

# Esercizi Laboratorio Programmazione

Prof. Stefano Guerrini  
Canale P-Z

12 aprile 2002

Gli esercizi dovranno essere inviati **entro le 24 di giovedì 18/04/02.**

## 1 Esercizio 5: primi esercizi su alberi

Sia data la seguente struttura per la memorizzazione di alberi binari etichettati con numeri interi:

```
typedef struct nodobin {
    int info;
    struct nodobin *sn, *dx;
} NODOBIN;
typedef NODOBIN *ALBBIN;
```

Si devono scrivere due funzioni ricorsive

```
int cercamax(ALBBIN radice);
int sommanodi(ALBBIN radice);
```

delle quali, `cercamax` cerca il valore dell'etichetta massima dell'albero, mentre `sommanodi` somma i valori delle etichette nell'albero.

Il file `es5.h` (vedi appendice A) contiene i prototipi delle suddette funzioni e la definizione della struttura.

Le due funzioni dovranno essere implementate in un modulo `es5.c` che verrà poi verificato per mezzo del main in `provaes5.c` (vedi appendice B).

Il modulo `es5.c` inviato dagli studenti verrà compilato insieme a `provaes5.c` col comando

```
gcc provaes5.c es5.c -o provaes5
```

## 2 Esercizio 6: primi esercizi su alberi binari (iterativi)

Si devono scrivere le versioni iterative

```
int cercamaxiter(ALBBIN radice);  
int sommanodiiter(ALBBIN radice);
```

delle funzioni dell'esercizio 5.

Il file `es6.h` (vedi appendice C) contiene i prototipi delle suddette funzioni e la definizione della struttura.

Le due funzioni dovranno essere implementate in un modulo `es6.c` che verrà poi verificato per mezzo del main in `provaes6.c` (vedi appendice E).

Per poter scrivere l'algoritmo in modo iterativo, si deve anche scrivere un modulo `pilep.c`, nel quale si implementa uno stack di puntatori. L'interfaccia di tale modulo è fornita nel file `pilep.h` (vedi appendice D). Il significato delle funzioni è quello usuale (ed è anche specificato nei commenti ai prototipi delle funzioni).

**NB:** Per rendere l'implementazione dello stack riutilizzabile, ogni elemento dello stack contiene un puntatore di tipo generico `void *`. Quindi, attenzione alle conversioni di tipo necessarie.

Il modulo `es6.c` inviato dagli studenti verrà compilato insieme a `provaes6.c` col comando

```
gcc provaes6.c es6.c pilep.c -o provaes6
```

# A es5.h

```
/* -*- Mode: C -*- */
/* Time-stamp: <es5.h 02/04/12 10:36:46 guerrini@sgport.dsi.uniroma1.it> */

/* alberi binari etichettati con interi */
typedef struct nodobin {
    int info;
    struct nodobin *sn, *dx;
} NODOBIN;
typedef NODOBIN *ALBBIN;

int cercamax(ALBBIN radice);
/* cerca l'etichetta max nell'albero binario radice
*/

int sommanodi(ALBBIN radice);
/* somma le etichetta dei nodi dell'albero binario radice
*/
```

## B provaes5.c

```
/* -- Mode: C -- */
/* Time-stamp: <provaes5.c 02/04/12 10:32:48 guerrini@sgport.dsi.uniroma1.it> */

int main(void) {

/*****

    Verra' fornito in seguito

*****/

}
```

# C es6.h

```
/* -*- Mode: C -*- */
/* Time-stamp: <es6.h 02/04/12 10:37:23 guerrini@sgport.dsi.uniroma1.it> */

/* alberi binari etichettati con interi */
typedef struct nodobin {
    int info;
    struct nodobin *sn, *dx;
} NODOBIN;
typedef NODOBIN *ALBBIN;

int cercamaxiter(ALBBIN radice);
/* cerca l'etichetta max nell'albero binario radice
 * versione iterativa
 */

int sommanodiiter(ALBBIN radice);
/* somma le etichetta dei nodi dell'albero binario radice
 * versione iterativa
 */
```

## D pilep.h

```
/* -- Mode: C -- */
/* Time-stamp: <pilep.h 02/04/12 10:45:43 guerrini@sgport.dsi.uniroma1.it> */

/*****
 * header file della
 * libreria per la manipolazione di stack/pila di puntatori
 *****/

/* stack realizzati come liste lineari di puntatori */
typedef struct nodopilap {
    void *info;
    struct nodopilap *next;
} NODOFILAP;
typedef NODOFILAP *FILAP;

void inizzpilap(FILAP *);
/* inizializza la pila passata per riferimento: mette a NULL */

int pilavuota(FILAP);
/* verifica se la pila e- vuota: verifica se e' NULL */

void *pushpilap(FILAP *, void *);
/* inserisce un elemento in testa alla pila passata
 * per riferimento; ritorna un puntatore all'elemento inserito
 * se inserimento OK, altrimenti, ritorna NULL
 */

void *poppilap(FILAP *);
/* toglie dalla pila passata per riferimento l'elemento di testa e ritorna
 * il valore del campo info dell'elemento tolto dalla pila
 */
```

# E provaes6.c

```
/* -- Mode: C -- */
/* Time-stamp: <provaes6.c 02/04/12 10:32:59 guerrini@sgport.dsi.uniroma1.it> */

int main(void) {

/*****

    Verra' fornito in seguito

*****/

}
```