

# PROGRAMMAZIONE 1 e LABORATORIO (A,B)

a.a. 2010-2011

## Prova scritta del 31 gennaio 2011

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

### ESERCIZIO 1 (punti 5)

Dato l'alfabeto  $\Lambda = \{1, 2, 3, 4\}$ , si definisca una grammatica libera che genera il seguente linguaggio  $\mathcal{L} \subseteq \Lambda^*$

$$\mathcal{L} = \{x\alpha y \mid x, y \in \Lambda \wedge y \geq x \wedge \alpha \in \Lambda^*\}$$

### ESERCIZIO 2 (punti 5)

Dato il tipo degli alberi binari visto a lezione

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

si definisca una funzione `count` con tipo

```
count : 'a btree -> ('a -> bool) -> int
```

in modo che `(count bt p)` sia il numero dei nodi in `bt` i cui figli soddisfano la proprietà `p`.

### ESERCIZIO 3 (punti 5)

Dato un intero `x` e un array `a` sia  $d(x, a)$  il valore minimo delle differenze in valore assoluto tra `x` e ciascun elemento di `a`. Ad esempio, dato l'array `a` rappresentato nella seguente tabella

20	4	25	11	5
----	---	----	----	---

$d(9, a) = 2$ .

Si definisca in C una procedura

```
void change (int a[], int b[], int dima, int dimb)
```

che rimpiazza ogni elemento `x` dell'array `a` (di dimensione `dima`) con il valore  $d(x, b)$ , dove `b` è un array di dimensione `dimb`.

### ESERCIZIO 4 (punti 5)

Si definisca in C una funzione

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

che restituisce `true` se il valore minimo nell'array `a` (di dimensione `dim`) occorre in esso più volte del valore massimo, e restituisce `false` altrimenti.

### ESERCIZIO 5 (punti 5)

Senza utilizzare ricorsione esplicita, ma solo funzioni di ordine superiore, si definisca in CAML una funzione

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

in modo che `(foo xs)` sia la lista ottenuta rimpiazzando ogni elemento di `xs` con il valore massimo della sottolista che inizia dall'elemento stesso.

### ESERCIZIO 6 (punti 5)

Si suppongano date le seguenti definizioni:

```
struct el { T info; struct el *next;};
typedef struct el ElementoLista;
typedef ElementoLista *ListaDiElementi;
```

dove `T` è un generico tipo. Scrivere in C una procedura che cancella da una lista passata come parametro l'elemento, se esiste, puntato da un puntatore fornito come ulteriore parametro.