

## Esercizio 1)

Il linguaggio non è regolare.

Lo dimostriamo con il pumping lemma.

Qualunque sia il valore di  $n$  prendiamo la stringa  $w = a^n b b a^n \in L$ .

Ovviamente  $|w| > n$ , tutte le possibili divisioni di  $w$  in  $x y z$ , tali che  $|x y| \leq n$  e  $y \neq \epsilon$ , sono rappresentate da

$$x = a^s \quad 0 \leq s < n$$

$$y = a^t \quad 0 < t \leq n - s$$

$$z = a^{n-t-s} b b a^n$$

per  $i=0$ , la stringa  $x y^0 z = a^{n-t} b b a^n \notin L$  dato che  $n-t < n$ .

Una grammatica che genera il linguaggio è la seguente:

$$S \rightarrow a S a \mid b S b \mid a a \mid b b$$

## Esercizio 2)

```

int contemin (int el, int a[], int inizio, int fine)
{
    int i;
    int conte = 0;
    for (i = inizio; i < fine; i++)
        if (a[i] < el) conte++;
    return conte;
}

```

```

int formule (int a[], int dim)
{
    int i;
    int conte = 0;
    for (i = 1; i < dim - 1; i++)
        if (contemin (a[i], a, 0, i) ==
            contemin (a[i], a, i + 1, dim)) conte++;
    return conte;
}

```

### Esercizio 3) possibile soluzione 1

```
let proc l =  
  let f x (s, l, b) =  
    if not b then (x, l, true)  
    else if x < s then (x, x :: l; b)  
        else (x, l, b)
```

in match l with

[ ] → [ ]

| x :: xs →

```
  let (s, l1, b) = foldl f (x, [ ], false) l  
  in l1 ;;
```

non verrà utilizzato. Serve solamente per dare un valore del tipo degli elementi delle liste.

Esercizio 3) possibile soluzione 2

let ultimo l = match l with

[x] → x

| x :: xs → ultimo xs;;

let rec l =

let f x (s, l) =

if x < s then (x, s :: l)  
 else (x, l)

in

let (s, l1) = foldl f (ultimo l, []) l  
 in l1;;

Esercizio 3) possibile soluzione 3

let  $\text{prec } l =$

let  $f\ x\ (l_1, l_2) = \text{match } l_2 \text{ with}$   
 $[\_] \rightarrow (l_1, [x])$

|  $y :: ys \rightarrow \text{if } x < y \text{ then } (y :: l_1, x :: l_2)$   
 else  $(l_1, x :: l_2)$

in let  $(l_1, l_2) = \text{foldr } f\ ([\ ], [\ ]) l$   
 in  $l_1 ;;$

# Esercizio 4)

let rec prec l = match l with

  [] → []

  |x] → []

  |x::y::ys →

    if x < y then

      y::prec (y::ys)

    else prec (y::ys);;