

PROGRAMMAZIONE 1 e LABORATORIO (A,B) - a.a. 2011-2012

Esercitazione del 15/12/2011

ESERCIZIO 1

Si consideri il seguente tipo di alberi binari.

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

Si definisca in CAML una funzione `count_one` con tipo

```
count_one : 'a btree -> int
```

in modo che `count_one alb` restituisca il numero di nodi di `alb` che hanno esattamente un figlio.

ESERCIZIO 2

Scrivere in C una funzione

```
boolean subarray (int a [], int b [], int dima, int dimb)
```

che, dati due array `a` e `b` di dimensione `dima` e `dimb` rispettivamente, restituisce `true` se l'array `a` è una sottosequenza dell'array `b`, `false` altrimenti (si assuma data la definizione `typedef enum {false,true} boolean`).

Ad esempio, dati i seguenti array `vet1` e `vet2`.

```
3 | 1 | 5 |           | 5 | 3 | 1 | 5 | 7 | 9 |
```

la chiamata `subarray(vet1, vet2, 3, 6)` deve restituire `true`. Dati invece i seguenti array `vet1` e `vet2`

```
3 | 1 | 5 |           | 5 | 3 | 1 | 7 | 5 | 9 |
```

la chiamata `subarray(vet1, vet2, 3, 6)` deve restituire `false`.

ESERCIZIO 3

Scrivere in C una procedura

```
void foo (int a[], int dim, int x)
```

che rimpiazza con il valore `x` tutti gli elementi di `a` che precedono l'ultimo valore dispari in `a`.

ESERCIZIO 4

Date le seguenti definizioni:

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

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

scrivere in C una procedura che, data in ingresso attraverso un opportuno parametro una lista di interi, elimina dalla lista tutti gli elementi che precedono l'ultimo elemento dispari della lista, liberando opportunamente la memoria corrispondente agli elementi eliminati.

ESERCIZIO 5

Si supponga di estendere la sintassi dei comandi con la seguente produzione:

```
Com ::= do Com incr Ide;
```

la cui semantica (rispetto al modello senza memoria dinamica) è data dalla seguente regola:

$$\mathcal{S}em_c \text{ do } C \text{ incr } id \rho \mu = \mu' [v+1 / \rho(id)]^{mod}$$

dove $\mu' = \mathcal{S}em_c C \rho \mu$

e $v = \mu'(\rho(id))$

Estendere opportunamente il tipo `Com` e la definizione della funzione `semc` nella implementazione CAML della semantica dei comandi, al fine di contemplare anche il nuovo comando.