

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

Prova scritta del 12/01/2015

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

ESERCIZIO 1 (6 punti)

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

$$\mathcal{L} = \{\$a^n\$b^n\$ \mid n > 0\}$$

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

```
insord : 'a -> 'a list -> 'a list
```

in modo che `(insord x lis)`, supponendo `lis` ordinata in modo non decrescente, restituisca la lista ottenuta inserendo `x` in `lis` e ordinata nello stesso modo. Ad esempio:

```
insord 4 [-1; 0; 3; 3; 6; 10] = [-1; 0; 3; 3; 4; 6; 10]
```

```
insord 'b' ['d'; 'g'; 'h'] = ['b'; 'd'; 'g'; 'h']
```

ESERCIZIO 3 (6 punti)

Si scriva in C una procedura che, data una lista di interi in cui tutti gli elementi con valore pari precedono tutti gli elementi con valore dispari, e dato un intero `n`, inserisce `n` nella lista in modo che anche nella lista risultante tutti gli elementi con valore pari precedano tutti gli elementi con valore dispari.

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

$$(\forall i \in [0, dim - 1]. (\exists j \in [i + 1, dim). a[i] > a[j]))$$

ESERCIZIO 5 (6 punti)

Scrivere in CAML una funzione

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

che, data una lista `lis` di elementi, restituisce una lista di coppie in cui ciascuna coppia contiene un elemento di `lis` e il numero di volte in cui tale elemento occorre in `lis`. Nella lista risultante tutti i primi elementi delle coppie devono essere distinti tra loro.

Ad esempio:

```
multiset [20;40;30;20;50;20;30] = [(20,3); (40,1); (30,2); (50,1)]
```

Suggerimento: si utilizzino definizioni di funzioni ausiliarie.