

# PROGRAMMAZIONE 1 e LABORATORIO (A,B) - a.a. 2014-2015

## Prova scritta del 9/02/2015

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

### ESERCIZIO 1 (6 punti)

Dato l'alfabeto  $\Lambda = \{a, b, c, d\}$  si definisca una grammatica libera che genera il seguente linguaggio:

$$\mathcal{L} = \{ab^n c^k d^m \mid k, n, m > 0 \wedge m > n\}$$

Si dimostri, utilizzando il pumping lemma, che tale linguaggio non è regolare.

### ESERCIZIO 2 (6 punti)

Si definisca in CAML, senza utilizzare ricorsione esplicita, una funzione

```
pfirst : ('a -> bool) -> 'a list -> 'a list
```

in modo che (pfirst p lis) restituisca una lista con gli stessi elementi di lis, in cui tutti gli elementi che soddisfano p precedono tutti quelli che non soddisfano p.

### ESERCIZIO 3 (6 punti)

Si scriva in C una procedura che, dati attraverso opportuni parametri una lista di interi ed un intero n, cancella dalla lista, se c'è, l'ultimo elemento che contiene n.

Si suppongano predefiniti i tipi

```
struct el {int info; struct el *next;};

typedef struct el ElementoDiLista;
typedef ElementoDiLista *ListaDiInteri;
```

### ESERCIZIO 4 (6 punti)

Si scriva una funzione C

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

che, dato un array a e la sua dimensione dim, restituisce il valore di verità della seguente formula

$$\#\{i \mid i \in [0, dim - 1] \wedge (\exists j \in [i + 1, dim). a[i] = a[j])\} \geq 2$$

Si ricorda che, dato un insieme finito A, #A denota il numero di elementi di A.

### ESERCIZIO 5 (6 punti)

Una lista di coppie (valore, numero), in cui valore compare nella lista al più una volta e numero > 0, può essere utilizzata per rappresentare un multinsieme (ovvero una struttura in cui gli elementi, contrariamente agli insiemi, possono essere ripetuti). Ad esempio la lista di tipo (string \* int) list

```
[("abc", 1); ("x", 3); ("pq", 2)]
```

può rappresentare il multinsieme di stringhe

```
{"x", "abc", "x", "pq", "pq", "x"}.
```

Scrivere in CAML una funzione

```
canc : ('a * int) list -> 'a -> ('a * int) list
```

in modo che (canc m el) restituisca una lista che rappresenta il multinsieme ottenuto da quello rappresentato da m cancellando **una sola occorrenza** di el (se quest'ultimo occorre in m).