Titolo nota 15/12/2015

1) Indicare il tipo delle seguenti funzioni

```
let f g h x = if g x then h x else 1 ;;
let f g h = g h + h 3 ;;
let f x = let g y = if y then x else x+1 in g ;;
```

2) Definire una funzione CAML

extract : 'a list list → 'a list

che prese una lista di liste , ll, restituisce la lista che contiene l'ultimo valore, se c'è, delle liste in ll.

Es. extract [[3;4]; []; [1]; [2;3;6]] = [4; 1; 6]

Titolo nota 15/12/2015

3) Dato il tipo degli alberi binari

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

si definisca una funzione

conc : 'a btree → ('a → bool) → 'a btree

tale che

conc bt p

cancella tutte le foglie dell'albero il cui contenuto soddisfa p.

4) Definire una funzione ricorsiva

$f: \mathbb{N} \times \mathbb{N} \to \mathbb{N}$

tale che $\forall x,y \in \mathbb{N}.\ f(x,y) = 3x + y + 4$

in modo che la relazione di precedenza indotta sia

$(\forall x,y,x',y' \in \mathbb{N}.\ \langle x,y\rangle \prec_f \langle x',y'\rangle \equiv x' = x+2 \wedge y = y'+1)$

Dimostrare la correttezza della funzione proposta.

4) Scrivere in C una procedura che cancella il primo e l'ultimo elemento da una lista di interi, se esistono.

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

typedef struct el Elemento_Lista;
typedef Elemento_Lista *Lista_El;
```

```
let f g h x = if g x then h x else 1 ;;
```

$f : ('a \to bool) \to ('a \to int) \to 'a \to int$

```
let f g h = g h + h 3 ;;
```

$f : ((int \to int) \to int) \to (int \to int) \to int$

```
let f x = let g y = if y then x else x+1 in g ;;
```

$f : int \to (bool \to int)$

```
# f 3 ;;
- : bool -> int = <fun>
```

```
# (f 3) false ;;
- : int = 4
```

2) Soluzione ricorsiva

```
let rec extract ll =
        let rec last l = match l with
                [x] -> x | _::xs when xs<>[] -> last xs
        in  match ll with
            [] -> [] |
        []::ls -> extract ls |
        l::ls when l<>[] -> (last l)::(extract ls);;
```

last : 'a list -> 'a

extract : 'a list list -> 'a list = <fun>

Domanda: è corretto

```
let extract ll = let rec last l = ..... in map last ll ;;
```

NO !! last non è definita su liste vuote.

```
let extract ll = let f x y = if x=[] then y
                             else (last x) :: y

                 in foldr f [] ll
```

Titolo nota 08/10/2015

3)
```
let rec conc bt p = match bt with
      Void -> Void
| Node (x, Void, Void) -> if p x then Void else Node (x, Void, Void)
| Node (x, lt, rt) when lt <> Void or rt <> Void ->
                        Node (x, conc lt p, conc rt p) ;;
```

Titolo nota 08/10/2015

4) $f(x,y) = 3x + y + 4$

$\forall x,y,x',y'.\ \langle x,y\rangle \prec_f \langle x',y'\rangle \equiv x' = x+2 \ \&\ y = y'+1$

$$f(n,m) = \begin{cases} m+4 & \text{se } n = \emptyset \\ m+7 & \text{se } n = 1 \\ f(n-2, m+1) + 5 & \text{se } n \geq 2 \end{cases}$$

ipotesi induttiva

$3(n-2) + (m+1) + 4$

$= 3n - 6 + m + 1 + 4$

$= 3n + m - 1$

$\langle n, m\rangle$

$\langle n-2, m+1\rangle$

$\langle n-4, m+2\rangle$

$\langle \emptyset, \ldots\rangle$

$\langle n, m\rangle$

$\langle n-2, m+1\rangle$

$\langle 1, \ldots\rangle$

Titolo nota 08/10/2015

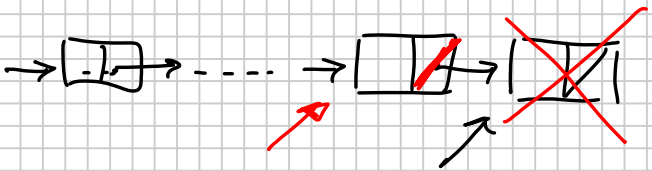
Dimostrazione: casi base sono ovvii

Caso induttivo $\underbrace{f(n-2, m+1) = 3(n-2) + (m+1) + 4}_{\text{Ip. induttiva}} \Rightarrow \underbrace{f(n,m) = 3n + m + 4}_{\text{da dimostrare}}$

$$
\begin{aligned}
& f(n,m) \\
= \; & \{ 3^\circ \text{ caso def. di } f \} \\
& f(n-2, m+1) + 5 \\
= \; & \{ \text{Ip. induttiva} \} \\
& 3(n-2) + m + 1 + 5 \\
= \; & \{ \text{calcolo} \} \\
& 3n + m + 4
\end{aligned}
$$

Titolo nota 08/10/2015

5) Procedura C che cancella il primo e l'ultimo elemento di una lista



```
void canc (ListaDiInteri *l)
 {

    if (*l != NULL)
     { corr = *l; *l = *l -> next; free (corr);
       prec = NULL; corr = *l;
       while (corr -> next != NULL)
          { prec = corr; corr = prec -> next; }
       if (prec == NULL)  *l = NULL;
         else prec -> next = NULL;
       free (corr);
     }
 }
```