

PROGRAMMAZIONE 1 e LABORATORIO (A,B) - a.a. 2010-2011

II Verifica scritta

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

ESERCIZIO 1 (6 punti)

Dato il tipo degli alberi binari

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

si definisca in CAML una funzione `replace` con tipo

```
replace : 'a btree -> 'a -> 'a btree
```

in modo che `(replace bt y)` sia l'albero ottenuto da `bt` rimpiazzando con `y` le etichette dei nodi non foglia che hanno un solo sottoalbero non vuoto.

ESERCIZIO 2 (6 punti)

Scrivere in C una funzione

```
int sumOdd (int a [], int from, int to, int dim)
```

che calcola la somma degli elementi dispari dell'array `a` (di dimensione `dim`) con indice `i` tale che $from \leq i \leq to$
Ad esempio, dato l'array `vet` seguente

20	4	25	11	5	2	9
----	---	----	----	---	---	---

la chiamata `sumOdd(vet,1,4,7)` deve restituire il valore 41, la chiamata `sumOdd(vet,4,0,7)` il valore 0.

ESERCIZIO 3 (6 punti)

Scrivere in C una procedura

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

che rimpiazza tutti gli elementi successivi all'ultima occorrenza del valore massimo con il valore massimo stesso.

ESERCIZIO 4 (6 punti)

Date le seguenti definizioni:

```
struct el { int info; struct el *next;};  
  
typedef struct el ElementoLista;  
typedef ElementoLista *ListaDiInteri;
```

scrivere in C una procedura

```
void maxmin (ListaDiInteri lista, int *pmax, int *pmin)
```

che lascia nelle variabili puntate da `pmax` e `pmin` rispettivamente il valore massimo e il valore minimo presenti nella lista passata come primo argomento (se la lista è vuota le variabili puntate dal secondo e dal terzo argomento devono rimanere inalterate).

ESERCIZIO 5 (6 punti)

Definire, senza utilizzare ricorsione esplicita e dunque utilizzando funzioni di ordine superiore, una funzione

```
foo : int list -> int list
```

in modo che `foo l` sia la lista ottenuta da `l` azzerando tutti gli elementi che sono preceduti da un numero negativo.
Ad esempio

```
foo [1 ; -2; -4; 5; 6; -7; 8] = [1; -2; 0; 0; 6; -7; 0]
```