

PROGRAMMAZIONE 1 e LABORATORIO (A,B) - a.a. 2012/13

Prova scritta del 10 gennaio 2013

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

ESERCIZIO 1 (punti 5)

Si definisca una grammatica libera che genera il linguaggio

$$\mathcal{L} = \Lambda^+ \setminus \{a\}$$

sull'alfabeto $\Lambda = \{a, b\}$.

ESERCIZIO 2 (punti 5)

Si consideri il seguente tipo di alberi *ternari* in cui ogni nodo ha al più tre figli.

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

Si definisca una funzione `check` con tipo

```
check : 'a ttree -> boolean
```

in modo che (`check tt`) restituisca `true` se ogni nodo non foglia ha al più un figlio non vuoto, e restituisca `false` altrimenti.

ESERCIZIO 3 (punti 5)

Dato un array a di dimensione dim ed un indice $i \in [1, dim - 1]$, siano:

- $dprec(i) = \max\{|a[i] - a[j]| \mid j \in [0, i)\}$
- $dsucc(i) = \max\{|a[i] - a[j]| \mid j \in [i + 1, dim)\}$

Si definisca in C una funzione

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

che restituisce il valore di verità della seguente formula:

$$\exists i. i \in [1, dim - 1) \wedge dprec(i) = dsucc(i)$$

(**N.B.** $|x - y|$ indica la differenza in valore assoluto tra x e y .)

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

ESERCIZIO 4 (punti 5)

Si completi la seguente definizione

```
let foo lis = let f x y = .... in foldr f ... lis
```

della funzione

```
foo: int list -> int list
```

in modo che (`foo lis`) sia la lista ottenuta da `lis` rimpiazzando ogni elemento con il numero di elementi uguali a 0 nella sottolista di `lis` che inizia dall'elemento stesso.

ESERCIZIO 5 (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 tutti gli elementi maggiori di x ed inserisce in testa alla lista il numero di elementi eliminati.

ESERCIZIO 6 (punti 5)

Si definisca una procedura C che, dato un array di interi a di dimensione `dim` ordinato in senso non decrescente, ed un intero x , inserisce x nell'array mantenendo l'ordinamento ed eliminando da a l'ultimo elemento. Non si possono utilizzare array di supporto.