1+i);
            j++;
        }
        i++;
    }
    *v2 = '\0';
    *(v1+j) = '\0';
    return;
}

int main(){
    char s1[10], s2[10];
    strcpy(s1, "aBgTyJ");
    split(s1, s2);
    printf("%s\n", s1);
    printf("%s\n", s2);
    return 0;
}
```

ESERCIZIO 4 (1 punti)

Il candidato illustri brevemente (eventualmente anche utilizzando brevi esempi) la differenza tra allocazione statica e dinamica della memoria utilizzando il linguaggio C.

Soluzioni

ESERCIZIO 1

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

list check(list l){
    float tempF;
    list tempL;
    int el;
    if(empty(l))
        return emptylist();
    else{
        tempL = tail(l);
        tempF = 0;
        el = 0;
        while( el<2 && !empty(tempL) ){
            tempF += head(tempL);
            tempL = tail(tempL);
            el++;
        }
        if( tempF < head(l) ){
            return cons(head(l), check(tail(l)));
        }
        else
            return check(tail(l));
    }
}

int main() {
    list res;
    list l = cons(5.3, cons(-6.1, cons(1.9, cons(-1.2, emptylist()))));
    res = check(l);
    printf("\n");
    while( !empty(res) ){
        printf("%f\n", head(res));
        res = tail(res);
    }
    return 0;
}
```

ESERCIZIO 2

La frase appartiene al linguaggio. In particolare, la si può ottenere tramite la seguente derivazione left-most:

$$S \rightarrow WXY \rightarrow 1XY \rightarrow 12Y \rightarrow 12ZX \rightarrow 123X \rightarrow 1234$$

ESERCIZIO 3

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

```
agy  
BTJ
```

La funzione `main()` copia nell'array `s1` la stringa ben formata "aBgTyJ" e poi invoca la funzione `split()`.

La funzione `split()` scansiona i caratteri della stringa ben formata `v1` fino al terminatore. Se il carattere in esame è una lettera maiuscola, la funzione `split()` inserisce il carattere in testa all'array `v2`, altrimenti lo inserisce in testa all'array `v1`. Infine la funzione `split()` inserisce il carattere di terminazione immediatamente dopo l'ultimo carattere inserito nei vettori `v1` e `v2`.

La funzione `main()` scrive sullo standard output il valore delle stringhe ben formate `s1` ed `s2`.