

PROGRAMMAZIONE 1 e LABORATORIO (A,B) - a.a. 2009-2010

Prova scritta del 1 giugno 2010

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

ESERCIZIO 1 (punti 5)

Si definisca una grammatica che genera il seguente linguaggio sull'alfabeto $\Lambda = \{a, b\}$

$$\mathcal{L} = \{a^n b^m \mid (n > 0 \wedge \text{pari}(n) \wedge m = n - 1) \vee (n > 0 \wedge \text{dispari}(n) \wedge m = n)\}$$

ESERCIZIO 2 (punti 5)

Dato il tipo degli alberi binari visto a lezione

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

si definisca una funzione

```
same_leafs : 'a btree -> 'a -> bool
```

in modo che (`same_leafs bt x`) restituisca *true* se tutte le foglie di `bt` sono uguali a `x`, *false* altrimenti.

ESERCIZIO 3 (punti 5)

Si definisca in C una funzione

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

che restituisce *true* se vale la seguente proprietà

$$\exists i \in [0, \text{dim}). a[i] = \sum_{\substack{j \in [0, \text{dim}) \\ j \neq i}} a[j]$$

e restituisce *false* altrimenti (si supponga dato il tipo `typedef enum {false, true} boolean;`).

ESERCIZIO 4 (punti 5)

Senza utilizzare ricorsione esplicita, si definisca in CAML una funzione

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

in modo che, se `foo l1 = l2`, la lista `l2` sia ottenuta da `l1` rimpiazzando ogni elemento `x` con la somma tra `x` e tutti gli elementi che seguono `x` in `l1`.

ESERCIZIO 5 (punti 5)

Un *multiinsieme* è una collezione di elementi in cui ciascun elemento può occorrere più volte. Ad esempio, $\{a, b, a, a, c, b\}$ è un multiinsieme di caratteri in cui *a* occorre tre volte, *b* due volte e *c* una volta. Un multiinsieme può essere rappresentato in C mediante una lista di coppie: ciascuna coppia nella lista rappresenta l'elemento del multiinsieme e il numero di sue occorrenze nel multiinsieme. Siano date allora le seguenti definizioni per la rappresentazione di multiinsiemi di caratteri

```
struct el { char c; int occ; struct el *next; }
typedef struct el ElementoMultiinsieme;
typedef ElementoMultiinsieme *Multiset;
```

Si scriva in C una procedura che, dato un multiinsieme di caratteri *m* ed un carattere *ch*, inserisca *ch* in *m*. Se *ch* non è già presente in *m*, la procedura dovrà inserirlo in testa al multiinsieme.

ESERCIZIO 6 (punti 5)

Scrivere in C una procedura che dato un array di interi *a* e la sua dimensione *dim*, ed un array di booleani *b* della stessa dimensione, modifichi *b* in modo che ciascun elemento *b*[*i*] sia uguale a *true* se il corrispondente elemento *a*[*i*] è maggiore di tutti gli elementi che lo seguono, sia uguale a *false* altrimenti.

(Si supponga dato il tipo `typedef enum {false, true} boolean;`)