PROGRAMMAZIONE 1 e LABORATORIO (A,B) - a.a. 2010/11

Prova scritta del 10 gennaio 2012

Scrivere **IN STAMPATELLO** COGNOME, NOME, MATRICOLA e CORSO su ogni foglio consegnato

ESERCIZIO 1 (punti 5)
Un albero *discendente a destra* (*dad*) è:

- una foglia
- un nodo con n figli ordinati (da sinistra a destra) tale che i primi $n-1$ sono foglie e l'n-esimo è la radice di un albero *dad*.

Si dia una grammatica G per il linguaggio $\{a^n b^m c^k \mid n,m,k \geq 0, n+m+k > 0\}$ sull'alfabeto $\Lambda = \{a,b,c\}$, in modo che tutti gli alberi di derivazione in G siano alberi *dad*.

ESERCIZIO 2 (punti 5)
Il taglio a profondità n di un albero binario è la lista delle etichette di tutti i nodi a profondità n, presi da sinistra a destra. Dato il tipo degli alberi binari visto a lezione

```
type 'a btree = Void | Node of 'a * 'a btree * 'a btree
```

si definisca una funzione `cut` con tipo

```
cut : 'a btree -> int -> 'a list
```

in modo che `(cut bt n)` sia il taglio a profondità `n` di `bt`.

ESERCIZIO 3 (punti 5)
Si definisca in `C` una funzione

```
boolean check (int a[], int d)
```

che restituisce il valore di verità della seguente formula:

$$\exists i.\ i \in [1, d-1) \ \wedge \sum_{j=0}^{i-1} a[j] = \sum_{j=i+1}^{d-1} a[j]$$

(Si assuma predefinito il tipo `typedef enum {false, true} boolean`).

ESERCIZIO 4 (punti 5)
Si definisca in `C` una funzione

```
int max_occ (int  a[], int dim)
```

che restituisce uno dei valori che occorrono in `a` il maggior numero di volte.

ESERCIZIO 5 (punti 5)
Si completi la seguente definizione

```
let revpari l = let f x y = ...  in foldr f ... l
```

in modo che `(revpari lis)` sia la lista degli elementi pari di `lis` in ordine inverso. Ad esempio:

```
revpari [1;5;2;4;7;8;10] = [10;8;4;2]          revpari [5;1;3] = []
```

ESERCIZIO 6 (punti 5)
Date le seguenti definizioni:

```
struct el {int info; struct el *next;};
typedef struct el ElementoDiLista;
typedef ElementoDiLista *ListaDiElementi;
```

scrivere in `C` una procedura che, dati in ingresso attraverso opportuni parametri una lista di interi ed un intero `x`, elimina dalla lista l'ultimo elemento uguale a `x` e quello che lo segue immediatamente (se esistono).