Laboratorio 6 – Implementazione del TDD Dizionario con alberi di ricerca binari

Si implementi in TDD Dizionario con *(chiave: string, valore:string)* usando alberi binari di ricerca, seguendo le indicazioni viste in classe e riportate sulle slide, ossia

- **implementazione delle operazioni con funzioni/procedure ricorsive che operino in theta(h) nel caso peggiore, con h=altezza dell'albero**
- **divieto di aggiunta del puntatore al padre alla struct che rappresenta un nodo**

Il TDD Dizionario così implementato deve fornire tutte le operazioni "standard" del TDD già viste a lezione e nel Laboratorio 5, i cui prototipi sono riportati in calce per comodità, e deve poter essere utilizzato nel main fornito con il Laboratorio 5, estendendo opportunamente lo header dictionary.h e compilando in modo adeguato.

Sarà necessario implementare le funzioni di lettura da file e stdin, nonché la **funzione di stampa corrispondente ad una visita DFS simmetrica**. I file su cui effettuare i test sono quelli già usati per il Laboratorio 5: le funzioni di lettura non saranno dissimili da quelle implementate dai docenti per il Laboratorio 5.

Il main deve essere consegnato così come è stato distribuito per il Laboratorio 5. In fase di sviluppo del codice è naturalmente possibile - anzi, è consigliato - modificare il main in modo da verificare separatamente le singole funzionalità implementate; in alternativa, è possibile creare un main nuovo nel quale vengono effettuate chiamate di funzioni ad hoc per il testing. Tuttavia alla fine il programma deve funzionare con il main fornito originariamente nel Laboratorio 5.

Il comando per compilare correttamente il programma dovrà essere esplicitamente indicato da qualche parte (nel main, oppure in un README) o contenuto in un Makefile.

La consegna di questo laboratorio deve includere anche il foglio excel dei tempi di esecuzione delle singole operazioni già predisposta per il Laboratorio 5, compilata in ogni sua parte incluse quelle richieste dal Laboratorio 5 ma comprendente anche le misure di prestazione relative all'albero binario di ricerca. Se non si è riusciti a implementare la tabella di hash con liste di collisione come richiesto dal Laboratorio 5, è possibile utilizzare la soluzione messa a disposizione dai docenti per completare il foglio excel.

Nello .zip consegnato devono essere presenti solo i file necessari a compilare ed eseguire (header, main e file ausiliari), eventuali file README (p.es. contenenti il comando di compilazione e la composizione del gruppo di

lavoro), eventuale Makefile, e il foglio excel completo in ogni sua parte. I file sorgenti ereditati dal laboratorio 5, se non necessari, vanno tolti.

/* Prototipi delle funzioni del TDD Dizionario e definizione dei tipi usati da esse; la definizione del tipo Dictionary è parte dell'esercitazione. Il tipo Dictionary dovrà rappresentare un albero binario di ricerca (si vedano slide) */

```
enum Error {OK, FAIL};

typedef string Key;
typedef string Value;

const Key emptyKey = "###RESERVED KEYWORD### EMPTY KEY";
const Value emptyValue = "###RESERVED KEYWORD### EMPTY VALUE";

struct dictionaryElem {
    Key    key;
    Value value;
};

typedef dictionaryElem Elem;

Error insertElem(const Key, const Value, Dictionary&);
Error deleteElem(const Key, Dictionary&);
Value search(const Key, const Dictionary&);
Dictionary createEmptyDict();
```