

PROGRAMMAZIONE 1 e LABORATORIO (A,B) - a.a. 2012-2013

Esercitazione del 18/12/2012

ESERCIZIO 1

Dato il tipo degli alberi binari

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

si definisca in CAML una funzione `check` con tipo

```
check : 'a btree -> int -> bool
```

in modo che `(check bt n)` restituisca `true` se la profondità di `bt` è al più `n`, `false` altrimenti.

ESERCIZIO 2

Scrivere in C una funzione

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

che, dati due array `a` e `b` di dimensione `dima` e `dimb` rispettivamente, verifica che ogni elemento di `a` con indice `i` occorra in `b` con indice `j > i`.

ESERCIZIO 3

Scrivere in C una procedura

```
void replace (int a[], int dim)
```

che rimpiazza ogni elemento `x` dell'array `a` con il numero di elementi di `a` uguali a `x` che lo precedono.

Dato ad esempio il seguente array `vet1`

8	5	8	8	6	5	2	8
---	---	---	---	---	---	---	---

la chiamata `replace(vet1,8)` deve modificare `vet1` come segue

0	0	1	2	0	1	0	3
---	---	---	---	---	---	---	---

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, modifica la lista portando nella parte iniziale della stessa tutti gli elementi negativi (**N.B.:** non è consentito utilizzare `malloc`).

ESERCIZIO 5

Senza utilizzare ricorsione esplicita, definire in CAML una funzione

```
foo : int list -> (int -> bool) -> (int * int)
```

in modo che `(foo lis p)` restituisca la coppia `(somma,num)` dove `somma` è la somma degli elementi di `lis` che soddisfano `p` e `num` è il numero di elementi di `lis` che non soddisfano `p`.